Answers

Chapter 1

1.1 All letters except for A work as generators. The problem can be expressed in modular calculations like this: g is a generator if there for each element x exists an element y such that $yg = x \pmod{29}$, since the Swedish alphabet has 29 letters. According to *Discrete Mathematics and Discrete Models*, this holds for gs that are coprime to 29, that is, all numbers from 1 to 28 since 29 is a prime number.

1.2

(a) PERFECT

(b) Both the messages M and the codes C consist of strings from the normal alphabet.

(c) KIMMO

1.3 $\Theta(N^2)$ and $\Theta(N)$, respectively.

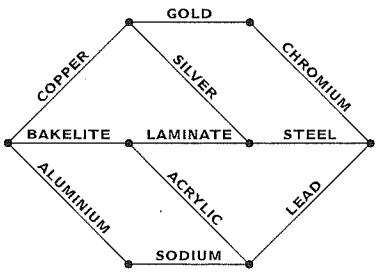
1.4 You just turn the cycles around, and the the letters are rotated back:

$$(G \quad N \quad R_2 \quad H \quad S \quad T \quad F \quad R_1 \quad P)$$
$$(I_2 \quad A \quad I_1 \quad O)$$

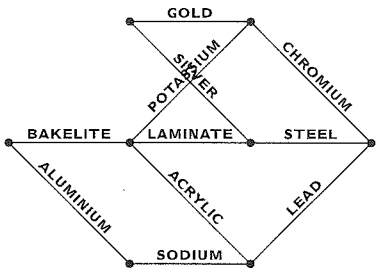
1.5 The partitions of 6 where the greatest part is 3 are: $3+3$, $3+2+1$, and $3+1+1+1$. The general statement is shown using conjugation in chapter 6.

1.6

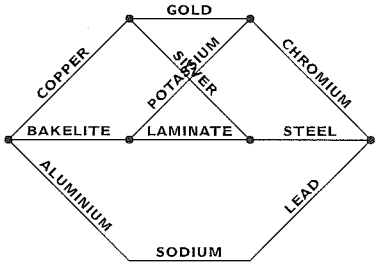
(a)

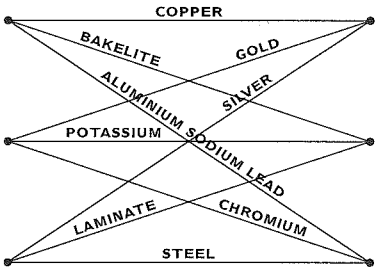


(b) No! Chapter 7 describes how this can be proved by finding six nodes connected to each other as in the complete bipartite graph $K_{3,3}$.

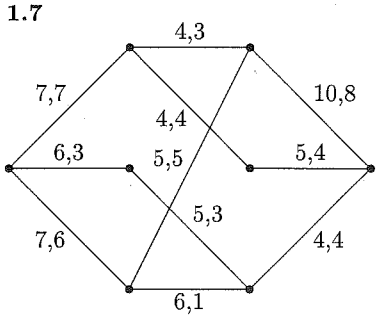


Remove the acrylate pipe and draw the aluminium-sodium-lead pipes as a single edge, and you'll see $K_{3,3}$.





1.7



(There are alternative, just as good, solutions. But in all of them, the copper,

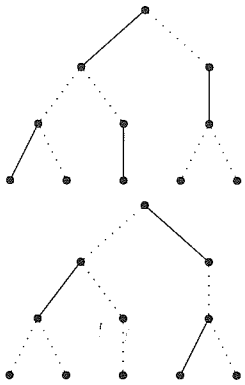
laminate, and lead pipes will be filled to max.) In chapter 8, an algorithm for finding maximum flows in a systematic way will be presented.

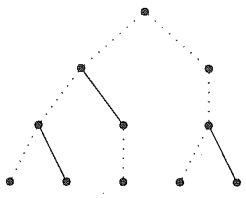
1.8

(a) Each discussion is beween a person on an odd level (level 1 or 3) and a person an an even level (2 or 4). At level 1 and 3 there are in total four persons, who then can participate in at most four discussions at a time.

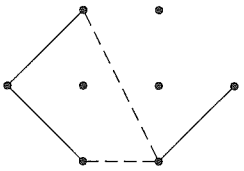
(b) Several nodes are of degree three, that is, the persons have to participate in three discussion. Thus at least three rounds of discussions are needed.

(c)





1.9 Since the competitor starts, she can't be stopped from placing her two pipes from the inlet. Her third pipe she wants to place at the outlet so that both ways to her earlier pipes still are usuable – then she is certain to win, since Good Connections Inc. only are able to hide one of them in their next move.



Such a pipe must be possible to put in place, since there are three alternatives and Good Connections Inc. can by their up until now hidden pipes only have sabotaged two of them.

Chapter 2

2.1

(a) Closed, since an integer minus an integer is an integer. Not associative, for instance $(1-2)-3 = -1-3 = -4$ while $1-(2-3) = 1-(-1) = 1+1 = 2$. Furthermore no element can be found that works as a unit both from the left and from the right. (Zero works from the right, since $x-0=x$, but not from the left.) If you don't have a unit discussing inverses becomes a bit tricky, but if we pretend that zero is the unit, every number is its own inverse, since $x-x=0$. Not a group.

(b) The calculations are closed, associative, and have a unit, namely one. But there is a number which lacks an inverse: zero. There is no number which multiplied by zero gives one. Not a group. (But $\langle \mathbb{Q} \setminus \{0\}, \cdot \rangle$ is a group!)

(c) The calculations are closed (the union of two sets is a set); the calculations are associative; there is a neutral element, namely the empty set. But no inverses exist! Not a group.

(d) Group.

(e) Group.

2.2 It consists of a single element, that is both neutral element and its own inverse. The thrilling calculation table is (if the element is named I) $I * I = I$.

2.3

(a) Assume that we have two identities, i_1 and i_2. Then what happens if we combine them?

$$\begin{aligned} i_1 &= i_1 * i_2 && \{\text{since } i_2 \text{ identity}\} \\ &= i_2 && \{\text{since } i_1 \text{ identity}\} \end{aligned}$$

They are identical!

(b) Assume that the element a has the two inverses b and c. Then we get

$$\begin{aligned} b = b * \mathrm{id} = b * (a * c) = \\ = (b * a) * c = \mathrm{id} * c = c \end{aligned}$$

2.4 You have to check that H is closed, so that it's impossible to end up outside it. Furthermore you have to check that all the elements of H got to bring their inverses. The associativity, on the other hand, is inherited from G, and doesn't need to be checked again. (If the operation is associative for all elements in G it must, among other things, be so for the elements in H.) And if H is closed and all elements have their inverses, then for instance $h \cdot h^{-1} = e$ has to belong to H, that is to say, the identity is guaranteed to be there *provided that* there is anything at all in H, that is to say, provided that H is non-empty.

2.5

(a) No, the elements (except for 1) lack inverses. There is for instance no integer n such that $17n = 1$.

(b) No, the set isn't closed. An odd number plus an odd number is an *even* number, not an odd one!

(c) Yes.

2.7 If there is a row where some element doesn't appear the row must contain some other element twice. That means that there are two multiplications by the same element, say $x * y$ och $x * z$ where y and z are different elements, that give the same result. Multiply both the expressions from the left by x^{-1} and we have $x^{-1} * x * y = x^{-1} * x * z$. This equality can be simplified to $I * y = I * z$, that is, to $y = z$, which contradicts the assumption that there were two equal elements in the same row in the group table.

2.8

$\circ$	Id	f
Id	Id	f
f	f	Id

2.9 All the orders were divisors of 6, that is to say, of the number of elements in the set. (This applies to all finite groups, see Lagrange's theorem below.) Furthermore we have the relationship $\text{order}(k) = 6/\gcd(6, k)$. (The same thing applies in all groups $\langle \mathbb{Z}_n, + \rangle$.)

2.10 Follows directly from the power rules.

2.11 It's enough to check that it's closed. If it is and the element g belongs to the subset then g^2, g^3, and so on will belong as well. And sooner or later one of the powers will be equal to g^{-1}.

2.12 No. 1 has the order 1, the remaining 3 elements have the order 2.

2.13 It's cyclic; as a generator you can use 3 or 7.

2.14 Each element generates a cyclic subgroup of the same size as the order of the element. The size of this subgroup divides according to Lagrange's theorem the size of the group.

2.15

(a) We have to show that for Abelian groups it always holds that $g * H = H * g$. The left hand side equals $\{g * h \mid h \in H\}$ and the right hand side equals $\{h * g \mid h \in H\}$ and since $g * h = h * g$ always holds in an Abelian group these two sets are identical.

(b) It's easily verified that $f * \langle g \rangle = \langle g \rangle * f$ and for reasons of symmetry this holds even if f is exchanced for some other element in $G \setminus \langle g \rangle$. Thus $\langle g \rangle$ is a normal subgroup. But $g * \langle f \rangle \neq \langle f \rangle * g$ so $\langle f \rangle$ isn't a normal subgroup.

(c) The closedness axiom is the central thing here: Since left and right cosets are alike it holds that

$$\begin{aligned}(H * g_1) * (H * g_2) &= \\ &= H * (g_1 * H) * g_2 \\ &= H * H * g_1 * g_2 \\ &= H * (g_1 * g * 2)\end{aligned}$$

for all $g_1, g_2 \in G$. Associativity is inherited by $*$. The identity is H, since

$$\begin{aligned}H * (H * g) = H * g = \\ = g * H = (g * H) * H\end{aligned}$$

for all $g \in G$. The inverse of $H * g$ is $H * g^{-1}$, since

$$\begin{aligned}(H * g^{-1}) * (H * g) = \\ = H * g^{-1} * g * H = H * H = H.\end{aligned}$$

2.16 The tables are

$+$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,0)$	$(1,1)$	$(1,0)$
$(1,0)$	$(1,0)$	$(1,1)$	$(0,0)$	$(0,1)$
$(1,1)$	$(1,1)$	$(1,0)$	$(0,1)$	$(0,0)$

$\cdot$	1	3	5	7
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

They have exactly the same structure as it is, so we just have to pair the elements in corresponding positions: $\phi\big((0,0)\big) = 1$, $\phi\big((0,1)\big) = 3$, $\phi\big((1,0)\big) = 5$, $\phi\big((1,1)\big) = 7$ is an isomorphism.

2.17

(a) For arbitrary $g \in G$ it holds that

$$\begin{aligned}&\phi(0)\cdot\phi(g) = \\ &\{\text{since isomorphism}\} \\ &= \phi(0+g) \\ &= \phi(g)\end{aligned}$$

which means that $\phi(0) = 1$.

(b) For arbitrary $g \in G$ it holds that

$$\begin{aligned}&\phi(g)\cdot\phi(-g) = \\ &\{\text{since isomorphism}\} \\ &= \phi\big(g+(-g)\big) \\ &= \phi(0) \\ &\{\text{according to previous exercise}\} \\ &= 1\end{aligned}$$

and thus $\phi(-g) = \big(\phi(g)\big)^{-1}$.

2.18 For each element g in the group G we define a bijection f_g on G as $f_g(x) = g * x$ for all $x \in G$. Let F_g denote the set of all f_g:s. Thereby F_g is a subset of the symmetric group S_g. Clearly the mapping $g \mapsto f_g$ is a bijection from G to G_g, and for all $x \in G$ it holds that

$$f_g \circ f_h(x) = g * h * x = f_{g*h}(x)$$

and thus $f_g \circ f_h = f_{g*h}$. Thereby F_g is a group isomorphic to G and a subgroup of S_G.

2.19 The isomorphism is simply logarithms, and the inverse is exponentiation. You find the logarithms of the two factors, add the logarithms, and exponentiate, like $100 \cdot 1000 = 10^2 \cdot 10^3 = 10^{2+3} = 10^5 = 100\,000$, to use two easily handled numbers as a demonstration. The conversions were usually done using tables. (It's not necessary to use the base 10, but it's practical when using decimal notation.)

2.20 The most famous marginal note is of course the one where Fermat formulated his famous hypothesis, see section 11.4.

2.21 Follows directly from exercise 2.6.

2.22 The set of polynomials with integer coefficients is closed under both addition and multiplication. The identity elements are 0 and 1, respectively. The additive inverse of a polynomial $p(x)$ is simply $-p(x)$. Thereby $\mathbb{Z}[x]$ is a ring. (It's not a field, though, since even if multiplication is commutative, the only elements in the set having a multiplicative inverse are ± 1.)

2.23 Nothing stops you from having *several* minimal elements, all of them placed along the bottom edge, which are unrelated to each other. On the other hand, there can only exist *one* element that is the least. A least element is always a minimal element, but the reverse doesn't hold.

2.24 Ason is the only maximal element, and thereby greatest as well. Eson, Lson, Mson, Gson, Nson, Oson, Ison, Json, Pson, Qson, and Rson are all minimal elements, no least element exists.

2.25

(a) That said person is the boss of both of them.

(b) That said person is a subordinate of both of them.

(c) Upper bounds: Cson and Ason. Lower bounds: None.

(d) Upper bounds: Dson and Ason. Lower bounds: Kson, Pson, Qson, and Rson.

2.26 A lower bound of m and n divides both the numbers, that is, it's a common divisor of them. An upper bound is divided by both the numbers, that is, it's a common multiple. The common divisors of 12 and 8 are 4, 2, and 1.

2.27 $\forall z\big[z = \text{glb}(x,y) \leftrightarrow$
$(z \preceq x) \wedge (z \preceq y) \wedge$
$\forall w\big\{(w \preceq x) \wedge (w \preceq y) \rightarrow (w \preceq z)\big\}\big]$

2.28 Least common multiple and greatest common divisor.

2.29 Cson and nobody, and Kson and Dson, respectively.

2.30 No. If you have for instance two different maximal elements no upper bound of them will exist, and the no least upper bound either. So you can have at most one maximal and one minimal element.

2.32 A lattice is a partially ordered set, and can be drawn as a Hasse diagram. If you have a greatest and a least element, the diagram will converge into a point at top and at bottom. But nothing prevents it from having an infinte number of levels inbetween (which the power set of $\mathbb{N}$ has) or there being an infinite number of elements on each level (which holds for the subset-relation on $\mathcal{P}(\mathbb{N})$ as well). Bounds don't guarantee finiteness.
On the other hand, a finite lattice has to be bounded.

2.33

(a) No group, since the operation isn't closed. A vector dot another vector equals a *number*.

(b) No group, since the operation isn't associative. For instance $\big((1,0,0)\times(1,0,0)\big)\times(0,1,0) = (0,0,0)\times(0,1,0) = (0,0,0)$ while $(1,0,0)\times\big((1,0,0)\times(0,1,0)\big) = (1,0,0)\times(0,0,1) = (0,-1,0)$. Nor does there exist any neutral element, since the cross product of two vectors is perpendicular to them both and thus not equal to either of them. And if there is no neutral element, talking about inverses isn't meaningful.

(c) Abelian group. The neutral element is the zero vector, $(0,0,0)$.

2.34 We denote the "invese of a" by a^{-1} and the unit element in the group by e

$$\begin{aligned} a*b &= a*c \\ a^{-1}*(a*b) &= a^{-1}*(a*c) \\ (a^{-1}*a)*b &= (a^{-1}*a)*c \\ e*b &= e*c \\ b &= c \end{aligned}$$

2.35 $(\pi*\sigma)^{-1}$ is by definition what you have to pair $\pi*\sigma$ with to get the identity. Let's see what happens if we pair $\pi*\sigma$ and $\sigma^{-1}*\pi^{-1}$:

$$\pi*\sigma*\sigma^{-1}*\pi^{-1} = \pi*\mathrm{id}*\pi^{-1} = \\ = \pi*\pi^{-1} = \mathrm{id}$$

Since we got the identity, what we used has to be the inverse!

2.36

(a) $2\mathbb{Z}$ does only have one coset besides itself, namely the numbers you get if you add a number outside $2\mathbb{Z}$ to a number in $2\mathbb{Z}$. The numbers outside are odd, and an odd number plus an even number is odd. We can denote the coset $2\mathbb{Z}+1$. The table is

	$2\mathbb{Z}$	$2\mathbb{Z}+1$
$2\mathbb{Z}$	$2\mathbb{Z}$	$2\mathbb{Z}+1$
$2\mathbb{Z}+1$	$2\mathbb{Z}+1$	$2\mathbb{Z}$

(b) The group, which consists of two elements, is isomorphic to $\langle\mathbb{Z}_2,+\rangle$.

(c) $\mathbb{Z}/2\mathbb{Z}$ is isomorphic to $\mathbb{Z}_2$, which isn't isomorphic to any subgroup of $\mathbb{Z}$, since all subgroups of $\mathbb{Z}$ except $\{0\}$ contain an infinite number of elements (since all numbers have an infinite order when using normal addition, since you never return to zero!)

2.37 An equivalence relation is reflexive, symmetric, and transitive. Two groups are isomorphic if there exists a bijection that preserves relationships between them.

- Each group is isomorphic to itself (with the identity functon as bijection), so the relation is reflexive.
- If there is a bijection ϕ from G_1 to G_2 then its inverse is a bijection in the other direction (bijections do always have an inverse), so the relation is symmetric.
- If there is a bijection ϕ from G_1 to G_2 and a bijection ψ from g_2 to G_3, then $\psi\circ\phi$ is a bijection from G_1 to G_3, and that it preserves the operations follows from the fact that the other bijections do. So the relation is transitive.

2.38

(a) $U_8 = \{1, 3, 5, 7\}$. $3^2 = 9 \equiv 1$, $5^2 = 25 \equiv 1$, $7^2 = 49 \equiv 1$. No, the group isn't cyclic, since no element exists that generates it.

(b) $U_9 = \{1, 2, 4, 5, 7, 8\}$. $2^2 = 4$, $2^3 = 8$, $2^4 = 16 \equiv 7$, $2^5 \equiv 14 \equiv 5$, $2^6 \equiv 10 \equiv 1$. We can already see that the group is cyclic, with 2 as a generator. The question is whether there are any more generators. We can without any calculations realise that order(4) = 3 and that order(8) = 2 (since $4 = 2^2$ and $8 = 2^3$) so it's enough to investigate the remaining elements:
$5^2 = 25 \equiv 7$, $5^3 \equiv 35 \equiv 8$. We can stop here, since the order has to be a divisor of the number of elements in the group, here 6, and we have by now passed all those except for 6 itself. 5 is a generator of the group.

Since $7 \equiv 5^2$ and 5 has the order 6, 7 has to have the order 3, and thus doesn't generate the group. The group is cyclic, with the two generators 2 and 5.

2.39

(a) S_6 contains $6! = 720$ elements. 7 doesn't divide 720, so it's impossible for a subgroup with this cardinality to exist.

(b) The subgroup then has to consist of two elements, out of which one has to be the identity function and the other one its own inverse. For instance $\{\mathrm{Id}, f\}$, where f is defined according to $1 \mapsto 2$, $2 \mapsto 1$, $3 \mapsto 3$, $4 \mapsto 4$, $5 \mapsto 5$, $6 \mapsto 6$ works fine. (There are several alternative answers!)

2.40

(a) Yes. Take two elements in $K \cap H$. They then belong to both K and H, so the combination of them will belong to both K and H as well (since the sets are closed) and thereby in $K \cap H$. And the same reasoning applies to the inverses. And the identity is included in every subgroup, and thereby in the intersection.

(b) No. The two subgroups are closed regarded separately, but nothing says that an element of K paired with an element of H has to end up anywhere special! If we for instance study the group $\langle \mathbb{Z}, + \rangle$ and its subgroups $2\mathbb{Z}$ (the even numbers) and $3\mathbb{Z}$ (the numbers divisible by 3) we see that $2 \in (2\mathbb{Z} \cup 3\mathbb{Z})$ and $3 \in (2\mathbb{Z} \cup 3\mathbb{Z})$ but $2+3 = 5 \notin (2\mathbb{Z} \cup 3\mathbb{Z})$. Not closed, not a group.

2.41

(a) This chapter has to be read first.

(b) This chapter can be read without having read anything else before it.

(c) The chapter requires that you have read both the others.

(d) Chapter 5 and chapter 9, which are unrelated. No lowest upper bound exists.

(e) There are no upper bounds at all, and definitly none that is lowest.

(f) Chapter 6 and chapter 4. Since chapter 6 is placed above chapter 4, chapter 6 is the greatest lower bound.

(g) Only chapter 4, which then of course is the greatest lower bound.

(h) Definitely not!

2.43 If $a \preceq b$ then $a \wedge b = a$ and $a \vee b = b$ (and the other way around if the relationship is inverted). Since one of the relationships is guaranteed to hold in a total order, we are guaranteed that $\wedge$ and $\vee$ will always be defined, which is the requirement for the order to be a lattice.

Chapter 3

3.1

(a) **Morse code**, for instance. It's made up from two symbols, long and short, and the lengths of the code words are different. The in the English language common letter e is encoded ·, while the less common letter h is encoded as the longer sequence · · · ·.

(b) For instance **ASCII-code** (American Standard Code for Information Interchange), which is commonly used to code characters in computers. There it's specified which sequence of ones and zeros that is to represent which letter.

3.2 $\log_2 n$ rounded upwards.

3.3

(a) If the probability of getting it wrong is 1 % then the probability of getting it right is 99 %. With ten digits that gives a probability of $0.99^{10} \approx 0.904 = 90.4$ %.

(b) We have to choose one out of the ten digit for the error, and the rest are to be correct: $10 \cdot 0.01 \cdot 0.99^9 \approx 0.091 = 9.1$ %.

(c) Choose two of the digits for the errors: $\binom{10}{2} \cdot 0.01^2 \cdot 0.99^8 \approx 0.004 = 0.4$ %.

We can see that it's true that it's more likely that we are affected by one error than by several. (We have already covered 99.9 %, so the probability of more that two errors has to be *very* low.)

3.4 That it contains different code words that are identical! Not an especially good idea, if you want people to understand what you mean. (It's true that the code *the English language* contains words that are written and/or pronounced in the same way but mean different things, but in spite of that you are usually able to understand the meaning based on the context.)

3.5

$\delta(010101, 101010) = 6$
$\delta(010101, 111000) = 4$
$\delta(010101, 000111) = 2$
$\delta(101010, 111000) = 2$
$\delta(101010, 000111) = 4$
$\delta(111000, 000111) = 6$

The minimum distance is 2.

3.8 000111, that differs from the recieved word at two places. The rest of the code words differ at at least three places.

3.9 We group the strings according to which code word they resembles the most:

$00000: \quad \{00000, 00001, 00010, 00100, 01000, 10000\}$
$00111: \quad \{00111, 00110, 00101, 00011, 01111, 10111\}$
$11100: \quad \{11100, 11101, 11110, 11000, 10100, 01100\}$
$11011: \quad \{11011, 11010, 11001, 11111, 10011, 01011\}$

The remaining strings can't be classified, since they are at the *same* distance from several of the code words. If we for instance look at 01001 we find that $\delta(01001, 00000) = 2 = \delta(01001, 11011)$. The given code has the minimum distance 3, which makes it possible to correct singel-bit errors but not necessarily two bit errors.

3.11 All the elements are their own inverses! (This is a rather unique thing, and simplifies later calculations a lot, when you've got used to it.)

3.13 Nothing says that the differences have to be different!

3.14 Dimension 1 means that we are to have $2^1 = 2$ words in the code. One of them has to be the neutral element of $(\mathbb{Z}_2)^4$: 0000. The second one (which at the same time is the inverse of itself) can be chosen freely. From an error analysis point of view we want to make the distance of the code as large as possible, that is to say, the two code words are to be as different as possible. Most stupid choice seems to be something on the line of $\{0000, 1000\}$, which has $\delta = 1$, most clever is $\{0000, 1111\}$, which has $\delta = 4$.

3.16 One method: firstly the subgroup has to contain the neutral element, the word with only zeros. Then we can add another word, which at the same time is it's own inverse. Then we have two code words. If that isn't enough we can add another word, and the word we get if we add the new word to the one we've already got. Then we have four words. If we want more than that, we take another word, and the words we get if we pair this word with the old words, and so on. If you want a small code this way of finding code words can be simpler than the one using the system of equations. But you won't get the features that will be described shortly.

3.17

(a) The product is 000, so the word belongs to the code.

(b) The product is 010, so the word doesn't belong to the code.

3.18 The product was 010, which is identical to the fourth column in the matrix. Thus it should be the fourth bit of the word that is wrong, and the correct message should have been 1111111 (which is a correct code word, which you'll see if you multiply it by the matrix).

3.19 Each column consists of four binary digits, so there are $2^4 = 16$ different versions. On of them consists of only zeros, thought. Furthermore we can't use two columns that are alike, so $16-1 = 15$ is the maximum number.

3.20

(a) If we have m rows, we can make at most $2^m - 1$ different non-zero columns. If the code is to have dimension k the number of rows, n, has to be k more than the number of rows, that is, $m = n - k$. To make it possible for all the columns to be different, the following condition has to be met:

$2^{n-k} - 1 \geq n$

The simplest way of solving an inequality of this kind is testing. $k = 3$ was given. $n = 5$ didn't work, since

$$2^{5-3} - 1 = 3 \not\geq 5$$

$n = 6$, on the other hand, works well:

$$2^{6-3} - 1 = 7 \geq 6$$

(b)

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

is a suggestion of a matrix. (There are several other matrices that work just as well.) This matrix generates the code {000000, 011001, 101010, 110011, 110100, 101101, 011110, 000111}.

(c) This time we are looking for the dimension, based on $n = 5$. $k = 3$ didn't work, as seen above. But

$$2^{5-2} - 1 = 7 \geq 5$$

A code with the dimension 2, that is to say, with $2^2 = 4$ code words, is possible to make.

(d)

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

is a suggestion. Gives the code {00000, 01101, 11010, 10111}.

(e) If we settle for the minimum distance two we can't have a column consisting of just zeros, but otherwise we have no restrictions. The more rows we have in proportion to the number of columns in the matrix, the smaller the dimension will be. The largest dimension we get when using the least number of rows, which ought to be one single row. One of the unknowns is expressed in the other four. That gives 16 code words.

(f) The somewhat degenerate matrix that achieves this (in this case there really is just one answer) is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

The code is {00000, 10001, 10010, 00011, 10100, 00101, 00110, 10111, 11000, 01001, 01010, 11011, 01100, 11101, 11110, 01111}. The first bit is the sum of the remaining four, which means that we have a code with even parity. (A code with odd parity won't be a *linear* code, since it won't contain the zero word.)

(g) If we don't care about error control we might as well take all 5-bit strings that exist, that is, all the $2^5 = 32$ ones! The seriously degenerate matrix that fix this is

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3.21 There are several different answers to the question, since both systems of equations can be written in several equivalent ways. Here is one version for the first code:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We see that there is a column consisting of zeros only, which gives that the code has the minimum distance 1. The other code can be generated from the matrix

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

No zero-column, no columns that are alike. Thus $\delta \geq 3$. (Actually, we have $\delta = 4$.)

3.22 We can use the same line of reasoning as for the single-bit errors, and partition the distance into a sum of words where a single bit is one. If we are to correct 2 errors the minimum distance has to be at least 5. Words containing 3 or 4 ones have to be disallowed. If a string having 3 ones is a solution to the system there has to exist three columns that summed give zero. Then one of them can be written as the sum of the other two. The same for 4 ones. Thus there mustn't be

- Columns consisting of zeros only.
- Two columns that are identical
- One column that is the sum of two others
- One column that is the sum of three others

The first two restrictions are easily checked using inspection, but the last two seem laborious to investigate. (One way of doing it is to have a list of possible columns, and every time you pick one column you strike out the columns that suddenly become disallowed from the list. If the matrix is finished before the list is exhausted it was possible to make up a code conforming to the specifications.) You can check that the second matrix in exercise 3.21 doesn't have

any column that is the sum of two others, while the last column is the sum of the other three, which gives that this code has $\delta = 4$.

3.23 Yes. It simply has to consist of one single code word. No matter what signal you recieve you interprete it as this word. It's perhaps not possible to express yourself in a very nuanced way, but there are applications for this thing as well. ("Are you alive?" is a typical case. If you get any answer whatsoever you can take for granted that the person is alive.)

3.24 Well, how do you reach an agreement? You can hardly discuss that via the unsafe channel! (Crypto keys have frequently been something that has been delivered by a courier.)

3.25 One method is trial and error. If it's shifted one step the first word of the message has to be "Wr", which doesn't seem very likely. We try two steps instead, then three, and so on. At four we hit something:

```
Xs fi sv rsx xs fi,
To be or not to be,
xlex mw xli uyiwxmsr
that is the question
```

3.26 The English alphabet consists of 26 letters. If you shift firstly 13 steps and then another 13 steps you are back at the starting point, so the operation is its own inverse! That means that the same program can be used both to encrypt and to decrypt, which is practical. The cipher is mostly used when you don't want anyone to be able to read something by *mistake*, since it isn't a least bit secure. (An example of an application is when you want to discuss the ending of a movie, but don't know if the other person has seen it or not. If they haven't, they don't want to know how it ends!)

3.27

(a) No. You can't be sure that you'll return to the beginning of the list. A simple example is if we want to calculate $2^{10} \pmod{16}$. Then we get the sequence 2, 4, 8, 0, 0, ...

(b) If we are calculating in a *group* we will, using this method, be generating a subgroup, and sooner or later return to the number we had at first. Multiplication modulo a prime number is a group operation (se exercise 2.6). Multiplication of the invertible numbers modulo a composite number is a group operation as well, so we'll return to the beginning of the list if (and only if) the modular base and the number of which we are seeking the powers are coprime.

3.28 If n is a power of two, let's say $n = 2^k$, we need k multiplikations. That's the best case. The worst case is if $n = 2^k - 1$. Then $k - 1$ multiplications are needed to find all the powers, and another $k-1$ to combine them. Best case is $\log_2 n$ operations, worst case $2\lfloor \log_2 n \rfloor$.

3.29 32 (mod 33)

3.33 We can for instance test encrypt all numbers between 1 and 516. Then we get a nice dictionary, with the help of which we can translate intercepted messages. This method can by the way also be used to confirm that an intercepted message is what you believe it is; you encrypt the assumed message. If you get the same thing as the intercepted message your guess was correct. (Military messages, which have rather standardised wordings, are susceptible to this kind of analysis. You can add an introduction consisting of nonsense to reduce the risk.)

3.34 $p = n/q = 1961/53 = 37$, which gives $m = (37 - 1)(53 - 1) = 1872$. Furthermore we have that $797 \cdot 101 = 1 + 43 \cdot 1872$, so the decryption key is $d = 101$. $444^{101} \equiv 777 \mod 1961$, so the secret message was 777.

3.35 $3^{90} \equiv 1 \pmod{91}$, but $2^{90} \equiv 64 \pmod{91}$. 91 is thus a pseudoprime, which by the way can be factorised as $7 \cdot 13$.

3.36 The sieve of Eratosthenes, writing a list of all the numbers up to the one of interest, and crossing out first every second one (the even numbers), then every third one, and so on until only the primes are left. Works efficiently up to the range 10^{10}, approximately.

Trial division, dividing the number by all numbers smaller than the square root of the number in case. If the division comes out evenly you have found a factor of the number. Can be speeded up if you happen to have a list of primes handy, because then you can settle for dividing by prime numbers. This as well works

efficiently for small numbers, but not on numbers in the range used in encryption.

Before starting any advanced test one can perform some very simple ones: If the last digit is even the number is divisible by two, if the last digit is zero or five the number is divisible by five, and if the sum of the digits is divisible by three the number is divisible by three. You remove a very large part of the failing candidates using these simple methods!

3.37

(a) All the 6 differences are codewords as well.

(b) $\delta = 5$, which means that the code corrects all errors on up to two bits.

(c) The code is characterised by the fact that the first five bits are identical, and the last five as well. The relationships $x_1 = x_2 = x_3 = x_4 = x_5$, $x_6 = x_7 = x_8 = x_9 = x_{10}$ can be summarised in the system of equations below, with the associated matrix:

$$\begin{cases} x_1 = x_5 \\ x_2 = x_5 \\ x_3 = x_5 \\ x_4 = x_5 \\ x_6 = x_{10} \\ x_7 = x_{10} \\ x_8 = x_{10} \\ x_9 = x_{10} \end{cases} \qquad \begin{pmatrix} 1\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1 \end{pmatrix}$$

(d) $\delta(0001100111, 0000000000) = 5$,
$\delta(0001100111, 1111111111) = 5$,
$\delta(0001100111, 0000011111) = 4$,
$\delta(0001100111, 1111100000) = 6$.
The word contains (at least) four errors, and should according to the nearest neighbour principle be corrected to 0000011111. We may observe that the answer in (b) guarantees that we can correct *all* errors on two bits; it doesn't say that it isn't possible to correct *some* errors on more bits than that.

(e) No. Then we would need $\delta = 7$, which means that all the words except for the zero has to contain at least 7 ones (at most 3 zeros). Two words with the length 10 and 7 ones can only differ at 6 places, which means that it's impossible to make up the last codeword.

(f) According to the sphere-packing theorem the following has to be true:

$$256 = 2^{10-2} \geq 1 + \binom{10}{1} + \binom{10}{2} + \binom{10}{3}$$
$$256 \geq 166$$

which is correct! We have to remember that the sphere-packing theorem is only able to say that something is impossible, not that it actually is possible.

3.38

(a) So we have $k = 8$ and $\delta = 3$. Then n has to satisfy

$$2^{n-8} - 1 \geq n$$

We test

$$n = 11 : 2^{11-8} - 1 = 7 \not\geq 11$$
$$n = 12 : 2^{12-8} - 1 = 15 \geq 12$$

The least possible length is 12 bits.

(b) We need $12 - 8 = 4$ rows, 12 columns, no zero-column, no columns that are alike, and 4 columns that are independent. That we can for instance fix using a Hamming matrix:

$$\begin{pmatrix} 0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,1 \\ 0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,0 \\ 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \end{pmatrix}$$

3.40

(a) $n_A = 91 = 7 \cdot 13$, so $m_A = 6 \cdot 12 = 72$. We are looking for d_A:

$$\begin{aligned} 72 &= 14 \cdot 5 + 2 \\ 5 &= 2 \cdot 2 + 1 \\ 1 &= 5 - 2 \cdot 2 \\ &= 5 - 2(72 - 14 \cdot 5) \\ &= 29 \cdot 5 - 2 \cdot 72 \end{aligned}$$
$$5 \cdot 29 = 1 + 2 \cdot 72$$
$$5^{-1} = 29 \pmod{72}$$

$n_B = 35 = 5 \cdot 7$, so $m_B = 4 \cdot 6 = 24$. We are looking for d_B:

$$\begin{aligned} 24 &= 3 \cdot 7 + 3 \\ 7 &= 2 \cdot 3 + 1 \\ 1 &= 7 - 2 \cdot 3 \\ &= 7 - 2(24 - 3 \cdot 7) \\ &= 7 \cdot 7 - 2 \cdot 24 \end{aligned}$$
$$7 \cdot 7 = 2 \cdot 24 + 1$$
$$7^{-1} = 7 \pmod{24}$$

$d_A = 29$, $d_B = 7$.

(b) B starts by decrypting using his own keys, and gets $23^7 = 2 \pmod{35}$. This result is encrypted using A:s keys. $2^5 = 32 \pmod{91}$. Thus B shall send 32.

3.41 The main trick is usually to mask the encrypted message so that it looks like something else. A message written in invisible ink in a completely normal letter, telegrams where the initial letters in the words spell the real message (usually fairly easy to detect), and other such methods are used.

One interesting modern version is to use pictures. A common way of storing pictures in a computer is to store it as a sequence of binary numbers, where every number tells what shade of gray a certain element in the picture is to have. If you use one byte per picture element (pixel) that gives 256 different levels of gray, which is more than what the human eye is able to differentiate. Because of this, you can borrow the least significant digit of each number, and use that for your message. The changes in the picture are not large enough for this to be noticeable. And files containing pictures from holidays and other stuff are regularly distributed! (One common criticism of NSA's efforts to keep the rest of the world from getting access to working crypto systems is precisely that the drug maffia and terrorists probably use methods of this kind and not encryptors approved by the state.)

Chapter 4

4.1 In M we store the largest value found this far. We start at one end of the sequence. At start, the largest value found is simply the first value (since it's the only one we've had the opportunity to look at). Then we inspect the elements one by one, and if the one we are looking at is larger than the largest one this far we put it down instead of the old value. When we are finished, we have the largest number of the sequence.

4.2

(a) $\Theta(n^3)$. We may note that for small values of n, the quadratic term with its large coefficient will dominate, but for large enough ns the cubic term will "drive by".

(b) $\Theta\big((\log n)^2\big)$

(c) $\Theta(n2^n)$

4.3 So we are to find two constants a and b such that $an \le 3\log n + 2n \le bn$. Firstly, we know that $0 \le 3\log n$ (at least if $n \ge 1$, which we can take for granted). That gives

$$2n = 0 + 2n \le 3\log n + 2n$$

Furthermore we know that $\log n \le n$ (that's a consequence of the definition of logarithm). That gives

$$3\log n + 2n \le 3n + 2n = 5n$$

We have found working values on the constants: $a = 2$, $b = 5$.

$$2n \le 3\log n + 2n \le 5n \qquad \text{if } n \ge 1$$

(There are lots of other values that work just as well.)

4.4

$$f_1(n) \ll g_1(n) \quad \Leftrightarrow \quad \lim_{n\to\infty} \frac{f_1(n)}{g_1(n)} = 0$$

$$f_2(n) \ll g_2(n) \quad \Leftrightarrow \quad \lim_{n\to\infty} \frac{f_2(n)}{g_2(n)} = 0$$

This gives (according to established rules for limits)

$$\lim_{n\to\infty} \frac{f_1(n)f_2(n)}{g_1(n)g_2(n)} =$$
$$= \lim_{n\to\infty} \frac{f_1(n)}{g_1(n)} \lim_{n\to\infty} \frac{f_2(n)}{g_2(n)}$$
$$= 0 \cdot 0 = 0$$

4.5

(a) $a_n = (-3)^n + 4^n$

(b) $a_n = 2 - 2^n + 3^n$

(c) $b_n = (3n+5)2^n$

(d) $f_n = \dfrac{1}{\sqrt{5}}\left(\dfrac{1+\sqrt{5}}{2}\right)^n - \dfrac{1}{\sqrt{5}}\left(\dfrac{1-\sqrt{5}}{2}\right)^n.$

It's rather funny that an integer problem can have an answer looking like this! All the irrational parts cancel out when you substitute an integer value of n.

4.6

(a) $a_n = 3^n - n + 2$

(b) $b_n = 2^n + 3^n + 4^n$

(c) $c_m = 2^m \cdot m$

4.8 The proportion is

$$\frac{1}{2}(1-\frac{7}{8}(\frac{4}{5})^{n-1})$$

The reason the results differ is that you can have a one first in a number, but you can't have a zero there.

4.10 For all $n \geq 0$, $-n \leq 0$, obviously holds. That substituted gives

$$\frac{1}{2}(n^2-n) \leq \frac{1}{2}(n^2-0) = \frac{1}{2}n^2$$

4.11 How much time it takes in the *best* case might be of interest as well. But most thrilling is actually how much time it takes in the *average* case! Besides, not only the usage of time is interesting; the usage of memory is important as well. (If the algorithm requires too much memory, operations for the handling of memory are needed as well, and then the computation may jump into a completely different complexity class.)

4.16 Merge-Sort uses exactly the same amount of time irrespective of whether the input is a complete mess or sorted! Bubble-Sort on the other hand won't perform any exchanges if the input is in correct order.

4.17 It depends a bit on how you have implemented the algorithm, but in principle twice the length of the sequence. One list where data is taken and one where it is placed. For large amounts of data, this can be problematic. There are other sorting algorithms that are as fast as Merge-Sort, but don't require more space than Bubble-Sort, and they are the ones mostly used. Merge-Sort is used when one has amounts of data that don't fit into primary memory anyway, and has to be read and written to a file during sorting.

4.18 To find the smallest element, one has to look through the whole sequence, which gives $n-1$ comparisons. (You compare the first element and the second, the smaller of the first and second and the third, and so on.) Finding the second smallest takes $n-2$ comparisons, ans so on. In total $(n-1)+(n-2)+\cdots+1 = n(n-1)/2$ comparisons, the same as for Bubble-Sort.

4.19 If you are unlucky, you have to move something in all the steps. That gives $n-1$ exchanges, to compare with the $n(n-1)/2$ used by Bubble-Sort.

4.20

(a) 1 ms and 0.2 ms, respectively.

(b) 0.3 s and 0.07 s respectively.

(c) 69 s and 17 s respectively.

4.21 Exacly what numbers you got we can't see here, but one finds that even if Bubble-Sort and Straight-Selection *in principle* are just as good, $O(n^2)$, *in practice* they are differently effective. You can't just look at the general time complexity when you evaluate an algorithm.

4.24 2^m categories.

4.25 $n!$.

4.26 We have to solve

$$2^m = n!$$

We take logarithms and use Stirling's formula:

$$\begin{aligned} m &= \log_2 n! \\ &\approx \log_2(\sqrt{2\pi n}(\frac{n}{e})^n) \\ &= \log_2\sqrt{2\pi} + \frac{1}{2}\log_2 n \\ &\quad + n\log_2 n - n\log_2 e \end{aligned}$$

The dominating term is $n\log n$.

4.27 The sorting problem doesn't get any easier by being generalised.

4.29 What you have to do is to check that the graph is connected and that all nodes (exept for maybe two) have an even degree. Checking the degrees of the nodes runs in linear time. (Look at one node at a time.) Checking that the graph is connected can be done by looking at a node. Write down all nodes connected to this node by an edge. For all these nodes, write down all the nodes that are reachable, and add the new ones to the sequence. For the new ones, write down which are reachable, and so on until no new nodes are added. Since the maximum number of edges in a graph is $n(n-1)/2$, this runs in polynomial time. And if the sequence when we are done consists of all the nodes, the graph is connected.

4.30 If somebody says that there is a Hamiltonian path, said person should be able to show us the Hamiltonian path. We have to check that it is a path (linear in the number of edges), that all the

edges used actually exist in the graph (linear in the number of edges), and that all the nodes are included (can be done while we investigate that it is a path). So it is possible to verify the solution in polynomial time, and thereby the problem belongs to NP.

4.31 "Does a clique of this size exist?" and "Does a route with this length exist?", respectively.

4.32 The precise time complexity depends a bit on the implementation. Here the graph is assumed to be stored in a matrix A, where $a_{i,j}$ is the weight of the edge between node i and node j. We note that if the matrix fits into primary memory, you can pick elements from it in constant time, if you know which one you are looking for.

We start by making a list of the edges in order of weight. The number of edges is $\binom{n}{2} = n(n-1)/2$, so the sorting ought to run in $\Theta(n^2 \log n)$. At the same time, we take the opportunity of listing the existing nodes and writing down their number. In the list over the nodes, we also note how many times they have been included in the path up to this point. (From start zero times, at the end they should all be included twice.) For those that are included we also note where in the list over taken edges they are (which means that we can search this in constant time).

In a list like this, where the elements are numbered consecutively, we can find the infomation about a specific node in constant time as well. Furthermore, we make a list over the edges used, and create a variable containing accumulated weight.

We start by picking the lightest edge, update the information about its nodes, add its weight to the weight variable, and remove the edge from the list over available edges. This runs in constant time.

After that, in each step we:

- Pick the shortest remaining edge. (Constant time.)
- Check how many edges that are connected to its ends (constant time). If two are connected to one of the ends, the edge is unusable, throw it away. If one or no edge is attached to either end and none to the other, the edge is usuable, keep it, increment the accumulator, increment the number of edges taken, insert it into the list of edges taken, and update the information about the ends (constant time). If edges are attached to both ends, the whole thing is a bit more complicated, since then the edge may tie the taken edges into a cycle. Take one end of the edge, find the connected edge in the list of edges, find the other end and keep on like that until we either reach a node that doesn't lead any further (in that case, the new edge has connected two pieces of path into a longer path) or until we get to the other end of the suggested edge (in that case it ties the ends of a path into a cycle, which we don't want). In the first case we keep the edge and update, in the latter case we throw it away. Searching the list of edges runs in linear time for each edge, and in the worst case we have to look through all the up until now chosen edges. So this may run in quadratic time.

When we have picked $n-1$ edges, we know that the last edge we need should connect the two remaining nodes with just one edge. Find them and the connecting edge. (Linear time.)

In the absolutely worst case, when the heaviest edge is part of the route, we have to look through all the $\Theta(n^2)$ edges. The investigation of an edge runs in the worst case in linear time in the number of taken edges. The number of taken edges can't exceed n. So worst case seem to be $O(n^3)$. In the best case, we simply have to use the first n edges on the list, and then the most time-consuming part is the making of the list.

4.35

(a) $\Theta(\log n)$ (note that $\log_2 n^2 = 2\log_2 n$).

(b) $\Theta(2^{2n})$.

(c) $\Theta(3^{2n})$.

4.36 For a start, both n^4 and $n \cdot 2^n$ are positive for positive n, so

$$0{,}01 \cdot 3^n = 0 + 0 + 0{,}01 \cdot 3^n \leq n^4 + n \cdot 2^n + 0{,}01 \cdot 3^n$$

if $n > 0$.

Furthermore, $n^4 \leq 3^n$ when $n \geq 8$ and $n \cdot 2^n < 3^n$ for all n, so

$$n^4 + n \cdot 2^n + 0{,}01 \cdot 3^n \leq 3^n + 3^n + 0{,}01 \cdot 3^n = 2{,}01 \cdot 3^n$$

when $n > 8$ so we have

$$0{,}01 \cdot 3^n \leq f(n) \leq 2{,}01 \cdot 3^n \quad n > 8$$

which means that $f(n) \in \Theta(3^n)$.

4.37

(a) $a_n = 2^n + 3^n + 4^n$.

(b) $b_n = 2^n - 5^n + n + 1$

(c) $c_m = (m+2) \cdot 3^m + m + 2$. Note that since the right-hand side is a first degree polynomial you should try a first degree polynomial, $c_m^{(p)} = am + b$, as the particular solution, $c_m^{(p)} = am$ won't work.

(d) $d_k = 2 - 2^k + k \cdot 2^k$. Since the reasonable guess $d_k^{(p)} = a \cdot 2^k$ appears as a solution of the homogenous equation, you have to correct it to $d_k^{(p)} = ak \cdot 2^k$.

(e) $e_n = 1 + n + n^2 + n^3$. This exercise combines the complications of (c) and (d).

4.38

(a) $b_8 = 93, b_9 = 130 \Rightarrow 9$ cuts are needed to get more than 100 pieces.

(b) $a_n = \frac{1}{2}n^2 + \frac{1}{2}n + 1$,
$b_n = \frac{1}{6}n^3 + \frac{5}{6}n + 1$.

(c) Explanation: The largest number of newly created volume regions with one cut is equal to the largest possible number of area regions in the cut surface $= a_n - 1$ when cut number n is made.

4.40 In ordinary exponentiation, the number of multiplications is one less than the exponent n. Time complexity $\Theta(n)$. When exponentiating using repeated squaring, in the best case $\log_2 n$ and in the worst case $2\log_2 n$ operations are needes. Time complexityt $\Theta(\log n)$.

4.41 The time needed to plait a plait is directly proportional to the length of the plait (which has to be considered as given) and inversely proportional to the width of the plait. The width of the plait is proportional to the square root of the cross sectional area. The cross sectional area is inversely proportional to the number of plaits. This means that the time needed for each plait is proportional to $\sqrt{n}$, which in total for all the plaits mean that we are in the time complexity class $\Theta(n\sqrt{n})$.

4.42

(a) In the worst case the tree doesn't branch out, and each new element glides down until it stops at a new level, after being compared to all the previous elements. Gives

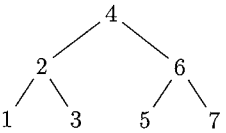
$$\sum_{k=1}^{n}(k-1) = \frac{n^2}{2}$$

that is, about the same number as for Bubble-Sort and Straight-Selection. This happens for instance if we get data that is already sorted. (You can modify the method so that the tree is rebuilt when needed, to a wider and flatter structure. If you have lots of data, and intend to use the tree for searching, this is worth the trouble.)

(b) In the best case we build a maximally wide, minimally deep tree. Then the height is $\log_2 n$. In a complete tree of this kind, we have a root, 2 nodes at the first level, $2^2 = 4$ nodes at the second level, and so on, that is, $n = 2^0 + 2^1 + \cdots + 2^k = 2^{k+1} - 1$ nodes. Each node has had to make the same number of comparisons as its level. Then in total

$$\begin{aligned} &2^0 \cdot 0 + 2^1 \cdot 1 + 2^2 \cdot 2 + \cdots = \\ &= \sum_{k=0}^{n} 2^k k \\ &= 2^{k+1}(k+1) - 2 \\ &\approx n \log_2 n \end{aligned}$$

operations have been used, same as for Merge-Sort. (The sum can be calculated using the same method as used to derive the sum of a geometric sequence, see *Discrete Mathematics and discrete models.*) One such a tree is



We can get a sequence that generates this tree by traversing it breadth-first: 4, 2, 6, 1, 3, 4, 7. (There are several other sequences that generates this tree as well. But all of them starts with 4.)

(c) $O(n^2)$.

4.43 If somebody delivers a suggested solution, it ought to consist of a packing list. We have to check that the listed objects exist (with the stated properties).

Can be done in linear time for each object, gives in worst case quadratic time for all the objects. Then we have to check that the sum of the volumes isn't too large (runs in linear time) and that the total value is enough (runs in linear time as well). So the suggested solution can be checked in polynomial time, so the problem belongs to NP. (It's by the way NP-complete. There exist encryption algorithms based on this problem, since just like prime factorisation it's a problem which is very easy to run in one direction, which here corresponds to unpacking a knapsack, while it's super hard the other way.)

Chapter 5

5.1 $[3\ \ 2\ \ 6\ \ 5\ \ 1\ \ 7\ \ 4]$ and $(1\ \ 3\ \ 6\ \ 7\ \ 4\ \ 5)\,(2)$.

5.2 $(1\ \ 5\ \ 4\ \ 7\ \ 6\ \ 3)\,(2)$

5.3 $[1\ \ 5\ \ 6\ \ 3\ \ 2\ \ 7\ \ 4]$

5.4 The three cycles can be placed in 3! ways. In the 2-cycle there are 2 possible places to start, in the 1-cycle you have 1 way, and in the 4 cycle you have 4 ways. According to the multiplication principle: $3! \cdot 2 \cdot 1 \cdot 4 = 48$ ways.

5.5 If we call the total number of cycles m we get (according to the same line of argument as in the previous exercise) $m! \cdot 1^{e_1} \cdot 2^{e_2} \cdot 3^{e_3} \cdot \ldots$.

5.6 We give all the answers on one-line form

(a) $[5\ \ 3\ \ 4\ \ 2\ \ 1]$
(b) $[5\ \ 3\ \ 4\ \ 2\ \ 1]$
(c) $[3\ \ 5\ \ 1\ \ 4\ \ 2]$
(d) $[4\ \ 2\ \ 3\ \ 5\ \ 1]$

5.7

(a) $(1\ \ 5\ \ 4\ \ 3\ \ 2)$
(b) $(1\ \ 3)\,(2)\,(4)\,(5)$
(c) $(2\ \ 5)\,(1)\,(3\ \ 4)$
(d) $(3\ \ 4\ \ 1\ \ 2\ \ 5)$

5.8 You get the inverse of a permutation on cycle form by inverting the order of each cycle.

5.9 $\mathrm{id} = [1\ \ 2\ \ \ldots\ \ n] = (1)\,(2)\ldots(n)$.

5.10

(a) An arbitrary permutation $\pi = [\pi_1\ \ \pi_2\ \ \ldots]$ is to correspond to the matrix M_π, where the one in the first column is on row π_1, the one in the second column is on row π_2, and so on. Clearly the matrix gets one one in each column, and since the permutation includes all elements from 1 up to n exactly once, the matrix gets exactly one one in each row as well. Conversely, each permutation matrix generates a permutation via the row coordinates of the ones. The correspondence $\pi \mapsto M_\pi$ between permutations and permutation matrices is thus a bijection.

(b) The matrix multiplication $M_\pi M_\tau$ gives a one in place ij if and only if row i in M_π has its one in the same position as column j in M_τ (and otherwise we get a zero in place ij). That corresponds exactly to the fact that the permutation multiplication $\pi\tau$ maps j on i if and only if τ maps j on the same element that π maps on i.

(c) Inverdion of a permutation means exchanging the upper and lower rows in the two-line form. The upper row gives the column coordinates and the lower row gives the row coordinates for the permutation matrix. Exchanging the rows thus corresponds exactly to an exchange of rows and columns, that is to say, to transposing the permutation matrix.

5.11

(a) Type: $\{2, 2, 2, 1\}$, order 2.
(b) Type: $\{4, 3\}$, order 12.
(c) Cycle form:
$(1)\,(2\ \ 7\ \ 3\ \ 6\ \ 5)\,(4)$.
Type: $\{1, 1, 5\}$, order 5.

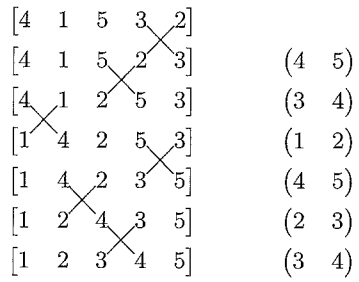
5.12 If you write down the placement of the cards after the first shuffle you find that you get a single cycle, with the length 52. 52 perfect riffle shuffles are thus needed to restore the pack.

5.13 Since the order is the lcm of the type, the same type gives the same order.

5.14 For instance $(1\ \ 2)\,(3\ \ 4\ \ 5)\,(6)$ and $(1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6)$, which both have the order 6.

5.15 If two permutations belong to the same conjugacy class they in principle do the same thing, they just use different names on the elements. And if we have a cycle in one of the permutations we must have a corresponding cycle in the other one, representing the same move using other names. Because of this, there has to be the same number of cycles of the same sizes, that is to say, the permutations are of the same type.

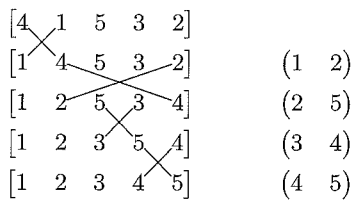
5.16 We run Bubble-Sort, and note at the side which transpositions we do.



The factorisation is

$$(4\ \ 5)\,(3\ \ 4)\,(1\ \ 2)\,(4\ \ 5)\,(2\ \ 3)\,(3\ \ 4)$$

5.17 The shortest factorisation into transposition can be found using Straight-Selection:



The factorisation is

$$(1\ \ 2)\,(2\ \ 5)\,(3\ \ 4)\,(4\ \ 5)$$

(There are other factorisations of the same length, but this one was easiest to find.)

5.18 If you perform a transposition of the numbers on places i and j in a permutaiton π the following pairs will be affected and switch from inversion to non-inversion or vice versa: For each element π_k between the places i and j two pairs will be affected, namely $\pi_i\pi_k$ and $\pi_k\pi_j$. Additionally, the pair $\pi_i\pi_j$ is affected. Thus an odd number of pairs are affeced, and thus the number of inversions are changed by an odd number.

If you start with the indentity permutation and create permutations by performing transpositions one at a time, then the number of inversions will all the time have the same parity as the number of transpositions used up to this point. Theorem 5.1 thereby follows.

5.20 We can count the inversions, or, as done here, write the permutations on cycle form:

(a) $(1\ \ 3\ \ 2\ \ 5\ \ 4)$. 4 transpositions, even permutation.

(b) $(1\ \ 3\ \ 5)\,(2\ \ 6\ \ 4)$. $2 + 2 = 4$ transpositions, even permutation.

(c) $(1\ \ 3\ \ 5\ \ 7\ \ 4\ \ 2\ \ 6)$. 6 transpositions, even permutation.

5.21 We write the positions on cycle form:

(a) $(1\ \ 15)\,(2)\,(3)\,(4)\,(5)\,(6)\,(7)\,(8)\,(9)\,(10)\,(11)\,(12)\,(13)\,(14)$. Odd permutations, unsolvable.

(b) We start by sliding the 13 to the side, so that the empty space end up where it should be. If we then use cycle form we get $(1\ \ 2\ \ 3\ \ 4)\,(5\ \ 6\ \ 7\ \ 8)\,(9\ \ 10\ \ 11\ \ 12)\,(13\ \ 14\ \ 15)$. $3 + 3 + 3 + 2 = 11$ transpositions, unsolvable.

5.23

(a) Breakpoints after elements 2 and 5.

(b) Two permutations in S_n can at the least have 0 breakpoint (if the permutations are identical) and at the most n breakpoints (all the elements have different successors in the two permutations.)

5.24

(a) Start from $[1\ \ 2\ \ 3\ \ 4\ \ 5]$. Block inversion of $[1\ \ 2\ \ 3\ \ 4]$ gives $[4\ \ 3\ \ 2\ \ 1\ \ 5]$. Block inversion of this whole permutation gives $[5\ \ 1\ \ 2\ \ 3\ \ 4]$.

(b) Start from $[1\ \ 2\ \ 3\ \ 4\ \ 5]$. Block transposition of $[2\ \ 3]$ to the end gives $[1\ \ 4\ \ 5\ \ 2\ \ 3]$. Block transposition of $[1\ \ 4]$ in between 2 and 3 gives $[5\ \ 2\ \ 1\ \ 4\ \ 3]$. Block transposition of $[2\ \ 1]$ to the end gives $[5\ \ 4\ \ 3\ \ 2\ \ 1]$.

5.27

(a) Inverses in groups are unique. It's given that ab is its own inverse. What do we get if we multiply ab and ba?

$$(ab)(ba) = a(bb)a = aIa = aa = I$$
$$\Rightarrow \qquad ba = (ab)^{-1}$$

ba is apparently inverse to ab as well, and must then be equal to ab.

(b) $abbaba = aIaba = aaba = Iba = ba$.

(c) For a start, all sequences with several instances of the same letter can be shortened to one letter or no letter (depending on whether the number is odd or even). What remains is a sequence of alternating letters (or nothing, I). If the sequence consists of more than two letters, turn every second pair back to front, and then everything except for maybe the first letter or pair of letters will cancel out.

(d) Since all words can be rewritten as one out of four alternatives, the group has just four elements. We compare the calculation table with the one for $\langle(\mathbb{Z}_2)^2, +\rangle$:

$\cdot$	I	a	b	ab
I	I	a	b	ab
a	a	I	ab	b
b	b	ab	I	a
ab	ab	b	a	I

$+$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,0)$	$(1,1)$	$(1,0)$
$(1,0)$	$(1,0)$	$(1,1)$	$(0,0)$	$(0,1)$
$(1,1)$	$(1,1)$	$(1,0)$	$(0,1)$	$(0,0)$

The tables have exactly the same structure. $I \mapsto (0,0)$, $a \mapsto (0,1)$, $b \mapsto (1,0)$, $ab \mapsto (1,1)$ is an isomorphism.

5.28

(a) The inverse is what you have to combine with to get the identity. If we combine a word with the same word backwards, the two middle elements will be equal and cancel out, at which point the next two elements get into contact and cancel out, and so on, until the only thing that remains is id.

(b) If the transpositions overlap, like $(3\ \ 4)$ and $(4\ \ 5)$, the composition is a three-cycle, an element with the order 3. Whereas if they are disjoint the composition is its own invers, that is, an element with the order 2.

5.29 *Closedness*: The concatenation of two sequences of rotations is clearly a new sequence of rotations. *Associativity*: Concatenation of two sequences is clearly an associative operation. *Identity*: If you multiply by *the sequence of zero rotations* nothing happens, so this sequence is an identity. *Inverse*: Each sequence of rotations has an inverse, since if you rotate everything back in the opposite direction so that the cube looks the same as it did from start (that is, solve the cube) you could just as well have done the sequence of zero rotations.

5.30 Every rotation of a single side (say U) can be written as a power of the side (U^1, U^2, or $U^3 = U^{-1}$). All rearrangements consist of sequences of rotations of sides.

5.31 $(R^2U^2)^6 = I$. $(RU^2)^{30} = I$. $(RU)^{105} = I$.

5.32 Read the sequence of rotations backwards and make the inverse of every side rotation. For instance, the inverse of $RU^2LR^{-1}U^2L^{-1}$ is thereby $LU^{-2}RL^{-1}U^{-2}R^{-1}$

5.34 The inversions are 31, 32, 51, 54, 52, 74, 72, 76, and 42, $\ell(3517426) = 9$.

5.35 The permutation can be regarded as instructions for how something is to be messed up, and the inverse as instructions for how to tidy it up again. The shortest possible tidying up has to consist of the same number of steps as the shortest possible messing up, and since ℓ measures the number of steps, the permutation and its inverse has to have the same value.

5.37

(a) The reverse permutation.

(b) $n(n-1)/2$.

(c) The level on which a permutation is placed is equal to the number of inversions. We get the maximal number of inversions when everything has as much as possible on the wrong side, and that can only be done in one way. (Put all numbers that are smaller on the wrong side.)

5.38 $\pi \leq \sigma$ means that π is found on the shortest path from the identity to σ. The shortest path only contains adjacent transpositions that generate inversions, none that remove any of them, so all inversions in π have to be included in σ as well. And the other way around, if π's inversions are part of σ, it has to be possible to get to σ via π, by firstly do the adjacent transpositions that generate the inversions in π and afterwards the remaining inversions. So π is found on the path to σ.

5.39 What we have to prove is firstly that every list of inversions has to fulfil the requirement and secondly that every list that fulfils the requirement is a list of inversions.

If the permutation exists: If (m_1, m_2) and (m_2, m_3) both are incuded in a list of inversions, that means that firstly $m_1 > m_2$ and $m_2 > m_3$ and secondly that m_1 stands to the left of m_2 and m_2 stands to the left of m_3. Both $>$ and "stands to the left of" are transitive relations, which gives that $m_1 > m_3$ and that m_1 is to the left of m_3 in the permutation. In that case, the pair (m_1, m_3) has to be included in the list of inversions as well.

To show that something is a valid list of inversions, we can show that it is possible to make a permutation based on it. The inversions are really information about "this one should be put before that one". The pairs that aren't included in the list are implicit information about "this one should not be put before that one". For a list to be impossible to follow, it has to give contradictory information. A contradiction of the type "a both should and should not be put before b" can't arise, since a pair either belongs to the list or doesn't. The next kind of contradiction would be something like a is in front of b which is in front of c which is in front of a. But this can't happen, since a "a in front of c" follows automatically from the given constraint.

5.42 Eight. (You have to be grateful that they don't have that shape!)

5.43 SIM-cards of mobile phones are rectangular, but one corner is cut off. That means that there is only one way to fit them into the intended hole. Something like that might be possible. Alternatively, you could redesign the machine so that it doesn't matter which way the card is inserted.

5.44

(a) $\langle (\mathbb{Z}_2)^2, + \rangle$, or multiplication of the invertible elements in $\mathbb{Z}_8$, see exercise 2.16 on page 20. Or the group in exercise 5.27.

(b) Since it's isomorphic to the group in exercise 5.27, it can be expressed in the same way with generators and relations:

$$\langle s_1, s_2 \mid (s_1)^2 = (s_2)^2 = (s_1 s_2)^2 = I \rangle$$

(c) If we start at the upper left corner and then go clockwise, the labels become

$$\begin{array}{ll} s_0: & (1)\,(2)\,(3)\,(4) \\ s_1: & (1\ \ 3)\,(2\ \ 4) \\ s_2: & (1\ \ 4)\,(2\ \ 3) \\ s_3: & (1\ \ 2)\,(3\ \ 4) \end{array}$$

5.45

(a) Six operations (so here we get all of S_n).
s_0: Put it back where you took it:
$(1)\,(2)\,(3)$
s_1: Rotate 120° clockwise:
$(1\ \ 2\ \ 3)$
s_2: Rotate 120° counterclockwise:
$(1\ \ 3\ \ 2)$
s_3: Rotate around corner 1 so that the wrong side is turned up:
$(1)\,(2\ \ 3)$
s_4: Rotate around corner 2 so that the wrong side is turned up:
$(2)\,(3\ \ 1)$
s_5: Rotate around corner 3 so that the wrong side is turned up:
$(3)\,(1\ \ 2)$

(b) Well, as noted in the previous exercise: S_3.

(c) The subgroups are

1. $\{s_0\}$, means that we aren't allowed to do anything. Generated by s_0.
2. $\{s_0, s_3\}$, $\{s_0, s_4\}$ and $\{s_0, s_5\}$. Allowed is to rotate around a corner, nothing else. (As if the triangle was attached by the corner in question.) The subgroups are generated by s_3, s_4, and s_5, respectively.
3. $\{s_0, s_1, s_2\}$. You may move the triangle in the plane, but not turn it wrong side up. Generated by either s_1 or s_2.
4. The whole group.

5.46 Not the answer, but the calculation would have been longer!

5.47

(a)

$$\begin{aligned} s_0 &= (1)\,(2)\,(3)\,(4) & 3^4 &= 81\\ s_1 &= (1\ \ 4\ \ 3\ \ 2) & 3^1 &= 3\\ s_2 &= (1\ \ 3)\,(2\ \ 4) & 3^2 &= 9\\ s_3 &= (1\ \ 2\ \ 3\ \ 4) & 3^1 &= 3 \end{aligned}$$

gives that there are

$$\frac{81+3+9+3}{4} = 24$$

different colourings.

(b) Now, we have another four operations:

$$\begin{aligned} s_0 &= (1)\,(2)\,(3)\,(4) & 3^4 &= 81\\ s_1 &= (1\ \ 4\ \ 3\ \ 2) & 3^1 &= 3\\ s_2 &= (1\ \ 3)\,(2\ \ 4) & 3^2 &= 9\\ s_3 &= (1\ \ 2\ \ 3\ \ 4) & 3^1 &= 3\\ s_4 &= (1\ \ 2)\,(3\ \ 4) & 3^2 &= 9\\ s_5 &= (1\ \ 4)\,(2\ \ 3) & 3^2 &= 9\\ s_6 &= (1)\,(2\ \ 4)\,(3) & 3^3 &= 27\\ s_7 &= (1\ \ 3)\,(2)\,(4) & 3^3 &= 27 \end{aligned}$$

which gives

$$\frac{81+3+9+3+9+9+27+27}{8} = 21$$

The colourings that use all three of the colours exist in one right and one left version, and if you are allowed to turn them upside down, these will be equivalent.

5.48

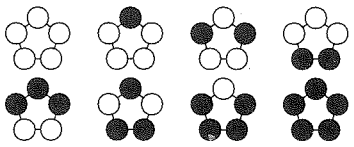
(a) Identity; rotation one, two, three, or four fifths of a turn in the plane; turning upside down with one of the five positions remaining in place. (The cycle form will be given in the next part.)

(b) Here are the different cycle forms and the number of configurations that they fix:

$$\begin{aligned} &(1)\,(2)\,(3)\,(4)\,(5) && 2^5 = 32\\ &(1\ \ 2\ \ 3\ \ 4\ \ 5) && 2^1 = 2\\ &(1\ \ 3\ \ 5\ \ 2\ \ 4) && 2^1 = 2\\ &(1\ \ 4\ \ 2\ \ 5\ \ 3) && 2^1 = 2\\ &(1\ \ 5\ \ 4\ \ 3\ \ 2) && 2^1 = 2\\ &(1)\,(2\ \ 5)\,(3\ \ 4) && 2^3 = 8\\ &(2)\,(1\ \ 3)\,(4\ \ 5) && 2^3 = 8\\ &(3)\,(1\ \ 5)\,(2\ \ 4) && 2^3 = 8\\ &(4)\,(1\ \ 2)\,(3\ \ 5) && 2^3 = 8\\ &(5)\,(1\ \ 4)\,(2\ \ 3) && 2^3 = 8 \end{aligned}$$

$$\frac{32+4\cdot 2+5\cdot 8}{10} = \frac{80}{10} = 8$$

(c)



(d) We have the transformations identity (997 cycles); 996 different rotations (one cycle each); 997 different turns (which each contains one 1 cycle and 489 2-cycles). In total 1994 different transformations working on 2^{997} configurations.

$$\frac{2^{997} + 996\cdot 2^1 + 997\cdot 2^{499}}{1994} \approx 6{,}7\cdot 10^{296}$$

(e) If the number of beads isn't prime, the rotations where the number of steps isn't coprime to the number of pearls will generate more than one cycle (just as when we rotated the pot coaster two steps). That means that we get a lot more cases to study. The upside down cases get more complicated as well.

5.50

(a) Figure 3 shows how the shuffle is made. In card shuffling the only thing of interest is which *places* it is that exchanges cards with each other. The row of values thus shows the inverse of what we are looking for. The permutation is shown in figure 4.

(b) The number of times = the order of the permutation = $\operatorname{lcm}\{1,3,4,3,1\} = 12$.

5.51 We write the permutations on cycle form: $\pi = (1\ \ 4)\,(2\ \ 5)\,(3\ \ 6\ \ 7)$, $\gamma = (1\ \ 7)\,(2\ \ 4\ \ 3)\,(5\ \ 6)$. The elements in the 3-cycles have to correspond to each other, the elements in the 2-cycles to each other. One such pairing (there are several) is

$$\begin{matrix} (1\ \ 7)\,(2\ \ 4\ \ 3)\,(5\ \ 6)\\ (1\ \ 4)\,(3\ \ 6\ \ 7)\,(2\ \ 5) \end{matrix}$$

$$\sigma = \begin{bmatrix} 1 & 7 & 2 & 4 & 3 & 5 & 6\\ 1 & 4 & 3 & 6 & 7 & 2 & 5 \end{bmatrix}$$

If we now combine the permutations we

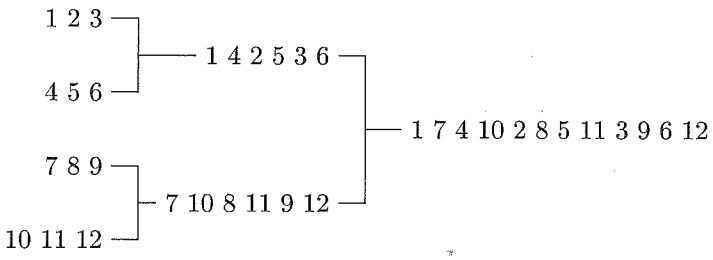


Figure 3: *Exercise 5.50.*

$$\begin{bmatrix} 1 & 7 & 4 & 10 & 2 & 8 & 5 & 11 & 3 & 9 & 6 & 12 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 5 & 9 & 3 & 7 & 11 & 2 & 6 & 10 & 4 & 8 & 12 \end{bmatrix}$$

$$= (1)\,(2 \quad 5 \quad 7)\,(3 \quad 9 \quad 10 \quad 4)\,(6 \quad 11 \quad 8)\,(12)$$

Figure 4: *Exercise 5.50.*

get

$1 \overset{\sigma}{\mapsto} 1 \overset{\pi}{\mapsto} 4 \overset{\sigma^{-1}}{\mapsto} 7$

$2 \mapsto 3 \mapsto 6 \mapsto 4$

$3 \mapsto 7 \mapsto 3 \mapsto 2$

$4 \mapsto 6 \mapsto 7 \mapsto 3$

$5 \mapsto 2 \mapsto 5 \mapsto 6$

$6 \mapsto 5 \mapsto 2 \mapsto 5$

$7 \mapsto 4 \mapsto 1 \mapsto 1$

which equals γ.

5.52 According to the same line or reasoning as used on the Fifteen puzzle you can only solve even permutations. Written on cycle form this becomes $(1)\,(2 \quad 4)\,(3 \quad 7)\,(5)\,(6 \quad 8)\,(9)$, an odd permutation. Thus, it's *im*possible!.

5.53 We know that the whole of S_7 can be generated using the adjacent transpositions. If we can show that is's possible to make up all the adjacent transpositions, we are done.

π makes the first two elements swap places, τ "cranks around" the elements one step. Furthermore, $\tau^6 = \tau^{-1}$. What happens if we calculate $\tau^{-1}\pi\tau$?

$1 \overset{\tau}{\mapsto} 1 \overset{\pi}{\mapsto} 2 \overset{\tau^{-1}}{\mapsto} 7$

$2 \mapsto 3 \mapsto 3 \mapsto 2$

$3 \mapsto 4 \mapsto 4 \mapsto 3$

$4 \mapsto 5 \mapsto 5 \mapsto 4$

$5 \mapsto 6 \mapsto 6 \mapsto 5$

$6 \mapsto 7 \mapsto 7 \mapsto 6$

$7 \mapsto 2 \mapsto 1 \mapsto 1$

The result is

$(1 \quad 7)\,(2)\,(3)\,(4)\,(5)\,(6)$

that is, we have made 1 and 7 swap places. In the same way we can exchange places of 1 and any other element. And every permutation can be carried out using two permutations that include 1; if we for instance want to swap 3 and 5 we can exchange the places of 1 and 3, then of 1 and 5, and lastly of 1 and 3 again. So it's possible to carry out all adjacent transpositions, and then you can generate the whole group!

5.54

(a) We can't be bothered to draw the graph, but it has three levels besides the bottom, six permutations on the first level, six on the third and the remaining $4! - 2 \cdot 6 - 1 = 11$ ones on the second.

(b) The level of the permutation is equal to the length of the shortest possible factorisation into transpositions, which can be directly determined from the cycle form, see example 5.7 on page 84. For instance, the permutatiton $(1 \quad 3 \quad 5)\,(2 \quad 4)\,(6)$ is placed on level $6 - 3 = 3$ in S_6, since it has 6 elements and 3 cycles. Alternatively, you can run the permutation through Straight-Selection, which finds the shortest possible factorisation into tranpositions, and check how many you get.

(c) At the top, we find the permutations that consist of just one cycle, and they are then on level $n-1$. There are $(n-1)!$ permutations of this kind; we can start by writing 1 first in the cycle and then permute the remaining positions. At level 1, we get as many permutations as there are transpositions, $\binom{n}{2}$.

(d) A transposition that takes us upwards in the tree "ties" two cycles into one, and one that takes us downwards "cuts" a cycle into two. If π has more cycles than σ and it's possible to get π from σ by cleaving a number of cycles, π is placed under, otherwise not. You have to remember that rotating the cycles before cutting is allowed ; it doesn't matter which one of the elements of the cycle that is placed first.

5.55

(a) We number the pancakes on the edge 1–6 and the pancake in the middle 7. There are six symmetries: rotation 0, 1, 2, 3, 4, or 5 times 60°. (Turning the plate upside down doesn't seem like a reasonable action.) On cycle form these become:

$$\begin{array}{rl} 0°: & (1)\,(2)\,(3)\,(4)\,(5)\,(6)\,(7) \\ 60°: & (1\ 2\ 3\ 4\ 5\ 6)\,(7) \\ 120°: & (1\ 3\ 5)\,(2\ 4\ 6)\,(7) \\ 180°: & (1\ 4)\,(2\ 5)\,(3\ 6)\,(7) \\ 240°: & (1\ 5\ 3)\,(2\ 6\ 4)\,(7) \\ 300°: & (1\ 6\ 5\ 4\ 3\ 2)\,(7) \end{array}$$

Using three kinds of pancakes, this gives

$$\frac{3^7+3^4+2\cdot 3^3+2\cdot 3^2}{6}=390$$

versions.

(b) We have to subtract the arrangements that only use two or one kind of pancakes. Two kinds can be picked in $\binom{3}{2}=3$ ways, there are in total three one-kind arrangements. For each pair of kinds, there are

$$\frac{2^7+2^4+2\cdot 2^3+2\cdot 2^2}{6}=28$$

arrangements. But if we multiply this by three, the one-kind kinds will be counted double, since for instance the vanilla plate is included in both spinach-vanilla and tomato-vanilla. This we have to compensate for. In total there are

$$390-3\cdot 28+3=309$$

arrangements containing all three kinds.

Chapter 6

6.1

$4+1+1$	o o o o|o|o
$3+2+1$	o o o|o o|o
$3+1+2$	o o o|o|o o
$2+3+1$	o o|o o o|o
$2+2+2$	o o|o o|o o
$2+1+3$	o o|o|o o o
$1+2+3$	o|o o|o o o
$1+1+4$	o|o|o o o o
$1+3+2$	o|o o o|o o
$1+4+1$	o|o o o o|o

6.2 Put down n dots. Now, there are $n-1$ spaces, and in each space we can choose to put a separator or not do it. Gives 2^{n-1} alternatives.

6.3

(a) *Version 1* If we have a partition of n into k non-negative parts, we can generate a partition of $n+k$ into k positive parts by adding 1 to each part, and vice versa. Thus there is the same number of partitions of n into k non-negative parts as there is partitions of $n+k$ into k positive parts, and we have already shown that there are $\binom{n+k-1}{k-1}$ ways to do the latter thing.

Version 2 A partition of n into k nonnegative parts can be coded in the same way as partitions into positive parts, but you are here allowed to place two sticks side by side. Every code of this kind corresponds to a permutation of n rings and $k-1$ sticks. Choose $k-1$ of the $n+k-1$ available positions for the sticks.

(b) Every solution of this kind corresponds to one of the compostitions above.

(c) The interesting thing here is how

many marbles there will be in each box. Each such distribution corresponds to a solution of the equation above.

(d) The interesting thing is how many of each kind we pick. Each choice of this kind corresponds to a solution of the equation in (b).

6.4

(a)

$$\begin{aligned} p(13) &= p(12) + p(11) - p(8) \\ &\quad - p(6) + p(1) \\ &= 77 + 56 - 22 - 11 + 1 = 101 \end{aligned}$$

(b)

$$\begin{aligned} p(14) &= p(13) + p(12) - p(9) \\ &\quad - p(7) + p(2) \\ &= 101 + 77 - 30 - 15 + 2 \\ &= 135 \end{aligned}$$

(c)

$$\begin{aligned} p(15) &= p(14) + p(13) - p(10) \\ &\quad - p(8) + p(3) + p(0) \\ &= 135 + 101 - 42 \\ &\quad - 22 + 3 + 1 \\ &= 176 \end{aligned}$$

6.5 There exists a conjugate class for each type of permutation in S_n. The type is defined by a partition of n. Because of this, the number of conjugate classes is the number of partitions of n, that is, $p(n)$.

6.7

(a)

(b) Introduce a coordinate system with origo in the lower left corner. A box with the coordinates (x, y) should be situated inside the diagram if and only if the contour path passes the box horisontally before it passes the box vertically, which is equivalent to R_x coming before U_y in the code.

(c) Corresponds to one line along the left edge and one along the upper edge, which won't add any boxes, and because of this may just as well be there.

(d) We add boxes at corners, and a newly added box means that you go one step to the right before going one step upwards. (This demands that you actually have those infinitely many start-Us and end-Rs.

6.8

$$\begin{aligned} 7+1 &\mapsto 7+1 \\ 5+3 &\mapsto 5+3 \\ 5+1+1+1 &\mapsto 5+2+1 \\ 3+3+1+1 &\mapsto 6+2 \\ 3+1+1+1+1+1 &\mapsto 4+3+1 \\ 1+1+1+1+1+1+1+1 &\mapsto 8 \end{aligned}$$

6.9 If we have an even number of odd parts, and merge parts that are equal until no equal parts are left, we will at every step still have an even number of odd parts, since we always remove two parts of the same kind, and an even number minus two is an even number. When we are finished, all parts will be of different sizes, and the number of odd parts is still even. And the same thing when we go in the opposite direction.

6.10

(a) Partitions where there are at most two parts of each size.

(b) When no parts divisible by three remains.

(c) The number of partitions where there are at most two parts of each size is equal to the number of partitions where no part is divisible by three.

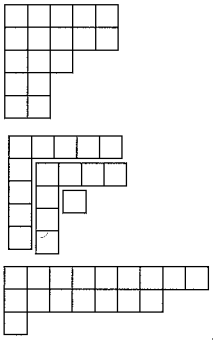
6.11 Each partition can be coded as a string of Rs and Us and each such string corresponds to a partition. If there are m Us there can be at most m parts. (There can be fewer than this, if U is placed first in the string). If there are k Rs no part can be greater than k (it may happen that all the parts are smaller than this, if there is an R last in the string.)

6.12 If we have a diagram representing a partition of n into parts of different sizes, each row in the diagram will be longer than the one below. The topmost row will protrude beyond the others. Read from the other direction, the protruding boxes correspond to terms of size 1. The second row protrudes beyond the third one, and the protruding boxes will combined with the boxes from the first row correspond to terms of size 2

when read from the other direction. And the protruding part of row 3 corresponds to 3s. And so on down to the last row. The largest term in the sum of parts of all sizes is equal to the number of terms in the sum of distinct parts.

6.13 So we are to find a bijection between self-conjugate partitions and partitions consisting of odd distinct parts. A self-conjugate partition doesn't have to consist of odd parts of different sizes; the partition $16 = 4+4+4+4$ is a good counter-example. So we have to invent some way of transforming a self-conjugate partition to one of odd distinct parts, a way that can be run in reverse.

In a self-conjugate partition the number of parts is equal to the greatest part. If we remove the largest part and the first box in the remaining rows, what we have left is a self-conjugate partition of a smaller number. The number of boxes removed is an odd number; if the number of terms is k we remove $2k-1$ boxes. We keep on like this, until no boxes remain. We have from the self-conjugate partition generated a sum of different odd numbers. (The numbers are of different sizes, since the number of terms get reduced by at least one in each step.) Conversely, we can from each partition of different odd numbers get a self-conjugate partition, by starting by bending the largest number to an L, after which we put the next number as an L inside the previous L, until no more parts remain. This can be done with all partitions consisting of distinct odd numbers, and if we just generated the partition from a self-conjugate partition we get the self-conjugate partition we started with back. Below we see how the self-conjugate partition 2+2+3+5+5 is transformed into 1+7+9, which consists of distinct odd parts.



6.14 If we can remove a Durfee square of size j from a partition of the number n, the part below the square (which contains, say, m boxes) will be a partition of m where no part is greater than j, while the part to the right will contain the remaining $n-j^2-m$ boxes from the number n. If we conjugate this part we see that it corresponds to a partition of $n-j^2-m$ where no part is greater than j. If we sum this for all possible values of m (which ought to be everything between zero and $n-j^2$) we get the total number of diagrams of this kind.

6.15

(a) If we add a box at the bottom of a Young diagram for $n-1$, we get a diagram for n, corresponding to a partition containing a one. Besides, all partitions of n containing a one can be generated in this way, in one and only one way. The remaining partitions of n don't contain ones. This gives the equality

$$p(n) = p(n-1) + p(n \mid \text{no part} = 1)$$

(b) All numbers from 2 and upwards can be partitioned without any part equal to one. That means that $p(n \mid \text{no part} = 1) > 0$, which means that

$$p(n) = p(n-1) + p(n \mid \text{no part} = 1) > p(n-1)$$

So the function is increasing.

6.17

$$\begin{aligned}\frac{1}{(1-x)^n} &= (1-x)^{-n} \\ &= \sum_{k=0}^{\infty} \binom{-n}{k} 1^{-n-k}(-x)^k \\ &= \sum_{k=0}^{\infty} \binom{-n}{k} (-1)^k x^k\end{aligned}$$

Now we take a closer look at the binomial coefficients:

$$\begin{aligned}&\binom{-n}{k} = \\ &= \frac{(-n)(-n-1)\cdots(-n-k+1)}{1\cdot 2\cdot\cdots\cdot k} \\ &= \frac{(-1)n(-1)(n+1)\cdots(-1)(n+k-1)}{1\cdot 2\cdot\cdots\cdot k} \\ &= (-1)^k \frac{(n+k-1)\cdots(n+1)n}{1\cdot 2\cdot\cdots\cdot k} \\ &= (-1)^k \binom{n+k-1}{k} \\ &= (-1)^k \binom{n+k-1}{n-1}\end{aligned}$$

Inserted into the formula, this gives

$$\frac{1}{(1-x)^n} =$$
$$= \sum_{k=0}^{\infty}(-1)^k \binom{n+k-1}{n-1}(-1)^k x^k$$
$$= \sum_{k=0}^{\infty}\binom{n+k-1}{n-1}x^k$$

6.18

(a)

$$1 + 5x + 5^2x^2 + \cdots = \frac{1}{1-5x}$$

(b)

$$-2 + (-2)^2x + (-2)^3x^2 + (-2)^4x^3 + (-2)^5x^4 + \cdots =$$
$$= -2\big(1 + (-2)^1x + (-2)^2x^2 + (-2)^3x^3 + (-2)^4x^4 + \ldots\big)$$
$$= -2\frac{1}{1-(-2x)}$$
$$= -\frac{2}{1+2x}$$

(c)

$$\binom{1/5}{0} + \binom{1/5}{1}x + \binom{1/5}{2}x^2 + \binom{1/5}{3}x^3 + \cdots =$$
$$= (1+x)^{1/5} = \sqrt[5]{1+x}$$

(d)

$$\binom{17}{17} + \binom{18}{17}x + \binom{19}{17}x^2 + \binom{20}{17}x^3 + \cdots =$$
$$= \frac{1}{(1-x)^{18}}$$

6.19

(a)

$$A(x) + B(x) =$$
$$= a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots + b_0 + b_1x + b_2x^2 + b_3x^3 + \ldots$$
$$= (a_0+b_0) + (a_1+b_1)x + (a_2+b_2)x^2 + (a_3+b_3)x^3 + \ldots$$

(b)

$$A(x)B(x) =$$
$$= (a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots)\cdot \cdot (b_0 + b_1x + b_2x^2 + b_3x^3 + \ldots)$$
$$= a_0b_0 + (a_0b_1 + a_1b_0)x + (a_0b_2 + a_1b_1 + a_2b_0)x^2 + \ldots$$
$$= \sum_{n=0}^{\infty}\sum_{k=0}^{n} a_k b_{n-k}x^n$$

(c)

$$A(x^2) =$$
$$= a_0 + a_1x^2 + a_2x^4 + a_3x^6 + \ldots$$
$$= a_0 + 0x + a_1x^2 + 0x^3 + a_2x^4 + 0x^5 + a_3x^6 + \ldots$$

(d)

$$\frac{d}{dx}A(x) =$$
$$= \frac{d}{dx}(a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots)$$
$$= 0 + a_1 + 2a_2x + 3a_3x^2 + \ldots$$

(e)

$$\frac{A(x) - a_0}{x} =$$
$$= \frac{(a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots) - a_0}{x}$$
$$= a_1 + a_2x + a_3x^2 + \ldots$$

(f)

$$xA(x) + a_{-1} =$$
$$= x(a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots) + a_{-1}$$
$$= a_{-1} + a_0x + a_1x^2 + a_2x^3 + a_3x^4 + \ldots$$

6.20

$$(1 + x + x^2 + x^3 + \ldots)\cdot$$
$$\cdot(1 + x^5 + x^{10} + x^{15} + \ldots)\cdot$$
$$\cdot(1 + x^{10} + x^{20} + x^{30} + \ldots)\cdot$$
$$\cdot(1 + x^{20} + x^{40} + x^{60} + \ldots) =$$
$$= \frac{1}{1-x}\frac{1}{1-x^5}\frac{1}{1-x^{10}}\frac{1}{1-x^{20}}$$

6.21 Since the number of crowns is an integer we have to use an even number

of 50-öre coins. You can make up one crown using 50-öre coins in exactly one way.

$$(1+x+x^2+x^3+\dots)\cdot$$
$$\cdot(1+x+x^2+x^3+\dots)\cdot$$
$$\cdot(1+x^5+x^{10}+x^{15}+\dots)\cdot$$
$$\cdot(1+x^{10}+x^{20}+x^{30}+\dots)=$$
$$=\frac{1}{(1-x)^2}\frac{1}{1-x^5}\frac{1}{1-x^{10}}$$

6.22 We can write the factors in

$$\prod_{k=1}^{\infty}\frac{1}{1-x^{2k}}\prod_{k=1}^{\infty}\frac{1}{1-x^{2k-1}}$$

as

$$\frac{1}{1-x^2}\frac{1}{1-x^4}\frac{1}{1-x^6}\cdots$$
$$\cdot\frac{1}{1-x^1}\frac{1}{1-x^3}\frac{1}{1-x^5}\cdots$$

and if we rearrange the factors we se that we have

$$\frac{1}{1-x^1}\frac{1}{1-x^2}\frac{1}{1-x^3}\cdots$$

which can be written as

$$\prod_{k=1}^{\infty}\frac{1}{1-x^k}$$

6.23

(a)

$$(1+x+x^2+\dots)\cdot$$
$$\cdot(1+x^2+x^4+\dots)\cdot$$
$$\cdot(1+x^3+x^6+\dots)\cdot\dots$$
$$\cdot(1+x^m+x^{2m}+\dots)=$$
$$=\frac{1}{1-x}\frac{1}{1-x^2}\frac{1}{1-x^3}\cdots\frac{1}{1-x^m}$$
$$=\prod_{k=1}^{m}\frac{1}{1-x^k}$$

(b) The same answer as in (a), since conjugation gives a bijection between partitions with all parts $\leq m$ and partitions with at most m parts!

(c)

$$(1+x+x^2+\cdots+x^m)\cdot$$
$$\cdot(1+x^2+x^4+\cdots+x^{2m})\cdot$$
$$\cdot(1+x^3+x^6+\cdots+x^{3m})\cdots=$$
{See example 6.5}
$$=\frac{1-x^{m+1}}{1-x}\frac{1-x^{2(m+1)}}{1-x^2}$$
$$\cdot\frac{1-x^{3(m+1)}}{1-x^3}\cdots$$
$$=\prod_{k=1}^{\infty}\frac{1-x^{k(m+1)}}{1-x^k}$$

6.24 The generating function of $p(n \mid$ at most $m-1$ parts of each size) follows from the previous exercise:

$$\prod_{k=1}^{\infty}\frac{1-x^{km}}{1-x^k}$$

The generating function of $p(n \mid$ no part divisible by $m)$ is

$$\frac{1}{1-x}\cdots\frac{1}{1-x^{m-1}}\frac{1}{1-x^{m+1}}\cdots$$
$$\cdots\frac{1}{1-x^{2m-1}}\frac{1}{1-x^{2m+1}}\cdots$$

If we fill in the factors that are "missing" we get

$$\frac{1}{1-x}\cdots\frac{1}{1-x^{m-1}}\frac{1-x^m}{1-x^m}\frac{1}{1-x^{m+1}}\cdot$$
$$\cdot\cdot\cdot\cdot\frac{1}{1-x^{2m-1}}\frac{1-x^{2m}}{1-x^{2m}}\frac{1}{1-x^{2m+1}}\cdots$$
$$=\prod_{k=1}^{\infty}\frac{1-x^{km}}{1-x^k}$$

6.25

(a)

$$(1+x)(1+x^2)(1+x^4)(1+x^8)\cdots=$$
$$=(1+x+x^2+x^3)$$
$$\cdot(1+x^4)(1+x^8)\dots$$
$$=(1+x+x^2+x^3+x^4$$
$$+x^5+x^6+x^7)\cdot$$
$$\cdot(1+x^8)\dots$$
$$=1+x+x^2+x^3+x^4+x^5$$
$$+x^6+x^7+x^8+\dots$$
$$=\frac{1}{1-x}$$

(b) Partitions where the number n is written as a sum of powers of two, with at most one term of each kind.

(c) There is only one way of doing this, which corresponds to the hopefully well-known fact that there is only one way of writing the number n using binary notation.

6.27 We can try to separate a and b into one equation each. From the second one we get

$$b_n=\frac{2}{3}b_{n-1}+\frac{1}{3}a_{n-1}$$
$$\Rightarrow\qquad a_{n-1}=3b_n-2b_{n-1}$$

which inserted into the first equation gives

$$\begin{aligned} a_n &= \frac{2}{3}(3b_n - 2b_{n-1}) + \frac{1}{3}b_{n-1} \\ &= 2b_n - b_{n-1} \\ &\Rightarrow \\ a_{n-1} &= 2b_{n-1} - b_{n-2} \end{aligned}$$

which reinserted into the lower equation gives

$$\begin{aligned} b_n &= \frac{2}{3}b_{n-1} + \frac{1}{3}(2b_{n-1} - b_{n-2}) \\ &= \frac{4}{3}b_{n-1} - \frac{1}{3}b_{n-2} \end{aligned}$$

From this equation we can in the "normal way" get an expression for b_n, which we can then insert into the equation for a_n, which in its turn is solved in the normal way.

6.28 $a_n = 2 \cdot 3^n + 4^n$, $b_n = 3^n + 4^n$.

6.29

(a)

$$\begin{aligned} &0!\frac{x^0}{0!} + 1!\frac{x^1}{1!} + 2!\frac{x^2}{2!} + 3!\frac{x^3}{3!} + \cdots = \\ &= x^0 + x^1 + x^2 + x^3 + x^4 + \ldots \\ &= \sum_{k=0}^{\infty} x^k \\ &= \frac{1}{1-x} \end{aligned}$$

(b)

$$\begin{aligned} &0\frac{x^0}{0!} + 1\frac{x^1}{1!} + 1\frac{x^2}{2!} + \\ &\quad + 1\frac{x^3}{3!} + 1\frac{x^4}{4!} + \cdots = e^x - 1 \end{aligned}$$

(c) The sequence is $a_k = 52!/(52-k)!$ for $k \leq 52$, $a_k = 0$ for the rest, and the exponential generating function

$$\begin{aligned} &\sum_{k=0}^{52} \frac{52!}{(52-k)!}\frac{x^k}{k!} = \\ &\quad = \sum_{k=0}^{52} \binom{52}{k} x^k = (1+x)^{52} \end{aligned}$$

6.30

(a) Unsolvable if k is even, otherwise about half of the in total 2^k strings.

(b)

$$\begin{aligned} &(1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots) \\ &\quad \cdot (\frac{x^1}{1!} + \frac{x^3}{3!} + \frac{x^5}{5!}) = \\ &= \frac{e^x + e^{-x}}{2}\frac{e^x - e^{-x}}{2} \\ &= \frac{e^{2x} - e^{-2x}}{4} \\ &= \frac{1}{2}(\frac{(2x)^1}{1!} + \frac{(2x)^3}{3!} + \frac{(2x)^5}{5!} + \ldots) \\ &= 2^0\frac{x^1}{1!} + 2^2\frac{x^3}{3!} + 2^4\frac{x^5}{5!} + \ldots \end{aligned}$$

(c) If you have an even number of zeros and an odd number of ones, you have an odd number of symbols, and no strings with even length and an odd number of symbols exist. Because of this, $a_{\text{even}} = 0$. For odd k, we can take zero or two or four or... of the k positions for the zeros (and the remaining ones for the ones), which gives

$$\sum_{n=0}^{\infty} \binom{k}{2n}$$

ways. (The binomial coefficients become zero when we've passed the maximal number of zeros.)

(d) Both expressions say that $a_{\text{even}} = 0$. The expression in (a) says that $a_k = 2^{k-1}$, the expression in (b) that a_k is the sum of the number of even subsets of a set with k elements. In *Discrete Mathematics and Discrete Models* the fact that a set has an equal number of odd and even subsets is shown, and furthermore that the power set consists of 2^k subsets. The even ones are then of half of those, that is, $2^k/2 = 2^{k-1}$ subsets.

6.31 According to the definition of all $g_i(x)$ we get

$$g_1(x)\cdots g_n(x) = \sum_{\substack{k_1 \in M_1 \\ \vdots \\ k_n \in M_n}} \frac{x^{k_1+\cdots+k_n}}{k_1!\cdots k_n!}.$$

After extension by $\frac{k!}{k!}$ we see that the coefficient in front of $\frac{x^k}{k!}$ is

$$\sum_{\substack{k_1 \in M_1 \\ \vdots \\ k_n \in M_n \\ k_1+\ldots k_n = k}} \frac{k!}{k_1!\cdots k_n!}.$$

Every term $\frac{k!}{k_1!\cdots k_n!}$ is the number of different strings with k symbols out of

which k_1 of kind No. 1, k_2 of kind No. 2, etc. Thus the sum gives the total number of different strings with k symbols and permitted numbers of each kind.

6.32 If the zeros can't stand side by side we have to distribute the available ones over the spaces. Furthermore we may put some ones at the beginning and the end. Each string thus corresponds to a solution of the equation

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 10,$$
$$x_1, x_6 \geq 0, \quad x_2, \ldots, x_5 \geq 1$$

The equation can be rewritten as

$$y_1 + x_2 + x_3 + x_4 + x_5 + y_6 = 12,$$
$$y_1, y_6 \geq 1, \quad x_2, \ldots, x_5 \geq 1$$

and an equation of this kind has $\binom{12-1}{6-1} = \binom{11}{5} = 462$ different solutions.

6.33 Generalise the bijection in exercise 6.10.

6.34 Sylvester's bijection takes every partition of n that doesn't consist of odd distinct parts and pair it with another such partition where the number of parts differs by one. Thus exactly half of these partitions have an even number of parts. To calculate the difference

$$p(n \mid \text{even number of parts}) - p(n \mid \text{odd number of parts})$$

we thus only need to study the partitions in odd distinct parts. Since the sum of an odd number of odd parts will always be odd, the parity of the number of parts of these partitions will be equal to the parity of the number n. In other words, all partitions into odd distinct parts will add to the positive term if n is even and to the negative term if n is odd.

6.35 If we give each part a negative sign in the normal generating function of the partition function, partitions with an even number of parts will be counted positively and partition with an odd number of parts be counted negatively. The generating function of the left-hand side of the identity will thus be

$$\prod_{k=1}^{\infty} \frac{1}{1+x^k}.$$

In the same way, the generating function of the right-hand side in the identity will be

$$\prod_{k=1}^{\infty} (1 - x^{2k-1})$$

if we use that the parity of n is equal to the parity of the number of parts in a partition into odd distinct parts. That these two generating function are equal follows in the same way as for Euler's identity – the expressions are simply inverted here!

6.37

(a) The "gable" of the whole box can in principle look in three ways:

In the first case, two blocks have been added to a box that was one unit shorter, and the blocks can be turned in two ways (horisontally and vertically), so there are $2a_{n-1}$ boxes of this kind of length n. In the second case, four blocks have been added to a box that was two units shorter, so there are a_{n-2} boxes of this kind. And in the last case, three blocks have been added to a box that ended with an "step". The step can be placed in four different ways, so there are $4s_{n-1}$ boxes of this kind. And more kinds that these don't exist.

The cut-down box can be made either by putting a block next to a whole box of length $n-1$ or by turning a cut-down box of length $n-1$ upside down and pushing the short sides of two blocks under the step.

(b)

$$a_n = \tfrac{1}{6}\big(2(-1)^n + (2+\sqrt{3})^{n+1} + (2-\sqrt{3})^{n+1}\big)$$
$$s_n = \tfrac{1}{6}\big((-1)^{n+1} + \tfrac{1+\sqrt{3}}{2}(2+\sqrt{3})^n + \tfrac{1-\sqrt{3}}{2}(2-\sqrt{3})^n\big)$$

6.38 We set up the generating functions for the different kinds of candles:

$$S(x) = \frac{x^0}{0!} + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \ldots = \frac{e^x + e^{-x}}{2}$$
$$T(x) = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}$$
$$F(x) = \frac{x^0}{0!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \ldots = e^x - (\frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!})$$
$$G(x) = \frac{x^1}{1!} + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots = \frac{e^x - e^{-x}}{2}$$

The number of ways of setting up n aquired candles is the coefficient in front of $\frac{x^n}{n!}$ in $S(x)T(x)F(x)G(x)$.

6.39

(a) It seems reasonable that about $\frac{1}{3}$ of the 2^n code words will contain a multiple of three ones. (It can't be exactly that number, since the division won't break even.)

(b) We set up the exponential generating function:

$$\begin{aligned}&(1+\frac{x}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\cdots)\\&\quad\cdot(1+\frac{x^3}{3!}+\frac{x^6}{6!}+\frac{x^9}{9!})\\&=e^x(1+\frac{x^3}{3!}+\frac{x^6}{6!}+\frac{x^9}{9!})\end{aligned}$$

The answer is the coefficient in front of $\frac{x^n}{n!}$ in the expression above.

(c) We pick the terms with the correct degree from the expression:

$$\begin{aligned}&\frac{x^7}{7!}1+\frac{x^4}{4!}\frac{x^3}{3!}+\frac{x^1}{1!}\frac{x^6}{6!}\\&=(1+\frac{7!}{4!\,3!}+\frac{7!}{6!\,1!})\frac{x^7}{7!}\\&=1+35+7\\&=43\end{aligned}$$

The proportion is then $43/2^7 \approx 0.33 \approx 1/3$.

(d) None. The minimal distance between the words is 2, for if you exchange the places of a one and a zero in a code word the result is a valid code word. So we can detect all errors in 1 bit, but that's not enough to get error correction.

Chapter 7

7.3 When laying pipes for water and drains, for instance. You can't let pipes of this kind pass through each other. It's possible to let a water pipe make a detour to avoid a crossing, but that increases the resistance to the flow and should, if possible, be avoided. And drains have to be laid in a straight line, and with a suitable gradient (otherwise you'll get blockages). (By the way, to lessen problems, the ground has been subdivided into different levels, so that the different kinds of pipes are placed at different depths, and thus don't have any problems crossing each other.)

7.4 If two graphs are homeomorphic you can transform one into the other by dissolving some nodes of degree 2 and adding others. Neither of these operations can either add or remove edges that cross, so if there were any crossings before it will still be the case afterwards, and if there weren't any before there won't be any afterwards.

7.5 A graph is planar if and only if it doesn't has any subgraph that is homeomorphic to $K_{3,3}$ or K_5.

7.6 A subgraph can be created by drawing the full graph and then ereasing the parts one doesn't want. If you have a planar drawing, ereasing can't generate any intersecting edges, so all subgraphs have to be planar as well.

On the other hand, the nonplanar graph $K_{3,3}$ for instance has the very planar subgraph K_1.

7.7 If all the cycles have the length at least k, and there are any cycles at all, then each face has to have at least k edges. From the point of view of the faces there are then at least fk edges, but each edge is seen from two faces (or twice in the same face, if it ends with a leaf) so we get the relationship $2e \geq fk$ between the numbers of edges and faces. Euler's formula then says that $f = 2-v+e$, which substituted into the inequality gives

$$\begin{aligned}2e &\geq (2-v+e)k\\2e &\geq (2-v)k+ek\\(v-2)k &\geq ek-2e\\(v-2)k &\geq e(k-2)\\(v-2)\frac{k}{k-2} &\geq e.\end{aligned}$$

Cycles have at least three edges, so $k-2$ has to be positive, which means that we can divide by $k-1$ without turning the inequality around.

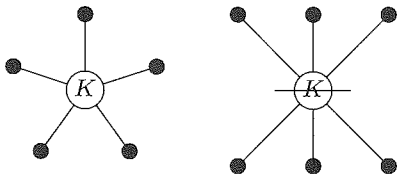
7.8 If someone says that a graph is nonplanar then said person has to be able to indicate a subgraph homeomorphic to either K_5 or $K_{3,3}$. You first have to check that the indicated really is a subgraph (that is to say, that the edges used are found in the original graph) which can be done in linear time.

Then you have to check that it's homeomorphic to a suitable graph. That can be done for instance by removing "unnecessay" nodes of degree 2. Each such node is removed and replace by a direct

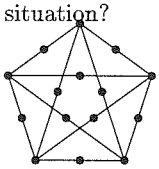
edge between the nodes in case. That should be possible to do in linear time in the number of edges. And when no unnecessary edges are left it's easy to verify whether what is left is a K_5, a $K_{3,3}$, or something else.

7.9

(a)



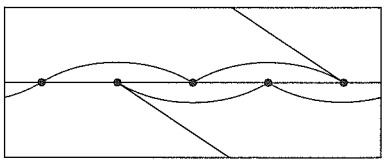
(b) How do you use Kimmo's nice nodes to untangle this highly nonplanar situation?



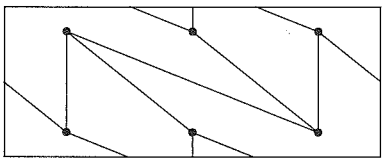
The problem is that the inventions are only able to handle subgraphs *isomorphic* to K_5 and $K_{3,3}$, not ones that are *homeomorphic*.

7.10 Most hobby shops have tori in styrofoam that are cheap to buy, if you find it hard to visualise the problem. Mark the nodes using pins and the edges using thread, to make it possible to try out different configurations.

(a)



(b)



This is, by the way, the graph drawn on the torus on page 137.

(c) Because the removal of a edge in a cycle doesn't automatically ensure that you get one face less. We may for instance draw a three-cycle around the torus. Then we have one face, and if we remove an edge we'll still have one face!

7.11 If you enflate the polyhedron somewhat it become sperical, without generating any intersecting edges. And things drawn on spheres can be drawn on paper, according to the previous exercise.

7.12 No. The graph has to be connected and mustn't be a tree, and all the nodes have to have a degree of at least 3 for the graph to correspond to a geometrical solid.

7.13 In a Platonan solid *all* the faces have the same number of edges, so we get an equality in the in exercise 7.7 derived relationship. Furthermore all the nodes have the same degree, $g = 2e/v$. This degree has additionally to be at least 3, otherwise we get a polygon. All the nunbers involved have to be integers.

$$e = \frac{k}{k-2}(v-2) \quad \Rightarrow$$

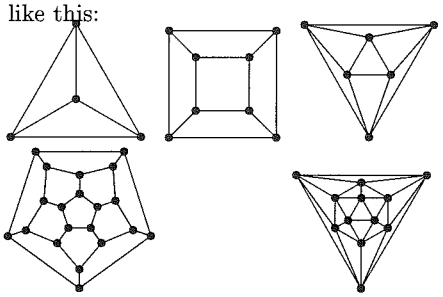
$$(k-2)e - k(v-2) = 0$$

So we have to find integer values of e, v, k, and g that satisfies the equality.
We try a number of combinations of k and g, starting at 3.

- $k = 3$, $g = 3$ give $e = 6$, $v = 4$, which is the tetrahedron.
- $k = 3$, $g = 4$ give $e = 12$, $v = 6$, which is the octahedron.
- $k = 3$, $g = 5$ give $e = 30$, $v = 12$, which is the icosahedron.
- $k = 3$, $g > 4$ doesn't give a positive number of nodes.
- $k = 4$, $g = 3$ give $e = 12$, $v = 8$, which is the cube.
- $k = 4$, $g > 3$ doesn't give a positive number of nodes.
- $k = 5$, $g = 3$ gives $e = 30$, $v = 20$, which is the dodecahedron.
- $k = 5$, $g > 3$ doesn't give a positive number of nodes.
- $k > 5$ doesn't give a positive number of nodes.

These are the only solutions!

Drawn in a planar way, the solids look like this:



Note by the way that the tetrahedron is a K_4.

7.14 Yes. Try for intance to "fold out" the node with degree 1 from the square in the graph in example 7.4.

7.16

(a) The tetrahedron.

(b) The octahedron.

(c) The cube.

(d) The icosahedron.

(e) The dodecahedron.

By the way, it's always the case that the dual of a polyhedron is a poyhedron!

7.17 Put one dot on each country, connect two dots if the countries are bordering to each other.

7.19 The original graph, provided that you haven't redrawn the dual to something isomorphic before the redualisation.

7.20 The graphs are **Eulerian**: you can find a circuit consisting of all the edges. From a practical point of view this means that the whole thing can be strung onto one single thread, where the ends can be tied together when it's finished. (To actually succeed in making such a circuit is a fairly large challenge in the three-loop construction...)

7.23 In a complete graph, all the nodes need different colours, which gives the same chromatic number as the number of nodes: n.

7.24 2, if n is even (colour every second node in one colour, every second one in the other). 3 if n is odd (one node will be squeezed between two differently coloured ones if you try to get by with two colours).

7.25

(a) If the algorithm takes the nodes in order, the first one will get colour 1, the second one colour 2, the third colour 1, and so on all the way around. So two colours will be used. If the algorithm on the other hand happens to take two opposing nodes at start, both will be assigned colour 1. The next node on the way between the coloured ones will be given colour 2, and the third, which now is placed between two nodes with different colours has to be assigned colour 3. Now three colours are used!

(b) Start with some colouring with the optimal number of colours and recolour the graph using the greedy algorithm in such an order that all nodes with colour 1 are coloured first, then all nodes with colour 2, and so on. This will give a colouring that to each node assigns a colour with at most the same number as in the original optimal colouring, and which thus is optimal as well.

(c) You can run the greedy colouring algorithm using all possible orders of the nodes and choose the colouring that was the best, since according to the previous exercise, it will be optimal. For a graph with n nodes, that gives $n!$ different orders to try, so this is not an efficient way of determining the chromatic number.

7.26 The largest clique corresponds to the largest complete graph that is part of the graph. A complete graph can't be coloured with less than n colours, so the chromatic number of the whole graph has to be at least n.

7.27

(a) The greedy algorithm always uses the lowest colour number possible. The highest number possible you get if the node is connected to a lot of different nodes, which already have been assigned different colours with lower numbers. There can be at most $\Delta(G)$ nodes connected to a node, so you'll never need a higher colour number than $\Delta(G) + 1$.

(b) We may for instance look upon a star: a node with a terribly large degree in the middle, and then protruding beams. This is a bipartite graph, which can be two-coloured, but the estimation will give one colour more than beams, which is a lot to much.

7.28 Let each form be represented by a node. Connect the nodes if there is a pair of siblings with one child in one of the forms and the other child in the other form. The problem is now to find a proper colouring with as few colours as possible. Each colour will represent a time-slot. (Note that finding the chromatic number isn't enough – you really need the colouring as well!)

7.29 We use theorem 7.4, and try to remove edges so that we get something easy to analyse left. We let the chromatic polynomials be represented by

pictures, that is simpler from a notational point of view.

$$= \lambda(\lambda-1)^4(\lambda-2) - \lambda(\lambda-1)^2(\lambda-2)^2$$

Substituting $\lambda = 3$ we get the answer 36 colourings.

7.30 The man seems to consists of hood, face, eyes, beard, mittens, sweater, trousers, and shoes, in total eight areas. One of the mittens is bordering to both trousers and sweater, the remaining things to just one area each. This gives the chromatic polynomial

$$P(\lambda) = \lambda(\lambda-1)^6(\lambda-2)$$

If we colour all the areas, this makes it possible to make $3 \cdot 2^6 \cdot 1 = 192$ different pictures. If we besides realise that it's possible to let some areas keep the original colour of the paper, we have actually four colours available, and are thus able to make $4 \cdot 3^6 \cdot 2 = 5832$ different pictures. (Whether a black-bearded father Christmas with red eyes and a white hood creates that much of an atmosphere is a different matter.)

7.31

(a) λ^n

(b) $\lambda(\lambda-1)(\lambda-2)\ldots(\lambda-n+1)$

(c) $\lambda(\lambda-1)^{(n-1)}$

7.32

$$\begin{cases} P_{C_1} = \lambda \\ P_{C_n} = \lambda(\lambda-1)^{n-1} - P_{C_{n-1}} \end{cases}$$

7.33 $\chi(G) = \min\{\lambda \mid P_G(\lambda) > 0\}$

7.34 Using the recursion, we can write $P_G(\lambda)$ as a sum of chromatic polynomials of graphs without edges, and the chromatic polynomials of graphs like that are of the form λ^m. Sums of terms like this are precisely polynomials.

7.35 If we from G remove one edge at a time until no more edges remain, the recursion gives that the chromatic polynomial of G is equal to the chromatic polynomial of n nodes without edges ($= \lambda^n$) minus the sum of a lot of chromatic polynomials of the graphs where one edge has been contracted and thus $n-1$ nodes remain. According to the inductive hypothesis, graphs with $n-1$ nodes have chromatic polynomials of the form $\lambda^{n-1} - a_{n-2}\lambda^{n-2} + a_{n-3}\lambda^{n-3} - a_{n-4}\lambda^{n-4} + \ldots$, where all $a_k \geq 0$. Subtraction of all these polynomials from λ^n gives the form of $P_G(\lambda)$ we were looking for.

7.36 Assume the opposite, that is, that there exists a planar graph G where all the nodes have a degree of at least six. Let v be the number of verticies in G. Since at least six edges originate from each vertex and each edge touches exactly two verticies, the number of edges have to satisfy the inequality $e \geq 3v$. But in each planar graph, according to corollary 7.2, we have $e \leq 3v-6$. This is a contradiction, so the assumption that there is a planar graph G where all the nodes have a degree of at least six has to be discarded.

7.37 Assume that all graphs with up to n nodes can be five-coloured. Now study a graph with $n+1$ nodes. In this graph there is some node with a degree of at most 5. If we remove it, the remaining graph can be five-coloured. Now put the node back. If its degree is less than five, its neighbours can have at most four different colours, and the node can be coloured using the fifth colour.

If the node actually is of degree five, we can see two cases: Either the five neighbours have less than five colours in total, and then we can just as before take the fifth colour for the node. Or all the neighbours have different colours.

Now study two of the neighbours, say v_1 och v_3, two of the adjacent nodes that aren't side by side. Let's say that they have the colours c_1 and c_3. Pick out a subgraph from the graph, the subgraph consisting of all nodes with the colours c_1 and c_3, and the connecting edges. If v_1 and v_3 belong to different components of this graph, we can swap the colours in one of the components, so that all the nodes with the colour c_1 instead get the colour c_3, and vice versa. This can't destoy the correctness of the previous colouring. And now the new node is surrounded by things with just four colours, and can be assigned the fifth one.

If v_1 and v_2 on the other hand belong to the same component, swapping the colours won't help, and then we instead have to study a different pair of neighbours, say v_2 and v_4 and reason in the same way. It then proves that they can't belong to the same component of the c_2-

c_4-subgraph, since the connected c_1-c_3-subgraph forms a barrier between them. And then we can re-colour one of the components.

7.38 We can take a K_5 and add a node to each edge, in the same way as in exercise 7.9. This gives a bipartite graph (with the original nodes and the added nodes as the two kinds). A bipartite graph can be two-coloured, and thus four-coloured as well.

7.39

(a) A graph that doesn't include K_1 can't have any nodes! And if there aren't any nodes, you can use zero colours without any problems concerning adjacent nodes getting the same colour.

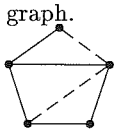
(b) A graph not including K_2 lacks edges, and only consists of isolated nodes. And then you can without any problems manage using one colour.

(c) A graph not including K_3 lacks cycles, and is thus a tree (or many). Trees are bipartite, and can be two-coloured.

(d) K_4 is in principle a cycle with two crossing diagonals. If our graph doesn't include anything like that it can be drawn with cycles side by side, with one or several edges in common. A cycle demands at most three colours, and the colouring of one cycle is then the start of the colouring of the adjacent cycle, and can't complicate anything.

(e) No, we refrain from writing down this proof. But if you figure it out, your teacher definitely wants to see it!

7.40 Every planar graph can be extended to an almost-triangulated graph by subdividing the faces with more than three edges, and connecting nodes on projecting branches to other nodes in the graph.

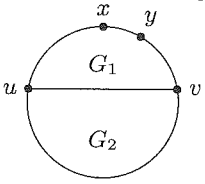


Now assume that we have a planar graph with five-lists. We extend it to an almost-triangular graph. If this extended graph can be five-list coloured, the original graph has to be five-list colourable as well (using the same colouring), since the removal of an edge can't ever destroy a colouring.

7.41

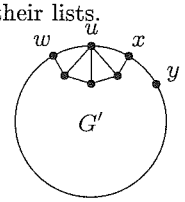
(a) When G has three nodes out of which x and y have the colours blue and red, respectively, and the third node has three colours on its list, one of these colours has to be free.

(b) The chord uv divides the graph into two parts, G_1 (containing the adjacent nodes x and y) and G_2.



Now G_1 satisfies the conditions and has less than n nodes and can thereby be list coloured according to the assumption. Thereby the neighbours u and v have been assigned two colours, so now G_2 satisfies the conditions as well and can therefore be coloured according to the assumption. Now all of G has been list coloured.

(c) Let u be the neighbour of x on the cycle C which isn't y. Since G lacks a chord, u has only two neighbours on C (x and w), the remainder of the neighbours are inner nodes and thereby have at least five colours on their lists.



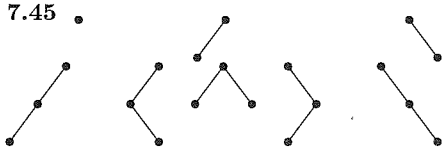
Choose two out of u's list colours that aren't the same as the colour of x and remove them from the lists of all inner neighbours of u. Then remove u and its edges, and what remains is a graph G' which satisfies the conditions and which has less than n nodes. Thereby G' can be list coloured. To get the whole of G coloured, colouring u is what's remains to do, and one of the two previously selected colours has to be free since w can only have one of them.

7.42 The previous exercise means that every almost-triangulated graph can be coloured using 5-lists for the inner nodes and 3-lists for the outer ones. It can't get any less colourable if you extend the outer lists to the length 5, so almost-triangulated graphs are five-list colourable. And since they are, all planar graphs are as well.

7.43 Let all the lists consist of the same five colours.

7.44 The statement is trivially true for graphs with one node. Assume that it's true for graphs with less than n nodes and let G be a stable directed graph with n nodes that satisfies the condition that each node has more colours than its number of outgoing edges. Choose some colour and let S be the set of all nodes that has said colour on their lists. Since G is stable there exists an independent subset $S' \subset S$ so that each node in $S \setminus S'$ has a directed edge to some node in S'. Since the nodes in S' are independende, that is, have no edges inbetween, we can colour all of them using the chosen colour. Remove the coloured nodes and remove the chosen colour from the lists of all the remaining nodes. The nodes that get one colour less are the nodes in $S \setminus S'$, but for them at least one edge vanishes according to the stability property, so the remaining graph as well satisfies the condition that each node has more colours on its list than the number of outgoing edges, and thereby it can be coloured using the remaining colours according to the inductive hypothesis.

7.45



7.46

$$\begin{aligned} c_4 &= c_0c_3 + c_1c_2 + c_2c_1 + c_3c_0 \\ &= 1\cdot 5 + 1\cdot 2 + 2\cdot 1 + 5\cdot 1 = 14 \\ c_5 &= c_0c_4 + c_1c_3 + c_2c_2 + c_3c_1 + c_4c_0 \\ &= 1\cdot 14 + 1\cdot 5 + 2\cdot 2 + 5\cdot 1 + 14\cdot 1 \\ &= 42 \\ c_6 &= c_0c_5 + c_1c_4 + c_2c_3 \\ &\quad + c_3c_2 + c_4c_1 + c_5c_0 \\ &= 1\cdot 42 + 1\cdot 14 + 2\cdot 5 \\ &\quad + 5\cdot 2 + 14\cdot 1 + 42\cdot 1 = 132 \end{aligned}$$

7.47 It's easily shown by induction that a binary tree with n nodes has $n+1$ places where new leaves can be added. Expand the binary trees with new leaves that are are precisely the factors x_1 to x_n from right to left. Then interprete the tree as a binary expression tree where all the inner nodes represent multiplications and we get a placement of parentheses. Conversely, each placement of parentheses gives a unique expression tree, which after removal of the leaves x_1 to x_n gives a binary tree with n nodes.

7.48

(a)

$$\sqrt{1-4x} = (1-4x)^{1/2}$$

{according to the generalised binomial theorem}

$$\begin{aligned} &= \sum_{n=0}^{\infty} \binom{1/2}{n} (-4x)^n \\ &= \sum_{n=0}^{\infty} \binom{1/2}{n} (-4)^n x^n \end{aligned}$$

(b) We carry out the calculations for an even value of n; odd values are handled in an analogue way, but some details are different.

$$\binom{1/2}{n}(-4)^n =$$

$$= \frac{1/2 \cdot (-1/2)(-3/2)\cdots(-(2n-3)/2)}{1\cdot 2\cdot 3\cdots n} \cdot (-4)^n$$

{Factor out the halves}

$$= \frac{(-1)\cdot 1\cdot 3\cdots(2n-3)}{1\cdot 2\cdot 3\cdots n}(-\frac{1}{2})^n \cdot (-4)^n$$

{Cancel the odd factors in the denominator}

$$= (-1)\frac{(n+1)(n+3)\cdots(2n-3)}{2\cdot 4\cdots n} \cdot \big((-\frac{1}{2})(-4)\big)^n$$

{Expand to "fill in the gaps" in the numerator }

$$= (-1)\frac{(n+1)(n+2)\cdots \cdot (2n-3)(2n-2)}{2\cdot 4\cdots n \cdot (n+2)\cdots(2n-2)} 2^n$$

{Factor out 2:s in the denominator}

$$= (-1)\frac{(n+1)(n+2)\cdots \cdot (2n-3)(2n-2)}{1\cdot 2\cdots(n-1)\cdot 2^{n-1}} 2^n$$

{Expand by $(2n-1)2n$}

$$= (-1)\frac{(n-1)\cdots(2n-2) \cdot (2n-1)2n}{1\cdot 2\cdots(n-1) \cdot (2n-1)2n\cdot 2^{n-1}} 2^n$$

$$= \frac{-1}{2n-1}\cdot\frac{(n-1)\cdots 2n}{1\cdot 2\cdots n\cdot 2^n}\cdot 2^n$$

$$= \frac{-1}{2n-1}\binom{2n}{n}$$

(c)

$$\begin{aligned}
&\frac{1-\sqrt{1-4x}}{2x} = \\
&= \frac{1}{2x}\left(1-\sqrt{1-4x}\right) \\
&= \frac{1}{2x}\left(1-\sum_{m=0}^{\infty}\binom{1/2}{m}(-4x)^m\right) \\
&= \frac{1}{2x}\left(1-\sum_{m=0}^{\infty}\frac{-1}{2m-1}\binom{2m}{m}x^m\right) \\
&= \frac{1}{2x}\left(1-\frac{-1}{2\cdot 0-1}\binom{2\cdot 0}{0}x^0\right. \\
&\qquad \left.-\sum_{m=1}^{\infty}\frac{-1}{2m-1}\binom{2m}{m}x^m\right) \\
&= \frac{1}{2x}\sum_{m=1}^{\infty}\frac{1}{2m-1}\binom{2m}{m}x^m \\
&= \sum_{m=1}^{\infty}\frac{1}{2x}\frac{1}{2m-1} \\
&\quad \cdot\frac{2m(2m-1)\cdots(m+1)}{1\cdot 2\cdots(m-1)m}x^m \\
&= \sum_{m=1}^{\infty}\frac{(2m-2)\cdots(m+1)m}{1\cdot 2\cdots(m-1)m}x^{m-1} \\
&= \sum_{m=1}^{\infty}\binom{2(m-1)}{m-1}\frac{1}{m}x^{m-1} \\
&\{\text{Insert } m-1=n\} \\
&= \sum_{n=0}^{\infty}\binom{2n}{n}\frac{1}{n+1}x^n
\end{aligned}$$

(d) Since the generating function of the Catalan numbers is

$$C(x) = \frac{1-\sqrt{1-4x}}{2x} = \sum_{n=0}^{\infty}\frac{1}{n+1}\binom{2n}{n}x^n$$

the Catalan numbers can be calculated using the formula

$$c_n = \frac{1}{n+1}\binom{2n}{n}$$

7.49

$$\begin{aligned}
&\binom{2n}{n}-\binom{2n}{n-1} \\
&= \frac{2n(2n-1)\ldots(n+2)(n+1)}{1\cdot 2\cdot\ldots\cdot(n-1)n} \\
&\quad -\frac{2n(2n-1)\ldots(n-2)}{1\cdot 2\cdot\ldots\cdot(n-1)} \\
&= \frac{2n(2n-1)\ldots(n+2)(n+1)}{1\cdot 2\cdot\ldots\cdot(n-1)n} \\
&\quad -\frac{2n(2n-1)\ldots(n+2)n}{1\cdot 2\cdot\ldots\cdot(n-1)n} \\
&= \frac{((n+1)-n)2n(2n-1)\ldots(n+2)}{1\cdot 2\cdot\ldots\cdot(n-1)n} \\
&= \frac{2n(2n-1)\ldots(n+2)}{1\cdot 2\cdot\ldots\cdot(n-1)n} \\
&= \frac{2n(2n-1)\ldots(n+2)(n+1)}{1\cdot 2\cdot\ldots\cdot(n-1)n(n+1)} \\
&= \frac{1}{n+1}\frac{2n(2n-1)\ldots(n+1)}{1\cdot 2\cdot\cdots\cdot n} \\
&= \frac{1}{n+1}\binom{2n}{n}
\end{aligned}$$

7.50 Each parentheses expression can be divided into one part up to the first point where one have the same number of right and left parentheses, and the rest of it. (So "(()())(())" is divided into "(()())" and "(())".) The part inside the first part corresponds to the left subtree, the rest corresponds to the right subtree. () corresponds to the tree with just one node. The first binary trees, which are shown in the answer to exercise 7.45, correspond, taken in order, to the parentheses expressions (), (()), ()(), ((())), (()()), (())(), ()(()), ()()().

7.51 The left parenthesis corresponds to a step upwards, the right parentesis to a step downwards.

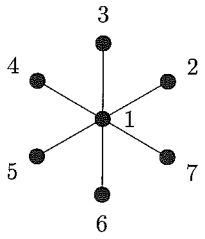
7.52

(a) (1, 4, 4, 1)

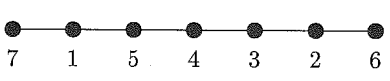
(b) (5, 2, 5, 2)

7.53

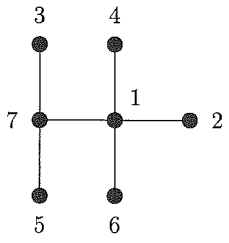
(a)



(b)



(c)



7.55 Nodes with just one neighbour are leaves, and they are removed without being included in the code, so it's true that they are included zero times. For the rest, we write down a node every time we remove its neighbour, and we can't remove the node until it has become a leaf, that is, only has one neighbour left. So all nodes that are removed are included the same number as the number of neighbours, minus one. The only nodes that haven't been removed are the two connected ones we have left when we are finished, and they as well have been put down once per removed neighbour, except for the other node which remains.

7.56 You draw one edge for each number in the code, and a final edge when the code is empty. That means that you generate $n-1$ edges. In each step, a node is picked from the list of leaves, and every node will sooner or later end up there, which means that there will be n nodes. A graph with $n-1$ edges and n nodes is either a tree or an unconnected graph with cycles. What does it take to make a cycle? It takes drawing an edge between two nodes that are already connected. But you easily realise using induction that during the whole process, the nodes that are mentioned in the code or in the list of leaves will keep on being in different components.

7.57

(a) 1. At least one employee and at most all of them must be working directly under a top manager (given that there are any employees): e ways.

2. From the e, appoint s submanagers: $\binom{e}{s}$ ways.

3. Repeat the reasoning, but now with the submanagers as top managers: $R(e-s,s)$ ways.

4. Each of the s submanagers has t topmanagers to choose from: t^s ways.

(b) The recursion becomes $R(e,t) = \sum_{s=1}^{e} \binom{e}{s} t^s R(e-s,s)$.

(c) $R(0,t)=1$, since organising one top manager without any employees can only be done in one way.

(d) Substitute $R(e,t) = t(t+e)^{e-1}$ and $R(e-s,s) = se^{e-s-1}$ into the recursion and establish that the left-hand side after expansion according to the binomial theorem is equal to the right-hand side. The left-hand side becomes

$$\sum_{k=0}^{e-1} \binom{e-1}{k} t^k e^{e-1-k},$$

while the right-hand side becomes

$$\sum_{s=1}^{e} \binom{e}{s} t^s s e^{e-s-1}.$$

Using that $\binom{e}{s}s = \binom{e-1}{s-1}e$ and substituting $s-1=k$ makes the right-hand side the same as the left-hand side.

(e) $R(n-1,1)$ is the number of possible hierarchies with one top manager and $n-1$ employees. Each such hierarchy can be modeled by a rooted tree, where the top manager (the root) gets the number 1, and the remaining persons are assigned numbers. And every numbered tree can be interpreted as an organisation chart for such a hierarchy.

7.61

(a) $p_{2m} = 2^{-2m}\binom{2m}{m}$

(b)

$$\begin{aligned}\frac{1}{\sqrt{1-x^2}} &= (1-x^2)^{-\frac{1}{2}} \\ &= 1 + \frac{1}{2}x^2 + \frac{1\cdot 3}{2\cdot 2}\frac{x^4}{2!} + \cdots \\ &= \sum_{m=0}^{\infty} p_{2m} x^{2m}\end{aligned}$$

(c) The sum $s_2 + s_4 + \ldots + s_{2m}$ is the total probability of returning to origo at some point inside the first $2m$ steps. That's of course equal to $1 - q_{2m}$.

(d) Inclusion and exclusion gives

$$\begin{aligned}s_{2m} = p_{2m} &- \sum_{i+j=2m} p_i p_j \\ &+ \sum_{i+j+k=2m} p_i p_j p_k - \ldots.\end{aligned}$$

We then have

$$\begin{aligned}&\sum s_{2m} x^{2m} \\ &= (p(x)-1) - (p(x)-1)^2 \\ &\quad + (p(x)-1)^3 - \ldots \\ &= \frac{p(x)-1}{1+(p(x)-1)} = 1 - \frac{1}{p(x)}\end{aligned}$$

(e) $S(x) = 1 - \sqrt{1-x^2}$.
$\sum s_{2m} = S(1) = 1$.

(f) $S'(x) = \sum 2ms_{2m}x^{2m-1}$,
so $S'(1) = \sum 2ms_{2m}$ which is the average number of steps before origo is revisited.

(g) $S'(x) = \frac{x}{\sqrt{1-x^2}}$, so $S'(1) = \infty$. So it's true that the risk of never returning to origo is zero, but it takes an unlimited amount of time before you get back!

7.63 The probability p_{2m} of a random walk in three dimensions ending at origo after $2m$ steps can be expressed using multinomial numbers:

$$p_{2m} = 6^{-2m} \sum \binom{2m}{i,i,j,j,k,k} =$$

$$= 2^{-2m}\binom{2m}{m} \sum 3^{-2m} \binom{m}{i,j,k}^2,$$

where we sum over all natural numbers i, j, k such that $i + j + k = m$. To simplify this, we make the overestimation

$$\binom{m}{i,j,k} \leq \binom{m}{\frac{m}{3},\frac{m}{3},\frac{m}{3}},$$

which gives

$$p_{2m} \leq 2^{-2m}\binom{2m}{m} 3^{-m} \binom{m}{\frac{m}{3},\frac{m}{3},\frac{m}{3}} \cdot$$

$$\cdot \sum 3^{-m} \binom{m}{i,j,k}.$$

Here we see that the last sum is exactly equal to one! Stirling's formula now gives that p_{2m} is less than some constant times $\frac{1}{m\sqrt{m}}$. Thereby $\sum p_{2m}$ converges, and the same line of reasoning as in the two-dimensional case now gives that the probability of a random walk in space never ever will return to origo *isn't zero*.

7.64

(a) Yes, that G is r-regular means that the complement graph $\overline{G}$ is $(n - r - 1)$-regular, since with n nodes in the graph there will always exist $n - r - 1$ nodes that a node in G isn't connected to and to which there thus are edges in $\overline{G}$.

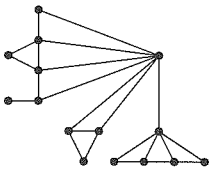
(b) No. There are counterexamples. The cube for instance is 3-regular, but the regions are not triangles there!

(c) No. There are counterexamples. For instance, the octahedron is 4-regular and planar.

(d) No. There are counterexamples. The cube is 3-regular and bipartite besides, that is, has $\chi(G) = 2$.

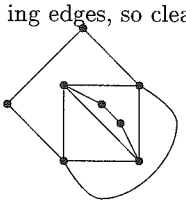
7.65

(a)



Chromatic polynomial: $\lambda(\lambda-1)^5(\lambda-2)^9$. Chromatic number: 3.

(b) It can be drawn without intersecting edges, so clearly it's planar!



(c) The two protruding wings don't really add anything to the difficulties, since these edges can be drawn parallel to the ones in the inner cross. The graph is thus in principle a K_4, which can't possibly have either a K_5 or a $K_{3,3}$ as a subgraph!

7.67

(a) In a planar graph the inequality $e \leq 3v - 6$ holds. If the graph had had 3001 edges we would have had $e = 3001$ and $3v - 6 = 3000$.

(b) Place the 1002 nodes in a circle. Use upp 1002 edges by connecting the circle into a cycle. Then use up a further 999 edges by connecting one of the nodes in the cycle to all the nodes to which it isn't already connected. (Can be done like a fan without any crossings.) Now look at the next node. Connect it to all the nodes to which it isn't already connected using 999 edges placed on the outside of the cycle. These can be drawn withot any crossings as well. $1002 + 999 + 999 = 3000$, so we have used all the edges.

7.69 Yes. Example: let the graphs be two **chains** of equal length, trees without branches. They can be two-coloured. Then connect the green nodes in one of them with the red nodes in the other. It's possible, so the joined graph has the chromatic number 2 as well.

7.70

(a) By using the recursion for the chromatic polynomial one gets that the graph has the chromatic polynomial $\lambda(\lambda-1)(\lambda-2)(\lambda^2-3\lambda+3)$. Using three frequencies (colours), there are thus $3\cdot 2\cdot 1\cdot 3 = 18$ acceptable frequency assignments. But we are not allowed to use the frequency 106 on x. For reasons of symmetry, exacly two thirds of the assignments must not be using 106 on x, that is, 12 of them.

(b) Six out of the acceptable assignments have frequency 106 on x and six of them have frequency 106 on y. It's easy to see that there are $2\cdot 2\cdot 1$ assignments that give frequency 106 to both x and y. According to inclusion/exclusion, there are $18-6-6+4 = 10$ acceptable assignments which don't have 106 on either x or y.

7.72 In this case, the easiest way is to show that the number of ways match the recursion for the Catalan numbers. Call the number of ways a_n. A triangle is already partitioned, som $a_3 = 1$. A "diagon" can also just look in one way, so $a_2 = 1$. If we now study an $n+1$-gon, we can fix one side in this. The side has to be part of some triangle, which means that it has to be connected to one of the $n-1$ remaining corners. On each side of the triangle, we then get an polygon with, say, k corners and one with $n-k+2$ cornes. k can be anything between 2 and n. That means that we have the recurrence equation

$$a_{n+1} = \sum_{k=2}^{n} a_k\, a_{n-k+2}$$

If we now carry out a renumbering, and call a_{n+2} c_m, the recursion can be written

$$c_{m+1} = \sum_{k=0}^{m} c_k c_{m-k}, \qquad c_0 = 1,\ c_1 = 1$$

This is the recursion of the Catalan numbers!

7.73 There are $n!$ node-labelings of each binary tree with n nodes. There are the Catalan number c_n such trees. Thus there are $n!c_n$ different node-labeled binary trees.

7.74

(a) Two. A chain and a branching graph looking like a Y.

(b) As many as the number of lists with two elements between 1 and 4, that is, $4^2 = 16$.

(c) One method: Generate all the list in a systematic way, then make up the trees based on them. (Ought to guarantee that you don't forget any of them.) Other method: Try in some systematic way to distribute the labels on the two basic trees that exist.

(d) No, we can't be bothered to draw the answer!

7.75

(a) We have two recursions: $a_{n+1} = b_n/2+c_n/2 = b_n$ and $b_{n+1} = a_n/2+c_n/2 = a_n/2+b_n/2$. From this we can conclude that $a_{n+1}-b_{n+1} = -\frac{1}{2}(a_n-b_n)$. Thus, a_n-b_n approaches zero when n increases towards infinity.

(b) In each bipartite graph, an even number of steps is needed to get back to the starting point. $K_{1000,1000}$ has $1000^2 = 1\,000\,000$ edges and 2000 nodes.

Chapter 8

8.1 If Erdős is placed in the middle of the graph, two persons placed at opposite sides may have the distance 27 to *each other* at the same time as they have the distance 15 to Erdős.

8.2 One has to find cooperative connections that in combination make up a subgraph homeomorphic to K_5 or $K_{3,3}$. A K_5 looking like this has been found: The nodes are Paul Erdős, Frank Harary, Ron Graham, Dan Kleitman, and Saharon Shelah. Erdős has edges to all the four others. From Harary to Graham there is a path via Stefan Burr, from Harary to Kleitman via Jin Akiyama and Noga Alon, from Harary to Shelah via Andreas Blass. Graham has an edge to Kleitman, and there is a path from Graham to Shelah via Joel Spencer. Fi-

nally, there is a path from Graham to Shela via Eric Milner.

A simpler, but less intriguing, solution is to take any article whatsoever which had five coathours, for instance *Sorting a Bridge Hand* by Eriksson, Eriksson, Karlander, Svensson, and Wästlund (see section 11.5). It generates a K_5-subgraph in the Erdős graph directly.

8.3

(a) If the graph has the diameter 1, every node needs to be connected to every other node, so we must have a complete graph. K_n has $\binom{n}{2}$ edges.

(b) Here the smallest graph seems to be a star-formed net, with a central node and the remaining vertices each on a beam. The central node is at the distance 1 from the others, the remaining ones are at the distance 2 from each other. A star like this has $n-1$ edges.

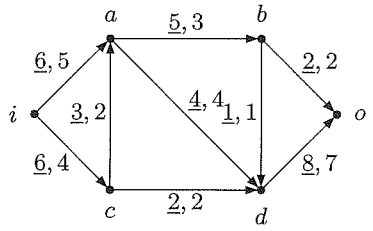
8.5 We substitute $n = 5 \cdot 10^5$ and $\Delta = 100$ into the formula, and note that it tells how many nodes you get at most:

$$5 \cdot 10^9 \leq \frac{100(100-1)^D - 2}{100-2} \quad \Rightarrow$$

$$D \geq \frac{\log\left(\frac{99 \cdot 5 \cdot 10^9 + 2}{100}\right)}{\log 99} \approx 4{,}86$$

Since the diameter is an integer, it thus has to be at least 5.

8.8



The flows in the non-saturated edges depend a bit on the order in which the flow-augmenting paths were taken. The maxflow and the flow in the saturated edges will be the same in all correct answers. The maxflow is 9, and the mincut is $P_i = \{i, a, b, c\}$, $P_o = \{d, o\}$.

8.10

(a) We know that the inflow into a node always is the same as the outflow. That means that the sum of the flows (signs included) is zero. A cut on one side cuts some of the edges; if the cut is moved to the opposite side, the complement set of the edges are cut, and besides the flows are regarded from the opposite point of view (so that positive becomes negative). The two flows give the total flow through the node, that is, zero, and must because of this be of equal size.

(b) The in the question described situation corresponds to us moving a cut a little bit, so that it ends up on the other side of the node a. The edges that were unconnected to a remains, and the flow through the ones on the other side of a is according to the previous exercise equal to the one on the first side, so the total flow in the cut can't have been changed by the operation.

(c) Every cut that exists can be made by taking a cut that cuts of the source from the rest, and step by step moving it past the nodes. Since the flow doesn't change when you move the cut past a node, all the cuts have to have the same flow.

8.11

(a) We can put in directed edges in both directions.

(b) We can add a "super source" and a "super sink" at the ends of the graph, and connect the actual sources and sinks to these using edges with unlimited capacities.

(c) We can divide the node into two: inflow side and outflow side, and connect these with a directed edge with a capacity equal to the transportation capacity of the node.

(d) From the nodes with leakage, we can draw edges directly to the sink, with capacities corresponding to the maximum deficit. (In practice, though, the deficit probably is proportional to the flow, which makes the calculations somewhat harder.)

8.12

(a) Using the maxflow algorithm, we get an integer flow. Since all the capacities are 1 this means that in all the edges there is either a flow of 1 or a flow of 0. When one flow unit has left i it thus never divides but follows a unique path from i to o. All these paths have to be edge disjoint since otherwise some edge would be trasporting more than one flow unit. From the other point of view, we can send unit flows

through every set of edge disjoint paths. Thus the maximum flow has to be equal to the maximum number of edge disjoint paths from i to o.

(b) Since all the edge capacities are 1, the least number of edges, the removal of which destroys all the paths from i to o, will be equal to the capacity of the minimum cut. This in its turn is, according to the theorem, equal to the maxflow, which according to the exercise above is equal to the maximum number of edge disjoint paths from i to o.

(c) Change all inner nodes to edges with capacity 1, see exercise 8.11(c). Previously inner-node disjoint paths now correspond to edge disjoint paths. Use the exercise above.

(d) Add a source i with edges to all nodes on the left and a sink o with edges from all nodes on the right. An edge covering is then the same thing as a set of inner nodes, the removal of which destroys all paths from i to o. A matching is the same thing as a set of inner-node disjoint paths. Menger's theorem now gives König's theorem.

8.14

(a) You can count on the message sooner or later arriving (at least if the network is connected), which isn't certain at fixed routing, which breaks down if the chosen path breaks down. On the other hand, the route can be very long, while fixed routing usually follows the shortest path.

(b) So t_{ij} is the probability of node i sending on the message to node j. If the nodes aren't connected, that is, if $a_{ij} = 0$, this probability is zero. Otherwise the probability is equal for all the $\deg(i)$ nodes that are connected to i, and then it is $1/\deg(i)$ for each one of them. The probability of the final addressee k sending the message on is zero, though!

(c) If the message is to end up at node i after $n+1$ steps, it must in the previous step have been at some neighbour of i, and from this point have chosen to go on to i. If we sum up for all the neighbours of i, we get

$$p_i^{(n+1)} = \sum_{j \text{ neighbour of } i} p_j^n t_{ij}$$

since $t_{ij} = 0$ for non-neighbours

$$= \sum_{j=1}^{n} p_j^n t_{ij}$$

which is the dot product of the vector $\mathbf{p}^{(n)}$ and column j in T. If we do this for all the rows we have simply multiplied $\mathbf{p}^{(n)}$ with the matrix T.

(d)
$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(e) You easily realise that the following relationship holds for all $n \geq 0$:

$$\begin{aligned} p_1^{(n+1)} &= \frac{1}{3}p_2^{(n)} + \frac{1}{3}p_3^{(n)} \\ p_2^{(n+1)} &= \frac{1}{2}p_1^{(n)} + \frac{1}{3}p_3^{(n)} \\ p_3^{(n)} &= p_2^{(n)} \\ p_4^{(n+1)} &= \frac{1}{3}p_2^{(n)} + \frac{1}{3}p_3^{(n)} + p_4^{(n)} \end{aligned}$$

Elimination of all quantities except for p_4 gives the sought for recursion.

(f) The message ought to arrive sooner or later, so $c_n \longrightarrow 1$. Both $|(1+\sqrt{13})/6|$ and $|(1-\sqrt{13})/6|$ are smaller than 1, so the exponential terms both approach zero. It the sum is to approach one, the constant term has to be one.

8.15

(a) Cheapest is going via Östgötabanken, that cost is $4 + 13 = 17$.

(b) It's possible that all the other banks except for Nordea went via Östgöta as well! The only route that we can be completely sure still remains is because of this the one via Nordea, which costs $9 + 12 = 21$. More expensive than that it can't be. (It's probable that some of the other routes remain as well, but that can't be determined from the given information.)

(c) What's of interest is the maximum number of *node disjoint* paths there are between Handelsbanken and Skandiabanken. (If you are to knock out all of them you have to knock out at least one node per path. It's quite possible that this isn't enough – there may remain an alternative path, that zigzags between what remains of the original paths.) One way is to replace each node with an edge with capacity 1 (the capacities of the other edges is irrelevant, as long as it's

at least 1). Then we force a maximum flow throught the network. The flow will be equal to the number of node disjoint paths.

8.16

(a) Let $c(x, y)$ mean the cost of the cheapest route from x to y, with $c(x, x) = 0$. The shortest route from A to some node v can be found from

$$\min\{c(A, B) + c(B, v), \\ c(A, C) + c(C, v)\}$$

since the neighbours of A are B and C. For each v, $c(B, v)$ and $c(C, v)$ are found from the given tables. The table of A is

B	3	B
C	5	C
D	8	C
E	7	C
F	8	C

(b) An articulation point is a node in the network, which if it breaks down makes some nodes no longer able to communicate with each other.

(c) None.

(d) If a graph is biconnected it can't have any articulation points, since any node that is removed destroys at most one of the two node disjoint paths between each pair of nodes, so the number of components doesn't increase. Conversely, if a connected graph isn't biconnected there is some pair of nodes where every pair of paths has some node in common. It's easy to realise that in this case, all path between the pair of nodes must have some node in common. This node is then an articulation point.

8.17

(a) All the nodes get the message at some point, and thus send it on. i sends the same number of copies as its degree, the remaining ones one copy less than the degree. In total:

$$\sum_{g \in G} (\deg(g) - 1) + 1$$

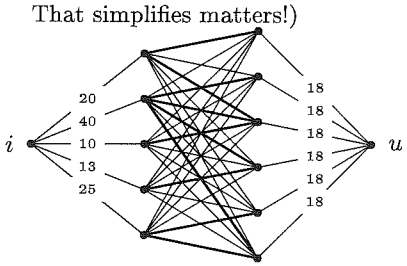
(b) If I'm unlucky I'm in direct contact with all the other nodes, and thereby get one copy from each before having the time to counter! $|G| - 1$ copies, then.

8.19 If we make a cut downwards to the right we cut edges with a total capacity of $35 + 17 + 17 + 29 = 98 < 100$.

8.20

(a) We can regard the common sum as a flow through a network. The r-numbers are the capacities of the edges from the inlet; the c-numbers are the edges to the outlet. The numbers in the matrix are flows in the intermediate edges. If we place a $K_{n,m}$ between the given edges, with an unlimited flow in each edge, we have the same number of edges as the number of elements that the matrix needs. The minimum cut in the network is through the ingoing edges or the outgoing edges (since the sum is the same) and according to Ford-Fulkerson's theorem this is equal to the maximum flow, so it is possible to make up the sum.

(b) This exercise isn't as hard as it looks at first glance, since the main part of the flow can be run straight through the network, which saturates a large number of edges and therby removes a number of others from the analysis. (Since there aren't any backwards edges, we will never have to decrease a given flow. That simplifies matters!)



If we choose to lead the flow through the edges marked in the graph, we get the matrix

18	0	2	0	0	0	20
0	18	6	5	0	11	40
0	0	10	0	0	0	10
0	0	0	13	0	0	13
0	0	0	0	18	7	25
18	18	18	18	18	18	108

8.21 If we work out a maximum flow, we see that the saturated edges define a minimum cut with the capacity $5 + 5 + 3 = 13$, which is less than the requirement.

Chapter 9

9.1

(a) One matching is *a*–*h*, *b*–*j*, *c*–*f*, *d*–*i*, *e*–*g*.

(b) One matching is *a*–*k*, *b*–*o*, *c*–*i*, *d*–*j*, *e*–*p*, *f*–*n*, *g*–*m*, *h*–*r*.

9.2

(a) The four nodes *b*, *d*, *e*, *f* are only connected to the three nodes *g*, *i*, *l*.

(b) The three nodes *a*, *c*, *e* are only connected to the two nodes *h* and *j*.

9.3

(a) One matching is A–c, B–e, C–a, D–f, E–b, F–g, G–d.

(b) One matching is A–b, B–d, C–g, D–f, E–a, F–c, G–e, H–h.

9.4

(a) The four elements {A, C, D, E} are only connected to the three elements {b, d, e}.

(b) The three elements {B, D, G} are only connected to the two elements {c, f}.

9.5 Draw the seven kinds to the left, and the seven tins to the right. Draw an edge from kind i to tin j if there is at least one biscuit of kind i in tin j. We want the biscuits to represent one tin each. Now regard an arbitrary set of k tins. In them, there are in total $20k$ biscuits. Since there are only 20 biscuits of each kind, this means that the k tins will be connected to *at least* k kinds. Thus there is no set of tins that isn't connected to a set of kinds of at least the same size, and according to Hall's theorem this is a sufficient condition for a complete matching to exist. And the matching means that each kind can represent a tin.

9.6

(a) That there is a complete matching if $\delta(G) = 0$ follows from Hall's theorem.

(b) The meaning of $\delta(G)$ gives that there exists a subset A among the left nodes that is only connected to $|A| - \delta(G)$ right nodes. Then $\delta(G)$ out of these left nodes can't be matched to anything, and if $\delta(G)$ nodes in the graph are guaranteed to be impossible to match then it's definitely impossible for more than $|X| - \delta(G)$ ones to be matched.

(c) If we now expand the graph by adding $\delta(G)$ new nodes on the right side that are connected to all nodes on the left, the condition in Hall's theorem will be met, and then a complete matching exists. Before the addition of these nodes there were (at least) $\delta(G)$ unmatched nodes, so all the new nodes must have been put to use in the complete matching. If we remove them, their partners will be mateless, while the rest of the nodes on the left are unaffected. The remaining matching must then contain $|X| - \delta(G)$ nodes on the left.

9.8 Assume that all nodes on the left have a degree of at least two and that all nodes on the right have a degree of at most four. If we study a subset A of the left set of nodes, and the nodes on the right that are connected to it, we see that this relationship holds here as well. (That you reduce the number of nodes on the left can't affect the degrees of the remaining ones, but the degrees of the nodes on the right may decrease.) How many nodes on the right does A have to be connected to? You ought to get the minimal number if all the nodes on the left have the minimum degree 2 at the same time as all the nodes on the right have the maximum degree 4 (the more nodes on the left a node on the right is connected to, the lower the total number of nodes on the right). Each edge has two ends, so in this extreme case it holds that $2|A| = 4|P(A)| \Rightarrow |P(A)| = \frac{1}{2}|A|$. And this was the minimal possible $P(A)$; in general it holds that $P(A) \geq \frac{1}{2}|A|$.

9.9 Using the same line of reasoning as in the previous exercise we get

$$|P(A)| \geq \frac{x}{y}|A|$$

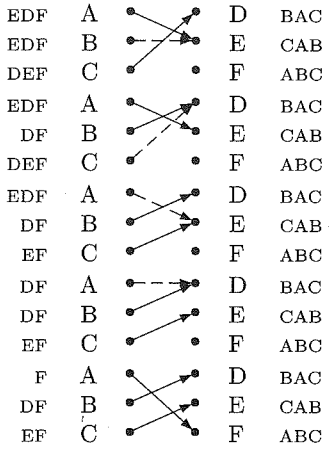
9.11 The really time consuming moment ought to be step 2: looking for augmenting paths. We check an easily analysed special case: We are looking for matchings in $K_{n,n}$, where we've already matched k nodes. Now we are to

1. Look for an unmatched node on the left (can be done in $n - k$ ways).
2. Find a matched node on the right, and go to this. (There are n nodes to check, out of which k are matched.)

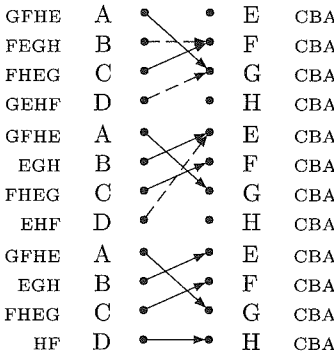
3. Go to the node on the left to which this one was matched. (Can be done in one way.)
4. Either find an unmatched node on the right that is also connected to this node on the left (there are n nodes to check, out of which $n-k-1$ are unmatched), or find a matched node on the right (there are $k-1$ ones to choose from) and restart at 3.

9.14

(a) To save space we give the persons who will be turned down a dashed arrow



(b) In this case, we can for a start realise that some lady will be dissatified by the outcome, since they can't all of them marry Cyrus (not according to the current marriage code, at least).



Observation: The most popular person has in principle free choice; the least popular has to take what's left over.

9.17

(a) If woman W turns down man M, this means that W has some other suitor S whom she prefers to M. Does there exist some other stable matching where W has been paired with M and S with som other female F whom he prefers to W? (If S doesn't prefer F to W, that matching would be unstable, since then S och W would break it up and take each other instead of M and F.)

The answer is no, since if S prefers F he would have proposed to her in an earlier stage of the algorithm, and thus has been turned down, but then we know from the inductive hypothesis that there isn't any stable matching that matches S with F.

(b) The man whom the woman takes is the best possible among the ones that came and proposed. What is the status of those who didn't come, because they got hold of someone else? The ones that didn't come and whom she likes less than her fiancé, can they be included in a stable matching? No, since she is the best one among those who don't turn the fiancé down. The ones who turn him down can according to the previous exercise not be paired with him in a stable matching, so he must because of this be matched to someone he likes less. But then we have an unstable pair, because these two prefer each other to their current partners. So nobody she likes less than this one can be considered, so he must be the worst alternative possible!

9.18 Run the algorithm until everyone of the underrepresented sex have been matched. If a woman is left over, that must be because nobody has proposed to her, which must be because all the men have been accepted by someone they like better. So no man is tempted to abandon his wife for the woman left over.

If a man is left over, he must have been turned down by all the women, who then must have a suitor they prefer to him, and because of this don't want to abandon for the man left over.

9.19 Let every woman's list consist of the men she can consider, in order of preference. Men not included on the list are turned down automatically. (If we want to be fair we can extend the privileges to the men who give up and stop proposing if they have been turned down by everone they found acceptable.)

9.20 We can take a simple example. We have a number of men and a number of women. Man A prefers woman C

and D in this order. Man B prefers woman D and C in this order. C prefers in her turn B and A, and D prefers A and B. In the first step of the algorithm A will propose to C and B to D, and C and D will never get any offers from those they want the most. C can't get it if D doesn't get a better offer first, and D won't get a better offer until C gets it. (Classical case of dead-lock.) But if C and D tinker with their lists and put the man they actually have as second choice last, they will (probably) be able to turn him down when he proposes for the first time, and thereby get the one they really want, when said person in his turn is turned down by the other woman.

9.21 We can introduce some kind of matriarcal polygamy. Each university ranks the students, and has a maximum number it accepts. If 150 students have a university with 100 places as first choice, the university turns down the 50 lowest ranking, who have to try somewhere else. Alternatively, we can let the universities propose to all the 100 students they most of all want, and then the students can turn them down if they get better offers from somewhere else. The first version favours the students, the latter the universities. (As a curiosity, we can mention that NIMP originally favoured the hospitals, but after complaints from student organisations, the algorithm in 1997 was turned around so that it favours the students.)

9.22

(a) We divide the problem into two parts: firstly we show that we actually get a matching, and then that this matching is stable.

If we don't get a matching, this means that some ladies end up as co-owners of the same man, while some other man is left over. Can this happen? Let's call the bigamous man m, the woman he was matched to in the first matching w_1 and the one he was matched to in the second matching w_2. Now, m must have ranked w_1 and w_2 in some way. If he had placed w_1 first, the second matching can't have been stable, because then m and w_1 would mutually have preferred each other to their partners. Same reasoning with the other possible ranking. So two ladies getting the same man can't happen.

Now it remains to show that the matching is stable. If it isn't, there exists a pair who prefer each other to their current partners. Since the woman now has been given her favourite in the two matchings, any unstabiliser has to have been outside the two matchings. He, on his part, must have already been coveting the lady before the regrouping. Can he in the regrouping have lost a lady whom he adores, so that this instability is news? No, if he has lost her he can't have had her in both the matchings, and in this case the one where he didn't have her would have been unstable.

(b) Matchings can according to the preferences of the women be ordered in a partial order, which can be called better-than. A matching is better-than another one if all the women are at least as happy as before. (If some are more happy and others less, the matchings aren't comparable.) The least upper bound of two matchings has to be a matching where every woman is at least as happy as in the two original matchings, and where all matchings that satisfies this as well are higher up. The matching where every woman takes her favourite in the two matchings satisfies this: it's better-than both of them, since everyone is at least as happy as before, and it's the lowest one satisfying this. (If you make someone less happy than this, what you get won't be comparable to the two matchings.) And since this combination is possible to calculate for every pair of matchings, the least upper bound is defined for all pairs of matchings. If we then in the same way makes a matching where every woman gets the one she likes least in the two matchings we also have a greatest lower bound. And if it's possible to finde $\wedge$ and $\vee$ for all pairs of elements in the partial order we have a lattice.

9.23

(a) No. If for instance all the persons involved have identical lists of preferences, it's unavoidable that someone gets their first choice and someone else their last choice.

(b) Badly. From the point of stability, totally satisfied pairs are to prefere, even if they are put together at the cost of other pairs, since totally satisfied pairs aren't interested

in switching and thereby can't cause any instabilities.

9.25

(a) A necessary conditions is for instance that the number of nodes is even.

(b) Make up a star graph, with a node in the middle and an odd number of beams. It has an even number of nodes, but it's impossible to match more than one pair (since all the outer nodes have the same potential partner).

9.26

(a) We look at a component in G'. If the component simply consists of one edge, it's undeniably a (degenerated) alternating path. (The edge has to come from one of the matchings.) If the component consists of several edges, we can start in one of them. In one of the ends (perhaps both) an edge is attached. This edge can't come from the same matching as the first edge, since matchings consist of several separate edges, so it has to come from the other matching. No more than one edge can be attached to the end, since the other matching doesn't have connected edges either. If we then go to the other end of this new edge, we can repeat the line of reasoning there: either nothing is attached, or exactly one edge from the first matching. And if we follow these edges, we have an alternating path (perhaps an alternating cycle).

(b) If an augmenting path exists, then the matching clearly isn't maximum: it can be augmented using the path.

Now assume that no augmenting path exists, and show that the matching has to be maximum. Alternatively, show that if the matching isn't maximum, then an augmenting path exists.

Study a non-maximum matching M. That it isn't maximum means that there exists another matching M' which is larger. Now make up a graph G' as above. It will consist of alternating paths. Since there are more edges in M', there has to exist a path which consists of more M'-edges than M-edges. (Possibly this path consists of just one edge, from M'.) This path has to start and end with edges from M', and is from the point of view of M an augmenting path.

9.28 If you succed in pairing two who are each others first choices, they at least won't cause any instabilities. Unfortunately no such preferences exist, but at least we can pair some of them with their second choices. If we pair A–E and C–F, E and C are totally satisfied. F would rather have A, but A doesn't want F but B. But B prefers the remaining D. D thus has no say in the matter.

9.29 Swap Adiles two first preferences around, so that she prefers Carola to Biljana. Then Adile and Carola form a pair that isn't interersted in changing with someone else, and then Biljana and Doris has no choice except for putting up with each other, in spite of being mutual last choices.

9.30

(a) The bipartite graph has to have the set of nodes $X = \{$ Kimmo, Viggo, Lasse, Ambjörn, Olle$\}$ and $Y = \{\text{serving } 1, \ldots, 5\}$. Edges are drawn from each person to the servings he is prepared to eat. We note that with $A = \{$Kimmo, Viggo, Lasse, Ambjörn$\}$ we get $|A| = 4$ while $|P(A)| = 3$. Thus we have the deficiency $\delta(A) = 1$ so a maximum matching has at most 4 edges. One of these is easily found.

(b) The graph contains $K_{3,3}$ as a subgraph and is thus nonplanar according to Kuratowski's theorem.

9.31 The theorem holds for a graph with one node on the left, that's easily checked. If it's connected to any node on the right there is a matching, otherwise not. (This was the base case.)

Now assume that the theorem is true for all graphs with up to n nodes on the left. Study a graph with $n + 1$ nodes on the left that satisfies the conditions of the theorem, that is, each subset of the nodes on the left is connected to an at least as large set of nodes to the right.

There are $n + 1$ subsets with n nodes of this set. All of these must be possible to match, according to the inductive hypothesis. We now want to show that there has to exist two matchings that together are connected to (at least) $n + 1$ nodes on the right. Assume the opposite, which is that all these $n+1$ matchings are to the same set of n nodes on the right. But the graph satisfies according

to the premisses that the $n+1$ nodes on the left are connected to (at least) $n+1$ nodes on the right. That means that one of the $n+1$ nodes on the left besides is connected to some node on the right outside the set of n, and that means that we can redesign one of the matchings where this node is involved by connecting it to the node outside instead. That redesigned matching combined with one of the original ones does in total involve $n+1$ nodes on the right.

Now study two matchings, M_1 and M_2, that combined include $n+1$ nodes on the right. We are going to, based on these, make up a matching that includes all the nodes on the left.

There must exist a node on the right that's only included in M_1. We pick this edge. Either it goes to a node on the left only included in M_1, and then M_2 combined with this edge will be a complete matching. Or it will go to a node on the left that is included in M_2 as well. In that case there is a M_2-edge leaving it, to some other node on the right. Either that node belongs to M_1 as well, and then we can repeat the argument. Or this right node only belongs to M_2, and then we can't continue the path. In that case we have a path consisting of k nodes on the left and $k+1$ nodes on the right, and there we can match the k nodes on the left, either according to M_1 or to M_2. Then we are left with the remaining $n+1-k$ nodes on the left and the right. The graph satisfies the condition in Hall's theorem, and according to the inductive hypothesis that means that we can find a complete matching of this subset of the nodes. These two matchings combined make up a complete matching of the graph.

9.33

(a) A complete matching in a cycle demands that there is an even number of nodes in the cycle. If there is, an arbitrary node can either be paired with the node in front or the one behind. When that choice is made, the remaining choices make themselves, so the total number of possibilities is two.

(b) For the path as well, a first requirement is that n is even. It that is the case, the first node has to be paired with the second one, the third with the fourth, and so on. There is thus only one way.

(c) This problem seems designed to be solved recursively. If $n = 1$ there is one way: match the two nodes to each other. If $n = 2$, there are two ways: match horisontally or vertically. For the rest, we can partition the matchings into two classes: those that end with a vertical matching and those that don't (but instead end with two horisontal pairs). The first kind we can make by adding a vertical matching to a matching of $n-1$; the latter by adding two horisontal ones to a matching of $n-2$. We get the recurrence equation

$$\begin{cases} a_1 = 1 \\ a_2 = 2 \\ a_n = a_{n-1} + a_{n-2}, & n > 2 \end{cases}$$

This is a shifted Fibonacci equation; $a_n = F_{n+1}$. (One out of many situations where the Fibonacci numbers unexpectedly appear!)

9.34 The maximum matching can at most match five star boys, since there are only five Lucias. Since every star boy fancies exactly two Lucias, in the worst case all of them fancies the same two, and in this case the maximum matching is of size two. But here we assume that all the star boys are so charming that the Lucias agree to the matching. Probably, the Lucias are choosy as well, and then in the worst case the maximum matching may even be empty!

9.36 The matching is finished immediately, which is caused by all on the proposing side having different preferences.

9.37

(a) Just shift the order one step for each row, for instance

```
ABCDE
BCDEA
CDEAB
DEABC
EABCD
```

(b) If an $n \times n$-square is given, together with a list of n approved letters for each place in the square, then it's always possible to choose an approved letter for each place so that no letter is found twice in any row or column.

(c) For $n = 1$, there is only one box and one approved letter. If we put the letter in the box, the conditions are satisfied.

(d) For $n = 2$, we can start by picking a letter for box (1,1), say A. Then

there are at least one alternative for both (2,1) and (1,2), say B and C, respectively. If (2,2) has some letter on its list other than B and C we are done. Otherwise, if there were an alternative to B for (2,1) or to C for (1,2), we take that one instead and then B or C is free to use for (2,2). Finally, if there aren't any alternatives for (1,2) or (2,1), both their lists have to include A. Then we choose the other letter for (1,1), say B, and let (2,1), which previously had B, get A. Then (2,2) can have B so we are done.

(e) Let the nodes represent the boxes and draw edges between pairs of boxes that are in the same row or column.

(f) Take an arbitrary latin $n \times n$-square made out of the numbers $0, 1, \ldots, n-1$, and then exchange every number j with the pair $j, n-1-j$. One example of size 4×4:

0,3	1,2	2,1	3,0
3,0	0,3	1,2	2,1
2,1	3,0	0,3	1,2
1,2	2,1	3,0	0,3

Let the first and second number in each box represent the number of outgoing edges to other boxes in the same row and column, respectively. Edges are drawn to all boxes in the same row with a lower first number, and to all boxes in the same row with a lower second number. Then clearly the number of outgoing edges will be $n-1$ for each box.

(g) That a set of nodes in the box graph is independent means that it doesn't include several boxes from any row or column. We can thus see an independent set of boxes as a matching between corresponding rows and columns. Let the direction of the edges represent the order of preference, so that row i prefers column j to column k if there is an edge directed from box (i,k) to box (i,j), and correspondingly for the columns order of preference of the rows. The theorem about stable marriages say that we can always find a stable maching in each bipartite matching situation. In our situation, this means that we for every subset S of boxes can find an independent set S' of boxes such that all other boxes have an edge to some box in S'. In other words, the directed box-graph is stable! We also know that every box has more colours on its list (namely n) than outgoing edges (namely $n-1$). Thereby the graph is list-colourable according to exercise 7.44.

Chapter 11

11.3

(a) It's always possible in the first step to move card $n+1$ so that it bonds to card n and then you can regard the situation as if you have n cards.

(b) Since $V(1) = 0$ it follows by elementary induction that $V(n) \leq n-1$.

(c) You can at most achieve three new bonds per block move, since a block move only changes three adjacent pairs (where the block starts, where the block ends, and where the block is placed).

(d) In the worst case you have to make up $n-1$ bonds before being done.

(e) Why can't you get rid of three descents in one step? Assume that it's possible. If we had three descents the situations looks like this:

$$[\ldots\ a\ b\ \ldots\ c\ d\ \ldots\ e\ f \ldots],$$

where $a > b$, $c > d$, and $e > f$. We can get rid of all these descents by placing the block $[d\ \ldots\ e]$ between a and b:

$$[\ldots\ a\ d\ \ldots\ e\ b\ \ldots\ c\ f \ldots],$$

but if we aren't to get any new descents instead we have to have $a < d$, $e < b$, and $c < f$. Combined with the three inequalities above this gives $a < d < c < f < e < b < a$, but $a < a$ is impossible. So it isn't possible to get rid of three descents in one go!

(f) If every move removes at most two descents and there are $n-1$ ones at start, at least $\lceil \frac{n-1}{2} \rceil$ moves are needed.

(g) Inspection.

(h) If two of the moves remove just one descent, the number of moves is at least $\lceil \frac{n-1-2}{2} \rceil + 2 = \lceil \frac{n-1}{2} \rceil + 1$ for $n \geq 3$.