Section 1.1

1.1.1. We can form n digit numbers by choosing the leftmost digit AND choosing the next digit AND $\cdots$ AND choosing the rightmost digit. The first choice can be made in 9 ways since a leading zero is not allowed. The remaining $n-1$ choices can each be made in 10 ways. By the Rule of Product we have $9 \times 10^{n-1}$.

To count numbers with at most n digits, we could sum up $9 \times 10^{k-1}$ for $1 \le k \le n$. The sum can be evaluated since it is a geometric series. This does not include the number 0. Whether we add 1 to include it depends on our interpretation of the problem's requirement that there be no leading zeroes. There is an easier way. We can pad out a number with less than n digits by adding leading zeroes. The original number can be recovered from any such n digit number by stripping off the leading zeroes. Thus we see by the Rule of Product that there are 10^n numbers with at most n digits. If we wish to rule out 0 (which pads out to a string of n zeroes), we must subtract 1.

1.1.3. List the elements of the set in any order: $a_1, a_2, \ldots, a_{|S|}$. We can construct a subset by

including a_1 or not AND
including a_2 or not AND
. . .
including $a_{|S|}$ or not.

Since there are 2 choices in each case, the Rule of Product gives $2 \times 2 \times \cdots \times 2 = 2^{|S|}$.

1.1.5. The answers are SISITS and SISLAL. We'll come back to this type of problem when we study decision trees.

Section 1.2

1.2.1. If we want all assignments of birthdays to people, then repeats are allowed in the list mentioned in the hint. This gives 365^{30}. If we want all birthdays distinct, no repeats are allowed in the list. This gives $365 \times 364 \times \cdots \times (365-29)$. The ratio is 0.29. How can this be computed? There are a lot of possibilities. Here are some.

- Use a symbolic math package.
- Write a computer program.
- Use a calculator. Overflow may be a problem, so you might write the ratio as $(365/365) \times (364/365) \times \cdots \times (336/365)$.
- Use (1.2). You are asked to do this in the next problem. Unfortunately, there is no guarantee how large the error will be.
- Use Stirling's formula after writing the numerator as $365!/335!$. Since Stirling's formula has an error guarantee, we know we are close enough. Computing the values directly from Stirling's formula may cause overflow. This can be avoided in various ways. One is to rearrange the various factors by using some algebra:

$$\frac{\sqrt{2\pi 365}(365/e)^{365}}{\sqrt{2\pi 335}(335/e)^{335}(365)^{30}} = \sqrt{365/335}\,(365/335)^{335}/e^{30}.$$

  Another way is to compute the logarithm of Stirling's formula and use that to estimate the logarithm of the answer.

1.2.3. Each of the 7 letters ABMNRST appears once and each of the letters CIO appears twice. Thus we must form an ordered list from the 10 distinct letters. The solutions are

$$\begin{aligned} k=2:&\quad 10\times 9 = 90\\ k=3:&\quad 10\times 9\times 8 = 720\\ k=4:&\quad 10\times 9\times 8\times 7 = 5040\end{aligned}$$

1.2.5 (a) Since there are 5 distinct letters, the answer is $5 \times 4 \times 3 = 60$.

(b) Since there are 5 distinct letters, the answer is $5^3 = 125$.

(c) Either the letters are distinct OR one letter appears twice OR one letter appears three times. We have seen that the first can be done in 60 ways. To do the second, choose one of L and T to repeat, choose one of the remaining 4 different letters and choose where that letter is to go, giving $2 \times 4 \times 3 = 24$. To do the third, use T. Thus, the answer is $60 + 24 + 1 = 85$.

1.2.7 (a) push, push, pop, pop, push, push, pop, push, pop, pop. Remembering to start with something, say a on the stack: $(a(bc))((de)f)$.

(b) This is almost the same as (a). The sequence is 112211212122 and the last "pop" in (a) is replaced by "push, pop, pop."

(c) $a((b((cd)e))(fg))$; push, push, push, pop, push, pop, pop, push, push, pop, pop, pop; 111010011000.

1.2.9. Stripping off the initial R and terminal F, we are left with a list of at most 4 letters, at least one of which is an L. There is just 1 such list of length 1. There are $3^2 - 2^2 = 5$ lists of length 2, namely all those made from E, I and L minus those made from just E and I. Similarly, there are $3^3 - 2^3 = 19$ of length 3 and $3^4 - 2^4 = 65$. This gives us a total of 90.

The letters used are E, F, I, L and R in alphabetical order. To get the word before RELIEF, note that we cannot change just the F and/or the E to produce an earlier word. Thus we must change the I to get the preceding word. The first candidate in alphabetical order is F, giving us RELF. Working backwards in this manner, we come to RELELF, RELEIF, RELEF and, finally, RELEEF.

1.2.11. There are $n!/(n-k)!$ lists of length k. The total number of lists (not counting the empty list) is

$$\frac{n!}{(n-1)!} + \frac{n!}{(n-2)!} + \cdots + \frac{n!}{1!} + \frac{n!}{0!} = n!\left(\frac{1}{0!} + \frac{1}{1!} + \cdots + \frac{1}{(n-1)!}\right) = n!\sum_{i=0}^{n-1} \frac{1^i}{i!}.$$

Since $e = e^1 = \sum_{i=0}^{\infty} 1^i/i!$, it follows that the above sum is close to e.

1.2.13. We can only do parts (a) and (d) at present.

(a) A person can run for one of k offices or for nothing, giving $k+1$ choices per person. By the Rule of Product we get $(k+1)^p$.

(d) We can treat each office separately. There are $2^p - 1$ possible slates for an office: any subset of the set of candidates except the empty one. By the Rule of Product we have $(2^p - 1)^k$.

Section 1.3

1.3.1. After recognizing that $k = n\lambda$ and $n - k = n(1 - \lambda)$, it's simply a matter of algebra.

1.3.3. Choose values for pairs AND choose suits for the lowest value pair AND choose suits for the middle value pair AND choose suits for the highest value pair. This gives $\binom{13}{3}\binom{4}{2}^3 = 61,776$.

1.3.5. Choose the lowest value in the straight (A to 10) AND choose a suit for each of the 5 values in the straight. This gives $10 \times 4^5 = 10240$.

Although the previous answer is acceptable, a poker player may object since a "straight flush" is better than a straight—and we included straight flushes in our count. Since a straight flush is a straight all in the same suit, we only have 4 choices of suits for the cards instead of 4^5. Thus, there are $10 \times 4 = 40$ straight flushes. Hence, the number of straights which are not straight flushes is $10240 - 40 = 10200$.

1.3.7. This is like Exercise 1.2.3, but we'll do it a bit differently. Note that EXERCISES contains 3 E's, 2 S's and 1 each of C, I, R and X. By the end of Example 1.18, we can use (1.4) with $N = 9$, $m_1 = 3$, $m_2 = 2$ and $m_3 = m_4 = m_5 = m_6 = 1$. This gives $9!/3!\,2! = 30240$.

It can also be done without the use of a multinomial coefficient as follows. Choose 3 of the 9 possible positions to use for the three E's AND choose 2 of the 6 remaining positions to use for the two S's AND put a permutation of the remaining 4 letters in the remaining 4 places. This gives us $\binom{9}{3} \times \binom{6}{2} \times 4!$.

The number of eight letter arrangements is the same. To see this, consider a 9-list with the ninth position labeled "unused."

1.3.9. Think of the teams as labeled and suppose Teams 1 and 2 each contain 3 men. We can divide the men up in $\binom{11}{3,3,2,2,1}$ ways and the women in $\binom{11}{2,2,3,3,1}$ ways.

We must now count the number of ways to form the ordered situation from the unordered one. Be careful—it's not $4! \times 2$ as it was in the example! Thinking as in the early card example, we start out two types of teams, say M or F depending on which sex predominates in the team. We also have two types of referees. Thus we have two M teams, two F teams, and one each of an F referee and an M referee. We can order the two M teams (2 ways) and the two F teams (2 ways), so there are only 2×2 ways to order and so the answer is $\binom{11}{3,3,2,2,1}^2 \frac{1}{4}$.

1.3.11. The theorem is true when $k = 2$ by the binomial theorem with $x = y_1$ and $y = y_2$. Suppose that $k > 2$ and that the theorem is true for $k - 1$. Using the hint and the binomial theorem with $x = y_k$ and $y = y_1 + y_2 + \cdots + y_{k-1}$, we have that

$$(y_1 + y_2 + \cdots + y_k)^n = \sum_{j=0}^{n} \binom{n}{j} (y_1 + y_2 + \cdots + y_{k-1})^{n-j} y_k^j.$$

Thus the coefficient of $y_1^{m_1} \cdots y_k^{m_k}$ in this is $\binom{n}{m_k} = n!/(n - m_k)!m_k!$ times the coefficient of $y_1^{m_1} \cdots y_{k-1}^{m_{k-1}}$ in $(y_1 + y_2 + \cdots + y_{k-1})^{n-m_k}$. When $n - m_k = m_1 + m_2 + \cdots + m_{k-1}$ the coefficient is $(n - m_k)!/m_1!m_2! \cdots m_{k-1}!$ and otherwise it is zero by the induction assumption. Multiplying by $\binom{n}{m_k}$, we obtain the theorem for k.

Section 1.4

1.4.1. The rows are 1,7,21,35,35,7,1 and 1,8,28,56,70,56,28,8,1.

1.4.3. Let $L(n,k)$ be the number of ordered k-lists without repeats that can be made from an n-set S. Form such a list by choosing the first element AND then forming a $k-1$ long list using the remaining $n-1$ elements. This gives $L(n,k) = nL(n-1,k-1)$.

Single out one item $x \in S$. There are $L(n-1,k)$ lists not containing x. If x is in the list, it can be in any of k positions AND the rest of the list can be constructed in $L(n-1,k-1)$ ways. Thus

$$L(n,k) \;=\; L(n-1,k) + kL(n-1,k-1).$$

1.4.5. The only way to partition an n element set into n blocks is to put each element in a block by itself, so $S(n,n) = 1$. The only way to partition an n element set into one block is to put all the elements in the block, so $S(n,1) = 1$.

The only way to partition an n element set into $n-1$ blocks is to choose two elements to be in a block together an put the remaining $n-2$ elements in $n-2$ blocks by themselves. Thus it suffices to choose the 2 elements that appear in a block together and so $S(n,n-1) = \binom{n}{2}$.

The formula for $S(n,n-1)$ can also be proved using (1.9) and induction. The formula is correct for $n = 1$ since there is no way to partition a 1-set and have no blocks. Assume true for $n-1$. Use the recursion, the formula for $S(n-1,n-1)$ and the induction assumption for $S(n-1,n-2)$ to obtain

$$S(n,n-1) = S(n-1,n-2) + (n-1)S(n-1,n-1) = \binom{n-1}{2} + (n-1)1 = \binom{n}{2},$$

which completes the proof.

Now for $S(n,2)$. Note that $S(n,k)$ is the number of unordered lists of length k where the list entries are nonempty subsets of a given n-set and each element of the set appears in exactly one list entry. We will count ordered lists, which is $k!$ times the number of unordered ones. We choose a subset for the first block (first list entry) and use the remaining set elements for the second block. Since an n-set has 2^n, this would seem to give $2^n/2$; however, we must avoid empty blocks. In the ordered case, there are two ways this could happen since either the first or second list entry could be the empty set. Thus, we must have $2^n - 2$ instead of 2^n.

Here is another way to compute $S(n,2)$. Look at the block containing n. Once it is determined, the entire two block partition is determined. The block one of the 2^{n-1} subsets of $\underline{n-1}$ with n adjoined. Since something must be left to form the second block, the subset cannot be all of $\underline{n-1}$. Thus there are $2^{n-1} - 1$ ways to form the block containing n.

The formula for $S(n,2)$ can also be proved by induction using the recursion for $S(n,k)$ and the fact that $S(n,1) = 1$, much as was done for $S(n,n-1)$.

1.4.7. There are $\binom{n}{k}$ ways to choose the subset AND k ways to choose an element in it to mark. This gives the left side of the recursion times k. On the other hand, there are n ways to choose an element to mark from $\{1,2,\ldots,n\}$ AND $\binom{n-1}{k-1}$ ways to choose the remaining elements of the k-element subset.

1.4.9 (b) Each office is associated with a nonempty subset of the people and each person must be in exactly one subset. This is a partition of the set of candidates with each block corresponding to an office. Thus we have an ordered partition of a n element set into k blocks. The answer is $k!S(n,k)$.

(c) This is like the previous part, except that some people may be missing. We use two methods. First, let i people run for no offices. The remaining $n-i$ can be partitioned in $S(n-i,k)$ ways and the blocks ordered in $k!$ ways. Thus we get $\sum_{i\ge 0}\binom{n}{i}k!S(n-i,k)$. For the second method, either everyone runs for an office, giving $k!S(n,k)$ or some people do not run. In the latter case, we can think of a partition with $k+1$ labeled blocks where the labels are the k offices and "not running." This give $(k+1)!S(n,k+1)$. Thus we have $k!S(n,k)+(k+1)!S(n,k+1)$. The last formula is preferable since it is easier to calculate from tables of Stirling numbers.

(e) Let $T(p,k)$ be the number of solutions. Look at all the people running for the first $k-1$ offices. Let t be the number of these people. If $t<p$, then at least $p-t$ people must be running for the kth office since everyone must run for some office. In addition, any of these t people could run for the kth office. By the Rule of Product, the number of ways we can have this particular set of t people running for the first $k-1$ offices and some people running for the kth office is $T(t,k-1)2^t$. The set of t people can be chosen in $\binom{p}{t}$ ways. Finally, look at the case $t=p$. In this case everyone is running for one of the first $k-1$ offices. The only restriction we must impose is that a nonempty set of candidates must run for the kth office. Putting all this together, we obtain

$$T(p,k) \;=\; \sum_{t=1}^{p-1}\binom{p}{t}T(t,k-1)2^t + T(p,k-1)(2^p-1).$$

This recursion is valid for $p\ge 2$ and $k\ge 2$. The initial conditions are $T(p,1)=1$ for $p>0$ and $T(1,k)=1$ for $k>0$.

Notice that if "people" and "offices" are interchanged, the problem is not changed. Thus $T(p,k)=T(k,p)$ and a recursion could have been obtained by looking at offices that the first $p-1$ people run for. This would give us

$$T(p,k) \;=\; \sum_{t=1}^{k-1}\binom{k}{t}T(p-1,t)2^t + T(p-1,k)(2^k-1).$$

Section 1.5

1.5.1. For each element, there are $j+1$ choices for the number of repetitions, namely anything from 0 to j, inclusive. By the Rule of Product, we obtain $(j+1)^{|S|}$.

1.5.3. To form an unordered list of length k with repeats from $\{1,2,\ldots,n\}$, either form a list without n OR form a list with n. The first can be done in $M(n-1,k)$ ways. The second can be done by forming a $k-1$ element list AND then adjoining n to it. This can be done in $M(n,k-1)\times 1$ ways. Initial conditions: $M(n,0)=1$ for $n\ge 0$ and $M(0,k)=0$ for $k>0$.

1.5.5. Interpret the points between the ith and the $(i+1)$st vertical bars as the balls in box i. Since there are $n+1$ bars, there are n boxes. Since there are $(n+k-1)-(n-1)=k$ points, there are k balls.

1.5.7. This exercise and the previous one are simply two different ways of looking at the same thing since an unordered list with repetitions allowed is the same as a multiset. The nth item must appear zero, one OR two times. The remaining $n-1$ items must be used to form a list of length k, $k-1$ or $k-2$ respectively. This gives the three terms on the left. We generalize to the case where each item is used at most j times: $T(n,k) \;=\; \sum_{i=0}^{j} T(n-1,k-i)$.

1.5.9 (a) We give two solutions. Both use the idea of inserting a ball into a tube in an arbitrary position. To physically do this may require some manipulation of balls already in the tube.

1. Insert $b-1$ balls into the tubes AND then insert the bth ball. There are $i+1$ possible places to insert this ball in a tube containing i balls. Summing this over all t tubes gives us $(b-1)+t$ possible places to insert the bth ball. We have proved that

$$f(b,t) \;=\; f(b-1,t)(b+t-1).$$

Since $f(1,t)=t$, we can establish the formula by induction.

2. Alternatively, we can insert the first ball AND insert the remaining $b-1$ balls. The first ball has the effect of dividing the tube in which it is placed into two tubes: the part above it and the part below. Thus

$$f(b,t) \;=\; tf(b-1,t+1),$$

and we can again use induction.

(b) We give two solutions:

Construct a list of length $t+b-1$ containing each ball exactly once and containing $t-1$ copies of "between tubes." This can be done in $\binom{t+b-1}{t-1}b!$ ways—choose the "between tubes" and then permute the balls to place them in the remaining b positions in the list.

Alternatively, imagine an ordered $b+t-1$ long list. Choose $t-1$ positions to be divisions between tubes AND choose how to place the b balls in the remaining b positions. This gives $\binom{b+t-1}{t-1} \times b!$.

Section 2.2

2.2.3. The interchanges can be written as (1,3), (1,4) and (2,3). Thus the entire set gives $1 \to 3 \to 2$, $2 \to 3$, $3 \to 1 \to 4$ and $4 \to 1$. In cycle form this is (1,2,3,4). Thus five applications takes 1 to 2.

2.2.5 (a) This was done in Exercise 2.2.2, but we'll redo it. If $f(k)=k$, then the elements of $\underline{n}-\{k\}$ can be permuted in any fashion. This can be done in $(n-1)!$. Since there are $n!$ permutations, the probability that $f(k)=k$ is $(n-1)!/n! = 1/n$. Hence the probability that $f(k) \neq k$ is $1-1/n$.

(b) By the independence assumption, the probability that there are no fixed points is $(1-1/n)^n$. One of the standard results in calculus is that this approaches $1/e$ as $n \to \infty$. (You can prove it by writing $(1-1/n)^n = \exp(\ln(1-1/n)/(1/n))$, setting $1/n = x$ and using l'Hôpital's Rule.)

(c) Choose the k fixed points AND construct a derangement of the remaining $n-k$. This gives us $\binom{n}{k}D_{n-k}$. Now use $D_{n-k} \approx (n-k)!/e$.

2.2.7. For $1 \le k \le n-1$, $\mathbf{E}(|a_k - a_{k+1}|) = \mathbf{E}(|i-j|)$, where the latter expectation is taken over all $i \neq j$ in $\underline{n}$. Thus the answer is $(n-1)$ times the average of the $n(n-1)$ values of $|i-j|$ and so

$$\begin{aligned}
\text{answer} &= \frac{n-1}{n(n-1)}\sum_{i\neq j}|j-i| \;=\; \frac{n-1}{n(n-1)}\sum_{i,j}|j-i| \;=\; \frac{2}{n}\sum_{1\le i\le j\le n}(j-i), \qquad \text{proving (a)}\\
&= \frac{2}{n}\sum_{j=1}^{n}\sum_{i=1}^{j}(j-i) \;=\; \frac{2}{n}\sum_{j=1}^{n}\left(j^2 - \frac{j(j+1)}{2}\right) \;=\; \frac{1}{n}\sum_{j=1}^{n}(j^2-j)\\
&= \frac{1}{n}\left(\frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2}\right) \;=\; \frac{n^2-1}{2}.
\end{aligned}$$

Section 2.3

2.3.3. We can form the permutations of the desired type by first constructing a partition of $\underline{n}$ counted by $B(n, \vec{b})$ AND then forming a cycle from each block of the partition. The argument used in Exercise 2.2.2 proves that there are $(k-1)!$ cycles of length k that can be made from a k-set.

2.3.5 (a) In the order given, they are 2, 1, 3 and 4

(b) If f is associated with a B partition of $\underline{n}$, then B is the coimage of f and so f determines B.

(c) See (b).

(d) The first is not since $f(1) = 2 \neq 1$.
The second is: just check the conditions.
The third is not since $f(4) - 1 = 2 > \max(f(1), f(2), f(3)) = 1$.
The fourth is: just check the conditions.

(e) In a way, this is obvious, but it is tedious to write out a proof. By definition $f(1) = 1$. Choose $k > 1$ such that $f(x) = k$ for some x. Let y be the least element of $\underline{n}$ for which $f(y) = k$. By the way f is constructed, y is not in the same block with any $t < y$. Thus y is the smallest element in its block and so $f(y)$ will be the smallest number exceeding all the values that have been assigned for $f(t)$ with $t < y$. Thus the maximum of $f(t)$ over $t < y$ is $k - 1$ and so f is a restricted growth function.

(f) The functions are given in one-line form and the partition below them

1 1 1 1 $\{1,2,3,4\}$	1 1 1 2 $\{1,2,3\}\,\{5\}$	1 1 2 1 $\{1,2,4\}\,\{3\}$	1 1 2 2 $\{1,2\}\,\{3,4\}$	1 1 2 3 $\{1,2\}\,\{3\}\,\{4\}$
1 2 1 1 $\{1,3,4\}\,\{2\}$	1 2 1 2 $\{1,3\}\,\{2,4\}$	1 2 1 3 $\{1,3\}\,\{2\}\,\{4\}$	1 2 2 1 $\{1,4\}\,\{2,3\}$	1 2 2 2 $\{1\}\,\{2,3,4\}$
1 2 2 3 $\{1\}\,\{2,3\}\,\{4\}$	1 2 3 1 $\{1,4\}\,\{2\}\,\{3\}$	1 2 3 2 $\{1\}\,\{2,4\}\,\{3\}$	1 2 3 3 $\{1\}\,\{2\}\,\{3,4\}$	1 2 3 4 $\{1\}\,\{2\}\,\{3\}\,\{4\}$

2.3.7. The coimage is a partition of A into at most $|B|$ blocks, so our bound is $1 + (|A| - 1)/|B|$.

2.3.9. If $s < t$ and $f(s) = f(t)$, that tells us that we cannot put a_s at the start of the longest decreasing subsequence starting with a_t to obtain a decreasing subsequence. (If we could, we'd have $f(s) \geq f(t) + 1$.) Thus, $a_s > a_t$. Hence the subsequence $a_i, a_j, \ldots$ constructed in the problem is increasing.

Now we're ready to start the proof. If there is a decreasing subsequence of length $n+1$ we are done. If there is no such subsequence, $f : \underline{\ell} \to \underline{n}$. By the generalized Pigeonhole Principle, there is sum k such that $f(t) = k$ for at least ℓ/n values of t. Thus it suffices to have $\ell/n > m$. In other words $\ell > mn$.

2.3.11. Let the elements be $s_1, \ldots, s_n$, let $t_0 = 0$ and let $t_i = s_1 + \ldots + s_i$ for $1 \leq i \leq n$. By the Pigeonhole Principle, two of the t's have the same remainder on division by n, say t_j and t_k with $j < k$. It follows that $t_k - t_j = s_{j+1} + \ldots + s_k$ is a multiple of n.

Section 2.4

2.4.1. $x(x+y) = xx + xy = x + xy = x$.

2.4.3. We state the laws and whether they are true or false. If false we give a counterexample.

(a) $x + (yz) \;=\; (x+y)(x+z)$ is true. (Proved in text.)

(b) $x(y \oplus z) \;=\; (xy) \oplus (xz)$ is true.

(c) $x + (y \oplus z) \;=\; (x+y) \oplus (x+z)$ is false with $x = y = z = 1$.

(d) $x \oplus (yz) \;=\; (x \oplus y)(x \oplus z)$ is false with $x = y = 1$, $z = 0$.

2.4.5. We use algebraic manipulation. Each step involves a simple formula, which we will not bother to mention. You could also write down the truth table, read off a disjunctive normal form and try to reduce the number of terms.

(a) $(x \oplus y)(x+y) \;=\; (xy' + x'y)(x+y) \;=\; xy' + x'yx + xy'y + x'y \;=\; xy' + x'y$. Note that this is $x \oplus y$.

(b) $(x+y) \oplus z \;=\; (x+y)z' + (x+y)'z \;=\; xz' + yz' + x'y'z$.

(c) $(x+y+z) \oplus z \;=\; (x+y+z)z' + (x+y+z)'z \;=\; xz' + yz' + x'y'z'z \;=\; xz' + yz'$.

(d) $(xy) \oplus z \;=\; xyz' + (xy)'z \;=\; xyz' + x'z + y'z$.

2.4.7. There are many possible answers. A complicated one comes directly from the truth table and contains 8 terms. The simplest form is $xw + yw + zw + xyz$. This can be obtained as follows. $(x+y+z)w$ will give the correct answer except when $x = y = z = 1$ and $w = 0$. Thus we could simply add the term $xyzw'$. By noting that it is okay to add xyz when $w = 1$, we obtain $(x+y+z)w + xyz$.

Section 3.1

3.1.1. From the figures in the text, we see that they are 123, 132 and 321.

3.1.3. We will not draw the tree. The root is 1, the vertices on the next level are 21 and 12 (left to right). On the next level, 321, 231, 213, 312, 132, and 123. Finally, the leaves are 4321, 3421, 3241, 3214, 4231, 2431, 2341, 2314, 4213, 2413, 2143, 2134, and so on.

(a) 7 and 16.

(b) 2,4,3,1 and 3,1,2,4.

3.1.5. We will not draw the tree. There are nine sequences: ABABAB, ABABBA, ABBABA, ABBABB, BABABA, BABABB, BABBAB, BBABAB and BBABBA.

3.1.7. We will not draw the tree.

(a) 5 and 18.

(b) 111 and 433.

(c) 4,4,4 has rank 19.

(e) The decision tree for the strictly decreasing functions is interspersed. To find it, discard the leftmost branch leading out of each vertex except the root and then discard those decisions that no longer lead to a leaf of the original tree.

3.1.9. We assume that you are looking at decision trees in the following discussion.

(a) The permutation of rank 0 is the leftmost one in the tree and so each element is inserted as far to the left as possible. Thus the answer is $n, (n-1), \ldots, 2, 1$.

The permutation of rank $n! - 1$ is the rightmost one in the tree and so each element is inserted as far to the right as possible. Thus the answer is $1, 2, 3, \ldots, n$.

We now look at $n!/2$. Note that the decision about where to insert 2 splits the tree into two equal pieces. We are interested in the leftmost leaf of the righthand piece. The righthand piece means we take the branch $1, 2$. To stay to the left after that, 3 through n are inserted in the leftmost position. Thus the permutation is $n, (n-1), \ldots, 4, 3, 1, 2$.

(b) The permutation of rank 0 is the leftmost one in the tree and so each element is inserted as far to the left as possible. It begins 2,1. Then 3 "bumps" 2 to the end: 3,1,2. Next 4 "bumps" 3 to the end: 4,1,2,3. In general, we have $n, 1, 2, 3, \ldots, (n-1)$.

The permutation of rank $n! - 1$ is the rightmost one in the tree and so each element is inserted as far to the right as possible. Thus the answer is $1, 2, 3, \ldots, n$.

We now look at $n!/2$. Note that the decision about where to insert 2 splits the tree into two equal pieces. We are interested in the leftmost leaf of the righthand piece. The righthand piece means we take the branch $1, 2$. To stay to the left after that, 3 through n are inserted in the leftmost position. This leads to "bumping" as it did for rank 0. Thus the permutation is $n, 2, 1, 3, 4, 5, \ldots, (n-1)$.

(c) You should be able to see that the permutation $(1, 2, 3, \ldots, n)$ has rank 0 in both cases and that the permutation $(n, \ldots, 3, 2, 1)$ has rank $n! - 1$ in both cases.

First suppose that $n = 2m$, an even number. It is easy to see how to split the tree in half based on the first decision as we did for insertion order: Choose $m + 1$ and then stay as left as possible. This means everything is in order except for $m + 1$. Thus the permutation is $m + 1$ followed by the elements of $\underline{n} - \{m + 1\}$ in ascending order.

Now suppose that $n = 2m - 1$. In this case, we must make the middle choice, m and split the remaining tree in half, going to the leftmost leaf of the right part. If you look at some trees, you should see that this leads to the permutation $m, m + 1$ followed by the elements of $\underline{n} - \{m, m + 1\}$ in ascending order.

3.1.11 (a) We'll make a decision based on whether or not the pair in the full house has the same face value as a pair in the second hand. If it does not, there are

$$\binom{11}{2}\binom{4}{2}^2(52-8-5) \;=\; 77{,}220$$

possible second hands. If it does, there are

$$11\binom{4}{2}(52-8-3) \;=\; 2{,}706$$

possible second hands. Adding these up and multiplying by the number of possible full houses (79,926) gives us about 3×10^8 hands.

(b) There are various ways to do this. The decision trees are all more complicated than in the previous part.

(c) The order in which things are done can be very important.

3.1.13. You can simply modify the decision tree in Figure 3.5 as follows: Decrease the "number of singles" values by 1 (since the desired word is one letter shorter). Throw away those that become negative; i.e., erase leaves C and H. Add a new path that has no triples, one pair and five singles.

Call the new leaf X. It is then necessary to recompute the numbers. Here are the results, which total to 113,540:

$$
\begin{aligned}
X: &\quad \binom{2}{0}\binom{5}{1}\binom{5}{5}\binom{8}{2,\,1,\,1,\,1,\,1,\,1} &= 12{,}600\\
A: &\quad \binom{2}{0}\binom{5}{2}\binom{4}{3}\binom{8}{2,\,2,\,1,\,1,\,1} &= 50{,}400\\
B: &\quad \binom{2}{0}\binom{5}{3}\binom{3}{1}\binom{8}{2,\,2,\,2,\,1} &= 18{,}900\\
D: &\quad \binom{2}{1}\binom{4}{0}\binom{5}{4}\binom{8}{3,\,1,\,1,\,1,\,1} &= 8{,}400\\
E: &\quad \binom{2}{1}\binom{4}{1}\binom{4}{2}\binom{8}{3,\,2,\,1,\,1} &= 20{,}160\\
F: &\quad \binom{2}{1}\binom{4}{2}\binom{3}{0}\binom{8}{3,\,2,\,2} &= 2{,}520\\
G: &\quad \binom{2}{2}\binom{3}{0}\binom{4}{1}\binom{8}{3,\,3,\,1} &= 560.
\end{aligned}
$$

Section 3.2

3.2.1. We use the rank formula in the text and, for unranking, a greedy algorithm.

(a) $\binom{10}{3} + \binom{5}{2} + \binom{3}{1} = 133. \qquad \binom{8}{4} + \binom{5}{3} + \binom{2}{2} + \binom{0}{1} = 81.$

(b) We have $35 = \binom{7}{4}$ so the first answer is 8,3,2,1. The second answer is 12,9,6,5 because

$$
\begin{aligned}
\binom{11}{4} &\le 400 < \binom{12}{4} & 400 - \binom{11}{4} &= 70\\
\binom{8}{3} &\le 70 < \binom{9}{3} & 70 - \binom{8}{3} &= 14\\
\binom{5}{2} &\le 14 < \binom{6}{2} & 14 - \binom{5}{2} &= 4\\
\binom{4}{1} &\le 4 < \binom{5}{1}.
\end{aligned}
$$

(c) 9,6,4,2,1 and 9,7,2,1.

(d) 9,5,4,3,2 and 9,6,5,3.

3.2.3. One can compute the ranks by looking at the decision tree or by using the formula in Theorem 3.3. We choose the latter approach. In case (j), we have $f(i) = k + j - i$. (This is easily checked since this f clearly decreases by 1 as i increases by 1 and it gives $f(1) = k$, $k+1$ and $k+2$ for $j = 1$, 2 and 3, respectively.) By the theorem,

$$\text{RANK}(f) = \sum_{i=1}^{k} \binom{f(i)-1}{k+1-i} = \sum_{i=1}^{k} \binom{k+j-i-1}{k+1-i}$$

When $j = 1$, all the binomial coefficients are 0 and so the answer for the first function is 0.
When $j = 2$, all the binomial coefficients are 1 and so the answer for the second function is k.
When $j = 3$, we have

$$\text{RANK}(f) = \sum_{i=1}^{k} \binom{k+2-i}{k+1-i} = \sum_{i=1}^{k} (k+2-i) = (k+1) + (k) + (k-1) + \cdots + (2).$$

Since the sum of the first n positive integers is $\frac{n(n+1)}{2}$, the rank is $\frac{(k+1)(k+2)}{2} - 1 = \frac{k(k+3)}{2}$.

				•						
	1			3				4		choice for (2,2) entry
	4		1		4			1		choice for (2,3) entry
	3				1			3		choice for (2,4) entry
	4		1		4			1		choice for (3,2) entry
1		2	2	1		2	2		4	choice for (3,3) entry
2		1		2					2	choice for (3,4) entry

Figure S.3.1 The decision tree for 4×4 standard Latin Squares in Exercise 3.3.1.

3.2.5 (a) $D_1 \times (n-1)! + D_2 \times (n-2)! + \cdots + D_{n-1} \times 1! \;=\; \sum_{k=1}^{n-1} D_k (n-k)!$.

(b) Denote the permutation by f. Let $L = \underline{n}$. For $i = 1, 2, \ldots, n-1$ in order: let D_i is the number of elements in L which are less than $f(i)$ and replace L with $L - \{f(i)\}$.

(c) The decision sequences are 4,4,0,1,1 and 5,1,2,0,0 and so the ranks are 579 and 636.

(d) By a greedy algorithm we get the decision sequences 1,1,1,0,1 and 2,2,2,0,0. The permutations are 2,3,4,1,6,5 and 3,4,5,1,2,6.

3.2.9. $00000000000000000000 = 0^{20}$; $11000000000000000000 = 1^2 0^{18}$; 0100; 10101100.

Section 3.3

3.3.1. When building and $n \times n$ Latin Square, if the first $n-1$ rows have been filled in, then the last row is determined. Thus we'll omit it from the decision tree. The tree is shown in Figure S.3.1.

3.3.3. You should find 14 solutions.

Section 4.1

4.1.1. The Venn diagrams each consist of two intersecting circles.

(a) $V_2 \cap V_3$ contains words of the form $CVVC$. We are interested in $V_2 \cup V_3$, the union of the circles. Thus

$$\begin{aligned} |V_2 \cup V_3| &= |V_2| + |V_3| - |V_2 \cap V_3| \\ &= 21^2 \times 5 \times 26 + 21^2 \times 5 \times 26 - 21^2 \times 5^2 \\ &= 21^2 \times 5 \times 47 \end{aligned}$$

(b) We want all 4 letter words beginning and ending with consonants that are not in $C_2 \cap C_3$, which is $21^2 \times 26^2 - 21^4$.

4.1.3 (a) If everyone who lost an eye also lost an arm, a leg and an ear, then there would be 70 people who lost all four.

(b) Let A be the set of people who lost an arm and L the set who lost a leg. How small can $A \cap L$ be? We have

$$|A \cap L| \;=\; |A| + |L| - |A \cup L| \;=\; 165 - |A \cup L| \;\geq\; 165 - 100 \;=\; 65.$$

We can now look at the set $D = A \cap L$ of double amputees and ask now many must have lost an eye. As above, we have

$$|D \cap I| \;=\; |D| + |I| - |D \cup I| \;\geq\; 65 + 70 - 100 \;=\; 35,$$

where I is the set of people who have lost an eye. Finally, we combine these people with the 75 who have lost an ear to conclude that at least $35 + 75 - 100 \;=\; 10$ must have lost all four. Thus $p \geq 10$. We can achieve this by insisting that everyone lost at least three things. If the people are numbered 1–100, we can do it as follows:

lost arm:	1–80
lost leg:	1–65 and 81–100
lost eye:	1–35 and 66–100
lost ear:	1–10 and 36–100

4.1.5 (a) A number x has a factor in common with N if and only if it is divisible by one of the primes that divide N. Thus an element of $\underline{N}$ has no factor in common with N if and only if it is in none of the sets S_k.

(b) The intersection on the left side is the set of $x \in \underline{N}$ that are multiples of $b = p_{i_1} \cdots p_{i_r}$. These are b, $2b$, $3b$ …, $(N/b)b$. Thus the set has N/b elements, as was to be proved.

(c) By (4.3) and the previous result, we have

$$\varphi(N) \;=\; \sum_{I \subseteq \underline{n}} (-1)^{|I|} \frac{N}{\prod_{i \in I} p_i} \;=\; N \sum_{I \subseteq \underline{n}} \prod_{i \in I} \left(\frac{-1}{p_i} \right).$$

Replacing x_i by $-1/p_i$ in Example 1.14, we obtain the desired result.

4.1.7. Let S_i be those lists in which c_i is adjacent to c_i. Consider a list in $S_{i_1} \cap \cdots \cap S_{i_r}$. Using the hint, this can be thought of as a list made from $2m - r$ symbols, where for the present we regard the two occurrences of the symbol c_i as different Since the list is a rearrangement of the symbols, there are $(2m - r)!$ such lists. However, $m - r$ pairs of the symbols are identical and we have treated them as different. There are 2^{m-r} ways to treat such symbols as different. Thus $N_r = \binom{m}{r}(2m - r)!/2^{m-r}$.

4.1.9. The proof is practically the same as that given for Theorem 4.1. Instead of asking how much $s \in S$ contributes to the sums, ask how much $\Pr(s)$ contributes.

4.1.11 (a) The products are 1 or 0 according as s belongs to precisely the sets S_i, $i \in K$ or not. Thus the inner sum is 1 or 0 according as s belongs to precisely k sets or not.

(b) Simply expand using the distributive law as in the previous exercise.

(c) The first part is just a rearrangement: Instead of choosing K and then J, first choose L (corresponding to $J \cup K$) and then choose K. The second part arises because there are $\binom{|L|}{k}$ ways to choose K.

(d) Move the sum over $s \in S$ inside the other sums and collect terms according to $|L|$.

4.1.13 (a) Let the notation be as in the proof of the Principle of Inclusion and Exclusion. The proof given in the text is easily adjusted to prove s contributes exactly $c_{t-1}(X)$ to $\sum_{i=0}^{t-1}(-1)^i S_i$. Thus the sum will be a lower bound when t is even and an upper bound when t is odd. Including the term $(-1)^t S_t$ in the sum changes upper bounds to lower bounds and vice versa since we are now considering $c_t(X)$. By considering the cases of t even and t odd separately, it is easy to see that the inequalities follow.

(b) This can be proved by induction on t using $\binom{|X|}{t} = \binom{|X|-1}{t} + \binom{|X|-1}{t-1}$.

4.1.15 (a) Let $m = 2$. Initially the N array contains

$$2: \quad N_2 \qquad 1: \quad N_1 \qquad 0: \quad N_0.$$

With $j = 0$, we do $i = 1$ and then $i = 0$. The N array now contains

$$2: \quad N_2 \qquad 1: \quad N_1 - N_2 \qquad 0: \quad N_0 - (N_1 - N_2).$$

With $j = 1$, we obtain

$$2: \quad N_2 \qquad 1: \quad (N_1 - N_2) - N2 \qquad 0: \quad N_0 - (N_1 - N_2).$$

Equation (4.16) gives

$$E_2 \;=\; N_2 \qquad E_1 \;=\; N_1 - 2N_2 \qquad E_0 \;=\; N_0 - N_1 + N_2,$$

which agrees with the values computed by the algorithm. You can carry out similar calculations for $m = 3$.

(b) This can be done by carefully carrying out the steps in the algorithm.

(c) After no iterations (that is, at the start of the algorithm), N_r contains s as many times as there is set of r indices for which (4.17) is true. If s appears in exactly p of the S_i, this number is $\binom{p}{r}$. We now use induction on t, having done the case $t = 0$. After $t - 1$ iterations, formula (4.18) is true when t is replaced by anything smaller in it. In particular, it holds with t replaced by $t - 1$.

We must now focus on the inner loop of the algorithm. What does it do? Since N_m never changes, neither does N_m^*. Formula (4.18) gives 0 or 1 for all t according as $p < m$ or $p = m$ ($p > m$ is impossible). This is the correct answer for both N_m and E_m.

Back to the action of the inner loop. Again we can prove it by induction, but now we are going from N_m^* down to N_0^*. We dealt with N_m^* in the previous paragraph. If the inner loop has done the correct thing with N_{r+1}^*, then the number of times s appears in the new version of N_r^* is $\mu(p, r, t-1) - \mu(p, r+1, t)$. There are various cases to consider. We'll just look at one, namely $\binom{p-(t-1)}{r-(t-1)} - \binom{p-t}{(r+1)-t}$. Using $\binom{n+1}{k} \;=\; \binom{n}{k} + \binom{n}{k-1}$, we have

$$\binom{p-(t-1)}{r-(t-1)} - \binom{p-t}{(r+1)-t} \;=\; \binom{p-t+1}{r-t+1} - \binom{p-t}{r-t+1} \;=\; \binom{p-t}{r-t},$$

which is what we needed to prove. We leave the other cases in (4.18) to you. The last sentence in the exercise follows from the fact that all the numbers we calculate are nonnegative. (This takes care of the problem of how we should interpret the multiset difference $A - B$ if s appears more often in B than it does in A.)

When $t \geq m$, the only time the binomial coefficient is used in (4.18) is when $t = p = m$ and it then has the value $\binom{0}{r-m}$, which is zero unless $r = m$, when it is 1. Thus, for $t \geq m$, $\mu(p, r, t)$ equals 1 if $r = p$ and 0 otherwise. Hence N_r^* is a set containing precisely those elements that are in exactly r of the S_i.

(d) This is implicit in the proof for (c).

4.1.17. Let S_i be the subset of $\underline{n}$ divisible by a_i. Then, N_t is the sum over all t-subsets T of A of $\lfloor n/\mathrm{lcm}(T)\rfloor$, where $\mathrm{lcm}(T)$ is the least common multiple of the elements of T and the floor $\lfloor x\rfloor$ is the largest integer not exceeding x. For the various parts of the exercise you need the following.

- If all elements of A divide n, then $\lfloor n/\mathrm{lcm}(T)\rfloor = n/\mathrm{lcm}(T)$.
- If no two elements of A have a common factor, then $\mathrm{lcm}(T) = \prod_{i\in T} i$.

(a) Using the previous comments in the special case $k = 0$, we obtain after some algebra

$$\sum_{i=0}^{m}(-1)^i N_i \;=\; n\prod_{a\in A}\left(1-\frac{1}{a}\right),$$

which is the Euler phi function when A is the set of prime divisors of n.

(b) The comment for (a) applies in this case as well.

(c) There is no simple formula even when $k = 0$ because the floor function cannot be eliminated.

(d) Now we cannot even eliminate the lcm function.

4.1.19. In all cases, what we must do is prove that (P-1), (P-2) and (P-3) hold. We omit most of them.

(d) Since $x/x = 1$, (P-1) is true. Suppose that $x\rho y$ and $y\rho x$. Then x/y and y/x are both integers. Since $(x/y)(y/x) = 1$, the only possible integer values for x/y and y/x are ± 1. Since x and y are positive, it follows that $x/y = 1$ and so (P-2) is true. Suppose that x/y and y/z are integers. Then so is x/z and so (P-3) is true.

4.1.21. Since every set is the union of itself, $x\rho x$. Suppose $x\rho y$ and $y\rho x$. Let b_y be a block of y. Since $x\rho y$, $b_x \subseteq b_y$ for some block b_x of x. Since $y\rho x$, $b'_y \subseteq b_x$ for some block b'_y of y. Since blocks of a partition are either equal or disjoint and since $b'_y \subseteq b_x \subseteq b_y$, we have $b'_y = b_y$ and so $b_x = b_y$. This proves that every block of y is a block of x. Hence $x = y$ and so (P-2) is true. It is easy to prove (P-3).

4.1.23 (a) With each element $s \in S$, associate a set $g(s)$ such that $s \in S_i$ if and only if $i \in g(s)$. Then E_k counts those $s \in S$ for which $|g(s)| = k$. Since the number of $s \in S$ with $g(s) = y$ is $e(y)$, the sum of $e(y)$ over $|y| = k$ also counts those s.

(b) An element s is counted in (4.14) if and only if it belongs to all S_i for which $i \in x$. This is the same as the definition of the set intersection.

(c) The sum of $e(x)$ over all x of since k is E_k. Putting this together with (4.15), we have

$$E_k \;=\; \sum_{|x|=k}\sum_{y\supseteq x}(-1)^{|y|-k}f(y) \;=\; \sum_{|y|\ge k}\sum_{\substack{x\subseteq y\\ |x|=k}}(-1)^{|y|-k}f(y) \;=\; \sum_{|y|\ge k}\binom{|y|}{k}(-1)^{|y|-k}f(y).$$

The sum of $f(y)$ in (b) over all y of size t is N_t. Collecting terms according to $|y|$, we have

$$E_k \;=\; \sum_{t=k}^{m}\binom{t}{k}(-1)^{t-k}N_t \;=\; \sum_{i=0}^{m-k}\binom{i+k}{k}(-1)^i N_{k+i},$$

where we set $t = i + k$. Now use $\binom{i+k}{k} \;=\; \binom{k+i}{i}$.

Section 4.2

4.2.1. The number of 6-long sequences made with B, R and W is $3^6 = 729$, which is much too long. The number of 6-long sequences in which adjacent beads differ in color is $3 \times 2^5 = 96$, which is more manageable, but still quite long. We won't list them. We could "cheat" by being a bit less mechanical: If the necklace contains a B, we could start with it. There are $2^5 = 32$ such necklaces, a manageable number. The only necklace without B must alternate R and W, so there is only one of them. Here are the 32 other necklaces, where a number preceding a necklace is the first place it appears in the list when considered circularly or flipped over. A zero means it was rejected because the first and last beads are the same.

1: BRBRBR	2: BRBRBW	0: BRBRWB	3: BRBRWR	2: BRBWBR	4: BRBWBW
0: BRBWRB	5: BRBWRW	0: BRWBRB	6: BRWBRW	0: BRWBWB	7: BRWBWR
3: BRWRBR	8: BRWRBW	0: BRWRWB	9: BRWRWR	2: BWBRBR	4: BWBRBW
0: BWBRWB	8: BWBRWR	4: BWBWBR	10: BWBWBW	0: BWBWRB	11: BWBWRW
0: BWRBRB	7: BWRBRW	0: BWRBWB	7: BWRBWR	5: BWRWBR	11: BWRWBW
0: BWRWRB	12: BWRWRW				

4.2.3 (a) Since 4 beads are used, at most 4 different kinds of beads are used. We can construct an arrangement of beads by choosing the number of types that must appear (1, 2, 3 OR 4), choosing that many types of beads from the r types AND then choosing an arrangement using all of the types of beads that we chose.

(b) Trivially, $f(1) = 1$. For $f(2)$, our decision will be the number of beads of the first type that appear. After that, it is easy. This gives us $1 + 2 + 1 = 4$. For $f(3)$, our decision will be which bead appears twice. This gives us $3 \times 2 = 6$ For $f(4)$, each bead appears once and there are 3 possibilities. Thus

$$F(r) = \binom{r}{1} + \binom{r}{2}4 + \binom{r}{3}6 + \binom{r}{4}3,$$

which can be rewritten as $r(r+1)(r^2+r^2)/8$, if desired.

4.2.5. The problem can be solved by either decision tree method. It is useful to note that all solutions must begin with h because any board that starts with v can be flipped about a NW-SE (135°) diagonal to give one that starts with h. Also note that a lexically least sequence that starts with hv determines the entire sequence. (To see this, note that it starts hvv and look at rotations of the board.)

We will use the second method. Our first decision will be the number of entire rows and/or columns that are covered by two whole dominoes. For example, two dominoes in the top row or two dominoes in the third column. Note that we cannot simultaneously cover a row and a column because they overlap. Let the number be L. The possible values of L are 0, 1, 2 and 4. (You should find it easy to see why $L = 3$ is impossible.) Note that we can always use the symmetries to make the first domino horizontal. For $L = 4$, there is obviously only one solution and its lex minimal form is *hhhhhhhh*. For $L = 0$, we use Method 1 to obtain *hvvhvvhh* as the the only solution. (Beware: reading the sequence in reverse does not correspond to a symmetry of the board.) For $L = 1$, we note that the entire row or column must be at the edge of the board. Suppose it is the first row. Refer back to Figure 3.15 to see that the only way to complete the board without increasing L is *hvvvvh*. This is already lex minimal: *hhhvvvvh*. Suppose $L = 2$. By rotation, we can assume we have two full rows and, because they cannot be in the middle, one of them is the first row. Again, refer to Figure 3.15 to find how many ways we can complete the board with one more horizontal row. This leads to six solutions: *hhhhhvvh*, *hhhvvhhh*, *hhhhvhvh*, *hhvhvhhh*, *hhhhvvvv* and *hhvvvvhh*. This gives a total of nine solutions.

4.2.7. When we write out our answers, they will be in the form suggested in the problem, without the surrounding boxes. To obtain the lex least solutions, we must linearly order the faces. Our order will be the line of four side faces from left to right, then the top and, finally, the bottom. We use B, R and W to denote the colors. and b, r and w to denote the number of faces of each color.

(a) Our first decision will be the number of black faces. By interchanging black and white, a solution with b black faces can be converted to one with $6-b$, so we only need look at $b = 0$ 1, 2 and 3. For $b = 0$ and $b = 1$, there are obviously only one solution. For $b = 2$, we must decide whether to put the second black face adjacent or opposite the first one. Here are the 4 solutions for $b < 3$.

W	W	W	W
WWWW	BWWW	BBWW	BWBW
W	W	W	W

For $b = 3$, our second decision is whether or not all three black faces share a common vertex. This leads to just 2 solutions:

B	W
BBWW	BBBW
W	W

Doubling the answers for $b < 3$ to get those for $b > 3$ gives us 10 solutions.

(b) In the previous solution, we can limit ourselves to $b \le 3$. When $b = 3$, we need to check whether or not one solution is converted to the other when black and white are interchanged. They are not, so $b = 3$ still gives 2 solutions for a total of 6.

(c) The mirror image of each of the 10 solutions is equivalent to itself, so there are still 10 solutions.

(d) Our first decision will be the list b, r, w. By interchanging colors, we need only consider the situations where $b \le r \le w$. This gives us 1,1,4, 1,2,3 and 2,2,2. Interchanging colors in all possible ways gives rise to 3, 6 and 1 solutions, respectively, for each solution found. For 1,1,4, our decision will be whether B and R are on adjacent or opposite faces. Each leads to one coloring. For 1,2,3 our first decision will be the number of R's that are adjacent to the B. One adjacency gives 1 solution and two give 2 solutions, depending on whether the R's are adjacent or opposite each other. For 2,2,2, our first decision will be whether or not the B's are adjacent or opposite. Our second decision will be whether or not the R's are adjacent or opposite. Each choice leads to 1 solution except when the B's are adjacent and the R's are adjacent. In this case there are more solutions. One possibility is to have the 4 sides be BBRR. Another possibility is to have the 4 sides be BBRW and then place the additional R on either the top or the bottom. These last two possibilities are mirror images of each other, but we cannot transform one to the other with just rotations. The solutions are given in Figure S.4.1. This gives us $2 \times 3 + 3 \times 6 + 6 = 30$ solutions.

(e) If all 3 colors appear, there are 30 solutions. If only 1 color appears, there are obviously 3 solutions. What if exactly 2 colors appear, we can first choose the 2 colors AND then use them. By the first part of this exercise, there are $10 - 2 = 8$ ways to use the colors so that both appear. Thus we have $30 + 3 + \binom{3}{2}8 = 57$ solutions.

(f) Note that no color can appear more than 3 times on any given cube. Also note that at most 6 colors appear on any given cube. By looking over our previous work, we find, in the notation of Exercise 4.2.4, that $f(0) = f(1) = 0$, $f(2) = 1$ and $f(3) = 8$. By looking at decision trees for the color counts 1,1,1,3 and 1,1,2,2, we find that $f(4) = \binom{4}{1}2 + \binom{4}{2}5 = 32$. Consider $f(5)$ which has just the one color count list 1,1,1,1,2. There is one way to place the repeated colors. The

1,1,4	W B R W W W	W B W R W W	
1,2,3	W B R R W W	R B R W W W	W B R W R W
2,2,2	R B B W W R	R B R B W W	W B R B R W
	W B B R R W	R B B R W W	W B B R W R

Figure S.4.1 The distinct painted cubes with various numbers of faces painted Black, Red and White.

partially colored cube can be transformed into itself be leaving it fixed or by rotating it so that the two colored faces are interchanged. This means that whenever we color the remaining 4 faces with 4 *distinct* colors, there will be exactly one other coloring that is equivalent to it. Thus $f(5) = \binom{5}{1}(4!/2) = 60$. If you experiment a bit, you will discover that there are 24 symmetries of the cube. If all the faces are colored differently, each of the symmetries leads to an equivalent coloring that looks different. Thus $f(6) = 6!/24 = 30$. Putting all this together, we have

$$F(r) = \binom{r}{2} + \binom{r}{3}8 + \binom{r}{4}32 + \binom{r}{5}60 + \binom{r}{6}30.$$

Section 4.3

4.3.1. The image of F is all k element subsets of $\underline{n}$. $F^{-1}(x)$ consists of all possible ways to arrange the elements of x in a list. Since we are able to count lists, we know that there are $k!$ such arrangements. We also know that $|A| = n!/(n-k)!$. Thus the coimage of F consists of $C(n,k)$ blocks all of size $k!$ and the union of these blocks has $n!/(n-k)!$ elements. Thus $C(n,k) = \frac{n!}{k!(n-k)!}$.

4.3.3. Note that $N(\gamma) = 0$ unless $\gamma \in P_8$ or $\gamma \in P_5$. In the former case, $N(\gamma) = \binom{8}{3} = 56$ and in the latter case, $N(\gamma) = \binom{2}{1}\binom{3}{1} = 6$. Thus there are $(56 + 4 \times 6)/16 = 5$ necklaces.

4.3.5 (a) The second line consists of the first line circularly shifted by c, an integer between 0 and $n-1$; i.e., the second line is $s_1, s_2, \ldots, s_n$, where $s_t = c + t$ if this is at most n and $c + t - n$, otherwise.

(b) In addition to the elements of the cyclic group, we have permutations whose second lines are cyclic shifts of $n, \ldots, 2, 1$.

(c) There are 0, 1 or 2 cycles of length 1 and the remaining cycles are all of length 2. If n is odd, there is always exactly one cycle of length 1. If n is even, there is never exactly one cycle of length 1. You can write down the cycles as follows. All numbers that are mentioned are understood to have an appropriate multiple of n added to (or subtracted from) them so that they lie between 1 and n inclusive. If n is odd, choose a cycle (k). The remaining cycles are

$(k-t, k+t)$ where $1 \le t < n/2$. If n is even, choose $k \le n/2$. There are two ways to proceed. First, we could have all cycles of the form $(k-t+1, k+t)$ where $1 \le t \le n/2$. Second, we could have (k), $(k+n/2)$ and all cycles of the form $(k-t, k+t)$ where $1 \le t < n/2$.

4.3.7. The proof in the text shows that the right side of the given equality is $|G| \sum_{g \in G} N(g)$. By (4.20), the left side is

$$\sum_{y \in S} |I_x| \;=\; |G| \sum_{y \in S} \frac{1}{|B_y|}.$$

The rest of the proof follows easily by adapting what was done in the text. This seems to be a shorter proof than the one in the text. Why didn't we use it? First, it's not particularly shorter; however, it is a bit cleaner. Unfortunately, it requires starting with the completely unmotivated double summation in which we have interchanged the order of the sums.

Section 5.1

5.1.1. The sum is the number of ends of edges since, if x and y are the ends of an edge, the edge contributes 1 to the value of $d(x)$ and 1 to the value of $d(y)$. Since each edge has two ends, the sum is twice the number of edges.

5.1.3. The graph with

$$\varphi \;=\; \begin{pmatrix} a & b & c & d & e & f & g & h & i & j & k \\ C & C & F & A & H & E & E & A & D & A & A \\ C & G & G & H & H & H & F & H & G & D & F \end{pmatrix}.$$

is isomorphic to Q. The correspondence between vertices is given by

$$\begin{pmatrix} A & B & C & D & E & F & G & H \\ H & A & C & E & F & D & G & B \end{pmatrix}$$

where the top row corresponds to the vertices of Q. The graph with

$$E \;=\; \{1,2,3,4,5,6,7,8,9,10,11\} \quad \text{and} \quad \varphi \;=\; \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ A & E & E & E & F & G & H & B & C & D & E \\ G & H & E & F & G & H & B & C & D & D & H \end{pmatrix}.$$

is not isomorphic to Q. One edge needs to be deleted from $P'(Q)$ and one added.

5.1.5 (a) There is no graph Q with degree sequence $(1,1,2,3,3,5)$ since the sum of the degrees is odd.

(b) There are such a graph. You should draw an example.

(c) Up to labeling, the graph is unique. Take $V = \{1,\ldots,6\}$ and

$$E \;=\; \big\{\{1,6\},\,\{2,6\},\,\{2,4\},\,\{3,6\},\,\{3,5\},\,\{4,6\},\,\{4,5\},\,\{5,6\}\big\}$$

(d) A graph with degree sequence $(3,3,3,3)$ has $(3+3+3+3)/2 = 6$ edges and, of course 4 vertices. That is the maximum $\binom{4}{2}$ of edges that a graph with 4 vertices can have. It is easy to construct such a graph. This graph is called the *complete* graph on 4 vertices.

(f) There is no simple graph (or graph without loops or parallel edges) with degree sequence $(3,3,3,5)$.

(g) Similar arguments to the $(3,3,3,3)$ case apply to the complete graph with degree sequence $(4,4,4,4,4)$.

A	B	all	injections	surjections
L	L	b^a	$b(b-1)\cdots(b-a+1)$	$b!S(a,b)$
L	U	$\sum_{k\le b} S(a,k)$	1	$S(a,b)$
U	L	$\binom{a+b-1}{a}$	$\binom{b}{a}$	$\binom{a-1}{a-b}$
U	U	$\sum_{k\le b} p(a,k)$	1	$p(a,b)$

Figure S.5.1 Some basic enumeration problems.

Section 5.2

5.2.1. Let ν and ε be the bijections.

(a) This follows from the fact that ν and ε are bijections.

(b) This can be seen intuitively from the drawing of the unlabeled graph. If you want a more formal proof, first note that the degree of a vertex v is the number of edges e such that $v \in \varphi(e)$. Now use the fact that $v \in \varphi(e)$ is equivalent to $\nu(v) \in \varphi'(\varepsilon(e))$.

5.2.3 (a) This is exactly like the next problem with the transpose, ^{t}, replaced by inverse, $^{-1}$, everywhere.

(b) Let I be the $n \times n$ identity matrix. Since $A = IAI^t$, $A \simeq A$. Suppose that $A \simeq B$. Then $B = PAP^t$ for some nonsingular P. Multiplying on the left by P^{-1} and on the right by $(P^{-1})^t = (P^t)^{-1}$, we have

$$(P^{-1})B(P^{-1})^t \;=\; (P^{-1}P)A(P^t(P^{-1})^t) \;=\; (P^{-1}P)A(P^t(P^t)^{-1}) = A.$$

Thus $B \simeq A$. Suppose that $A \simeq B \simeq C$. Then we have nonsingular P and Q such that $B = PAP^t$ and $C = QBQ^t$. Thus $C = Q(PAP^t)Q^t = (QP)A(P^tQ^t) = (QP)A(QP)^t$. This proves transitivity.

5.2.5. Let $E \in \mathcal{P}_2(V)$ and $E' \in \mathcal{P}_2(V')$. Write $G = (V,E) \simeq (V',E') = G'$ if and only if there is a bijection $\nu: V \to V'$ such that $\{u,v\} \in E$ if and only if $\{\nu(u),\nu(v)\} \in V'$.

We could show that this is an equivalence relation by adapting the proof in Example 5.5. An alternative is to show how this definition leads to the equivalence relation for G and G' interpreted as graphs. We'll take this approach. In this case φ and φ' are identity maps. Define $\varepsilon(\{u,v\}) = \{\nu(u),\nu(v)\}$. By our definition in the previous paragraph, $\varepsilon: E \to E'$ is a bijection. Since φ and φ' are the identity, the requirement that $\varphi'(\varepsilon(e)) = \nu(\varphi(e))$ in the definition of graph isomorphism is satisfied.

5.2.7. The table is shown in Figure S.5.1. The entries which are 1 follow when you realize what is being counted. The LL row corresponds to ordered samples and the UL row to unordered samples, which have been considered in Chapter 1. The UL-surjection entry comes from the realization that our sample allows repetition but must include every element in b so that we are only free to choose $a-b$ additional elements. In the LU row, the fact that the range is unlabeled means that we can only distinguish functions that have different coimages. The UU row is associated with partitions of numbers. We use $p(n,k)$ to denote the number of partitions of n having exactly k parts.

Section 5.3

5.3.1. Since $E \subseteq \mathcal{P}_2(V)$, we have a simple graph. Regardless of whether you are in set C or S, following an edge takes you into the other set. Thus, following a path with an odd number of edges takes you to the opposite set from where you started while a path with an even number of edges takes you back to your starting set. Since a cycle returns to its starting vertex, it obviously returns to its starting set.

5.3.3 (a) Let $e = \{u, v\}$ and let $f = \{v, w\}$ be the other edge. Since G is simple, $u \neq w$. Since e is a cut edge, u and v are in separate components of $(V, E - \{e\})$. Thus so are u and w. Since the graph induced by $V - \{v\}$ is a subgraph of $(V, E - \{e\})$, u and w are in separate components of it as well.

(b) Take two triangles and identify their tops. The merged top is a cut vertex but the graph has no isthmus.

(c) We will prove that e is a cut edge if and only if its ends u and v, say, lie in different components of $G' = (V, E - \{e\})$. The result will then follow because, first, if C is a cycle containing e, removal of e does not leave its ends in different components, and, second, if u and v are in the same components of G', then there is a path P connecting them in G' and P and e form a cycle in G.

Now back to the original claim. If u and v are in different components of G', then e is a cut edge. Suppose e is a cut edge of G. Since G is connected and every path in G that is not a path in G' contains e, it follows that if x and y are in different components of G' any path connecting them in G contains e. Let P be such a path and let u be the end of e first reached on P when starting from x. It follows that x and u are in one component of G' and that y and v (the other end of e) are one component, too. Since x and y are in different components, so are u and v.

(d) We claim that $v \in V$ is a cut vertex of G if and only if there are two edges e and e' both containing v such that no cycle of G contains both e and e'.

Proof. Suppose that v is a cut vertex. Let x and y belong to different components of the graph G'' induced by $V - \{v\}$. Any path from x to y in G must include v. Let P be such a path and let e and e' be the two edges in P that contain v. If e and e' were on a cycle C in G, then we could remove e and e' from P and add on $C - \{e, e'\}$ to obtain a route from x to y that does not go through v. Since this contradicts the fact that x and y are in different components of G", it follows that e and e' do not lie in a cycle.

The steps can be reversed to prove that if e and e' are edges incident with v that do not lie on a cycle, then v is a cut vertex: Let x and y be the other vertices on e and e'. Since e and e' do not lie on a cycle, every path from x to y must include either e or e' (or both), and hence includes v. Since there is no path from x to y not including v, they are in different components of G".

5.3.5 (a) The graph is not Eulerian. The longest trail has 5 edges, the longest circuit has 4 edges.

(b) The longest trail has 9 edges, the longest circuit has 8 edges.

(c) The longest trail has 13 edges (an Eulerian trail starting at C and ending at D). The longest circuit has 12 edges.

(d) This graph has an Eulerian circuit (12 edges).

Section 5.4

5.4.1. We first prove that (b) and (c) are equivalent. We do this by showing that the negation of (b) and the negation of (c) are equivalent. Suppose $u \neq v$ are on a cycle of G. By Theorem 5.3, there are two paths from u to v. Conversely, suppose there are two paths from u to v. Call them $u = x_0, x_1, \ldots, x_k = v$ and $u = y_0, y_1, \ldots, y_m = v$. Let i be the smallest index such that $x_i \neq y_i$. We may assume that $i = 1$ for, if not, redefine $u = x_{i-1}$. On the new paths, let $x_a = y_b$ be the smallest $a > 0$ for which some x_j is on the y path. The walk

$$u = x_0,\ x_1, \ldots, x_a = y_b,\ y_{b-1}, \ldots, y_0 = u$$

has no repeated vertices except the first and last and so is a cycle. (A picture may help you visualize what is going on. Draw the x path intersecting the y path several times.)

We now prove that (d) implies (b). Suppose that G has a cycle, $v_0, v_1, \ldots, v_k, v_0$. Remove the edge $\{v_0, v_k\}$. In any walk that uses that edge, replace it with the path $v_0, v_1, \ldots, v_k$ or its reverse, as appropriate. Thus the graph is still connected and so the edge $\{v_0, v_k\}$ contradicts (d).

5.4.3 (a) By Exercise 5.1.1, we have $\sum_{v\in V} d(v) = 2|E|$. By 5.4(e), $|E| = |V| - 1$. Since

$$2|V| = \sum_{v\in V} 2, \quad \text{we have} \quad 2 = 2|V| - 2|E| = \sum_{v\in V} (2 - d(v)).$$

(b) We give three solutions. The first uses the previous result. The second uses the fact that each tree except the single vertex has at least two leaves. The third uses the fact that trees have no cycles.

Suppose that T is more than just a single vertex. Since T is connected, $d(v) \neq 0$ for all v. Let n_k be the number of vertices of T of degree k. By the previous result, $\sum_{k\geq 1}(2-k)n_k = 2$. Rearranging gives $n_1 = 2 + \sum_{k\geq 2}(k-2)n_k$. If $n_m \geq 1$, the sum is at least $m - 2$.

For the second solution, remove the vertex of degree m to obtain m separate trees. Each tree is either a single vertex, which is a leaf of the original tree, or has at least two leaves, one of which must be a leaf of the original tree.

For the third solution, let v be the vertex of degree m and let $\{v, x_i\}$ be the edges containing v. Each path starting v, x_i must eventually reach a leaf since there are no cycles. Call the leaf y_i. These leaves are distinct since, if $y_i = y_j$, the walk $v, x_i, \ldots, y_i = y_j, \ldots, x_j, v$ would lead to a cycle.

(c) Let the vertices be u and v_i for $1 \leq i \leq m$. Let the edges be $\{u, v_i\}$ for $1 \leq i \leq m$.

(d) Let $N = n_3 + n_4 + \cdots$, the number of vertices of degree 3 or greater. Note that $k - 2 \geq 1$ for $k \geq 3$. By our earlier formula, $n_1 \geq 2 + N$. If $n_2 = 0$, $N = |V| - n_1$ and so we have $n_1 \geq 2 + |V| - n_1$. Thus $n_1 \geq 1 + |V|/2$. Similarly, if $n_2 = 1$, $N = |V| - n_1 - 1$ and, with a bit of algebra, $n_1 \geq (1 + |V|)/2$.

(e) A careful analysis of the previous argument shows that the number of leaves will be closest to $|V|/2$ if we avoid vertices with high degrees. Thus we will try to make our vertices of degree three or less. We will construct some RP-trees, T_k with k leaves. Let T_1 the isolated vertex. For $k > 1$, let T_k have two children, one a single vertex and the other the root of T_{k-1}. Clearly T_k has one more leaf and one more nonleaf than T_{k-1}. Thus the difference between the number of leaves and nonleaves is the same for all T_k. For T_1 it is one.

5.4.5. Since the tree has at least 3 vertices, it has at least $3 - 1 = 2$ edges. Let $e = \{u, v\}$ be an edge. Since there is another edge and a tree is connected, at least one of u and v must lie on another edge besides e. Suppose that u does. It is fairly easy to see that u is a cut vertex and that e is a cut edge.

5.4.7 (a) The idea is that for a rooted planar tree of height h, having at most 2 children for each non-leaf, the tree with the most leaves occurs when each non-leaf vertex has exactly 2 children. You should sketch some cases and make sure you understand this point. For this case $l = 2^h$ and so $\log_2(l) = h$. Any other rooted planar tree of height h, having most 2 children for each non-leaf, is a subtree (with the same root) of this maximal-leaf binary tree and thus has fewer leaves.

(b) The height h can be arbitrarily large.

(c) $h = l - 1$.

(d) $\lceil \log_2(l) \rceil$ is a lower bound for the height of *any* binary tree with l leaves. It is easy to see that you can construct a full binary tree with l leaves and height $\lceil \log_2(l) \rceil$.

(e) $\lceil \log_2(l) \rceil$ is the minimal height of a binary tree.

5.4.9 (a) A binary tree with 35 leaves and height 100 is possible.

(b) A full binary tree with 21 leaves can have height at most 20. So such a tree of height 21 is impossible.

(c) A binary tree of height 5 can have at most 32 leaves. So one with 33 leaves is impossible.

(d) A full binary tree with 65 leaves has minimal height $\lceil \log_2(65) \rceil = 7$. Thus a full binary tree with 65 leaves and height 6 is impossible.

5.4.11 (a) Breadth-first: $MIAJKCEHLBFGD$,
Depth-first: $MICIEIHFHGHDHIMAMJMKLKBKM$,
Pre-order: $MICEHFGDAJKLB$,
Post-order: $CEFGDHIAJLBKM$.

(b) The tree is the same as in part (a), reflected about the vertical axis, with vertices A and J removed.

(c) It is not possible to reconstruct a rooted plane tree given just its pre-order vertex list. A counterexample can be found using just three vertices.

(d) It is possible to reconstruct a rooted plane tree given its pre-order and post-order vertex list. If the root is X and the first child of the root is Y, it is possible to reconstruct the pre-order and post-order vertex lists of the subtree rooted at Y from the pre-order and post-order vertex lists of the tree. In the same manner, you can reconstruct the pre-order and post-order vertex lists of the subtrees rooted at the other children of the root X. Now do the same trick on these subtrees. Try this approach on an example.

Section 5.5

5.5.1. Let D be the domain suggested in the hint and define $f\colon D \to \mathcal{P}_2(V)$ by $f((x,y)) = \{x,y\}$. Let $G(D) = (V,\psi)$ where $\psi(e) = f(\varphi(e))$.

5.5.3. Let $V = \{u,v\}$ and $E = \{(u,v),(v,u)\}$.

5.5.5. You can use the notation and proof of Example 5.5 provided you change all references to two element sets to references to ordered pairs. This means replacing $\{x,y\}$ with (x,y), $\{\nu(x),\nu(y)\}$ with $(\nu(x),\nu(y))$ and $\mathcal{P}_2(V_i)$ with $V_i \times V_i$.

5.5.7. "The statements are all equivalent" means that, given any two statements v and w, we have a proof that v implies w. Suppose D is strongly connected. Then there is a directed path $v = v_1, v_2, \ldots, v_k = w$. That means we have proved v_1 implies v_2, that v_2 implies v_3 and so on. Hence v_1 implies v_k.

5.5.9. Let $e = (u_1,u_2)$. For $i = 2,3,\ldots$, as long as $u_i \neq u_1$ choose an edge (u_i,u_{i+1}) that has not be used so far. It is not hard to see that $d_{\text{in}}(u_i) = d_{\text{out}}(u_i)$ implies this can be done. In this way we obtain a directed trail starting and ending at u_1. This may not be a cycle, but a cycle containing e can be extracted from it by deleting some edges.

5.5.11 (a) It's easy to see this pictorially: Suppose there were an isthmus $e = \{u,v\}$. Then G consists of the edge e, a graph G_1 containing u, and another graph G_2 containing v. Suppose e is directed as (u,v). Clearly one can get from G_1 to G_2 but one cannot get back along directed edges, contradicting strongly connectedness.

Here is a more formal proof. Suppose there were such a path, say $v = v_1, v_2, \ldots, v_k = u$ It does not contain the directed edge e (since e goes in the wrong direction). Now look at the original undirected graph. We claim removal of $\{u,v\}$ does not disconnect it. The only problem would be a path that used $\{u,v\}$ to get from, say x to y, say $x,\ldots,x',u,v,y',\ldots,y$. The walk $x,\ldots,x',v_k,\ldots,v_2,v_1,y',\ldots,y$ connects u and v without using the edge $\{u,v\}$.

(b) See Exercise 6.3.14 (p. 170).

5.5.13 (a) For all $x \in S$, $x|x$. For all $x,y \in S$, if $x|y$ and $x \neq y$, then y does not divide x. For all $x,y,z \in S$, $x|y$, $y|z$ implies that $x|z$.

(b) The covering relation

$$H = \{(2,4),(2,6),(2,10),(2,14),(3,6),(3,9),(3,15),(4,8),(4,12),(5,10),(5,15),(6,12),(7,14)\}.$$

5.5.15 (a) There are n^{n-2} trees. Since a tree with n vertices has $n-1$ edges, the answer is zero if $q \neq n-1$. If $q = n-1$, there are $\binom{\binom{n}{2}}{n-1}$ graphs. Thus the answer is $n^{n-2}\binom{\binom{n}{2}}{n-1}^{-1}$ when $q = n-1$.

(b) We have

$$\binom{\binom{n}{2}}{n-1} < \frac{\binom{n}{2}^{n-1}}{(n-1)!} = \frac{n^{n-1}(n-1)^{n-1}}{2^{n-1}\,(n-1)!} < \frac{n^{n-1}}{2^{n-1}/e^{n-1}} = \left(\frac{ne}{2}\right)^{n-1}.$$

Using this in the answer to (a) gives the result we want. It turns out that

$$n^{n-2}\binom{\binom{n}{2}}{n-1}^{-1} \sim \sqrt{\pi/2n}\,(2/e)^n,$$

which differs from our estimate by a constant times $n^{1/2}$.

Section 5.6

5.6.1. Let A and B be the partition of the vertices guaranteed by the definition of a bipartite graph. Let $k = |A|$, number the vertices in A with 1 to k and those is B with $k+1$ to n. Since no edges connect vertices in A to each other, $A(G)$ has a $k \times k$ block of zeroes in its upper left corner. Similarly B gives a block in the lower right corner.

5.6.3 (a) $a_{i,j}^{(k)}$ is the sum over all $t_1, \ldots, t_{k-1}$ of $a_{i,t_1} a_{t_1,t_2} \cdots a_{t_{k-1},j}$. Each of these products is 0 or 1, so the sum is nonzero if and only if some product is nonzero. This happens if and only if each factor in the product is nonzero. This happens if and only if the vertices $i, t_1, \ldots, t_{k-1}, j$ form a walk.

(b) We can construct a path from a walk by jumping over pieces that form cycles. Thus the shortest walk from i to j is a path. Here's a more formal argument. Suppose that $W = (i, t, \ldots, v, j)$ is the shortest walk from i to j. If it is not a path, then there must be repeated vertices in the list. Let u be such a vertex. Remove all vertices from the sequence after the first occurrence of u up to and including the last occurrence of u. The result is a shorter walk, contradicting the minimality of W.

(c) The obvious idea is to repeat the previous statement with $i = j$: "The shortest walk from i to i is a cycle." This is not true. If $\{i, j\}$ is an edge, then i, j, i is the shortest walk from i to j but it is not a cycle. The result would be true if we were looking at oriented simple graphs because an edge can be traversed in only one direction. All we can claim is that any odd length walk from i to i contains a cycle.

We can modify the situation a bit by looking at an edge $\{i, j\}$ of the graph. Let H be the graph obtained by removing it; i.e., by setting $a_{i,j} = a_{j,i} = 0$. The shortest walk from j to i in H together with the edge $\{i, j\}$ is a cycle of G. This follows from the previous result and the definitions of path and cycle.

(d) Following the hint, $B^k = \sum_{t=0}^{k} \binom{k}{t} B^t$ by the binomial theorem. Since $\binom{k}{t} > 0$, $b_{i,j}^{(k)}$ is nonzero if and only if $a_{i,j}^{(t)} \neq 0$ for some t with $0 \leq t \leq k$. $t = 0$ gives the identity matrix, so $b_{i,i}^{(k)} \neq 0$ for all k. For $i \neq j$, $b_{i,j}^{(k)} \neq 0$ if and only if there is a walk from i to j for some $t \leq k$, and thus if and only if there is a path for some $t \leq k$. Since paths of length t contain $t+1$ distinct vertices, no path is longer than $n-1$. Thus there is a path from i to $j \neq i$ if and only if $b_{i,j}^{(k)} \neq 0$ for all $k \geq n-1$.

5.6.5. We claim that $A(D)$ is nilpotent if and only if there is no vertex i such that there is a walk from i to i (except the trivial walk consisting of just i).

First suppose that there is a nontrivial walk from i to i containing k edges. Let $C = A(D)^k$. It follows that all entries of C are nonnegative and $c_{i,i} \neq 0$. Thus $c_{i,i}^{(m)} \neq 0$ for all $m > 0$. Hence $A(D)$ is not nilpotent.

Conversely, suppose that $A(D)$ is not nilpotent. Let n be the number of vertices in D and suppose that i and j are such that $a_{i,j}^{(n)} \neq 0$, which we can do since $A(D)$ is not nilpotent. There must be a walk $i = v_0, v_1, v_2, \ldots, v_n = j$. Since this sequence contains $n+1$ vertices, there must be a repeated vertex. Suppose that $k < l$ and $v_k = v_l$. The sequence $v_k, v_{k+1}, \ldots, v_l$ is a nontrivial walk from v_k to itself.

Section 6.1

6.1.1 (a) One description of a tree is: a connected graph such that removal of any edge disconnects the tree. Since an edge connects only two vertices, we will obtain only two components by removing it.

(b) Note that T with e removed and f added is a spanning tree. Since T has minimum weight, the result follows.

(c) The graph must have a cycle containing e. Since one end of e is in T_1 and the other in T_2, the cycle must contain another connector besides e.

(d) Since T^* with e removed and f added is a spanning tree, the algorithm would have removed f instead of e if $\lambda(f) > \lambda(e)$.

(e) By (b) and (d), $\lambda(f) = \lambda(e)$. Since adding f connects T_1 and T_2, the result is a spanning tree.

(f) Suppose T^* is not a minimum weight spanning tree. Let T be a minimum weight spanning tree so that the event in (a) occurs as late as possible. It was proven in (e) that we can replace T with another minimum weight spanning tree such that the disagreement between T and T^*, if any, occurs later in the algorithm. This contradicts the definition of T.

6.1.3 (b) Let Q_1 and Q_2 be two bicomponents of G, let v_1 be a vertex of Q_1, and let v_2 be a vertex of Q_2. Since G is connected, there is a path in G from v_1 to v_2, say $x_1, \ldots, x_p$. You should convince yourself that the following pseudocode constructs a walk $w_1, w_2, \ldots$ in $\mathcal{B}(G)$ from Q_1 to Q_2.

```
Set w_1 = Q_1, j=2, and k = 0.
While there is an x_i ∈ P(G) with i > k.
   Let i > k be the least i for which x_i ∈ P(G).
   If i = p
      Set Q = Q_2.
   Else
      Let Q be the bicomponent containing {x_i, x_{i+1}}.
   End if
   Set w_j = x_i, w_{j+1} = Q, k = i, and j = j + 2.
End while
```

(c) Suppose there is a cycle in $\mathcal{B}(G)$, say $v_1, Q_1, \ldots, v_k, Q_k, v_1$, where the Q_i are distinct bicomponents and the v_i are distinct vertices. Set $v_{k+1} = v_1$. By the definitions, there is a path in Q_i from v_i to v_{i+1}. Replace each Q_i in the previous cycle with these paths after removing the endpoints v_i and v_{i+1} from the paths. The result is a cycle in G. Since this is a cycle, all vertices on it lie in the same bicomponent, which is a contradiction since the original cycle contained more than one Q_i.

(d) Let v be an articulation point of the simple graph G. By definition, there are vertices x and y such that every path from x to y contains v. From this one can prove that there are edges $e = \{v, x'\}$ and $f = \{v, y'\}$ such that every path from x' to y' contains v. It follows that e and f are in different bicomponents. Thus v lies in more than one bicomponent.

Suppose that v lies in two bicomponents. There are edges $e = \{v, w\}$ and $f = \{v, z\}$ such that $e \not\sim f$. It follows that every path from w to z contains v and so v is an articulation point.

6.1.5 (a) Since there are no cycles, each component must be a tree. If a component has n_i vertices, then it has $n_i - 1$ edges since it is a tree. Since $\sum n_i$ over all components is n and $\sum (n_i - 1)$ over all components is k, $n - k$ is the number of components.

(b) By the previous part, H_{k+1} has one less component than G_k does. Thus at least one component C of H_{k+1} has vertices from two or more components of G_k. By the connectivity of C, there must be an edge e of C that joins vertices from different components of G_k. If this edge is added to G_k, no cycles arise.

(c) By the definition of the algorithm, it is clear that $\lambda(g_1) \le \lambda(e_1)$. Suppose that $\lambda(g_i) \le \lambda(e_i)$ for $1 \le i \le k$. By the previous part, there is some e_j with $1 \le j \le k+1$ such that G_k together with e_j has no cycles. By the definition of the algorithm, it follows that $\lambda(g_{k+1}) \le \lambda(e_j)$. Since $\lambda(e_j) \le \lambda(e_{k+1})$ by the definition of the e_i's, we are done.

6.1.7 (a) Hint: For (1) there are four spanning trees. For (2) there are 8 spanning trees. For (3) there are 16 spanning trees.

(b) Hint: For (1) there is one. For (2) there are two. For (3) there are two.

(c) Hint: For (1) there are two. For (2) there are four. For (3) there are 6.

(d) Hint: For (1) there are two. For (2) there are three. For (3) there are 6.

6.1.9 (a) Hint: There are 21 vertices, so the minimal spanning tree has 20 edges. Its weight is 30.

(b) Hint: Its weight is 30..

(c) Hint: Its weight is 30.

(d) Hint: Note that K is a the only vertex in common to the two bicomponents of this graph. Whenever this happens (two bicomponents, common vertex), the depth-first spanning tree rooted at that common vertex has exactly two "principal subtrees" at the root. In other words, the root of the depth-first spanning tree has degree two. Finding depth first spanning trees of minimal weight is, in general, difficult. You might try it on this example.

Section 6.2

6.2.1. This is just a matter of a little algebra.

6.2.3 (a) To color G, first color the vertices of H AND then color the vertices of K. By the Rule of Product, $P_G(x) = P_H(x)P_K(x)$.

(b) Let v be the common vertex. There is an obvious bijection between pairs of colorings (λ_H, λ_K) of H and K with $\lambda_H(v) = \lambda_K(v)$ and colorings of G. We claim the number of such pairs is $P_H(x)\big(P_K(x)/x\big)$. To see this, note that, in the colorings of K counted by $P_K(x)$, each of the x ways to color v occurs equally often and so $1/x$ of the colorings will have $\lambda_K(v)$ equal to the color given by $\lambda_H(v)$.

(c) The answer is $P_H(x)P_K(x)(x-1)/x$. We can prove this directly, but we can also use (b) and (6.4) as follows. Let $e = \{v, w\}$. By the construction of G, $P_{G-e}(x) = P_H(x)P_K(x)$. By (b), $P_{G_e}(x) = P_H(x)P_K(x)/x$. Now apply (6.4).

6.2.5. Let the solution be $P_n(x)$. Clearly $P_1(x) = x(x-1)$, so we may suppose that $n \ge 2$. Apply deletion and contraction to the edge $\{(1,1),(1,2)\}$. Deletion gives a ladder with two ends sticking out and so its chromatic polynomial is $(x-1)^2 P_{n-1}(x)$. Contraction gives a ladder with the contracted vertex joined to two adjacent vertices. Once the ladder is colored, there are $x-2$ ways to color the contracted vertex. Thus we have

$$P_n(x) \;=\; (x-1)^2 P_{n-1}(x) - (x-2)P_{n-1}(x) \;=\; (x^2 - 3x + 3)P_{n-1}(x).$$

The value for $P_n(x)$ now follows easily.

6.2.7. The answer is

$$x^8 - 12x^7 + 66x^6 - 214x^5 + 441x^4 - 572x^3 + 423x^2 - 133x.$$

There seems to be no really easy way to derive this. Here's one approach which makes use of Exercise 6.2.3 and $P_{Z_n}(x)$ for $n = 3, 4, 5$. Label the vertices reading around one face with a, b, c, d and around the opposite face with A, B, C, D so that $\{a, A\}$ is an edge, etc. If the edge $\{a, A\}$ is contracted, call the new vertex α. Introduce β, γ and δ similarly.

Let $e_1 = \{a, A\}$ and $e_2 = \{b, B\}$. Note that $G - e_1 - e_2$ consists of three squares joined by common edges and that $H = G_{e_1} - e_2$ is equivalent to $(G - e_1)_{e_2}$. We do H in the next paragraph. In $K = G_{e_1 e_2}$, let $f = \{\alpha, \beta\}$. $K - f$ is two triangles and a square joined by common edges and K_f is a square an a vertex v joined to the vertices of the square. By first coloring v and then the square, we see that $P_{K_f}(x) = xP_{Z_4}(x - 1)$.

Let $f_1 = \{c, C\}$, $f_2 = \{d, D\}$ and $f_3 = \{\beta, \gamma\}$. Then

- $H - f_1 - f_2$ is two Z_5's sharing β;
- $(H - f_1)_{f_2}$ is easy to do if you consider two cases depending on whether β and δ have the same or different colors, giving $x(x - 1)(x - 2)^4 + x(x - 1)^4$;
- $H_{f_1} - f_3$ is a Z_5 and a triangle with a common edge and
- $H_{f_1 f_3}$ are three triangles joined by common edges.

6.2.9. This can be done by induction on the number of edges. The starting situation involves some number n of vertices with no edges. Since the chromatic polynomial is x^n, the result is proved for the starting condition.

Now for the induction. Deletion does not change the number of vertices, but reduces the number of edges. By induction, it gives a polynomial for which the coefficient of x^k is a nonnegative multiple of $(-1)^{n-k}$. Contraction decreases both the number of vertices and the number of edges by 1 and so gives a polynomial for which the coefficient of x^k is a nonnegative multiple of $(-1)^{n-1-k}$. Subtracting the two polynomials gives one where the coefficient of x^k is an nonnegative multiple of $(-1)^{n-k}$.

Section 6.3

6.3.1. Every face must contain at least four edges and each side of an edge contributes to a face. Thus $4f \geq$ (edge sides) $= 2e$. From Euler's relation,

$$2 \;=\; v - e + f \;\geq\; v - e + e/2 \;=\; (2v - e)/2$$

and so $e \geq 2v - 4$.

6.3.3 (a) We have $2e = fd_f$ and $2e = vd_v$. Use this to eliminate v and f in Euler's relation.

(b) They are cycles.

(c) If $d_f \geq 4$ and $d_v \geq 4$, we would have $0 < 2/d_f + 2/d_v - 1 \leq 0$, a contradiction. Thus at least one of d_v and d_f is 3. Since $d_v \geq 3$, we have $2/d_v \leq 2/3$. Thus

$$0 \;<\; \frac{2}{d_f} + \frac{2}{d_v} - 1 \;\leq\; \frac{2}{d_f} - \frac{1}{3}$$

and so $d_f < 2/(1/3) = 6$. Since d_f is an integer, $d_f \leq 5$. Since $d_f \geq 3$ for a simple graph, interchanging f and v in the above gives us $d_v \leq 5$.

(d) Altogether there are 5 possibilities for the pair (d_v, d_f) by the previous part of the exercise. Given d_f and d_v, we can solve (6.9) for e. Then $vd_v = 2e$ and $fd_f = 2e$ give v and f. The five graphs turn out to be the Platonic solids with the interiors removed. (They are the tetrahedron, cube, octahedron, dodecahedron and icosahedron.)

6.3.5. The value of c is zero. Suppose when we cut as directed we cut through k edges. Each of these edges now becomes two, giving us k new edges. The same happens with the k faces. On each of the circles that we fill in with, we also get k edges and k vertices. The two circles give us 2 new faces. In summary, if we originally had $|V|$ vertices, $|E|$ edges and f faces on the torus, we now have a graph embedded on the sphere with $V|+2k$ vertices, $|E|+k+2k$ edges, and $f+k+2$ faces. From Euler's relation on the sphere,

$$2 \;=\; (|V|+2k) - (|E|+3k) + (f+k+2) \;=\; |V| - |E| + f.$$

Thus $|V| - |E| + f \;=\; 0$.

There's a subtle issue here: We described the cut as if each edge and face it encountered was different. This may not be the case, an edge (and face) can twist around the torus so that the cut meets it more than once; however, the counts are still correct. One way to see this is to imagine what happens if we cut around the face and stretch it flat. Stretching will distort our "bracelet cut" into some sort of curve that may cut through the face several times. Every time it passes through the face it creates another face, two edges and two vertices.

6.3.7. One method is to list all the simple planar graphs with $V = \underline{5}$ and find the least colorings for them. We use a theoretical argument instead.

The lex least proper coloring of $\underline{k} \subseteq V$ uses at most the first k colors. If it uses all k colors, then vertex k must be connected to each of the other vertices and the first $k-1$ vertices must use all of the first $k-1$ colors.

Let's apply these observations with $k = 5$, 4, 3 and 2 to a graph whose lex least coloring takes 5 colors. With $k = 5$, we see that vertex 5 is connected to each of the first 4 vertices and they use 4 different colors. Now, with $k = 4$, we see that vertex 4 is connected to each of the first 3 vertices and they use 3 different colors. Doing the same thing with $k = 3$ and $k = 2$, we finally see that every vertex is connected to every other; i.e., the graph is K_5, which is not planar.

6.3.9. The argument for degree 4 is correct. For degree 5, we can assume, perhaps after rotating and or flipping the graph, that $y_1, \ldots, y_5$ are assigned colors c_1, c_2, c_3, c_4 and c_2, respectively. Suppose we look at y_1 and y_3 as in the text. The argument given there is okay if we get y_1 and y_3 in separate components. If they are in the same component, we end up switching colors c_2 and c_4 in the component of the subgraph colored by c_2 and c_4 that contains v_4. The colors of $y_1, \ldots, y_4$ are now c_1, c_2, c_3 and c_2. If y_5 was not in the same component with y_4, it is colored c_2 and we are done. Unfortunately, if y_4 and y_5 are in the same component, *its color is switched to* c_4. You should convince yourself that there is no way to arrange things to avoid this possibility.

6.3.11 (a) We start with $\lambda = \begin{pmatrix} 1&2&3&4&5&6&7 \\ 1&&&&&&7 \end{pmatrix}$. A cycle is $(1,2,3,4,7)$, so we now have $\lambda = \begin{pmatrix} 1&2&3&4&5&6&7 \\ 1&2&3&4&&&7 \end{pmatrix}$. Another cycle is $(1,2,5,6,7)$, so we look at the path $2,5,6,7$ and choose $\lambda = \begin{pmatrix} 1&2&3&4&5&6&7 \\ 1&2&3&4&a&b&7 \end{pmatrix}$ where $2 < a < b < 7$ and λ is an injection. Depending on the choice of a and b compared to 3 and 4, we have get five different $1,7$-labelings. There could be others.

(b) Any $1,7$-labeling can be converted to a $7,1$-labeling simply by defining $\lambda_{7,1}(x) = 8 - \lambda_{1,7}(x)$.

6.3.13. We'll find all s,t-labelings. Suppose $K_{3,3}$ consists of all possible edges between $1,2,3$ and a,b,c. By symmetry, we may assume that $\lambda(1) = 1$ and either $\lambda(2) = 6$ or $\lambda(a) = 6$. In the former case, condition (c) requires that $\lambda(3)$ be less than 5 and more than 2. Up to symmetry, this gives us two answers:

$$\lambda \;=\; \begin{pmatrix} 1&2&3&a&b&c \\ 1&6&3&2&4&5 \end{pmatrix} \qquad \lambda \;=\; \begin{pmatrix} 1&2&3&a&b&c \\ 1&6&4&2&3&5 \end{pmatrix}.$$

Now suppose $\lambda(a) = 6$. In this case, one of $\lambda(2)$ and $\lambda(3)$ must be greater than $\lambda(b)$ and $\lambda(c)$. Thus, up to symmetry, we have $\lambda(2) = 5$. Similarly $\lambda(b) = 2$. This leads to two more answers:

$$\lambda \;=\; \begin{pmatrix} 1&2&3&a&b&c \\ 1&5&3&6&2&5 \end{pmatrix} \qquad \lambda \;=\; \begin{pmatrix} 1&2&3&a&b&c \\ 1&5&4&6&2&5 \end{pmatrix}.$$

6.3.15. We know from the text that a biconnected graph has an st-labeling. If $|V| = 2$, the result is trivial. Suppose that we have an st-labeling and that $\{x, y\}$ is an edge different from $\{s, t\}$. We may assume that $\lambda(x) < \lambda(y)$. By (iii) in the definition of st-labeling, we can find a sequence $y = w_1, w_2, \ldots = t$ such that $\lambda(w_i)$ is strictly increasing and such that $\{w_i, w_{i+1}\} \in E$. Similarly, we can find $x = u_1, u_2, \ldots = s$. These two paths with $\{s, t\}$ and $\{x, y\}$ form a cycle of G and so $\{x, y\}$ and $\{s, t\}$ are in the same bicomponent.

Section 6.4

6.4.1 (a) The value of a maximum flow is 45. Every maximum flow f will have $f(q, f) = 10$. Some other values of f are also determined uniquely, but many are not; for example, the flow into r can have any value from 15 to 20. Of course, the flows on the minimum cut set are unique. There are four minimum cut sets. The one found using $\mathcal{A}(f)$ is

$$\{\{r, h\}, \{f, a\}, \{k, e\}, \{y, u\}, \{z, u\}\}.$$

The others are obtained

(i) by deleting $\{r, h\}$ and adding $\{h, a\}$ and $\{h, c\}$,

(ii) by deleting $\{y, u\}$ and $\{z, u\}$ and adding $\{u, n\}$, or

(iii) by doing both (i) and (ii).

(b) See the previous solution.

(c) The value of a maximum flow is 25. Every maximum flow f will have $f(v, q) = 10$. Some other values of f are also determined uniquely, but many are not. There is just one minimum cut set:

$$\{\{c, d\}, \{k, e\}, \{r, x\}, \{w, x\}\}.$$

(d) See the previous solution. Since we do not have tools for finding all minimum cut sets, you may not have been able to prove that the minimum cut set was unique.

6.4.3. Since no complete augmentable path exists, $\mathcal{D}_{\text{in}} \subseteq A \subseteq V - \mathcal{D}_{\text{out}}$. Since $b(v) = 0$ for $v \notin \mathcal{D}$, it follows that $\sum_{v \in A} b(v) = \sum_{v \in \mathcal{D}_{\text{in}}} b(v)$, which is the definition of the value of a flow. Recall that $b(v)$ is the sum of all flows out of v minus the sum of all flows into v. It follows that for $e = (x, y) \in E$, $b(x)$ has a contribution of $f(x, y)$ and $b(y)$ has a contribution of $-f(e)$. We distinguish four cases according as x and y are in A or B and ask what $f(e)$ contributes to $\sum_{v \in A} b(v)$.

(i) $x \in B$, $y \in B$: Then $f(e)$ contributes nothing to the sum.

(ii) $x \in A$, $y \in A$: Then $f(e)$ contributes both $f(e)$ and $-f(e)$, which gives a net contribution of zero.

(iii) $x \in A$, $y \in B$; i.e., $(e) \in \text{FROM}(A, B)$: Then $f(e)$ contributes $f(e)$ to the sum.

(iv) $x \in B$, $y \in A$; i.e., $(e) \in \text{FROM}(B, A)$: Then $f(e)$ contributes $-f(e)$ to the sum.

6.4.5 (a) Without examining the network in detail, we would need to let c'_1 and c'_2 (resp. c'_3 and c'_4) be the sum of the capacities of edges leaving (resp. entering) the corresponding P'_i. That way we can guarantee the capability of supplying (resp. removing) as much fluid as the pump could possibly send out to (resp. get in from) other other sources. If we know all the maximum flows for the original network, we may be able to improve on this: We need to set c'_i to the largest net flow out of (resp. into) D_i for all maximum flows in the original network. This leads to no improvement in this case.

(b) Yes. Let f' be a flow in the new network shown for the exercise. With the c_i' edges removed and the P_i' pumps converted back to depots. If we eliminate these edges from f' we obtain a flow f in the network of Figure 6.6. We'll have value(f) = value(f') because the sum of the net flows out of D_1 and D_2 for f equals the net flow out of D_0 for f' because $b(P_1') = b(P_2') = 0$ for f'.

6.4.7. Let f and g be two maximum flows and let $A = \mathcal{A}(f)$. By the proof of the Augmentable Path Theorem, we see that value(g) = value(f) if and only if $g(e) = c(e)$ for all $e \in \text{FROM}(A, B)$ and $g(e) = 0$ for all $e \in \text{FROM}(B, A)$. It is tempting to conclude that therefore $A = \mathcal{A}(g)$, but this does not follow immediately.

Here is a correct proof. As above, let $A = \mathcal{A}(f)$. If $A \neq \mathcal{A}(g)$, we can assume that there is some $v \in \mathcal{A}(g)$ with $v \notin A$. (If not, interchange the names of f and g.) Let $u_1, u_2, \ldots$ be an augmentable path for g that ends at v. Let δ be its increment. Since $v \notin A$, and $u_1 \in \mathcal{D}_{\text{in}} \subseteq A$, there is an i with $u_i \in A$ and $u_{i+1} \in B$. If $e = (u_i, u_{i+1})$ is the directed edge of G, then $g(e) \leq c(e) - \delta$ and $e \in \text{FROM}(A, B)$. If $e = (u_{i+1}, u_i)$ is the directed edge of G, then $g(e) \geq \delta$ and $e \in \text{FROM}(B, A)$. In either case, the idea in the previous paragraph proves that value$(g) <$ value(f), contradicting the assumption that g is a maximum flow.

6.4.9 (a) This is trivial.

(b) Consider the sets when a is removed from them and the set A_n is removed. We have reduced n by 1 and (6.12) still holds (but may the inequalities may not be strict). By induction, we are done.

(c) By induction, there is an SDR for the A_i, $i \in I$. If the claimed inequality is true, then there is also an SDR for the B_i, $i \in \underline{n} - X$. Taken together, these give us our representatives. It remains to prove the inequality. We have

$$\bigcup_{i \in I \cup R} A_i \;=\; \Bigl(\bigcup_{i \in R} B_i\Bigr) \cup X,$$

where the last union is disjoint. Thus

$$\Bigl|\bigcup_{i \in R} B_i\Bigr| \;=\; \Bigl|\bigcup_{i \in R \cup I} A_i\Bigr| - |X| \;\geq\; |R \cup I| - |I| \;=\; |R|.$$

6.4.11. The result in the previous exercise is valid when all edges are taken to be undirected. To see this, construct a directed graph by replacing each edge $\{x, y\}$ of G with the two edges (x, y) and (y, x). The first part of the previous proof goes through. If a directed path $e_1, e_2, \ldots$ is constructed from a flow, replace each edge (x, y) in the directed path with $\{x, y\}$. This gives what we will call a pseudo-path. The same edge may appear twice in the pseudo-path because there may be two directed edges $e_i = (x, y)$ and $e_j = (y, x)$ which give the same undirected edge. We may assume that $i < j$. Replace the pseudo-path with the pseudo-path obtained from $e_1, \ldots, e_{i-1}, e_{j+1}, \ldots$. Iterating this process eventually leads to a path from u to v. (You may want to fill in some details about that.)

Section 6.5

6.5.1 (a) The probability that a vertex v has degree d is $\binom{n-1}{d}p^d(1-p)^{n-1-d}$ since we must choose d of the remaining $n-1$ vertices to connect to v, then multiply by the probability of an edge being present (p) or absent ($1-p$). Probabilities multiply since edges are independent in $\mathcal{G}_p(n)$. Using linearity of expectation and summing over all n vertices, we get $n\binom{n-1}{d}p^d(1-p)^{n-1-d}$.

(b) If C is a potential 4-cycle of 4 vertices, let $X_C = 1$ if the cycle is present and $X_C = 0$ if it is not. Then $\mathbf{E}(X_C) = p^4$. We must multiply this by the number of choices for C; that is, the number of potential 4-cycles. This number is $\binom{n}{4} \times 3 \;=\; \frac{n(n-1)(n-2)(n-3)}{4\times 2}$, which can be derived in at least two ways:

- Note that there are 3 ways to make a 4-cycle out of a set of 4 vertices.
- Choose an ordered list of 4 vertices that represent walking around a cycle. There are 4 vertices that could have been chosen as the starting vertex and 2 ways we could have gone around the cycle.

(c) This is the same as the previous situation, except that now we must make sure the two edges that cut across the 4-cycle are not present. Hence the answer is $3\binom{n}{4}p^4(1-p)^2$.

6.5.3 (a) The probability of a cycle is the probability of the union of the sets $\mathcal{G}_C$. The probability of the union of sets, is less than or equal to the sum of their separate probabilities; that is, $\Pr(A \cup B \cup \cdots) \;\le\; \Pr(A) + \Pr(B) + \cdots$.

(b) The denominator is $|\mathcal{G}(n,k)|$. The numerator counts graphs as follows. There are $(c-1)!$ directed cycles. Since each cycle can be made directed in two ways, there are $(c-1)!/2$ cycles. Since we have used up c edges making the cycle, we must choose $k-c$ edges from the remaining $N-c$ unused edges.

(c) Collect terms in (a) according to $c = |C|$ and use (b). There are $\binom{n}{c}$ c-subsets of $\underline{n}$.

(d) The left side comes from writing $\binom{x}{m} = \frac{x(x-1)\cdots(x-m+1)}{m!}$ and doing some algebra. The inequality comes from $\frac{k!}{(k-c)!} < k^c$ and $\frac{x-j}{y-j} < \frac{x}{y}$ when $y > x \ge j$.

6.5.5 (a) Let T contain a close to half the vertices as possible. If $|V| = 2n$, $|T| = n$ and $|V-T| = n$. Since G contains all edges, this choice of T gives us a bipartite subgraph with n^2 edges. When $|V| = 2n+1$, we take $|T| = n$ and $|V-T| = n+1$, obtaining a bipartite subgraph with $n(n+1)$ edges.

(b) The example bound is $|E|/2$ and $|E| = |\mathcal{P}_2(V)| = |V|(|V|-1)/2$. For $|V| = 2n$, we have $|E|/2 \;=\; n(2n-1)/2 = n^2 - n/2$. Hence the bound is off by $n/2$. This may sound large, but the relative error is small: Since $(n^2 - n/2)/n^2 = 1 - 1/2n$, the relative error is $1/|V|$. We omit similar calculations for $|V| = 2n+1$.

(c) The idea is to construct the largest possible complete graph and then add edges in any manner whatsoever. Let m be the largest integer such that $k \ge \binom{m}{2}$, choose $S \subseteq V$ with $|S| = m$, construct a complete graph on m vertices using $\binom{m}{2}$ edges, and insert the remaining $k - \binom{m}{2}$ edges in any manner to form a simple graph $G(V,E)$. By (a), the number of edges in a bipartite subgraph of the complete graph on T has at least $(m/2)^2 - m$ edges for some constant C Since m is as large as possible, $k < \binom{m+1}{2} < \frac{(m+1)^2}{2}$. Thus $m+1 > \sqrt{2k}$. Also, since $k \ge \binom{m}{2} > \frac{(m-1)^2}{2}$, $m-1 < \sqrt{2k}$. Hence the number of edges in bipartite subgraph is at least

$$(m/2)^2 - m \;>\; \frac{\left(\sqrt{2k}\, - 1\right)^2}{4} - \sqrt{2k} - 1,$$

Which equals k minus terms involving $k^{1/2}$ and constants.

	σ	$\bullet$	E	δ	comments on state
b	s1	sd	z	1a	starting
s1	z	sd	z	1a	part 1 sign seen
1a	z	2	e	1a	part 1 digits seen; accepting
sd	z	z	z	1b	decimal seen, no digits yet
1b	z	z	e	1b	part 1 after decimal; accepting
e	s2	z	z	2	E seen
s2	z	z	z	2	part 2 sign seen
2	z	z	z	2	part 2 digits seen; accepting
z	z	z	z	z	error seen

Figure S.6.1 The transition table for a finite automaton that recognizes floating point numbers, the possible inputs are sign (σ), decimal point ($\bullet$), digit (δ) and exponent symbol (E). The comments explain the states.

(d) Call the colors 1,2,3. Let V_i be the set of vertices colored with color i and let $E_{i,j}$ be the set of edges in G that connect vertices in V_i to vertices in V_j. Since $|E| = |E_{0,1}| + |E_{0,2}| + |E_{1,2}|$, at least one of $|E_{i,j}|$ is at most $|E|/3$. Suppose it is $E_{1,2}$. The bipartite subgraph whose edges connect vertices in V_0 to vertices in $V_1 \cup V_2$ contains $E - |E_{1,2}| \geq 2|E|/3$ edges.

Section 6.6

6.6.1. The left column gives the input and the top row the states.

	0	1	2	3	4
0	0	2	4	1	3
1	1	3	0	2	4

6.6.3. The states are 0, O1, E1 and R. In state 0, a zero has just been seen; in O1, an odd number of ones; in E1, an even number. The start state is 0 and the accepting states are 0 and O1. The state R is entered when we are in E1 and see a 0. Thereafter, R always steps to R regardless of input. You should be able to finish the machine.

6.6.5. In our input, we let δ stand for any digit, since the transition is independent of which digit it is. Similarly, σ stands for any sign. There is a bit of ambiguity as to whether the integer after the E must have a sign. We assume not. The automaton contains three states that can transit to themselves: recognizing digits before a decimal, recognizing digits after a decimal and recognizing digits after the E. We call them 1a, 1b and 2. There is a bit of complication because of the need to assure digits in the first part and, if it is present, in the second part. The transition table is given in Figure S.6.1.

6.6.7 (a) We need states that keep track of how much money is held by the machine. This leads us to states named $0, 5, \ldots, 30$. The output of the machine will be indicated by An, Bn, Cn and n, where n indicates the amount of money returned and A, B and C indicate the item delivered. There may be no output. The start state is 0.

(b) See Figure S.6.2.

	5	10	25	A	B	C	R
0	5	10	25	0	0	0	0
5	10	15	30	5	5	5	0, R5
10	15	20	10, R25	10	10	10	0, R10
15	20	25	15, R25	0, A0	15	15	0, R15
20	25	30	20, R25	0, A5	0, B0	20	0, R20
25	30	25, R10	25, R25	0, A10	0, B5	0, C0	0, R25
30	30, R5	30, R10	30, R25	0, A15	0, B10	0, C5	0, R30

Figure S.6.2 The transitions and outputs for an automaton that behaves like a vending machine. The state is the amount of money held and the input is either money, a purchase choice (A, B, C) or a refund request (R).

Section 7.1

7.1.1. $\mathcal{A}(m)$ (note m, not n) is the statement of the rank formula. The inductive step and use of the inductive hypothesis are clearly indicated in the proof.

7.1.3. Let $\mathcal{A}(k)$ be the assertion that the coefficient of $y_1^{m_1}\cdots y_k^{m_k}$ in $(y_1+\cdots+y_k)^n$ is $n!/m_1!\cdots m_k!$ if $n=m_1+\cdots+m_k$ and 0 otherwise. $\mathcal{A}(1)$ is trivial. We follow the hint for the induction step. Let $x=y_1+\cdots+y_{k-1}$. By the binomial theorem, the coefficient of $x^m y_k^{m_k}$ in $(x+y_k)^n$ is $n!/m!m_k!$ if $n=m+m_k$ and 0 otherwise. By the induction hypothesis, the coefficient of $y_1^{m_1}\cdots y_{k-1}^{m_{k-1}}$ in x^m is $m!/m_1!\cdots m_{k-1}!$ if $m=m_1+\cdots+m_{k-1}$ and zero otherwise. Combining these results we see that the coefficient of $y_1^{m_1}\cdots y_k^{m_k}$ in $(y_1+\cdots+y_k)^n$ is

$$\frac{n!}{m!\,m_k!}\,\frac{m!}{m_1!\cdots m_{k-1}!}$$

if $n=m_1+\cdots+m_k$ and 0 otherwise.

7.1.5 (a) $x_1'x_2'+x_1'x_2 \;=\; x_1'$.

(b) $x_1'x_2+x_1x_2'$.

(c) $x_1'x_2'x_3+x_1'x_2x_3+x_1x_2'x_3'+x_1x_2x_3' \;=\; x_1'x_3+x_1x_3'$.

(d) $x_1'x_2'x_3+x_1'x_2x_3'+x_1'x_2x_3+x_1x_2'x_3' \;=\; x_1'x_2+x_1'x_3+x_1x_2'x_3'$.

7.1.7. If you are familiar with de Morgan's laws for complementation, you can ignore the hint and give a simple proof as follows. By Example 7.3, one can express f' in disjunctive form: $f'=M_1+M_2+\cdots$. Now $f=(f')'=M_1'M_2'\cdots$ by de Morgan's law and, if $M_i=y_1y_2\cdots$, then $M_i'=y_1'+y_2'+\cdots$ by de Morgan's law.

To follow the hint, replace (7.5) with

$$f(x_1,\ldots,x_n) \;=\; \bigl(g_1(x_1,\ldots,x_{n-1})+x_n'\bigr)\,\bigl(g_0(x_1,\ldots,x_{n-1})+x_n\bigr)$$

and practically copy the proof in Example 7.3.

7.1.9. We can induct on either k or n. It doesn't matter which we choose since the formula we have to prove is symmetric in n and k. We'll induct on n. The given formula is $\mathcal{A}(n)$. For $n=0$, the formula becomes $F_{k+1} = F_{k+1}$, which is true.

$$\begin{aligned} F_{n+k+1} &= F_{(n-1)+(k+1)+1} && \text{using the hint} \\ &= F_n F_{k+2} + F_{n-1} F_{k+1} && \text{by } \mathcal{A}(n-1) \\ &= F_n (F_{k+1} + F_k) + F_{n-1} F_{k+1} && \text{by definition of } F_{k+2} \\ &= (F_n + F_{n-1}) F_{k+1} + F_n F_k && \text{by rearranging} \\ &= F_{n+1} F_{k+1} + F_n F_k && \text{by definition of } F_{n+1}. \end{aligned}$$

Section 7.2

7.2.1. Given that p and q are positive integers, it does not follow that p' and q' are positive integers. (For example, let $p = 1$.) Thus $\mathcal{A}(n-1)$ may not apply.

7.2.3. You may object that the induction has not been clearly phrased, but this can be overcome: Let I be the set of interesting positive integers and let $\mathcal{A}(n)$ be the assertion $n \in I$. If $\mathcal{A}(1)$ is false, then even 1 is not interesting, which is interesting. The inductive step is as given in the problem: If $\mathcal{A}(n)$ is false, then since $\mathcal{A}(k)$ is true for all $k < n$, n is the smallest uninteresting number, which is interesting.

Then what *is* wrong? It is unclear what "interesting" means, so the set of interesting positive integers is not a well defined concept. Proofs based on foggy concepts are always suspect.

7.2.5. To show the equivalence, we must show that an object is included in one definition if and only if it is included in the other. We do this by induction on the number of vertices. Before doing this, however, we observe that the objects constructed in Example 7.9 are trees:

- They are connected since $T_1, \ldots, T_k$ are connected by induction.
- They have no cycles since $T_1, \ldots, T_k$ have no cycles by induction and have no vertices in common by assumption.

We now turn to the inductive proof of equivalence.

(i) You should be able to see that both definitions include the single vertex.

(ii) Now for the inductive step in one direction: Suppose T has $n > 1$ vertices and is included in the definition in Example 7.9. By the induction hypothesis, $T_1, \ldots, T_k$ are included in Definition 5.12 (p. 139). By the construction in Example 7.9, the roots of $T_1, \ldots, T_k$ are ordered and are the children of the root of the new tree. Furthermore, joining $T_1, \ldots, T_k$ to a new root preserves the orderings and parent-child relationships in the T_i. Hence this tree satisfies Definition 5.12.

(iii) Now for the other direction. Let $r_1, \ldots, r_k$ be the ordered children of the root r of the tree T in Definition 5.12. Following on down through the children of r_i, we obtain an RP-tree T_i which is included in Example 7.9 since it has fewer than n vertices. By the argument in (ii), the construction forms an RP-tree from $T_1, \ldots, T_k$ which can be seen to be the same as T.

Section 7.3

7.3.1 (a) We must compare as long as both lists have items left in them. After all items have been removed from one list, what remains can simply be appended to what has been sorted. All items will be removed from one list the quickest if each comparison results in removing an item from the shorter list. Thus we need at least $\min(k_1, k_2)$ comparisons.

On the other hand, suppose we have k_1+k_2 items and the smallest ones are in the shorter list. In this case, all the items are removed from the shorter list and none from the longer in the first $\min(k_1, k_2)$ comparisons, so we have achieved the minimum.

(b) Here's the code. Note that the two lists have lengths m and $n-m$ and that $\min(m, n-m) = m$ because $m \le n/2$.

```
Procedure c(n)
    c = 0
    If (n = 1), then Return c
    Let m be n/2 with remainder discarded
    c = c + c(m)
    c = c + c(n − m)
    c = c + m
    Return c
End
```

(c) We have $c(2^0) = 0$ and $c(2^{k+1}) = 2c(2^k) + 2^k$ for $k \ge 0$. The first few values are

$$c(2^0) = 0, \quad c(2^1) = 2^0, \quad c(2^2) = 2 \times 2^1, \quad c(2^3) = 3 \times 2^2, \quad c(2^4) = 4 \times 2^3.$$

This may be enough to suggest the pattern $c(2^k) = k \times 2^{k-1}$; if not, you can compute more values until the pattern becomes clear.

We prove it by induction. The conjecture $c(2^k) = k \times 2^{k-1}$ is the induction assumption. For $k = 0$, we have $c(2^0) = 0$, and this is what the formula gives. For $k > 0$, we use the recursion to reduce k and then use the induction assumption:

$$c(2^k) = 2c(2^{k-1}) + 2^{k-1} = 2 \times (k-1) \times 2^{k-2} + 2^{k-1} = k \times 2^{k-1},$$

which completes the proof.

When k is large,

$$\frac{c(2^k)}{C(2^k)} = \frac{k \times 2^{k-1}}{(k-1)2^k + 1} = \frac{k/2}{k-1+2^{-k}} \sim 1/2.$$

This shows that the best case and worst case differ by about a factor of 2, which is not very large.

7.3.3. Here is code for computing the number of moves.

```
Procedure M(n)
    M = 0
    If (n = 1), then Return M
    Let m be n/2 with remainder discarded
    M = M + M(m)
    M = M + M(n − m)
    M = M + n
    Return M
End
```

This gives us the recursion $M(2^k) = 2M(2^{k-1}) + 2^k$ for $k > 0$ and $M(2^0) = 0$. The first few values are

$$M(2^0) = 0, \quad M(2^1) = 2^1, \quad M(2^2) = 2 \times 2^2, \quad M(2^3) = 3 \times 2^3, \quad M(2^4) = 4 \times 2^4.$$

Thus we guess $M(2^k) = k2^k$, which can be proved by induction.

7.3.5 (a) Here's one possible procedure. Note that the remainder must be printed out *after* the recursive call to get the digits in the proper order. Also note that one must be careful about zero: A string of zeroes should be avoided, but a number which is zero should be printed.

```
OUT(m)
     If m < 0, then
          Print ‘‘−’’
          Set m = −m
     End if
     Let q and 0 ≤ r ≤ 9 be determined by m = 10q + r
     If q > 0, then OUT(q)
     Print r
End
```

(b) Single digits

(c) When OUT calls itself, it passes an argument that is smaller in magnitude than the one it received, thus OUT(m) must terminate after at most $|m|$ calls.

7.3.7. The description for $k = 1$ is on the left and that for $k > 1$ is on the right:

$$n^{\underline{1}} \qquad\qquad\qquad\qquad n^{\underline{k}}$$

$$1 \quad \cdots \quad n \qquad\qquad k,\ \underline{k-1}^{\underline{k-1}} \quad \cdots \quad n,\ \underline{n-1}^{\underline{k-1}}$$

7.3.9 (a) Let $\mathcal{A}(n)$ be the assertion "H(n, S, E, G) takes the least number of moves." Clearly $\mathcal{A}(1)$ is true since only one move is required. We now prove $\mathcal{A}(n)$. Note that to do $S \xrightarrow{n} G$ we must first move all the other washers to pole E. They can be stacked only one way on pole E, so moving the washers from S to E requires using a solution to the Tower of Hanoi problem for $n-1$ washers. By $\mathcal{A}(n-1)$, this is done in the least number of moves by H$(n-1, S, G, E)$. Similarly, H$(n-1, E, S, G)$ moves these washers to G in the least number of moves.

(b) Simply replace H$(m, \ldots)$ with $S(m)$ and replace a move with a 1 and adjust the code a bit to get

```
Procedure S(n)
     If (n = 1)  Return 1.
     M = 0
     M = M + S(n − 1)
     M = M + 1
     M = M + S(n − 1)
     Return M
End
```

The recursion is $S(1) = 1$ and $S(n) = 2S(n-1) + 1$ when $n > 1$.

(c) The values are 1, 3, 7, 15, 31, 63, 127.

(d) Let $\mathcal{A}(n)$ be "$S(n) = 2^n - 1$." $\mathcal{A}(1)$ asserts that $S(1) = 1$, which is true. By the recursion and then the induction hypothesis we have

$$S(n) \ = \ 2S(n-1) + 1 \ = \ 2(2^{n-1} - 1) + 1 \ = \ 2^n - 1.$$

(e) By studying the binary form of k and the washer moved for small n (such as $n = 4$) you could discover the following rule.

> If $k = \cdots b_3 b_2 b_1$ is the binary representation of k, $b_j = 1$, and $b_i = 0$ for all $i < j$, then washer j is moved.

(This simply says that b_j is the lowest nonzero binary digit.) No proof was requested, but here's one. Let $\mathcal{A}(n)$ be the claim for $\mathtt{H}(n, \ldots)$. $\mathcal{A}(1)$ is trivial. We now prove $\mathcal{A}(n)$. If $k < 2^{n-1}$, it follows from $S(m)$ that $\mathtt{H}(n-1, \ldots)$ is being called and $\mathcal{A}(n-1)$ applies. If $k = 2^{n-1}$, then we are executing $S \xrightarrow{n} G$ and so this case is verified. Finally, if $2^{n-1} < k < 2^n$, then $\mathtt{H}(n-1, \ldots)$ is being executed at step $k - 2^{n-1}$, which differs from k only in the loss of its leftmost binary bit.

(f) Suppose that we are looking at move $k = \cdots b_3 b_2 b_1$ and that washer j is being moved. (That means b_j is the rightmost nonzero bit.) You should be able to see that this is move number $\cdots b_{j+2} b_{j+1} = (k - 2^{j-1})/2^j$ for the washer. Call this number k'. To determine source and destination, we must study move patterns.

The pattern of moves for a washer is either

$$P_0 \colon S \to G \to E \to S \to G \to E \to \cdots \text{ repeating or}$$
$$P_1 \colon S \to E \to G \to S \to E \to G \to \cdots \text{ repeating.}$$

Which washer uses which pattern? Consider washer j it is easily verified that it is moved a total of 2^{n-j} times, after which time it must be at G. A washer following P_i is at G only after move numbers of the form $3t + i + 1$ for some t. Thus $i + i$ is the remainder when 2^{n-j} is divided by 3. The remainder is 1 if $n - j$ is even and 0 otherwise. Thus washer j follows pattern P_i where i and $n - j$ have the same parity. If we look at the remainder after dividing k' by 3, we can see what the source and destination are by looking at the start of P_i. For those of you familiar with congruences, the remainder is congruent to $(-1)^j k + 1$ modulo 3.

7.3.11 (a) We have

$$H^*(n, S, E, G)$$

$$H^*(n-1, S, E, G) \qquad S \xrightarrow{n} E \qquad H^*(n-1, G, E, S) \qquad E \xrightarrow{n} G \qquad H^*(n-1, S, E, G)$$

(b) The initial condition is $h_1^* = 2$. For $n > 1$ we have $h_n^* \ = \ 3h_{n-1}^* + 2$.
Alternatively, $h_0^* = 0$ and, for $n > 0$, $h_n^* \ = \ 3h_{n-1}^* + 2$.

(c) The general solution is $h_n^* \ = \ 3^n - 1$. To prove it, use induction. First, it is correct for $n = 0$. Then, for $n > 0$,

$$h_n^* \ = \ 3h_{n-1}^* + 2 \ = \ 3(3^{n-1} - 1) + 2 \ = \ 3^n - 1.$$

7.3.13 (a) We omit the picture.

(b) Induct on n. It is true for $n = 1$. If $n > 1$, $a_2, \ldots, a_n \in G(k_2, \ldots, k_n)$ by the induction hypothesis. Thus $a_1, a_2, \ldots, a_n$ is in a_1, H and $a_1, R(H)$.

(c) Induct on n. It is true for $n = 1$. Suppose $n > 1$ and let the adjacent leaves be $b_1, \ldots, b_n$ and $c_1, \ldots, c_n$, with c following b. If $b_1 = c_1$, then apply the induction hypothesis to $G(k_2, \ldots, k_n)$ and the sequences $b_2, \ldots, b_n$ and $c_2, \ldots, c_n$. If $b_1 \neq c_1$, it follows from the local description that $c_1 = b_1 + 1$, that $b_2, \ldots, b_n$ is the rightmost leaf in H (or $R(H)$) and that $c_2, \ldots, c_n$ is the leftmost leaf in $R(H)$ (or H, respectively). In either case, $b_2, \ldots, b_n$ and $c_2, \ldots, c_n$ are equal because they are the same leaf of H.

(d) Let $R_n(\alpha)$ be the rank of $\alpha_1, \ldots, \alpha_n$. Clearly $R_1(\alpha) = \alpha_1 - 1$. If $n > 1$ and $\alpha_1 = 1$, then $R_n(\alpha) = R_{n-1}(\alpha_2, \ldots, \alpha_n)$. If $n > 1$ and $\alpha_1 = 2$, then $R_n(\alpha) = 2^n - 1 - R_{n-1}(\alpha_2, \ldots, \alpha_n)$. Letting $x_i = \alpha_i - 1$, we have $R_n(\alpha) = (2^n - 1)x_1 + (-1)^{x_1} R_{n-1}(\alpha_2, \ldots, \alpha_n)$ and so

$$R_n(\alpha) = (2^n-1)x_1+(-1)^{x_1}(2^{n-1}-1)x_2+(-1)^{x_1+x_2}(2^{n-2}-1)x_3+\cdots+(-1)^{x_1+x_2+\cdots+x_{n-1}}x_n.$$

(e) If you got this, congratulations. Let j be as large as possible so that $\alpha_1, \ldots, \alpha_j$ contains an even number of 2's. Change α_j. (Note: If $j = 0$, $\alpha = 2, 1, \ldots, 1$, the sequence of highest rank, and it has no successor.)

Section 7.4

7.4.1. Let $M(n)$ be the minimum number of multiplications needed to compute x^n. We leave it to you to verify the following table for $n \leq 9$

n	2	3	4	5	6	7	8	9	15	21	47	49
$M(n)$	1	2	2	3	3	4	3	4	5	6	8	7

Since $15 = 3 \times 5$, it follows that $M(15) \leq M(3) + M(5) = 5$. Likewise, $M(21) \leq M(3) + M(7) = 6$. Since the binary form of 49 is 110001_2, $M(49) \leq 7$. Since $47 = 101111_2$, we have $M(47) \leq 9$, but we can do better. Using $47 = 2 \times 23 + 1$, gives $M(47) \leq M(23) + 2$, which we leave for you to work out. A better approach is given by $47 = 5 \times 9 + 2$. Since x^2 is computed on the way to finding x^5, it is already available and so $M(49) \leq M(5) + M(9) + 1 = 8$. It turns out that these are minimal, but we will not prove that.

7.4.3. Let $\vec{v}_n$ be the transpose of $(a_n, \ldots, a_{n+k-1})$. Then $\vec{v}_n = M\vec{v}_{n-1}$ where

$$M = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & a_{k-2} & \cdots & a_1 \end{pmatrix}.$$

7.4.5. Finding a maximum of n items can be done in $\Theta(n)$, so it's the computation of all the different $F(v)$ values is the problem. Thus we could compute the values of F separately from finding the maximum. However, since it's convenient to compute the maximum while we're computing the values of F, we'll do it.

The root r of T has two sons, say s_L and s_R. Observe that the answer for the tree rooted at r must be either the answer for the tree rooted at s_L or the answer for the tree rooted at s_R or $F(r)$. Also

$$F(r) = f(r) + F(s_L) + F(s_R).$$

Here's an algorithm that carries out this idea.

```
/* r is the root of the tree in what follows. */

Procedure BestSum(r)
    Call Recur(r, Fvalue, best)
    Return best
End

Procedure Recur(r, F, best)
    If r is a leaf then
        F = f(r)
        best = f(r)
    Else
        Let s_L and s_R be the sons of r.
        Recur(s_L, F_L, b_L)
        Recur(s_R, F_R, b_R)
        F = f(r) + F_L + F_R
        best = max(F, b_L, b_R)
    End if
    Return
End
```

Since $f(r)$ is only used once, the running time of this algorithm is $\Theta(n)$ for n vertices, an improvement over $\Theta(n \ln n)$.

7.4.7 (a) We can use induction: It is easily verified for $n = 1$ and $n = 2$. For $n > 2$ we have

$$\begin{aligned} a_n &= a_{n-1} + a_{n-2} \;=\; (a_0F_{n-2} + a_1F_{n-3}) + (a_0F_{n-3} + a_1F_{n-4}) \\ &= a_0(F_{n-2} + F_{n-3}) + a_1(F_{n-3} + F_{n-4}) \;=\; a_0F_{n-1} + a_1F_{n-2}. \end{aligned}$$

(b) Since the a_i satisfy the same recursion as the Fibonacci numbers, it is easily seen that the a sequence is just the F sequence shifted by k.

Section 8.1

8.1.1. This is exactly the situation in the text, *except* that there is now one additional question when one reaches a leaf.

8.1.3 (a) If T is not a full binary tree, there is some vertex v that has only one child, say s. Shrink the edge (v, s) so that v and s become one vertex and call the new tree T'. If there are k leaves in the subtree whose root is v, then $\mathrm{TC}(T') = \mathrm{TC}(T) - k$.

(b) We follow the hint. Let k be the number of leaves in the subtree rooted at v. Since T is a binary tree and v is not a root, $k \geq 2$ Let $d = h(v) - h(l_2)$ and note that $d = (h(l_1) - 1) - h(l_2) \geq 1$. The distance to the root of every vertex in the subtree rooted at v is decreased by d and the distance of l_2 to the root is increased by d. Thus TC is decreased by $kd - d > 0$.

(c) By the discussion in the proof of the theorem, we know that the height of T must be at least m because a binary tree of height $m-1$ or less has at most 2^{m-1}. Suppose T had height $M > m$. By the previous part of this exercise, the leaves of T have heights M and, perhaps, $M - 1$. Thus, every vertex of v with $h(v) < M-1$ has two children. It follows that T has 2^{M-1} vertices w with $h(w) = M - 1$. If these were all leaves, T would have $2^{M-1} \geq 2^m$ leaves; however, at least one vertex u with $d(u) = M - 1$ is not a leaf. Since it has two children, T has at least $2^m + 1$ leaves, a contradiction.

(d) By the previous two parts, all leaves of T have height at most m. If T' is principal subtree of T, its leaves have height at most $m - 1$ in T'. Hence T' has at most 2^{m-1} leaves.

The argument hints at how to construct the desired tree: Construct T', a principal subtree of T, having all its leaves at height $m - 1$ in T'. Construct a binary tree T'' having $n - 2^{m-1}$ such that $\mathrm{TC}(T'')$ is as small as possible. The principal subtrees of T will be T' and T''.

8.1.5 (a) Suppose the answer is S_n. Clearly $S_0 = 1$ since the root is the only vertex. We need a recursion for S_n. One approach is to look at the two principal subtrees. Another is to look at what happens when we add a new "layer" by replacing each leaf with 

For the first approach, $S_n = 1 + 2S_{n-1}$, where each S_{n-1} is due to a principal subtree and the 1 is due to the root. The result follows by induction:

$$S_n = 1 + 2S_{n-1} = 1 + 2(2^n - 1) = 2^{n+1} - 1.$$

For the second approach, $S_n = S_{n-1} + 2^n$ and so $S_n = (2^n - 1) + 2^n = 2^{n+1} - 1$. By the way, if we have both recursions, we can avoid induction since we can solve the two equations

$$S_n = 1 + 2S_{n-1} \qquad \text{and} \qquad S_n = S_{n-1} + 2^n$$

to obtain the formula for S_n. Thus, by counting in two ways (the two recursions), we don't need to be given the formula ahead of time since we can solve for it.

(b) Let the value be $\mathrm{TC}^*(n)$. Again, we use induction and there are two approaches to obtaining a recursion. Clearly $\mathrm{TC}^*(1) = 0$, which agrees with the formula.

The first approach to a recursion: Since the principal subtrees of T each store S_{n-1} keys and since the path lengths all increase by 1 when we adjoin the principal subtrees to a new root, $\mathrm{TC}^*(n) = 2(S_{n-1} + \mathrm{TC}^*(n-1))$. Thus

$$\mathrm{TC}^*(n) = 2(2^n - 1 + (n-2)2^n + 2) = 2((n-1)2^n + 1) = (n-1)2^{n+1} + 2.$$

For the second approach, $\mathrm{TC}^*(n) = \mathrm{TC}^*(n-1) + n2^n$. Again, we can prove the formula for $\mathrm{TC}^*(n)$ by induction or, as in (a), we can solve the two recursions directly.

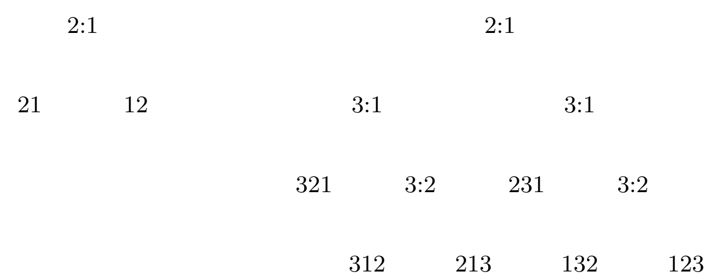


Figure S.8.1 The decision trees for binary insertion sorts. Go to the left at vertex $i : j$ if $u_i < s_j$ and to the right otherwise. (You may have done the reverse and gotten the mirror images. That's fine.)

Section 8.2

8.2.1. Here are the first few and the last.

1. Start the sorted list with 9.
2. Compare 15 with 9 and decide to place it to the right giving 9, 15.
3. Compare 6 with 9 to get 6, 9, 15.
4. Compare 12 with 9 and then with 15 to get 6, 9, 12, 15.
5. Compare 3 with 9 and then with 6 to get 3, 6, 9, 12, 15.

   . . . . . .

16. We now have the sorted list 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16. Compare 8 with 7, with 12 with 10 and then with 9 to decide where it belongs.

8.2.3. See Figure S.8.1. To illustrate, suppose the original list is 3,1,2. Thus $u_1 = 3$, $u_2 = 1$ and $u_3 = 2$.

- We start by putting u_1 in the sorted list, so we have $s_1 = 3$.
- Now u_2 must be inserted into the list s_1. We compare u_2 with s_1, the 2:1 entry. Since $s_1 = 3 > 1 = u_2$, we go to the left and our sorted list is $1, 3$ so now $s_1 = 1$ and $s_2 = 3$.
- Now u_3 must be inserted into the list s_1, s_2. Since we are at 3:1, we compare $u_3 = 2$ with $s_1 = 1$ and go to the right. At this point we know that u_3 must be inserted into the list s_2. We compare $u_3 = 2$ with $s_2 = 3$ at 3:2 and go to the left.

8.2.5 (a) Suppose that the alphabet has L letters and let the ith letter (in order) be a_i. Let u_j be a word with exactly k letters. The following algorithm sorts $u_1, \ldots, u_n$ and returns the result as $x_1, \ldots, x_n$.

```
BUCKET(u_1,...,u_n)
     Copy u_1,...,u_n to x_1,...,x_n.
     /* t is the position in the word. */
     For t = k to 1
     /* Make the buckets. */
          Create L empty ordered lists.
          For j = 1 to n
               If the tth letter of x_j is a_i,
                    then place x_j at the end of the ith list.
               End for
```

```
        Copy the ordered lists to x_1,...,x_n, starting with
             the first item in the first list and ending with the
             last item in the Lth list.
        End for
End
```

(b) Extend all words to k letters by introducing a new letter called "blank" which precedes all other letters alphabetically. Apply the algorithm in (a).

8.2.7. First divide the list into two equally long tapes, say

A: 9, 15, 6, 12, 3, 7, 11 5   B: 14, 1, 10, 4, 2, 13, 16, 8.

Think of each tape as containing a series of 1 long (sorted) lists. (The commas don't appear on the tapes, they're just there to help you see where the lists end.) Merge the first half of each tape, list by list, to tape C and the last halves to D. This gives us the following tapes containing a series of 2 long sorted lists:

C: 9 14, 1 15, 6 10, 4 12   D: 2 3, 7 13, 11 16, 5 8.

Now we merge these 2 long lists to get 4 long lists, writing the results on A and B:

A: 2 3 9 14, 1 7 13 15   B: 6 10 11 16, 4 5 8 12.

Merging back to C and D gives

C: 2 3 6 9 10 11 14 16   D: 1 4 5 7 8 12 13 15.

These are merged to produce one 16 long list on A and nothing on B.

8.2.9. A split requires $n-1$ comparisons since the chosen item must be compared with every other item in the list In the worst case, we may split an n long list into one of length 1 and another of length $n-1$. We then apply Quicksort to the list of length $n-1$. If $W(n)$ comparisons are needed, then $W(1) = 0$ and $W(n) = n - 1 + w(n-1)$ for $n > 1$. Thus $W(n) = \sum_{k=1}^{n-1} k = n(n-1)/2$.

Suppose that $n = 2^k$ and the lists are split evenly. Let $E(k)$ be the number of comparisons. Since Quicksort is applied to two lists of length $n/2$ after splitting, $E(k) = n - 1 + 2E(k-1)$ for $k > 0$ and $E(0) = 0$. A little computation gives us $E(1) = 1$, $E(2) = 5$, $E(3) = 17$, $E(4) = 49$ and $E(5) = 129$. From the statement of the problem we expect $E(k)$ to be near $k2^k$, which has the values 0, 2, 8, 24 and 64. Comparing these sequences we discover that $E(k) = 2(k-1)2^{k-1} + 1$ for $k < 6$. This is easily proved by induction

Section 8.3

8.3.1. Since there are only 3 things, you cannot compare more than one pair of things at any time. By the Theorem 8.1, we need at least $\log_2(3!)$ comparisons; i.e., at least three. A network with three comparisons that sorts is given in Figure 8.2.

8.3.3. As argued in the previous two solutions, we will need at least seven comparisons and we can do at least two per time. This means it will take at least four time units. It has been shown (but not in this text!) that at least five time units are required. A brick wall sort works.

8.3.5. One possibility is the type of network shown in Figure 8.3. For n inputs, this has

$$1 + 2 + \cdots + (n-1) \;=\; \frac{n(n-1)}{2}$$

comparators. It was noted in the text that a brick wall for n items must have length n. If n is even there are $(n/2)(n-1)$ comparators and if n is odd there are $n((n-1)/2)$ comparators. Thus this is the same as Figure 8.3. We don't know if it can be done with less.

8.3.7. By the Adjacent Comparisons Theorem, we need only check the sequence $n, \ldots, 2, 1$. Using the argument that proves the Zero-One Principle, it follows that this sequence is sorted if and only if all sequences that consist of a string of ones followed by a string of zeroes are sorted.

8.3.9. It is evident that the idea in the solution to (a) of the previous exercise works for any n. This can be used as the basis of an inductive proof.

An alternative proof can be given using sequences that consist of ones followed by zeroes. (See Exercise 8.3.7.) Note that when the lowest 1 starts moving down through the comparators, it moves down one line each time unit until it reaches the bottom. The 1 immediately above it starts the same process one time unit later. The 1 immediately above this one starts one more time unit later, and so forth. If there are j ones and the lowest 1 reaches the bottom after an exchange at time t, then the next 1 reaches proper position after an exchange at time $t+1$. Continuing in this way, all ones are in proper position after the exchanges at time $t+j-1$. Suppose the jth 1 (i.e., lowest 1) starts moving by an exchange at time i. Since it reaches position after $n-j$ exchanges, $t \;=\; i+(n-j)-1$. Thus all ones are in position after the exchanges at time $(i+(n-j)-1)+j-1 \;=\; n+i-2$. The jth 1 starts moving when it is compared with the line below it. This happens at time 1 or 2. Thus $n+i-2 \;\le\; n$.

8.3.11. Use induction on n. For $n = 2^1$, it works. Suppose that it works for all powers of 2 less than 2^t. We use the variation of the Zero-One Principle mentioned in the text. Suppose that the first half of the x_i's contains α zeroes and the second half contains β zeroes. `BMERGE` calls `BMERGE2` with $k = j = 2^{t-1}$ By the induction assumption, `BMERGE2` rearranges the "odd" sequence $x_1, x_3, \ldots, x_{2^t-1}$ in order and the "even" sequence $x_2, x_4, \ldots, x_{2^t}$ in order. The number of zeroes in the odd sequence minus the number of zeroes in the even sequence in 0, 1 or 2; depending on how many of α and β are odd. When the difference is 0 or 1, the result of `BMERGE2` is sorted. Otherwise, the last zero in the odd sequence, $x_{\alpha+\beta+1}$, is after the first one in the even sequence, $x_{\alpha+\beta}$, and all other x_i's are in order. The comparator in `BMERGE` with $i = (\alpha+\beta)/2$ fixes this.

8.3.13 (a) Since a one long list is sorted, nothing is done and so $S(0) = 0$. The two recursive calls of `BSORT` can be implemented by a network in which they run during the same time interval. This can then be followed by the `BMERGE` and so $S(N) \;\le\; S(N-1) + M(N)$.

(b) As for $S(0) = 0$, $M(0) = 0$ is trivial. Since all the comparators mentioned in `BMERGE` and be run in parallel at the same time, $M(N) \;\le\; M(N-1) + 1$.

(c) From the previous part, it easily follows by induction on N that $M(N) \le N$. Thus $S(N) \;\le\; S(N-1) + N$ and the desired result follows by induction on N.

(d) If $2^{N-1} < n \le 2^N$, then the above ideas show that $S(n) \le N(N+1)/2$. Thus

$$S(n) \;<\; \frac{1}{2}\,(1+\log_2 n)(2+\log_2 n).$$

Section 9.1

9.1.1. We do PREV(T).

```
PREV(T)
     Let r be the root of T
     Let T_1,...,T_k be the principal subtrees of T
     Output r
     For i = 1,...,k    Prev(T_i)
End
```

9.1.3. We give pseudocode for vertex visitation.

```
BFV(T)
     Initialize queue
     INQUEUE(T)
     While queue not empty
          S = OUTQUEUE( )
          Let r be the root of S
          Let S_1,...,S_k be the principal subtrees of S
          Output r
          For i = 1,...,k  INQUEUE(S_i)
     End while
End
```

9.1.5. The proof can be done by induction on the size of the tree by showing that the comments in the algorithm are correct. To do this, we need to notice a couple of things.

- By removing r from G before constructing S, we guarantee that S will not contain r. Thus it will contain precisely the vertices that are reachable on a path from r, starting with the edge $\{r,s\}$.
- Because we remove the root vertex of the tree from G and do this recursively, whenever a tree is ready to return, all its vertices have been removed from G. As a result, none of the vertices in S are left in G when we construct R.

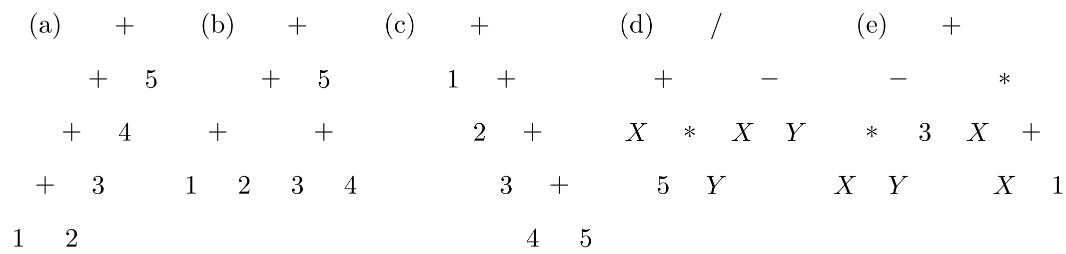
9.1.9 (a) $D(T)$ is $+1, D(T_1), -1, +1, D(T_2), -1, \ldots, +1, D(T_m), -1$.

(b) Each edge is traversed twice, proving the sum. The rest can be proved by induction on the number of vertices using the formula in (a). Actually, one can show more: The sum up to k is the length of the path from the root to the vertex that is reached after k steps.

(c) The "if" part follows from (b). The "only if" part can be done by showing that there is a unique way to construct a tree associated with such a sequence. This can be done recursively if we use the observation that the sum up to k is 0 if and only if we have returned to the root after k steps: Let k be the first index for which the sum is 0. We must have $s_1 = +1$, $s_k = -1$ and the subsequences $s_2, \ldots, s_{k-1}$ and $s_{k+1}, \ldots, s_n$ are associated with unique RP-trees. There is just one way to piece these trees together.

Section 9.2

9.2.1.



9.2.3. We use `value`$(\cdots)$ to indicate the value of a variable or constant.

(a) The first method:

```
EVALUATE(exp)
     If (exp has no op)   Return value(exp).
     If (exp = −exp1)   Return −value(exp1).
     Let exp = (exp1 op ; exp2).
     Return EVALUATE(exp1) op EVALUATE(exp2).
End
```

(b) The second method:

```
EVALUATE(T)
     Let r be the root of T.
     Let k be the number of principal subtrees of T
          and let T_i be the ith of them.
     If (k = 0), Return value(r).
     For i = 1,...,k    Let v_i = EVALUATE(T_i).
     /* If k = 1, r should be unary minus. */
     If k = 1, Return r v_1.
     If k = 2, Return v_1 r v_2.
End
```

9.2.5. We will indicate what needs to be added. Other solutions are possible.

(a) *exp* $\rightarrow$ $-$ *term*

(b) *term* $\rightarrow$ *power* and *power* $\rightarrow$ *factor* | *factor* ∗∗ *power*

(c) Let *subst* be the start symbol now and add *subst* $\rightarrow$ *exp* | **id :=** *exp*

(d) This is a bit trickier because the **:=** must reach as far to the right as possible. In particular, you cannot replace the last three items in the following list with just *factor* $\rightarrow$ *subst*. Let *start* be the start symbol.

start	$\rightarrow$	*exp* | *subst*
subst	$\rightarrow$	**id :=** *exp* | **id :=** *subst*
exp	$\rightarrow$	*exp*+*subst* | *exp*−*subst*
term	$\rightarrow$	*term*∗*subst* | *term*/*subst*
factor	$\rightarrow$	(*subst*)

Section 9.3

9.3.1. The construction starts with •. The first iteration gives • and all trees that have children produced in the starting step. Thus we get

In the next iteration, we obtain the following new trees with at most 4 vertices.

In the next step, the only new tree is a 4-vertex tree consisting of a path from the root to a single leaf. After this, no new trees with less than 5 vertices are obtained.

9.3.3. For $k \le 7$, the values are in the text. $b_8 = 429$, $b_9 = 1430$ and $b_{10} = 4862$.

9.3.5. We'll use $n\colon r$ to mean a tree with n leaves and rank r and $(n_1\colon r_1,\ n_2\colon r_2)$ to mean a tree with left son $n_1\colon r_1$ and right son $n_2\colon r_2$. We use formula (9.5) and the greedy approach: First make $|T_1|$ as large as possible, then make RANK(T_1) as large as possible. Here are the calculations. You should be able to construct the trees easily from the results as long as you remember (a) that $n\colon 0$ describes a tree in which all the left sons are leaves (since that is the leftmost tree in the list of trees) and (b) that since there is only one tree with 1 leaf and only one with 2 leaves, they each have rank 0.

$$\begin{aligned}
8\colon 100 &= (1\colon 0,\ 7\colon 100) && \text{since } b_1b_7 = 132 > 100, \text{ we have } |T_1| = 1 \text{ and } 100 = 0b_7 + 100\\
7\colon 100 &= (6\colon 10,\ 1\colon 0) && \text{since } b_1b_6 + \cdots + b_5b_2 = 90 \text{ and } 10 = 10b_1 + 0\\
6\colon 10 &= (1\colon 0,\ 5\colon 10) && \text{since } b_1b_5 = 14 > 10 \text{ and } 10 = 0b_5 + 10\\
5\colon 10 &= (4\colon 1,\ 1\colon 0) && \text{since } b_1b_4 + \cdots + b_3b_2 = 9 \text{ and } 1 = 1b_1 + 0\\
4\colon 1 &= (1\colon 0,\ 3\colon 1)\\
3\colon 1 &= (2\colon 0,\ 1\colon 0)
\end{aligned}$$

$$\begin{aligned}
8\colon 200 &= (3\colon 1,\ 5\colon 12) && \text{since } b_1b_7 + b_2b_6 = 174 \text{ and } 26 = 1b_5 + 12\\
5\colon 12 &= (4\colon 3,\ 1\colon 0) && \text{since } b_1b_4 + b_2b_3 + b_3b_2 = 9\\
4\colon 3 &= (3\colon 0,\ 1\colon 0) && \text{since } b_1b_3 + b_2b_2 = 3
\end{aligned}$$

$$\begin{aligned}
8\colon 300 &= (7\colon 3,\ 1\colon 0)\\
n\colon 3 &= (1\colon 0,\ n-1\colon 3) && \text{for } n \ge 5 \text{ since } b_1b_{n-1} > 3\\
4\colon 3 &= (3\colon 0,\ 1\colon 0)
\end{aligned}$$

$$8\colon 400 = (7\colon 103,\ 1\colon 0) \quad 7\colon 103 = (6\colon 13,\ 1\colon 0) \quad 6\colon 13 = (1\colon 0,\ 5\colon 13)$$
$$5\colon 13 = (4\colon 4,\ 1\colon 0) \quad 4\colon 4 = (3\colon 1,\ 1\colon 0)$$

9.3.7 (a) We omit the pictures.

(b) In the notation introduced in Exercise 9.3.5, with $n = 2m + 1$ and $k = b_n/2$, we claim that $\mathcal{M}_n = n\colon k = (m + 1\colon 0,\ m\colon 0)$. To prove this, note that the rank of this tree is $b_1b_{2m} + \cdots + b_mb_{m+1}$ and that

$$b_n \;=\; b_1b_{2m} + \cdots + b_{2m}b_1 \;=\; 2(b_1b_{2m} + \cdots + b_mb_{m+1}).$$

(c) If $n = 2m$, then $b_n = 2(b_1b_{2m-1} + \cdots + b_{m-1}b_{m+1}) + b_m^2$, which is divisible by 2 if and only if b_m is. Thus there is no such tree unless b_m is even. In this case you should be able to show that $\mathcal{M}_{2m} = (m\colon b_m/2,\ m\colon 0)$.

9.3.9. Here is one way to define an equivalence relation $\equiv$ by induction on the number of vertices. Let $\mathcal{T}_n$ be the set of labeled n-vertex RP-trees. Define all the trees in $\mathcal{T}_1$ to be equivalent. If $n > 1$, suppose $T, T' \in \mathcal{T}_n$. Let T be built from $T_1, \ldots, T_k$ and T' from $T'_l, \ldots, T'_\ell$ according to the recursive construction in Example 7.9 (p. 206). We define $T \equiv T'$ if and only $k = \ell$ and $T_i \equiv T'_i$ for $1 \le i \le k$.

9.3.11 (a) An x_i belongs in a parenthesis pair that has nothing inside it. Number the empty pairs from left to right and insert x_i into the ith pair.

(b) This is just a translation of what has been said. If you are confused, remember that $\mathtt{B}(n)$ should is to be thought of as all possible parentheses patterns for $x_1, \ldots, x_n$.

(c) This simply involves the replacement described in (b): Make $\bullet$ correspond to ( ) and make the tree with sons T_1 and T_2 correspond to (P_1P_2), where P_i is the parentheses pattern corresponding to T_i.

9.3.13 (a) The leaves in an RP-tree are distinguishable because the tree is ordered. Thus, each marking of the n leaves leads to a different situation. The same comments applies to vertices and there are $2n - 1$ vertices by Exercise 9.3.6.

(b) Mark the single vertex that arises in this way to obtain an element of $\mathcal{V}_n$ Interchanging x and the tree rooted at b gives a different element of $\mathcal{L}_{n+1}$ that gives rise to the same element of $\mathcal{V}_n$.

Conversely, given any element of $\mathcal{V}_n$, the marked vertex should be split into two, f and b with b a son of f. Introduce another son x of f which is a marked leaf. There are two possibilities–make f a left son or a right son.

(c) By (a), $|\mathcal{L}_n| = nb_n$ and $|\mathcal{V}_n| = (2n - 1)b_n$. By (b), $|\mathcal{L}_{n+1}| = 2|\mathcal{V}_n|$.

(d) By the recursion,

$$b_n = \frac{2(2n-3)}{n} b_{n-1} = \frac{2(2n-3)}{n}\,\frac{2(2n-5)}{n-1} b_{n-2} = \cdots = \frac{2^{n-1}(2n-3)(2n-5)\cdots 1}{n(n-1)\cdots 2} b_1.$$

Using $b_1 = 1$, we have a simple formula; however, it can be written more compactly:

$$\begin{aligned} b_n &= \frac{2^{n-1}(2n-3)(2n-5)\cdots 1}{n!} = \frac{2^{n-1}(n-1)!\,(2n-3)(2n-5)\cdots 1}{(n-1)!\,n!} \\ &= \frac{(2n-2)!}{(n-1)!\,n!} = \frac{1}{n}\binom{2n-2}{n-1}. \end{aligned}$$

Section 10.1

10.1.1. The p and q calculations can be done by multiplication. If so, and we are asked for the coefficient of x^3, say, then we can ignore any power of x greater than x^3 that appear in intermediate steps.

(a) Letting $\equiv$ mean equality of coefficients of x^n for $n \le 3$, we have

$$\begin{aligned} p^2 &= (1+x+x^2+x^3)^2 \equiv 1+2x+3x^2+4x^3 \\ p^3 &\equiv (1+x+x^2+x^3)(1+2x+3x^2+4x^3) \equiv 1+3x+6x^2+10x^3 \\ p^4 &\equiv (1+x+x^2+x^3)(1+3x+6x^2+10x^3) \equiv 1+4x^2+10x^3+20x^4. \end{aligned}$$

(b) By the opening remarks in the solution, this will be the same as (a).

(c) We can do this by writing $r \equiv 1+x+x^2+x^3$ or we can write, for example, $f(x) = (1-x)^{-2}$ and use Taylor's Theorem to compute the coefficients.

(d) Note that $r = 1+x+x^2+x^3+\cdots$. Whenever you add, subtract or multiply power series and look for the coefficient of some power of x, say x^n, only those powers of x that do not exceed n in the original series matter. Each of p, q and r begin $1+x+x^2+x^3$.

10.1.3. We have

$$(x^2+x^3+x^4+x^5+x^6)^8 \;=\; x^{16}(1+x+x^2+x^3+x^4)^8 \;=\; x^{16}\left(\frac{1-x^5}{1-x}\right)^8.$$

The coefficient of x^{21} in this is the coefficient of x^5 in the eighth power on the right hand side. Since $(1-x^5)^8 = 1-8x^5+\cdots$, this is simply the coefficient of x^5 in $(1-x)^{-8}$ minus 8 times the coefficient of x^0 (the constant term) in $(1-x)^{-8}$. Thus our answer is

$$(-1)^5\binom{-8}{5} - 8 \;=\; \binom{12}{5} - 8 \;=\; 784.$$

10.1.5. We'll do just the general k.

(a) We have $x^k A(x) = \sum_{m\ge 0} a_m x^{m+k} = \sum_{n\ge k} a_{n-k}x^n$. Thus the coefficient of x^n is 0 for $n < k$ and a_{n-k} for $n \ge k$.

(b) We have

$$\left(\frac{d}{dx}\right)^k A(x) \;=\; \sum_{m=0}^{\infty} a_m \left(\frac{d}{dx}\right)^k x^m \;=\; \sum_{m=k}^{\infty} a_m (m)(m-1)\cdots(m-k+1)x^{m-k}.$$

Set $n = m-k$ to obtain the answer: $a_{n+k}(n+k)(n+k-1)\cdots(n+1) \;=\; a_{n+k}\frac{(n+k)!}{n!}$.

(c) Since $\left(x\frac{d}{dx}\right)A(x) = \sum_{m=0}^{\infty} m a_m x^m$, repeating the operation k times leads to $\sum_{m=0}^{\infty} m^k a_m x^m$. Thus the answer is $n^k a_n$.

10.1.7. This is simply the derivation of (10.4) with r used instead of 1/3. The generating function for the sum is $S(x) = 1/(1-r(1+x))$ and the coefficient of x^k is

$$\frac{\big(r/(1-r)\big)^k}{1-r} \;=\; \frac{\big(r/(1-r)\big)^{k+1}}{r} \;=\; \frac{r^k}{(1-r)^{k+1}}.$$

To verify convergence, let $a_n = \binom{n}{k}r^n$ and note that

$$\lim_{n\to\infty}\left|\frac{a_{n+1}}{a_n}\right| \;=\; \lim_{n\to\infty}\frac{(n+1)|r|}{n-k+1} \;=\; |r| \;<\; 1.$$

10.1.9. This is very similar to the Exercise 10.1.8 With $a_j = (-1)^j\binom{m}{j}$ and $b_j = \binom{m}{j}$, we can apply the convolution formula. The result is $C(x) = (1-x)^m(1+x)^m = (1-x^2)^m$. By the binomial theorem, $(1-x^2)^m = \sum(-1)^j\binom{m}{j}x^{2j}$. Thus, the sum we are to simplify is zero if k is odd and $(-1)^j\binom{m}{j}$ if $k = 2j$.

10.1.11. The essential fact is that $\sum_{s=0}^{k-1}\omega^{rs}$ is k if r is multiple of k and 0 otherwise.

10.1.13. This is multisection with $k = 3$ and $j = 0, 2, 1$, respectively. The basic facts that are needed are $e^{i\theta} = \cos\theta + i\sin\theta$ and the sine and cosine of various angles in the 30°-60°-90° right triangle.

Section 10.2

10.2.1 (a) Let $a_n = 5a_{n-1} - 6a_{n-2} + b_n$ where $b_1 = 1$ and $b_n = 0$ for $n \neq 1$. Then

$$A(x) = \sum_{k=0}^{\infty}(5xa_{k-1}x^{k-1} - 6x^2a_{k-2}x^{k-2}) + x = 5xA(x) - 6x^2A(x) + x.$$

Thus

$$A(x) = \frac{x}{1-5x+6x^2} = \frac{1}{1-3x} - \frac{1}{1-2x}$$

and $a_n = 3^n - 2^n$.

(b) To correct the recursion, add c_{n+1} to the right side, where $c_0 = 1$ and $c_n = 0$ for $n \neq 0$. Multiply both sides by x^{n+1} and sum to obtain $A(x) = xA(x) + 6x^2A(x) + 1$. With some algebra,

$$A(x) = \frac{1}{1-x-6x^2} = \frac{3/5}{1-3x} + \frac{2/5}{1+2x}$$

and so $a_n = (3^{n+1} - (-2)^{n+1})/5$.

(c) To correct the recursion, add b_n where $b_1 = 1$ and $b_n = 0$ otherwise. Thus $A(x) = xA(x) + x^2A(x) + 2x^3A(x) + x$ and so

$$A(x) = \frac{x}{1-x-x^2-2x^3} = \frac{x}{(1-2x)(1+x+x^2)}.$$

By the quadratic formula, we can factor $1 + x + x^2$ as $(1 - \omega x)(1 - \overline{\omega}x)$, where $\omega = (-1 + \sqrt{-3})/2$ and $\overline{\omega}$ is the complex conjugate of ω. Using partial fractions,

$$A(x) = \frac{2/7}{1-2x} - \frac{(3-2\sqrt{-3})/21}{1-\omega x} - \frac{(3+2\sqrt{-3})/21}{1-\overline{\omega}x}$$

and so

$$a_n = \frac{2^{n+1}}{7} - \frac{(3-2\sqrt{-3})\omega^n}{21} - \frac{(3+2\sqrt{-3})\overline{\omega}^n}{21}.$$

The last two terms are messy, but they can be simplified considerably by noting that $\omega^3 = 1$ and so they are periodic with period 3. Thus

$$a_n = \frac{2^{n+1}}{7} + \begin{cases} (-2/7) & \text{if } n/3 \text{ has remainder } 0; \\ 3/7 & \text{if } n/3 \text{ has remainder } 1; \\ (-1/7) & \text{if } n/3 \text{ has remainder } 2. \end{cases}$$

(d) The recursion holds for $n = 0$ as well. From the recursion, $A(x) = 2xA(x) + \sum nx^n$. By Exercise 10.1.5, the sum is $x\frac{d}{dx}\sum x^n$, which is $x/(1-x)^2$. Thus

$$A(x) = \frac{x}{(1-x)^2(1-2x)} = \frac{2}{1-2x} - \frac{1}{1-x} - \frac{1}{(1-x)^2}.$$

After some algebra with these, we obtain $a_n = 2^{n+1} - n - 2$.

10.2.3. Start with a string of $n - i$ zeroes. Choose without repetition i of the $n + 1 - i$ positions (before all the zeroes or after any zero) and insert a one in each position chosen. The result is an n long string with i ones, none of them adjacent. The process is reversible: The position of a one is the number of zeroes preceding it. The formula for F_n follows immediately.

10.2.5 (a) Replacing A, B and C with their definitions and rearranging leads to

$$L_1L_2 + L_1H_2x^m + L_2H_1x^m + H_1H_2x^{2m} \;=\; (L_1 + H_1x^m)(L_2 + H_2x^m).$$

(b) The number of multiplications required by any procedure is an upper bound on $M(2m)$. There are three products of polynomials of degree m or less in our "less direct" procedure. If they are done as efficiently as possible, we will have $M(2m) \le 3M(m)$.

(c) Let $s_k = M(2^k)$. We have $s_0 = 1$ and $s_k \le 3s_{k-1}$ for $k > 0$. If we set $t_0 = 1$ and $t_k = 3t_{k-1}$ for $k > 0$, then $s_k \le t_k$. The recursion gives $T(x) = 3xT(x) + 1$ and so $t_k = 3^k$. Thus, with $n = 2^k$, $M(n) \le 3^k = (2^{\log_2 3})^k = n^{\log_2 3}$. From tables or a calculator, $\log_2 3 = 1.58\cdots$.

(d) To begin with, $L_1(x) = 1 + 2x$, $H_1(x) = -1 + 3x$, $L_2(x) = 5 + 2x$ and $H_2(x) = -x$. The product $L_1L_2 = (1 + 2x)(5 + 2x)$ is computed using the algorithm. The values are

$$m = 1, \quad A \;=\; (2)(2) \;=\; 4, \quad B \;=\; (1)(5) \;=\; 5 \quad \text{and} \quad C = (1 + 2)(5 + 2) = 21.$$

Thus $L_1L_2 = 5 + 12x + 4x^2$. In a similar way, the products $(-1 + 3x)(-x) = x - 3x^2$ and $(5x)(5 + x) = 25x + 5x^2$ are computed these are combined to give the final result:

$$(5 + 12x + 4x^2) + (x - 3x^2)x^4 + ((25x + 5x^2) - (5 + 12x + 4x^2) - (x - 3x^2))x^2,$$

which is $5 + 12x - x^2 + 12x^3 + 4x^4 + x^5 - 3x^6$.

(e) We'll just look at the case in which $n = 2m = 2^k$. Let a_k be the number of additions and subtractions needed. We have $a_0 = 0$ and, for $k > 0$, a_k equals $3a_{k-1}$ plus the number of additions and subtractions needed to prepare for and use the three multiplications. Preparation requires two additions of polynomials of degree $m - 1$. The results are three polynomials of degree $2m - 2$. We must perform two subtractions of such polynomials. Finally, the multiplication by x^m and x^{2m} arranges things so that there is some overlap among the coefficients. In fact, there will be $2m - 2$ additions required because of these overlaps (unless some coefficients happen to turn out zero). Since a polynomial of degree d has $d+1$ coefficients, there are a total of

$$2(m - 1 + 1) + 2(2m - 2 + 1) + (2m - 2) \;=\; 4n - 4.$$

Thus $a_k = 3a_{k-1} + 4 \times 2^k - 4$ and so $A(x) = 3xA(x) + 4\sum_{k>0}(2x)^k - 4\sum_{k>0} x^k$. Consequently, $A(x) = 4x/(1 - x)(1 - 2x)(1 - 3x)$ and $a_k = 2 \times 3^{k+1} - 2^{k+3} + 2$. Comparing this with the multiplication result, we see that we need about three times as many additions and/or subtractions as we do multiplications, which is still much smaller than n^2 for large n.

10.2.7. When we use the initial conditions and solve for $A(x)$ we get $A(x) = R(x) + \frac{N(x)}{D(x)}$ where $R(x)$ is some polynomial, $D(x) = 1 - c_1x - \cdots - c_kx^k$ and $N(x)$ is a polynomial of degree less than k. By the Fundamental Theorem of Algebra, we can factor $D(x)$ as given in the exercise. By the theory of partial fractions, there are constants $b_{i,j}$ such that

$$\frac{N(x)}{D(x)} \;=\; \sum_{i=1}^{m}\sum_{j=1}^{d_i} b_{i,j}(1 - r_ix)^{-j}.$$

Equating coefficients an assuming n is larger than the degree of $R(x)$, we have

$$a_n \;=\; \sum_{i=1}^{m}\sum_{j=1}^{d_i} b_{i,j}\binom{-j}{n}(-r_i)^n \;=\; \sum_{i=1}^{m}\sum_{j=1}^{d_i} b_{i,j}\binom{n+j-1}{j-1}r_i^n.$$

Since $\binom{n_j-1}{j-1}$ is a polynomial in n of degree $j-1$, it follows that $\sum_{j=1}^{d_i} b_{i,j}\binom{n+j-1}{j-1}$ is a polynomial in n of degree at most $d_i - 1$.

Let d be the degree of $R(x)$, where the degree of 0 is $-\infty$. If ℓ is the largest value of n for which an initial value of n must be specified, then $d \le \ell - k$. To find the coefficients of the polynomials $P_i(n)$, it suffices to know the values of $a_t, \ldots, a_{t_k}$ for any t larger than the degree of $R(x)$.

Section 10.3

10.3.1 (a) $(1-x)D' - D = -e^{-x} = -(1-x)D$ and so $(1-x)D' - xD = 0$.

(b) The coefficient of x^n on the left of our equation in (a) is

$$\frac{D_{n+1}}{n!} - \frac{D_n}{(n-1)!} - \frac{D_{n-1}}{(n-1)!}.$$

The initial conditions are $D_0 = 1$ and $D_1 = 0$.

10.3.3 (a) We are asked to solve $Q'(x) - 2(1-x)^{-1}Q(x) = 2x(1-x)^{-3}$. The integrating factor is

$$\exp\left(\int -2(1-x)^{-1}dx\right) = \exp\bigl(2\ln(1-x)\bigr) = (1-x)^2.$$

Thus

$$Q(x)(1-x)^2 = \int \frac{2x}{1-x}dx = \int \left(-2 + \frac{2}{1-x}\right)dx = -2x - 2\ln(1-x) + C.$$

(b) We have $-2\ln(1-x) - 2x = \sum_{k\ge 2} 2x^k/k$ and

$$(1-x)^{-2} = \sum_{k\ge 0}\binom{-2}{k}(-x)^k = \sum_{k\ge 0}(k+1)x^k.$$

By the formula for the coefficients in a product of generating functions,

$$\begin{aligned} q_n &= \sum_{k=2}^{n}\frac{2(n-k+1)}{k} = 2(n+1)\sum_{k=2}^{n}\frac{1}{k} - \sum_{k=2}^{n}2 \\ &= 2(n+1)\sum_{k=1}^{n}\frac{1}{k} - 2(n+1) - 2(n-1) = 2(n+1)\sum_{k=1}^{n}\frac{1}{k} - 4n. \end{aligned}$$

Section 10.4

10.4.1 (a) This is nothing more than a special case of the Rule of Product—at each time we can choose anything from $\mathcal{T}$.

(b) Simply sum the previous result on k.

(c) The hint tells how to do it. All that is left is algebra.

(d) The solution is like that in the previous part, except that we start with

$$\prod_{T\in\mathcal{T}}\left(\sum_{i=0}^{\infty}\left(\mathbf{x}^{\mathbf{w}(T)}\right)^i\right) \;=\; \prod_{T\in\mathcal{T}}(1-\mathbf{x}^{\mathbf{w}(T)})^{-1}.$$

10.4.3 (a) This is simply $2^*\ \{0,\, 12^*\}^*$. Thus the generating function is

$$A(x) \;=\; \frac{1}{1-x}\,\frac{1}{1-\left(x+x\dfrac{1}{1-x}\right)} \;=\; \frac{1}{1-3x+x^2}.$$

Multiply both sides by $1-3x+x^2$ and equate coefficients of x^n to obtain the recursion

$$a_n \;=\; 3a_{n-1}-a_{n-2} \quad \text{for} \quad n>1$$

with initial conditions $a_0 = 1$ and $a_1 = 3$.

(b) You should be able to see that this is described by $0^*(11^*0^k0^*)^*$. Since

$$\mathsf{G}_{11^*0^k0^*} \;=\; x\frac{1}{1-x}x^k\frac{1}{1-x} \;=\; \frac{x^{k+1}}{(1-x)^2},$$

the generating function we want is

$$A(x) \;=\; \frac{1}{1-x}\,\frac{1}{1-x^{k+1}/(1-x)^2} \;=\; \frac{1-x}{1-2x+x^2-x^{k+1}}.$$

Clearing of fractions and equating coefficients, we obtain the recursion

$$a_n \;=\; 2a_{n-1}-a_{n-2}+a_{n-k-1} \quad \text{for} \quad n>1,$$

with the understanding that $a_j = 0$ for $j < 0$. The initial conditions are $a_0 = a_1 = 1$.

(c) A possible formulation is

$$0^*\ (1(11)^*00^*)^*\ \{\lambda,\ 1(11)^*\}.$$

This says, start with any number of zeroes, then append any number of copies of the patterns of type Z (described soon) and then follow by either nothing or an odd number of ones. A pattern of type Z is an odd number of ones followed by one or more zeroes. The translation to a generating function gives

$$\frac{1}{1-x}\,\frac{1}{\mathsf{G}_Z(x)}\left(1+x\frac{1}{1-x^2}\right) \quad \text{where} \quad \mathsf{G}_Z(x) \;=\; x\frac{1}{1-x^2}x\frac{1}{1-x}.$$

After some algebra, the generating function reduces to

$$A(x) \;=\; \frac{1+x-x^2}{1-x-2x^2+x^3},$$

which gives $a_n \;=\; a_{n-1}+2a_{n-2}-a_{n-3}$ for $n>2$, with initial conditions $a_0 = 1$, $a_1 = 2$ and $a_2 = 3$.

10.4.5. Here's a way to construct a pile of height h. Look at the number of blocks in each column. The numbers increase to h, possibly stay at h for some time, and then fall off. The numbers up to but not including the first h form a partition of a number with largest part at most $h-1$ and the numbers after the first h form a partition of a number with largest part at most h. The structures are these partitions. By the Rule of Product and Exercise 10.4.4

$$\sum_{n\geq 0} s_{n,h}(x) \;=\; \left(\prod_{i=1}^{h-1}\frac{1}{1-x_i}\right) x^h \left(\prod_{i=1}^{h}\frac{1}{1-x_i}\right) \;=\; \frac{x^h}{(1-x^h)\prod_{i=1}^{h-1}(1-x^i)^2}.$$

Summing this over all $h>0$ and adding 1 gives $\sum s_n x^n$. No simple formula is known for the sum.

10.4.7 (a) We can build the trees by taking a root and joining to it zero, one or two binary RP-trees. This gives us $T(x) \;=\; x(1+T(x)+T(x)^2)$.

(b) There is no simple expansion for the square root; however, various things can be done. One possibility is to use $\sqrt{1-2x-3x^2} \;=\; \sqrt{1-3x}\,\sqrt{1+x}$. You can then expand each square root and multiply the generating functions together. This leads to a summation of about n terms for p_n. The terms alternate in sign. A better approach is to write

$$\sqrt{1-2x-3x^2} \;=\; \sum_k \binom{1/2}{k}(-1)^k(2x+3x^2)^k \;=\; \sum_{k,j}(-1)^k\binom{1/2}{k}\binom{k}{j}2^{k-j}3^j x^{k+j}.$$

This leads to a summation of about $n/2$ positive terms for p_n. It's also possible to get a recursion by constructing a first order, linear differential equation with polynomial coefficients for $T(x)$ as done in Exercise 10.2.6. Since the recursion contains only two terms, it's the best approach if we want to compute a table of values. It's also the easiest to program on a computer.

10.4.9. The key to working this problem is to never allow the root to have exactly one son.

(a) Let the number be r_n. The generating function for those trees whose root has degree k is $R(x)^k$. Since $\sum_{k\geq 0} R(x)^k \;=\; 1/(1-R(x))$, we have $R(x) \;=\; x\frac{1}{1-R(x)} - xR(x)$. Clearing of fractions and solving the quadratic,

$$R(x) \;=\; \frac{1+x-\sqrt{1-2x-3x^2}}{2(1+x)}.$$

(The minus sign is the correct choice for the square root because $R(0) \;=\; r_0 \;=\; 0$.) These numbers are closely related to p_n in Exercise 10.4.7. By comparing the equations for the generating functions,

$$(1+x)R(x) \;=\; x(P(x)+1)$$

and so $r_n + r_{n-1} \;=\; p_{n-1}$ when $n>1$.

(b) We modify the previous idea to count by leaves:

$$R(x) \;=\; x + \sum_{k\geq 2} R(x)^k \;=\; x + \frac{R(x)^2}{1-R(x)}.$$

Solving the quadratic:

$$R(x) \;=\; \frac{1+x-\sqrt{1-6x+x^2}}{4}.$$

(c) From (a) we have $2(1+x)R - 1 - x = -\sqrt{1-2x-3x^2}$ and so

$$2xR' + 2R - 1 = \frac{1-3x}{\sqrt{1-2x-3x^2}}.$$

Thus $(1-2x-3x^2)(2xR'+R-1) = -2(1+x)R+1+x$. Equating coefficients of x^n gives us

$$(2n+1)r_n - (4n-2)r_{n-1} - (6n-9)r_{n-2} = -2r_n - 2r_{n-1} \quad \text{for} \quad n \geq 3.$$

Rearranging and checking initial conditions we have

$$r_{n+1} = \frac{4nr_n + 3(2n-1)r_{n-1}}{2n+5} \quad \text{for} \quad n \geq 2,$$

with $r_0 = r_2 = 0$ and $r_1 = 1$. You should be able to treat (b) in a similar manner. The result is $r_0 = 0$, $r_1 = r_2 = 1$ and, for $n \geq 2$,

$$r_{n+1} = \frac{3(2n-1)r_n - (n-2)r_{n-1}}{n+1}.$$

10.4.11 (a) A tree of outdegree D, consists of a root and some number $d \in D$ of trees of outdegree D joined to the root. Use the Rules of Sum and Product.

(b) Let $f_0(x) = 1$. Define $f_{n+1}(x) = x\sum_{d\in D} f_n(x)^d$. We leave it to you to prove by induction that $f_n(x)$ agrees with $T_D(x)$ through terms of degree n.

(c) Except for the $1 \in D$ question, this is handled as we did $T_D(x)$. Why must we have $1 \notin D$? You should be able to see that there are an infinite number of trees with exactly one leaf—construct a tree that is just a path of length n from the root to the leaf.

10.4.13. We can build these trees up the way we built the full binary RP-trees: join two trees at a root. If we distinguish right and left sons, every case will be counted twice, except when the two sons are the same. Thus $B(x) = x + \frac{1}{2}(B(x)^2 - E(x)) + E(x)$, where $E(x)$ counts the situation where both sons are the same and nonempty. We get this by choosing a son and then duplicating it. Thus each leaf in the son is replaced by two leaves and so $E(x) = B(x^2)$.

10.4.15 (a) Either the list consists of repeats of just one item OR it consists of a list of the proper form AND a list of repeats of one item. In the first case we can choose the item in s ways and use it any number of times from 1 to k. In the second case, we can choose the final repeating item in only $s-1$ ways since it must differ from the item preceding it.

(b) After a bit of algebra,

$$A_k(x) = \frac{s/(s-1)}{1-(s-1)(x+x^2+\cdots+x^k)} - \frac{s}{s-1} = \frac{s(1-x)/(s-1)}{1-sx+(s-1)x^{k+1}} - \frac{s}{s-1}.$$

(c) Multiplying both sides of the formula just obtained for $A_k(x)$ by $1-sx+(s-1)x^{k+1}$ gives the desired result.

(d) Call a sequence of the desired sort acceptable. Add anything to the end of an n-long acceptable sequence. This gives $sa_{n,k}$ sequences. Each of these is either an acceptable sequence of length $n+1$ or an $(n-k)$-long acceptable sequence followed by $k+1$ copies of something different from the last entry in the $(n-k)$-long sequence.

10.4.17. We have $1 - 3x + x^2 = (1 - ax)(1 - bx)$ where $a = \frac{3+\sqrt{5}}{2}$ and $b = \frac{3-\sqrt{5}}{2}$. Thus

$$\frac{x}{1 - 3x + x^2} = \frac{1/(a-b)}{1 - ax} - \frac{1/(a-b)}{1 - bx}$$

and so, since $a - b = \sqrt{5}$,

$$r_n = \frac{a^n - b^n}{\sqrt{5}}.$$

10.4.19 (a) The accepting states are unchanged except that if the old start state was accepting, both the old and new start states are accepting. If there was an edge from the old start state to state t labeled with input i, then add an edge from the new start state to t labeled with i. (The old edge is *not* removed.) We can express this in terms of the map $f : S \times I \to 2^S$ for the nondeterministic automaton. Let $s_o \in S$ be the old start state and introduce a new start state s_n. Let $T = S \cup \{s_n\}$ and define $f^* : T \times I \to 2^T$ by

$$f * (t, i) = \begin{cases} f(t, i), & \text{if } t \in S, \\ f(s_o, i), & \text{if } t = s_n. \end{cases}$$

(b) Label the states of $\mathcal{A}$ and $\mathcal{B}$ so that they have no labels in common. Call their start states s_A and s_B. Add a new start state s_n that has edges to all of the states that s_A and s_B did. In other words, $f * (s_n, i)$ is the union of $f_A(s_A, i)$ and $f_B(s_B, i)$, where f_A and f_B are the functions for $\mathcal{A}$ and $\mathcal{B}$. If either s_A or s_B was an accepting state, so is s_n; otherwise the accepting states are unchanged.

(c) Add the start state of $S(\mathcal{A})$ to the accepting states. (This allows the machine to accept the empty string, which is needed since * means "*zero* or more times.") Run edges from the accepting states of $S(\mathcal{A})$ to those states that the start state of $S(\mathcal{A})$ goes to. In other words, if s is the start state,

$$f^*(t, i) = \begin{cases} f(t, i), & \text{if } t \text{ is not an accepting state,} \\ f(t, i) \cup f(s, i), & \text{if } t \text{ is an accepting state.} \end{cases}$$

(d) From each accepting state of $\mathcal{A}$, run an edge to each state to which the start state of $\mathcal{B}$ has an edge. The accepting states of $\mathcal{B}$ are accepting states. If the start state of $\mathcal{B}$ is an accepting state, then the accepting states of $\mathcal{A}$ are also accepting states, otherwise they are not. The start state is the start state of $\mathcal{A}$.

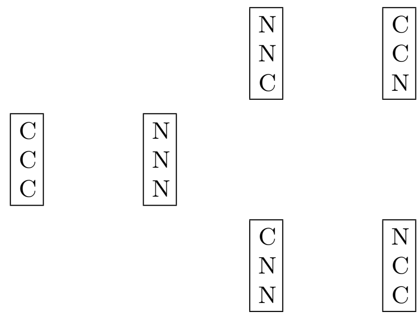


Figure S.11.1 The state transition digraph for covering a 3 by n board with dominoes. Each vertex is labeled with a triple that indicates whether commitment has been made in that row (C) or not made (N). The start and end states are those with no commitments.

Section 11.1

11.1.1. The problem is to eliminate all but the c's from the recursion. One can develop a systematic method for doing this, but we will not since we have generating functions at our disposal. In this particular case, let $p_n = f_n + s_n$ and note that $p_n - p_{n-1} = 2c_{n-1}$ by (11.5). Thus, by the first of (11.4), this result and the last of (11.5),

$$\begin{aligned} c_{n+1} - c_n &= (2c_n + p_n + c_{n-1}) - (2c_{n-1} + p_{n-1} + c_{n-2}) \\ &= 2c_n - c_{n-1} - c_{n-2} + (p_n - p_{n-1}) \\ &= 2c_n + c_{n-1} - c_{n-2}. \end{aligned}$$

11.1.3 (a) Figure S.11.1 gives a state transition digraph.

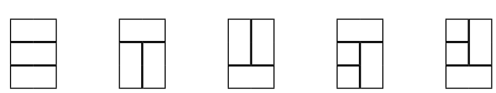
Let $a_{n,s}$ be the number of ways to take n steps from the state with no commitments and end in a state s. Let $A_s(x) = \sum_n a_{n,s}x^n$. As in the text, the graph lets us write down the linked equations for the generating functions. From the graph it can be seen that A_s depends only on the number k of commitments in s. Therefore we can write $A_s = B_k$. The linked equations are then

$$\begin{aligned} B_0(x) &= x(B_3(x) + 2B_1(x)) + 1 \\ B_1(x) &= x(B_0(x) + B_2(x)) \\ B_2(x) &= xB_1(x) \\ B_3(x) &= xB_0(x), \end{aligned}$$

which can be solved fairly easily for $B_0(x)$.

(b) Equate coefficients of x^n on both sides of $(1 - 4x^2 + x^4)A(x) = 1 - x^2$.

(c) By looking at the dominoes in the last two columns of a board, we see that it can end in five mutually exclusive ways:



This shows that a_n equals $3a_{n-2}$ plus whatever is counted by the last two of the five cases. A board of length $n - 2$ ends with either (i) one vertical domino and one horizontal dominoes or (ii) three horizontal dominoes. If the vertical dominoes mentioned in (i) are changed to the left ends of horizontal dominoes, they fit with the last two cases shown above. If the three horizontal mentioned in (ii) are removed, we obtain all boards of length $n - 4$. Thus the sum of the last two cases in the picture plus a_{n-4} equals a_{n-2}.

11.1.5. Call the start state α and let $L_{i,j}$ be the number of different single letter inputs that allow the machine to move from state i to state j. Let $a_{n,i}$ be the number of ways to begin in state α, recognize n letters and end in state i and let $A_i = \mathsf{G}^1(a_{n,i})$. The desired generating function is the sum of A_i over all accepting states. A linked set of recursions can be obtained from the automaton that leads to the generating function equations

$$A_i(x) \;=\; \sum_j L_{i,j} x A_j(x) + \begin{cases} 1 & \text{if } i = \alpha; \\ 0 & \text{otherwise.} \end{cases}$$

11.1.7 (a) We will use induction. It is true for $n = 1$ by the definition of $m_{x,y} = m^{(1)}_{x,y}$. (Its also true for $n = 0$ because the zeroth power of a matrix is the identity and so $m^{(0)}_{x,y} = 1$ if $x = y$ and 0 otherwise.) Now suppose that $n > 1$. By the definition of matrix multiplication, $m^{(n)}_{x,y} = \sum_z m^{(n-1)}_{x,z} m_{z,y}$. By the induction hypothesis and the definition of $m_{z,y}$ each term in the sum is the number of ways to get from x to z in $n-1$ steps times the number of ways to get from z to y in one step. By the Rules of Sum and Product, the proof is complete.

(b) If α is the initial state, $\mathbf{i}M^n\mathbf{a}^t = \sum m^{(n)}_{\alpha,y}$, the sum ranging over all accepting states y.

(c) By the previous part, the desired generating function is

$$\sum_{n=0}^{\infty} \mathbf{i}M^n\mathbf{a}^t\, x^n \;=\; \mathbf{i}\sum_{n=0}^{\infty} x^n M^n \mathbf{a}^t \;=\; \mathbf{i}\sum_{n=0}^{\infty} (xM)^n \mathbf{a}^t \;=\; \mathbf{i}(I - xM)^{-1}\mathbf{a}^t.$$

(d) The matrix M is replaced by the table given in the solution to the previous exercise.

Section 11.2

11.2.1. Theorem. Suppose each structure in a set $\mathcal{T}$ of structures can be constructed from an ordered partition (K_1, K_2) of the labels, two nonnegative integers ℓ_1 and ℓ_2, and some ordered pair (T_1, T_2) of structures using the labels K_1 in T_1 and K_2 in T_2 such that:

(i) The number of ways to choose a T_i with labels K_i and ℓ_i unlabeled parts depends only on i, $|K_i|$ and ℓ_i.

(ii) Each structure $T \in \mathcal{T}$ arises in exactly one way in this process.

(We allow the possibility of $K_i = \emptyset$ if $\mathcal{T}_i$ contains structures with no labels and likewise for $\ell_i = 0$.) It then follows that

$$T(x, y) \;=\; T_1(x, y) T_2(x, y),$$

where $T_i(x, y) \;=\; \sum_{n=0}^{\infty} t_{i,n,m}(x^n/n!)y^m$ and $t_{i,n,m}$ is the number of ways to choose T_i with labels $\underline{n}$ and k unlabeled parts. Define $T(x, y)$ similarly.

The proof is the same as that for the original Rule of Product except that there is a double sum:

$$t_{n,m} \;=\; \sum_{K_1 \subseteq \underline{n}} \sum_{\ell_1=0}^{m} t_{1,|K_1|,\ell_1}\, t_{2,n-|K_1|,m-\ell_1} \;=\; \sum_{k=0}^{n} \sum_{\ell_1=0}^{m} \binom{n}{k} t_{1,k,\ell_1}\, t_{2,n-k,m-\ell_1}$$

11.2.3 (a) By the text,

$$\sum_k z(n, k) y^k \;=\; y(y + 1) \cdots (y + n - 1).$$

Replacing all but the last factor on the right hand side gives us

$$\sum_k z(n,k)y^k \;=\; \Bigl(\sum_k z(n-1,k)y^k\Bigr)(y+n-1).$$

Equate coefficients of y^k.

(b) For each permutation counted by $z(n,k)$, look at the location of n. There are $z(n-1,k-1)$ ways to construct permutations with n in a cycle by itself. To construct a permutation with n not in a cycle by itself, first construct one of the permutations counted by $z(n-1,k)$ AND then insert n into a cycle. Since there are j ways to insert a number into a j-cycle, the number of ways to insert n is the sum of the cycle lengths, which is $n-1$.

11.2.5 (a) For any particular letter appearing an odd number of times, the generating function is

$$\sum_{n \text{ odd}} \frac{x^n}{n!} \;=\; \frac{e^x - e^{-1}}{2} \qquad \text{with Taylor's theorem and some work.}$$

We must add 1 to this to allow for the letter not being used. The Rule of Product is then used to combine the results for A, B and C.

(b) Multiplying out the previous result:

$$\begin{aligned}\left(1+\frac{e^x-e^{-x}}{2}\right)^3 &= 1+3(e^x-e^{-x})/2+3(e^x-e^{-x})^2/4+(e^x-e^{-x})^3/8\\ &= 1+\Bigl(3e^x/2-3e^{-x}/2\Bigr)+\Bigl(3e^{2x}/4-3/2+3e^{-2x}/4\Bigr)+\Bigl(e^{3x}/8-3e^x/8+3e^{-x}/8-e^{-3x}/8\Bigr)\\ &= -1/2+\Bigl(e^{3x}/8-e^{-3x}/8\Bigr)+\Bigl(3e^{2x}/4+e^{-2x}/4\Bigr)+\Bigl(9e^x/8-9e^{-x}/8\Bigr).\end{aligned}$$

Now compute the coefficients.

11.2.7. We saw in this section that $B(x) \;=\; \exp(e^x-1)$. Differentiating:

$$B'(x) \;=\; \exp(e^x-1)\,(e^x-1)' \;=\; B(x)e^x.$$

Equating coefficients of x^n:

$$\frac{B_{n+1}}{n!} \;=\; \sum_{k=0}^{n} \frac{B_k}{k!}\,\frac{1}{(n-k)!},$$

which gives the result.

11.2.9 (a) Let $g_{n,k}$ be the number of graphs with n vertices and k components. We have $\sum_{n,k} g_{n,k}(x^n/n!)y^k \;=\; \exp(yC(x))$, by the Exponential Formula. Differentiating with respect to y and setting $y=1$ gives us

$$\sum_n \Bigl(\textstyle\sum_k k g_{n,k}\Bigr)x^n/n! \;=\; \left.\frac{\partial \exp(yC(x))}{\partial y}\right|_{y=1} \;=\; H(x).$$

(b) $\sum_k g_{n,k}$ is the number of ways to choose an n-vertex graph and mark a component of it. We can construct a graph with a marked component by selecting a component (giving $C(x)$) AND selecting a graph (giving $G(x)$).

(c) For permutations, there are $(n-1)!$ connected components of size n and so

$$C(x) \;=\; \sum \frac{(n-1)!x^n}{n!} \;=\; \sum x^n/n \;=\; -\ln(1-x).$$

Since there are $n!$ permutations of n, $G(x) = \frac{1}{1-x}$ and the average number of cycles in a permutation is

$$\frac{h_n}{n!} \;=\; n!\,[x^n] \sum_{k=1}^{\infty} \frac{x^n}{n} \frac{1}{1-x} \sum_{k=1}^{n} \frac{1}{k}.$$

(d) Since $C(x) = e^x - 1$, we have $H(x) = (e^x - 1)\exp(e^x - 1)$ and so

$$h_n \;=\; \sum_{k=1}^{n} \binom{n}{k} B_{n-k} \;=\; \sum_{k=0}^{n} \binom{n}{k} B_{n-k} - B_n,$$

which is $B(n+1) - B_n$ by the previous exercise.

(e) Since $C(x) = x + x^2/2$, we have $H(x) = (x + x^2/2)I(X)$, where $I(x)$ is the EGF for i_n, the number of involutions of $\underline{n}$. Thus the average number of cycles in an involution of $\underline{n}$ is

$$\frac{\binom{n}{1}i_{n-1} + \binom{n}{2}i_{n-2}}{i_n} = \frac{n}{2}\left(1 + \frac{i_{n-1}}{i_n}\right),$$

where the right side comes from the recursion $i_n = i_{n-1} + (n-1)i_{n-2}$.

11.2.11. Suppose $n > 1$. Since f is alternating,

- k is even;
- $f(1), \ldots, f(k-1)$ is an alternating permutation of $\{f(1), \ldots, f(k-1)\}$;
- $f(k+1), \ldots, f(n)$ is an alternating permutation of $\{f(k+1), \ldots, f(n)\}$.

Thus, an alternating permutation of $\underline{n}$ for $n > 1$ is built from an alternating permutation of odd length AND an alternating permutation, such that the sum of the lengths is $n-1$. We have shown that

$$\sum_{n>1} \frac{a_n x^{n-1}}{(n-1)!} \;=\; B(x)A(x)$$

and so $A'(x) = B(x)A(x) + 1$. Similarly, $B'(x) = B(x)B(x) + 1$.

Separate variables in $B' = B^2 + 1$ and use $B(0) = 0$ to obtain $B(x) = \tan x$. Use the integrating factor $\cos x$ for

$$A'(x) = (\tan x)A(x) + 1$$

and the initial condition $A(0) = 1$ to obtain $A(x) = \tan x + \sec x$.

11.2.13 (a) The square of a k-cycle is

- another cycle of length k if k is odd;
- two cycles of length $k/2$ if k is even.

Using this, we see that the condition is necessary. With further study, you should be able to see how to take a square root of such a permutation.

(b) This is simply putting together cycles of various lengths using (a) and recalling that there are $(k-1)!$ k-cycles.

(c) By bisection $\displaystyle\sum_{\substack{k=1 \\ k \text{ odd}}}^{\infty} \frac{x^k}{k} = \tfrac{1}{2}\Big(\{-\ln(1-x)\} - \{-\ln(1-(-x))\}\Big)$.

(d) We don't know of an easier method.

11.2.15 (a) We'll give two methods. First, we use the Exponential Formula approach in Example 11.14. When we add a root, the number of leaves does not change—except when we started with nothing and ended up with a single vertex tree. Correcting for this exception gives us the formula.

Without the use of the Exponential Formula, we could partition the trees according to the degree k of the root, treating $k=0$ specially because in this case the root is a leaf:

$$L(x,y) = xy + \sum_{k=1}^{\infty} L(x,y)^k/k!.$$

(b) This type of problem was discussed in Section 10.3. Recall that there are n^{n-1} n-vertex rooted labeled trees.

(c) Differentiate the equation in (a) with respect to y and set $y=1$ to obtain

$$U(x) = xe^{T(x)}U(x) + x = T(x)U(x) + x,$$

where we have used the fact that $L(x,1) = T(x) = xe^{T(x)}$. Solving for U: $U = \frac{x}{1-T}$. Differentiating $T(x) = xe^{T(x)}$ and solving for $T'(x)$ gives us $T' = \frac{T}{x(1-T)}$. Thus $x^2T' + x = \frac{x}{1-T}$, which gives the equation for $U(x)$.

We know that $t_n = n^{n-1}$. It follows from the equation for $U(x)$ that

$$\frac{u_n}{n!} = \frac{(n-1)^{n-2}}{(n-2)!}.$$

Thus $u_n/t_n = n/(1+x)^{1/x}$, where $x = \frac{1}{n-1}$. As $n \to \infty$, $x \to 0$ and, by l'Hôpital's Rule

$$(1+x)^{1/x} = \exp\left(\frac{\ln(1+x)}{x}\right) \to \exp(1) = e.$$

11.2.17 (a) There are several steps

- Since g is a function, each vertex of $\varphi(g)$ has outdegree 1. Thus the image of $\underline{n}^{\underline{n}}$ lies in $\mathcal{F}_n$.
- φ is an injection: if $\varphi(g) = \varphi(h)$, then $(x, g(x)) = (x, h(x))$ for all $x \in \underline{n}$ and so $g = h$.
- Finally φ is onto $\mathcal{F}_n$: If $(V, E) \in \mathcal{F}_n$, for each $x \in \underline{n}$ there is an edge $(x, y) \in E$. Define $g(x) = y$.

(b) Let's think in terms of a function g corresponding to the digraph. Let $k \in \underline{n}$. If the equation $g^t(k)$ has a solution, then k is on a cycle and will be the root of a tree. The other vertices of the tree are those $j \in \underline{n}$ for which $g^s(j) = k$ for some s.

(c) This is simply an application of Exercise 11.2.2.

(d) In the notation of Theorem 11.6, $T(x)$ is $T(x)$, $f(t) = e^T$ and $g(u) = -\ln(1-u)$. Thus $n(f_n/n!)$ is the coefficient of u^n in $e^{nu}(1-u)^{-1}$. Using the convolution formula for the coefficient of a product of power series, we obtain the result.

Section 11.3

11.3.1 (a) $AB = CD$ follows from $a_i + b_i = \min(a_i, b_i) + \max(a_i, b_i)$. C divides A if and only if $c_i \le a_i$ for all i. Thus C divides both A and B if and only if $c_i \le \min(a_i, b_i)$ for all i. Thus $C = \gcd(A, B)$. The claim that $D = \operatorname{lcm}(A, B)$ follows similarly.

(b) It follows from the definition of gcd that $\gcd(n, i)$ must divide both n and i. This completes the first part. From (a), we have $\gcd(ab, ac) = a \gcd(b, c)$. Apply this with $ab = n$ and $a = k$: $\gcd(n, i) = k \gcd(n, i/k)$.

(c) By letting $j = n/k$, we can see that the two forms of the sum are equivalent, so we need only prove one. Let g generate C_n, that is $g(i) = i + 1$ modulo n. Let $h = g^t$ so that $h(i) = i + t$ modulo n. Thus $h^k(i) = i + kt$. The smallest $k > 0$ such that h^k is the identity is thus the smallest k such that kt is a multiple of n. Also, the cycle form of h consists of k-cycles. Since there are n elements, there must be n/k k-cycles. Thus h contributes $z_k^{n/k}$ to the sum. We need to know how many values of t give a particular value of k. By looking at prime factorization, you should be able to see that $k \gcd(n, t) = n$ and so $\gcd(n, t) = n/k$. We can now use (b) with k replaced by n/k to conclude that the number of such t is $\varphi(n/(n/k)) = \varphi(k)$.

11.3.3 (a) A regular octahedron can be surrounded by a cube so that each vertex of the octahedron is the center of a face of the cube. The center of the octahedron is the center of the cube. A line segment from the center of the cube to a vertex of the cube passes through the center of the corresponding face of the octahedron. A line segment from the center of the cube to the center of an edge of the cube passes through the center of the corresponding edge of the octahedron.

(b) By the above correspondence, the answer will be the same as the symmetries of the cube acting on the faces of the cube. See (11.31).

(c) By the above correspondence it is the same as the answer for the edges of the cube. See the previous exercise.

11.3.5. The group is usually called S_4. Here are its $4! = 24$ elements:

- The identity, which gives x_1^4.
- $(4-1)! = 6$ elements which are 4-cycles, which give $6x_4$.
- $\binom{4}{2} = 6$ elements which consist of two 1-cycles and a 2-cycle, giving $6x_1^2x_2$.
- $\binom{4}{1)\times(3-1)!=8}$ elements which consist of a 1-cycle and a 3-cycle, giving $8x_1x_3$.
- $\frac{1}{2}\binom{4}{2} = 3$ elements which consist of two 2-cycles, giving $3x_2^2$.

Thus

$$Z_{S_n} \;=\; \frac{x_1^4 + 6x_4 + 6x_1^2x_2 + 8x_1x_3 + 2x_2^2}{24}.$$

Now apply Theorem 11.9:

$$\begin{array}{rcccl} x_1^4 & \Longrightarrow & x_1^{16} & & \\ x_4 & \Longrightarrow & x_4^4 & & \\ x_1^2x_2 & \Longrightarrow & x_1^4x_2^{2+2+2} & = & x_1^4x_2^6 \\ x_1x_3 & \Longrightarrow & x_1x_3^{1+1+3} & = & x_1x_3^5 \\ x_2^2 & \Longrightarrow & x_2^8 & & \end{array}$$

11.3.7. If the vertices belong to different cycles of length i and $j > i$ we get $x_{\operatorname{lcm}(i,j)}^{\gcd(i,j)}$ as in the digraph case and so we get

$$\prod_{i<j} \left(x_{\operatorname{lcm}(i,j)}\right)^{\nu_i \nu_j \gcd(i,j)}.$$

If the two cycles have the same length, we must be careful not to overcount because the edge is not directed. When the two vertices are in different cycles of length i we get x_i^i and there are $\binom{\nu_i}{2}$ such pairs of cycles. When the two vertices belong to the same cycle of length i, we must be extra careful: If the separation between the vertices on the cycle is $i/2$, the edge $\{u,v\}$ comes back after $i/2$ steps around the cycle as $\{v,u\}$, so to speak. Otherwise, it must go i steps around the cycle. Thus we get a contribution of either

$$x_i^{(i-1)\nu_i/2} \text{ for odd } i, \qquad x_i^{(i-2)\nu_i/2} x_{i/2}^{\nu_i} \text{ for even } i.$$

Putting this all together, $\prod x_i^{\nu_i}$ becomes

$$\left(\prod_{i<j} (x_{\operatorname{lcm}(i,j)})^{\nu_i \nu_j \gcd(i,j)}\right)\left(\prod_i x_i^{i\nu_i(\nu_i-1)/2}\right)\left(\prod_{i \text{ odd}} x_i^{(i-1)\nu_i/2}\right)\left(\prod_{i \text{ even}} x_i^{(i-2)\nu_i/2} x_{i/2}^{\nu_i}\right).$$

Section 11.4

11.4.1 (a) We have $r^2 = 2r+1$ and so $r = 1+\sqrt{2}$ and $m = 1$. By the principle, we expect there is some constant A such that $a_n \sim A(1+\sqrt{2})^n$.

(b) Since $A(x) = \frac{1+x}{1-2x-x^2}$, we have $p(x) = 1+x$, $q(x) = 1-2x-x^2$, $r = \sqrt{2}-1 = 1/(1+\sqrt{2}\,)$ and $q'(r) = -2\sqrt{2}$. Thus $k = 1$ and we have

$$a_n \sim \frac{(-1)^1 \sqrt{2}\, n^{1-1}}{-2\sqrt{2}\, r^{n+1}} \;=\; \tfrac{1}{2}\big(1+\sqrt{2}\,\big)^{n+1}.$$

(c) We have $1-2x-x^2 \;=\; (1-ax)(1-bx)$ where $a = 1+\sqrt{2}$ and $b = 1-\sqrt{2}$. Expanding by partial fractions:

$$\begin{aligned} \frac{1+x}{1-2x-x^2} &= \frac{1}{1-2x-x^2} + \frac{x}{1-2x-x^2} \\ &= \frac{\frac{a}{a-b}}{1-ax} - \frac{\frac{b}{a-b}}{1-bx} \\ &\quad + \frac{\frac{1}{a-b}}{1-ax} - \frac{\frac{1}{a-b}}{1-bx} \\ &= \frac{(2+\sqrt{2}\,)/(2\sqrt{2}\,)}{1-ax} - \frac{(2-\sqrt{2}\,)/(2\sqrt{2}\,)}{1-bx}. \end{aligned}$$

Thus $a_n \;=\; \frac{1}{2}(1+\sqrt{2}\,)^{n+1} + \frac{1}{2}(1-\sqrt{2}\,)^{n+1}$.

11.4.3. From the discussion in the example, you can see that merging two lists of lengths i and $j > i$ takes at least i comparison. Thus the example shows that the number of comparisons for merge sorting satisfies $T_n \;=\; f(n) + T(m) + T(n-m)$ where $m = \lfloor n/2 \rfloor$ and $m \le f(n) < n$. Apply Principle 11.3.

11.4.5. We'll use Principle 11.4 (p. 345) so $t_{n,k}$ will denote the kth term of the sum we're given.

(a) Since

$$\frac{t_{n,k+1}}{t_{n,k}} \;=\; \frac{n-k}{n}$$

is less than 1 and is close to 1 when k/n is small, we'll use Principle 11.5 (p. 346). Since

$$\frac{1-r_k}{k} \;=\; \frac{1-(n-k)/n}{k} = \frac{1}{n},$$

(11.38) gives the estimate

$$\sqrt{\frac{\pi n}{2}}\,\frac{n!\,n^{n+1}}{n!}.$$

(b) This is a bit more complicated than (a) since

$$\frac{t_{n,k+1}}{t_{n,k}} \;=\; \frac{k+1}{k}\,\frac{n-k}{n}$$

is greater than 1 for small k and less than 1 for k near n. Thus $t_{n,k}$ achieves its maximum somewhere between 1 and n, namely, when the above ratio equals 1. This leads to a quadratic equation for k which has the solution

$$k \;=\; \frac{-1+\sqrt{1+4n}}{2}.$$

Since this differs from $\sqrt{n}$ by at most a constant, we'll split the sum into two pieces at $k=\sqrt{n}$ and use Principle 11.5 (p. 346) for each half. Since each half has the same estimate, we simply double one result. Ignoring the fact that $\sqrt{n}$ is not an integer, we set $k=\sqrt{n}+j$ and use $j\ge 0$ as the new index of summation. Call the new terms $t'_{n,j}$. We have

$$\begin{aligned}
r_j \;&=\; \frac{t'_{n,j+1}}{t'_{n,j}} = \frac{t_{n,k+1}}{t_{n,k}} \;=\; \frac{k+1}{k}\,\frac{n-k}{n}\\
&=\; \frac{\sqrt{n}+j+1}{\sqrt{n}+j}\,\frac{n-\sqrt{n}-j}{n}\\
&=\; \left(1+\frac{1}{\sqrt{n}+j}\right)\left(1-\frac{\sqrt{n}+j}{n}\right)\\
&=\; 1+\frac{n-\big(\sqrt{n}+j\big)^2-\big(\sqrt{n}+j\big)}{n\big(\sqrt{n}+j\big)}\\
&=\; 1-\frac{2j\sqrt{n}+j^2+\sqrt{n}+j}{n\big(\sqrt{n}+j\big)}\\
&\approx\; 1-\frac{2j\sqrt{n}}{n\sqrt{n}} = 1-2j/n.
\end{aligned}$$

Thus $(1-r_j)/j\approx 2/n$ and so we obtain the following approximation (the factor of 2 is due to the presence of two sums)

$$2\sqrt{\pi n/4}\,t'_{n,0} \;=\; \frac{\sqrt{\pi}\,n!}{n^{\sqrt{n}}(n-\sqrt{n}\,)!},$$

where $(n-\sqrt{n}\,)!$ should be approximated using Stirling's formula (Theorem 1.5 (p. 12)) since we have no formula for $x!$ when x is not a positive integer.

11.4.7. Use Principle 11.6 (p. 349) with $r=1$, $b=0$ and $c=-1$ to obtain

$$a_n \sim n!\,\exp\Bigg(-\sum_{k\in S} 1/k\Bigg).$$

11.4.9 (a) The function has radius of convergence $r = 1$ and has a singularity at $-1 = -r$. Another reason we can't use it for $A_e(x)$ is that $a_{e,n} = 0$ whenever n is odd.

(b) For both cases, $r = 1$, $b = 0$ and $c = -1/2$. We obtain $L = \sqrt{2}$ for $A_o(x)$ and $L = 1/\sqrt{2}$ for $A_e(x)$.

(c) By power series, $a_{e,2n} = (-1)^n \binom{-1/2}{n}(2n)!$, which can be rearranged to give the answer. By Stirling's formula, $a_{e,2n} \sim (2n)!/\sqrt{\pi n} \sim 2(2n/e)^{2n}$.

(d) Since $A_o(x) = (1+x)A_e(x)$, we have $a_{o,2n} = a_{e,2n}$ and $a_{o,2n+1} = (2n+1)a_{e,2n}$. By the previous part, $a_{o,2n} = \binom{2n}{n}(2n)!\,4^{-n}$ and $a_{o,2n+1} = \binom{2n}{n}(2n+1)!\,4^{-n}$.

11.4.11. We use Principle 11.6 (p. 349) with

$$A(x) \;=\; (1-2x-3x^2)^{-1/2}, \quad b = 0 \;\text{ and }\; c = -1/2.$$

Since $1 - 2x - 3x^2 = (1-3x)(1+x)$, $r = 1/3$ and

$$L \;=\; \lim_{x\to 1/3} \frac{(1-2x-3x^2)^{-1/2}}{(1-3x)^{-1/2}} \;=\; \lim_{x\to 1/3} \frac{1}{\sqrt{1+x}} \;=\; \frac{\sqrt{3}}{2}.$$

Thus

$$a_n \;\sim\; \frac{\sqrt{3}\,3^n n^{-1/2}}{2\Gamma(1/2)} \;=\; \frac{3^{n+1/2}}{2\sqrt{\pi n}}.$$

11.4.13. We use Principle 11.6 (p. 349). Since $(1+x^2)^2 - 4x$ vanishes at $x = r = 0.295597742\ldots$ and is positive whenever $-r \le x < r$, we have found r. Thus we take

$$f(x) \;=\; (1-x/r)^{1/2}, \qquad g(x) \;=\; \frac{-1}{2}\sqrt{\frac{(1+x^2)^2-4x}{1-x/r}} \qquad \text{and} \qquad h(x) \;=\; \frac{1+x^2}{2}.$$

We have

$$L \;=\; \lim_{x\to r} \frac{-\sqrt{(1+x^2)^2-4x}}{2\sqrt{1-x/r}} \;=\; -\frac{1}{2}\sqrt{\lim_{x\to r}\frac{(1-x^2)^2-4x}{1-x/r}}.$$

By using l'Hôpital's Rule, we obtain

$$L \;=\; -\frac{1}{2}\sqrt{\frac{-4r(1-r^2)-4}{-1/r}} \;=\; -\sqrt{r+r^2(1-r^2)} \;=\; -0.61265067.$$

11.4.15. By techniques we have used before,

$$H(x) \;=\; x\sum_{k\ge 2} H(x)^k + x.$$

Sum the geometric series and use algebra to obtain the desired quadratic equation for $H(x)$.

This quadratic could be treated as an implicit equation for $H(x)$ and we could apply Principle 11.7 (p. 353). Alternatively, we could solve the quadratic for $H(x)$ and use Principle 11.6 (p. 349). For Principle 11.7, let $F(x,y) = y^2 - y + \frac{x}{1+x}$. Then $F_y(x,y) = 2y - 1$ and so $s = 1/2$ and $\frac{r}{1+r} = 1/2 - (1/2)^2$, which yields $r = 1/3$. For Principle 11.6,

$$H(x) \;=\; \frac{1-\sqrt{1-4x/(1+x)}}{2} \;=\; \frac{1-\sqrt{(1-3x)/(1+x)}}{2}.$$

Thus $r = 1/3$, $f(x) = (1 - x/r)^{1/2}$, $g(x) = -1/2(1 + x)^{1/2}$ and $h(x) = 1/2$. In any case, the answer is

$$a_n \sim \frac{\sqrt{3}\, 3^n}{4\sqrt{\pi n^3}}.$$

11.4.17. We will use Principle 11.6 (p. 349). We want to solve

$$\sum_{k \in D} r^k/k! = 1$$

for r because then we have a singularity due to division by zero. This can always be done:

- The radius of convergence of the sum is ∞.
- The sum vanishes at $r = 0$.
- The sum is increasing and unbounded as $r \to +\infty$.

Having found r, we let $f(x) = (1 - x/r)^{-1}$. Then $g(x) = (1 - x/r)A(x)$ and, by l'Hôpital's Rule,

$$L = \lim_{x \to r} \frac{1 - x/r}{1 - \sum_{k \in D} x^k/k!} = \frac{1}{\sum_{k \in D} r^k/(k-1)!}$$

Let $d = \gcd(D)$. You should be able to see that $a_n = 0$ when n is not a multiple of d. Hence we'll need to assume that $d = 1$. (Actually you can get around this by setting $x^d = z$, a new variable.) The answer for general d is

$$a_n \sim \frac{d\, n!\, r^{-n}}{\sum_{k \in D} r^k/(k-1)!} \quad \text{when } d \text{ divides } n$$

and $a_n = 0$, otherwise.

11.4.19 (a) We have

$$B(x) = \sum_{t=0}^{k} \binom{k}{t} f(x)^t g(x)^t h(x)^{k-t}.$$

Apply the principle to each term in the sum. Since $c < 0$, the largest contribution comes from the $t = k$ term. Thus $b_n \sim g(r)^k n^{-ck-1}/\Gamma(-ck)$.

(b) Proceed as in the previous part. Since $c > 0$, the largest contribution is now from $t = 1$ and so $b_n\ k a_n h(r)^{k-1}$.

(c) The formula for $B(x)$ is just a bit of algebra. The only singularity on $[-r, r]$ is due to $f(x) = (1 - x/r)^{1/2}$.

(d) Since $A(x)$ is a sum of nonnegative terms, it is an increasing function of x and so $A(x) = 1$ has at most one positive solution. We take $b = 0$, $c = -1$ and $f(x) = (1 - x/s)^{-1}$ in Principle 11.6. Then

$$\lim_{x \to s} \frac{(1 - A(x))^{-1}}{(1 - x/s)^{-1}} = \lim_{x \to s} \frac{1 - x/s}{1 - A(x)} = \frac{1}{sA'(s)}$$

by l'Hôpital's Rule.

Suppose that $c < 0$. Note that $A(0) = 0$ and that $A(x)$ is unbounded as $x \to r$ because $A(x)/(1 - x/r)^c$ approaches a nonzero limit. Thus $A(x) = 1$ has a solution in $(0, r)$ by the Mean Value Theorem for continuous functions.

(e) If we could deal with $e^{A(x)}$ using Principle 11.6, we could multiply $g(x)$ and $h(x)$ in the principle by $e^{s(x)}$, where $s(x)$ is either of the sums given in this exercise. Then we can apply Principle 11.6.

11.4.21 (b) Let $U(x) = T(x)/x$. The equation for U is $U = \sum x^d U^d/d!$. Replacing x^k with z, we see that this leads to a power series for U in powers of $z = x^k$. Thus the coefficients of x^m in $U(x)$ will be 0 when m is not a multiple of k.

(c) We apply Principle 11.7 (p. 353) with

$$F(x,y) \;=\; y - x\sum_{d\in D} y^d/d! \quad\text{and}\quad F_y(x,y) = 1 - x\sum_{\substack{d\in D\\ d\neq 0}} y^{d-1}/(d-1)!.$$

Using $F(r,s) = 0 = F_y(r,s)$ and some algebra, we obtain

$$\sum_{\substack{d\in D\\ d\neq 0}} (d-1)s^d/d! \;=\; 1 \quad\text{and}\quad r \;=\; \Bigg(\sum_{\substack{d\in D\\ d\neq 0}} s^{d-1}/(d-1)!\Bigg)^{-1}.$$

Once the first of these has been solved numerically for s, the rest of the calculations are straightforward.

Appendix A

A.1. $\mathcal{A}(n)$ is the formula $1+3+\cdots+(2n-1) = n^2$ and $n_0 = n_1 = 1$. $\mathcal{A}(1)$ is just $1 = 1^2$. To prove $\mathcal{A}(n+1)$ in the inductive step, use $\mathcal{A}(n)$:

$$(1+3+\cdots+(2n-1)) + (2n+1) = n^2 + (2n+1) = (n+1)^2.$$

A.3. Let $\mathcal{A}(n)$ be $\sum_{k=1}^{n-1}(-1)^k k^2 \;=\; (-1)^{n-1}\sum_{k=1}^{n} k$. By (A.1), we can replace the right hand side of $\mathcal{A}(n)$ by $(-1)^{n-1}n(n+1)/2$, which we will do. It is easy to verify $\mathcal{A}(1)$. As usual, the induction step uses $\mathcal{A}(n-1)$ to evaluate $\sum_{k=1}^{n-1}(-1)^{k-1}k^2$ and some algebra to prove $\mathcal{A}(n)$ from this.

What would have happened if we hadn't thought to use (A.1)? The proof would have gotten more complicated. To prove $\mathcal{A}(n)$ we would have needed to prove that

$$(-1)^{(n-1)-1}\sum_{k=1}^{n-1} k + (-1)^{n-1}n^2 \;=\; (-1)^{n-1}\sum_{k=1}^{n} k.$$

At this point, we would have to prove this result separately by induction or prove in using (A.1).

A.5. The claim is true for $n = 1$. For $n+1$, we have

$$(x^{n+1})' \;=\; (x^n x)' \;=\; (x^n)'x + (x^n)x' = (nx^{n-1})x + x^n,$$

where the last used the induction hypothesis. Since the right side is $(n+1)x^n$, we are done.

A.7. The inductive step only holds for $n \geq 3$ because the claim that P_{n-1} belongs to both groups requires $n-1 \geq 2$; however, $\mathcal{A}(2)$ was never proved. (Indeed, if $\mathcal{A}(2)$ is true, then $\mathcal{A}(n)$ is true for all n.)

A.9. This is obviously true for $n = 1$. Suppose we have a numbering when $n-1$ lines have been drawn. The nth line divides the plane into two parts, say A and B. Assign all regions in A the same number they had with $n-1$ lines and reverse the numbering in B.

Section B.1

B.1.1. We'll omit most cases where the functions must be nonnegative. Also, the proofs for O properties are omitted because they are like those for Θ without the "A" part of the inequalities. All inequalities are understood to hold for some A's and B's and all sufficiently large n.

(a) Note that $g(n)$ is $\Theta(f(n))$ if and only if there are positive constants B and C such that $|g(n)| \le B|f(n)|$ and $|f(n)| \le C|g(n)|$. Let $A = 1/C$. Conversely, if $A|f(n)| \le |g(n)| \le B|f(n)|$, let $C = 1/A$ and reverse the previous steps.

(b) These follow easily from the definition.

(c) These follow easily from the definition.

(d) We do Θ using A. Let $A' = A|C/D|$ and $B' = B|C/D|$. Then $A'|Df(n)| \le |Cg(n)| \le B'|Df(n)|$.

(e) Use (a): We have $(1/B)|g(n)| \le |f(n)| \le (1/A)|g(n)|$.

(f) See proof in the text.

(g) We do $\Theta()$ and use (a). We have $A_i|f_i(n)| \le |g_i(n)| \le B_i|f_i(n)|$. Multiplying and using (a) with $A = A_1A_2$ and $B = B_1B_2$ gives the first result. To get the second part, divide the $i = 1$ inequalities by the $i = 2$ inequalities (remember to reverse direction!) and use (a) with $A = A_1/B_2$ and $B = B_1/A_2$.

(h) See proof in the text.

(i) This follows immediately from (h) if we drop the subscripts.

B.1.3. This is not true. For example, n is $O(n^2)$, but n^2 is not $O(n)$.

B.1.5 (a) Hint: There is an explicit formula for the sum of the squares of integers.

(b) Hint: There is an explicit formula for the sum of the cubes of integers.

(c) Hint: If you know calculus, upper and lower Riemann sum approximations to the integral of $f(x) = x^{1/2}$ can be used here.

B.1.7 (a) Here's a chart of values.

	5	10	30	100	300
n^2	25	10^2	9×10^2	10^4	9×10^4
$100n$	5×10^2	10^3	3×10^3	10^4	3×10^4
$100(2^{n/10} - 1)$	41	10^2	7×10^2	10^5	10^8
fastest	A	$A,\ C$	C	$A,\ B$	B
slowest	B	B	B	C	C

(b) When n is very large, B is fastest and C is slowest. This is because, (i) of two polynomials the one with the lower degree is eventually faster and (ii) an exponential function grows faster than any polynomial.

B.1.9. Let $p(n) = \sum_{i=0}^{k} b_i n^i$ with $b_k > 0$.

(a) Let $s = \sum_{i=0}^{k-1} |b_i|$ and assume that $n \ge 2s/b_k$. We have

$$|p(n) - b_k n^k| \le \left|\sum_{i=0}^{k-1} b_i n^i\right| \le \sum_{i=0}^{k-1} |b_i| n^i \le \sum_{i=0}^{k-1} |b_i| n^{k-1} = sn^{k-1} \le b_k n^k/2.$$

Thus $|p(n)| \ge b_k n^k - b_k n^k/2 \ge (b_k/2)n^k$ and also $|p(n)| \le b_k n^k + b_k n^k/2 \le (3b_k/2)n^k$.

(b) This follows form (a) of the theorem.

(c) By applying l'Hospital's Rule k times, we see that the limit of $p(n)/a^n$ is $\lim_{n\to\infty}(k!/(loga)^k)/a^n$, which is 0.

(d) By the proof of the first part, $p(n) \le (3b_k/2)n^k$ for all sufficiently large n. Thus we can take $C \ge 3b_k/2$.

(e) For $p(n)$ to be $\Theta(a^{Cn^k})$, we must have positive constants A and B such that $A \le a^{p(n)}/a^{Cn^k} \le B$. Taking logarithms gives us $\log_a A \le p(n) - Cn^k \le \log_a B$. The center of this expression is a polynomial which is not constant unless $p(n) = Cn^k + D$ for some constant D, the case which is ruled out. Thus $p(n) - Cn^k$ is a nonconstant polynomial and so is unbounded.

B.1.11 (a) The worst time possibility would be to run through the entire loop because the "`If`" always fails. In this case the running time is $\Theta(n)$. This actually happens for the permutation $a_i = i$ for all i.

(b) Let N_k be the number of permutations which have $a_{i-1} < a_i$ for $2 \le i \le k$ and $a_k > a_{k+1}$. (There is an ambiguity about what to do for the permutation $a_i = i$ for all i, but it contributes a negligible amount to the average running time.) The "`If`" statement is executed k times for such permutations. Thus the average number of times the "`If`" is executed is $\sum kN_k/n!$. If the a_i's were chosen independently one at a time from all the integers so that no adjacent ones are equal, the chances that all the k inequalities $a_1 < a_2 < \cdots < a_k > a_{k+1}$ hold would be $(1/2)^k$. This would give $N_k/n! = (1/2)^k$ and then $\sum_{k=0}^{\infty} kN_k/n!$ would converge by the "ratio test." This says that the average running time is bounded for all n. Unfortunately the a_i's cannot be chosen as described to produce a permutation of $\underline{n}$.

We need to determine N_k. With each arbitrary permutation $a_1, a_2, \ldots$ we can associate a set of permutations $b_1, b_2, \ldots$ counted by N_k. We'll call this the set for $a_1, a_2, \ldots$. For $i > k+1$, $b_i = a_i$, and $b_1, \ldots, b_{k+1}$ is a rearrangement of $a_1, \ldots, a_{k+1}$ to give a permutation counted by N_k. How many such rearrangements are there? b_{k+1} can be any but the largest of the a_i's and the remaining b_i's must be the remaining a_i's arranged in increasing order. Thus there are k possibilities and so the set for $a_1, a_2, \ldots$ has k elements. Hence the set associated with $a_1, a_2, \ldots$ contains k permutations counted by N_k. Since there are $n!$ permutations, we have a total of $n!k$ things counted by N_k; however, each permutation $b_1, b_2, \ldots$ counted by N_k appears in many sets. In fact it appears $(k+1)!$ since any rearrangement of the first $k+1$ b_i's gives a permutation that has $b_1, b_2, \ldots$ in its set. Thus the number of things in all the sets is $N_k(k+1)!$. Consequently, $N_k = n!k/(k+1)!$.

By the previous paragraphs, the average number of times the "`If`" is executed is $\sum k^2/(k+1)!$, which approaches some constant. Thus the average running time is $\Theta(1)$.

(c) The minimum running time occurs when $a_n > a_{n+1}$ and this time is $\Theta(n)$. By previous results the maximum running time is also $\Theta(n)$. Thus the average running time is $\Theta(n)$.

Section B.3

B.3.1 (a) If we have know $\chi(G)$, then we can determine if c colors are enough by checking if $c \ge \chi(G)$.

(b) We know that $0 \le \chi(G) \le n$ for a graph with n vertices. Ask if c colors suffice for $c = 0, 1, 2, \ldots$. The least c for which the answer is "yes" is $\chi(G)$. Thus the worst case time for finding $\chi(G)$ is at most n times the worst case time for the NP-complete problem. Hence one time is O of a polynomial in n if and only if the other is.