such colorings on A, we obtain

$$p(E_A) = \frac{2}{2^d} = \frac{1}{2^{d-1}}\,.$$

Thus with $|\mathcal{F}| \le 2^{d-1}$ (where the events are not disjoint),

$$p\Big(\bigcup_{A\in\mathcal{F}} E_A\Big) < \sum_{A\in\mathcal{F}} p(E_A) = m\frac{1}{2^{d-1}} \le 1\,.$$

Now $\bigcup_{A\in\mathcal{F}} E_A$ is the event that *some* set in $\mathcal{F}$ is of a single color, and we conclude from $p(\bigcup_{A\in\mathcal{F}} E_A) < 1$ that there *must* exist a two-coloring of S without single-colored sets, which is just what we wanted to show.

An upper bound for $m(d)$ of order of magnitude $d^2 2^d$ is known, where this time random sets and a fixed coloring are used. As for exact values, only the first two are known: $m(2) = 3$ and $m(3) = 7$.

Exercises for Chapter 1

1.1 Suppose that Dean B of County College determines that every student must enroll in exactly four courses in the history of mathematics from among the seven that are offered. The professors of the various courses specify the maximum enrollments in their courses as $51, 30, 30, 20, 25, 12, 18$. What can one conclude from this?

1.2 Suppose we are given n disjoint sets S_i. Let the first have a_1 elements, the second a_2, and so on. Show that the number of sets that contain at most one element from each S_i is equal to $(a_1{+}1)(a_2{+}1)\cdots(a_n{+}1)$. Apply this result to the following number-theoretic problem: Let $n = p_1^{a_1}p_2^{a_2}\cdots$ be the prime decomposition of n. Then n has exactly $t(n) = \prod(a_i + 1)$ divisors. Conclude that n is a square if and only if $t(n)$ is odd.

▷ **1.3** Let $N = \{1, 2, \ldots, 100\}$ and let A be a subset of N with $|A| = 55$. Show that A contains two numbers a and b such that $a - b = 9$. Does this hold as well for $|A| = 54$?

1.4 Number the twelve edges of a cube with the numbers 1 through 12 in such a way that the sum of the three edges meeting at each vertex is the same for each vertex.

▷ **1.5** In the parliament of country X there are 151 seats and three political parties. How many ways (i, j, k) are there of dividing up the seats such that no party has an absolute majority?

1.6 How many different words can be made from permutations of the letters in *ABRACADABRA*?

1.7 Show that $1! + 2! + \cdots + n!$ for $n > 3$ is never a square.

1.8 Show that for the binomial coefficients $\binom{n}{k}$ the following holds:

$$\binom{n}{0} < \binom{n}{1} < \cdots < \binom{n}{\lfloor n/2\rfloor} = \binom{n}{\lceil n/2\rceil} > \cdots > \binom{n}{n}\,,$$

where for even n the two middle coefficients coincide.

▷ **1.9** Prove an analogous result for the Stirling numbers of the second kind. For each $n \geq 1$ there exists $M(n)$ such that $S_{n,0} < S_{n,1} < \cdots < S_{n,M(n)} > S_{n,M(n)+1} > \cdots > S_{n,n}$ or $S_{n,0} < S_{n,1} < \cdots < S_{n,M(n)-1} = S_{n,M(n)} > \cdots > S_{n,n}$, where $M(n) = M(n-1)$ or $M(n) = M(n-1)+1$. The same result holds also for $s_{n,k}$. Hint: Use the recurrence relation for $S_{n,k}$ and finish the proof by induction.

▷ **1.10** Show that every natural number n possesses a unique representation $n = \sum_{k\geq 0} a_k k!$ with $0 \leq a_k \leq k$.

1.11 Derive the following recurrence relation for partition numbers $P_{n,k}$: $P_{n,1} = P_{n,n} = 1$ and $P_{n,k} = P_{n-k,1} + P_{n-k,2} + \cdots + P_{n-k,k}$.

1.12 The *Bell number* $\tilde{B}_n$ is the number of *all* set partitions of an n-set; that is, $\tilde{B}_n = \sum_{k=0}^{n} S_{n,k}$ with $\tilde{B}_0 = 1$. Show that

$$\tilde{B}_{n+1} = \sum_{k=0}^{n} \binom{n}{k} \tilde{B}_k \,.$$

▷ **1.13** Let $f_{n,k}$ be the number of k-subsets of $\{1, \ldots, n\}$ that contain no pairs of adjacent numbers. Show that

$$\text{(a)} \quad f_{n,k} = \binom{n-k+1}{k} \qquad \text{(b)} \quad \sum_{k} f_{n,k} = F_{n+2}\,,$$

where F_n is the nth Fibonacci number (that is, $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ $(n \geq 2)$).

1.14 Show that the sum of the binomial coefficients in Pascal's triangle along a diagonal from upper right to lower left is always the Fibonacci number F_{n+k+1}. Example: Starting with $n = 4$, $k = 3$ gives $4 + 10 + 6 + 1 = 21 = F_8$.

1.15 Prove that $\binom{n}{r}\binom{r}{k} = \binom{n}{k}\binom{n-k}{r-k}$ and conclude that $\sum_{k=0}^{m} \binom{n}{k}\binom{n-k}{m-k} = 2^m \binom{n}{m}$.

▷ **1.16** An ordinary deck of 52 playing cards is well shuffled. What is the probability that both the top and bottom cards are queens (on the assumption that all 52! permutations are equally probable)?

1.17 In a lottery, six numbers from the set $\{1, 2, \ldots, 49\}$ are selected. What is the probability that the chosen set of numbers contains two adjacent integers?

1.18 Suppose the random variable X can assume only the values 0 and 1. Show that $VX = EX \cdot E(1-X)$.

▷ **1.19** Prove that in every set of $n+1$ integers from the collection $\{1, 2, \ldots, 2n\}$ there is always a pair of integers that are relatively prime, and always a pair in which one of the pair is a divisor of the other. Does the result hold for n integers?

1.20 Construct a sequence of n^2 distinct numbers that contains neither a monotonically increasing subsequence nor a monotonically decreasing sequence of length $n+1$.

▷ **1.21** The Euler φ-function is defined by

$$\varphi(n) = |\{k : 1 \leq k \leq n,\ k \text{ relatively prime to } n\}|.$$

Prove that

$$\sum_{d|n} \varphi(d) = n\,.$$

1.22 Each square of a 4×7 checkerboard is colored either white or black. Show that there is always a rectangle whose four corner squares have the same color. Does the result hold for a 4×6 checkerboard?

▷ **1.23** Let $N = \{1, 2, 3, \ldots, 3n\}$. Matilda removes n elements of N. Show that she can now remove a further n numbers in such a way that the remaining n appear in the order odd, even, odd, even, ...

1.24 Matilda selects n points on the circumference of a circle and colors each point red or blue. Show that there are at most $\left\lfloor \frac{3n-4}{2} \right\rfloor$ chords connecting points of different colors that do not intersect in the interior of the circle.

▷ **1.25** Consider an $n \times n$ checkerboard with rows and columns numbered from 1 to n. A set T of n squares will be called a *transversal* if none of its squares are in the same row or column. In other words, $T = \{(1, \pi_1), \ldots, (n, \pi_n)\}$, where $(\pi_1, \ldots, \pi_n)$ is a permutation of $\{1, \ldots, n\}$. Now let $n \geq 4$ be an even number and place a number in each of the n^2 squares in such a way that every number appears exactly twice. Show that there is always a transversal containing n distinct numbers. Hint: The checkerboard contains r pairs of squares (in different rows and columns) that contain the same number. Construct the $(n!) \times r$ incidence matrix (m_{ij}) with $m_{ij} = 1$ if the ith transversal contains the jth pair. Now count in two different ways.

1.26 We would like to consider how we might effectively list all $n!$ permutations of $\{1, \ldots, n\}$. The most usual method is the lexicographic ordering. We say that $\pi = (\pi_1, \ldots, \pi_n)$ is lexicographically smaller than $\sigma = (\sigma_1, \ldots, \sigma_n)$ if for the smallest i with $\pi_i \neq \sigma_i$, we have $\pi_i < \sigma_i$. For example, for $n = 3$ we obtain the ordering $123, 132, 213, 231, 312, 321$. Show that the following algorithm finds the successor permutation σ to the permutation $\pi = (\pi_1, \ldots, \pi_n)$:

1. Search for the largest index r with $\pi_r < \pi_{r+1}$. If no such r exists, then $\pi = (n, \ldots, 2, 1)$ is the last permutation.
2. Search for the index $s > r$ with $\pi_s > \pi_r > \pi_{s+1}$.
3. $\sigma = (\pi_1, \ldots \pi_{r-1}, \pi_s, \pi_n, \ldots, \pi_{s+1}, \pi_r, \pi_{s-1}, \ldots, \pi_{r+1})$ is the successor permutation.

1.27 Analogously to the previous exercise, we would like to list all 2^n subsets of an n-set. As usual, we represent the subsets by $(0,1)$-words of length n. The following list is called a Gray code. Suppose $G(n) = \{G_1, \ldots, G_{2^n}\}$ is the list for n. Then $G(n+1) = \{0G_1, 0G_2, \ldots, 0G_{2^n}, 1G_{2^n}, 1G_{2^n-1}, \ldots, 1G_1\}$. Prove the following: (a) Every pair of neighboring $(0,1)$-words in $G(n)$ differ in exactly one place. (b) Let $G(n,k)$ be the subsequence of $G(n)$ with exactly k 1's. Show that consecutive words in $G(n,k)$ differ in exactly two places.

1.28 There are $4n$ participants in a bridge tournament playing at n tables. Each player requires one other player as partner, and every pair of partners requires

another pair of partners as their opponents. In how many ways can the selection of partners and opponents occur?

▷ **1.29** In how many ways can the numbers $1, \dots, n$ be arranged in a row so that aside from the first element, each number k has $k-1$ or $k+1$ as one of its predecessors (not necessarily the immediate one)? Examples: 3 2 4 5 1 6 and 4 3 5 2 1 6.

1.30 In how many ways can the numbers $1, \dots, n$ be arranged in a circle so that adjacent numbers always differ by 1 or 2?

1.31 For a permutation $a_1 a_2 \dots a_n$ of $\{1, \dots, n\}$, an *inversion* is a pair a_i, a_j with $i < j$ but $a_i > a_j$. Example: 1 4 3 5 2 has the inversions $4,3; 4,2; 3,2; 5,2$. Let $I_{n,k}$ be the number of n permutations with exactly k inversions. Prove the following:

a. $I_{n,0} = 1$.
b. $I_{n,k} = I_{n,\binom{n}{2}-k}$ $(k = 0, \dots, \binom{n}{2})$.
c. $I_{n,k} = I_{n-1,k} + I_{n,k-1}$ for $k < n$. Is this true as well for $k = n$?
d. $\sum_{k=0}^{\binom{n}{2}} (-1)^k I_{n,k} = 0$, $n \geq 2$.

▷ **1.32** Let $a_1, a_2, \dots, a_n$ be a permutation of $\{1, \dots, n\}$. We let b_j denote the number of elements to the left of j that are greater than j (and thus form an inversion with j). The sequence $b_1, \dots, b_n$ is called the *inversion table* of $a_1, \dots, a_n$. Show that $0 \leq b_j \leq n - j$ $(j = 1, \dots, n)$, and prove that conversely, every sequence $b_1, \dots, b_n$ with $0 \leq b_j \leq n - j$ $(\forall j)$ is the inversion table for exactly one permutation.

1.33 We return to the lexicographic ordering of permutations introduced in Exercise 26. We assign the number 0 to the smallest permutation $(1, 2, \dots, n)$. We assign 1 to the next permutation, and so on, giving the last permutation $(n, n-1, \dots, 1)$ the number $n! - 1$. The problem is to determine the number $\ell_n(\pi)$ assigned to a given permutation $\pi = (\pi_1, \dots, \pi_n)$. Show that $\ell_1(1) = 0$, $\ell_n(\pi) = (\pi_1 - 1)(n-1)! + \ell_{n-1}(\pi')$, where $\pi' = (\pi'_1, \dots, \pi'_{n-1})$ is derived from π by deleting π_1 and reducing all $\pi_j > \pi_1$ by 1. Example: $\ell_4(2314) = 3! + \ell_3(213) = 3! + 2! + \ell_2(12) = 8$.

1.34 The converse of the previous exercise. Let ℓ, $0 \leq \ell \leq n! - 1$ be given. Determine the permutation π for which $\ell_n(\pi) = \ell$. Hint: From Exercise 10, we can represent ℓ in the form $\ell = a_{n-1}(n-1)! + a_{n-2}(n-2)! + \dots + a_1 1!$ with $0 \leq a_k \leq k$.

1.35 Prove the following recurrence formulas for the Stirling numbers:

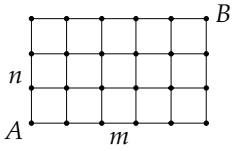
(a) $s_{n+1,k+1} = \sum_i \binom{i}{k} s_{n,i}$, (b) $S_{n+1,k+1} = \sum_i \binom{n}{i} S_{i,k}$.

▷ **1.36** The *Euler numbers* $A_{n,k}$ count the permutations π of $\{1, \dots, n\}$ with precisely k increases, that is, k places i for which $\pi_i < \pi_{i+1}$. For example, for $n = 3$, we have $A_{3,0} = 1$, $A_{3,1} = 4$, $A_{3,2} = 1$. Prove the recurrence

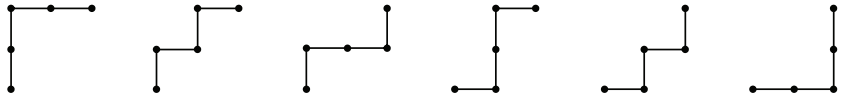
$$A_{n,k} = (n-k)A_{n-1,k-1} + (k+1)A_{n-1,k}$$

$(n > 0)$ with $A_{0,0} = 1, A_{0,k} = 0$ $(k > 0)$.

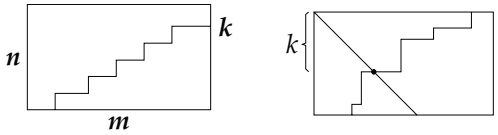
1.37 Many identities for binomial coefficients can be obtained by counting paths in lattices. Consider an $m \times n$ lattice such as the one shown in the figure for $m = 5$, $n = 3$. (Note that m and n are the numbers of edges.)



Show that the number of different paths from A to B that always move upward or to the right is equal to $\binom{m+n}{n}$. For example, for $m = 2$, $n = 2$ we have the $\binom{4}{2} = 6$ paths



1.38 Two examples of the lattice method. (a) Classify the lattice paths according to a path's first meeting the right vertical, and derive equation (1.9) (see the figure on the left). (b) Prove the Vandermonde identity (1.13) by classifying the lattice paths according to their point of intersection with the diagonal shown in the right-hand figure.



▷ **1.39** In how many ways can a king move from the lower left corner of a chess board to the upper right corner if it always moves up, to the right, or diagonally up and to the right? Hint: Let r equal the number of diagonal moves and sum over r.

1.40 Show that $r^{\underline{k}}(r - \frac{1}{2})^{\underline{k}} = \frac{(2r)^{\underline{2k}}}{2^{2k}}$ and conclude the validity of the formula $\binom{-\frac{1}{2}}{n} = (-\frac{1}{4})^n \binom{2n}{n}$.

▷ **1.41** The following problem goes back to J. L. F. Bertrand (1822–1900). Two candidates A and B receive a and b votes in an election, with $a > b$. In how many ways can the ballots be arranged so that in counting, one vote after the other, A always has more votes than B. For example, for $a = 4$, $b = 2$, we have the following possibilities: $AAAABB$, $AAABAB$, $AAABBA$, $AABAAB$, $AABABA$. Show that the answer is $\frac{a-b}{a+b}\binom{a+b}{a}$. Hint: Draw a sequence as points (x, y), where y is the number of votes for A minus the number of votes for B when x votes have been counted. The desired sequences are then the paths from $(0, 0)$ to $(a+b, a-b)$ that after $(0, 0)$ never again touch the x-axis.

1.42 Show that $(1+\sqrt{3})^{2n+1}+(1-\sqrt{3})^{2n+1}$ represents a natural number for every $n \geq 0$. Hint: Use the binomial theorem. Since $0 < \left|1 - \sqrt{3}\right| < 1$, it must be that $-(1 - \sqrt{3})^{2n+1}$ is the fractional part of $(1 + \sqrt{3})^{2n+1}$. Conclude that the integer part of $(1 + \sqrt{3})^{2n+1}$ always contains 2^{n+1} as a factor.

▷ **1.43** Let $a_n = \frac{1}{\binom{n}{0}} + \frac{1}{\binom{n}{1}} + \cdots + \frac{1}{\binom{n}{n}}$. Show that $a_n = \frac{n+1}{2n} a_{n-1} + 1$ and determine $\lim_{n\to\infty} a_n$ (if the limit exists). Hint: Show that $a_n > 2 + \frac{2}{n}$ and $a_{n+1} < a_n$ for $n \geq 4$.

1.44 It is clear that n points on a line divide the line into $n+1$ parts.

a. Let L_n denote the maximum number of pieces into which the plane can be divided by n lines. Determine a recurrence for L_n and calculate L_n.
b. Let M_n denote the maximum number of three-dimensional pieces into which 3-space $\mathbb{R}^3$ can be divided by n planes. Determine a recurrence for M_n and calculate M_n.
c. Generalize to $\mathbb{R}^n$.

▷ **1.45** Pascal's triangle (shifted somewhat) yields an astonishing primality test. We number the rows as usual with $0, 1, 2, \ldots, n, \ldots$, and likewise the columns. In the nth row we write the $n+1$ binomial coefficients $\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}$, but shifted into the columns with numbers $2n$ through $3n$ inclusive. Finally, we draw a circle around any of these $n+1$ numbers that are multiples of n. The first rows and columns look like this:

n \ k	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1			(1)	(1)									
2					1	(2)	1						
3							1	(3)	(3)	1			
4									1	(4)	6	(4)	1

Show that a number k is prime if and only if all the elements in the kth column are circled. Hint: For k even the problem is easy, and for k odd, first prove that the element in the nth row and kth column is $\binom{n}{k-2n}$.

▷ **1.46** Two dice have the same probability distribution. Show that the probability of throwing the two dice and obtaining the same number on each is always at least $\frac{1}{6}$.

1.47 The following important inequalities estimate the distance from X to the expectation EX. Prove Markov's inequality: Let X be a random variable that takes on only nonnegative values. Then $p(X \geq \alpha) \leq \frac{EX}{\alpha}$ for $\alpha \geq 0$. Conclude from this Chebyshev's inequality: $p(|X - EX| \geq \alpha) \leq \frac{VX}{\alpha^2}$ for a random variable X and $\alpha \geq 0$.

1.48 Estimate, with the help of the previous exercise, the probability that a permutation has $k+1$ fixed points (all permutations of equal probability).

▷ **1.49** A group of n hunters shoot simultaneously at r rabbits, with every shot hitting a rabbit. Every hunter hits each rabbit with equal probability, and if a rabbit is hit, it is killed. What is the expectation for the number of surviving rabbits? Show with the help of Markov's inequality that for $n \geq r(\log r + 5)$, the probability that no rabbit survives is greater than 0.99.

1.50 A random number generator chooses one of the numbers $1, 2, \ldots 9$, all with equal probability. Determine the probability that after n selections ($n > 1$), the

product of these numbers is divisible by 10. Hint: Consider the random variables X_k, the number of times k is chosen, $1 \le k \le 9$.

▷ **1.51** A deck of playing cards with n cards contains three aces. The deck is shuffled, with all permutations equally probable. The cards are then dealt out one after the other until two aces have appeared. Show that the expectation for the number of dealt cards is $\frac{n+1}{2}$.

1.52 Let $(p_1, \ldots, p_6)$ and $(q_1, \ldots, q_6)$ be the probability distributions for two dice. Show that p_i, q_j can never be chosen such that the different possible sums $2, 3, \ldots, 12$ of the throws are equally probable (that is, equal to $\frac{1}{11}$).

▷ **1.53** Let x be a real number. Then among the numbers $x, 2x, 3x, \ldots, (n-1)x$ there is at least one that differs from an integer by at most $\frac{1}{n}$.

1.54 Prove the following general theorem of Ramsey: Let k and $\ell_1, \ldots, \ell_r$ be given. Then there is a least number $R(k; \ell_1, \ldots, \ell_r)$ such that the following holds: If N is an n-set with $n \ge R(k; \ell_1, \ldots, \ell_r)$ and if the k-subsets of N are colored in some way with the colors $1, \ldots, r$, then there is a color i such that in some ℓ_i-subset of N, all k-subsets are colored with i. Hint: Induction.

▷ **1.55** Show that if the Ramsey numbers $R(k-1, \ell)$ and $R(k, \ell-1)$ are both even, then $R(k, \ell) < R(k-1, \ell) + R(k, \ell-1)$. Calculate $R(3, 4)$.

1.56 For the two-coloring problem of families of sets discussed in Section 1.6, prove that $m(2) = 3$, $m(3) = 7$.

▷ **1.57** Prove that $R(k, k) \ge 2^{k/2}$. Hint: $R(2, 2) = 2$, $R(3, 3) = 6$. So assume that $k \ge 4$. Let $n < 2^{k/2}$. Altogether, there are $2^{\binom{n}{2}}$ patterns of acquaintance. Let A denote the event, $|A| = k$, that all the persons are mutually acquainted. Now use the probabilistic method of Section 1.6.

1.58 In country Z, every pair of cities is linked by exactly one of the three modes of transportation bus, train, air, where all three possibilities occur. No city is served by all three forms of transportation, and no three cities are linked by the same mode of transport. Determine the maximum number of cities in country Z. Hint: Consider the possible modes of transport from a fixed city.

▷ **1.59** Suppose that n different numbers (n is very large) are written on n scraps of paper and then the scraps are mixed together in a hat. We now pull one scrap of paper after another out of the hat. Our task is to determine the largest number. When the scrap on which we believe the largest number to be written has been drawn, we are to announce, "That is the largest number." It is not allowed to name a number that was previously drawn. Since we know nothing about the size of the numbers or their arrangement, the task seems hopeless. Nevertheless, there is an algorithm that names the correct number with probability greater than $\frac{1}{3}$. Hint: Let s numbers be drawn, and then choose the first number to be drawn that is larger than all the previous numbers drawn.

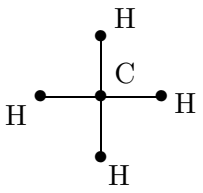
1.60 Let $a_1, a_2, a_3, \ldots$ be an infinite sequence of natural numbers. Then there exists either an infinite strictly monotonically increasing subsequence $a_{i_1} < a_{i_2} < a_{i_3} < \cdots$ (where $i_1 < i_2 < i_3 < \cdots$) or a strictly monotonically decreasing subsequence or an infinite constant subsequence $a_{j_1} = a_{j_2} = a_{j_3} = \cdots$.

Chapter 7

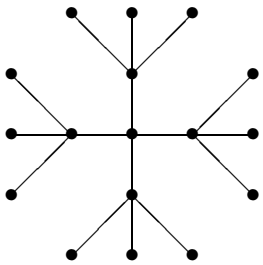
Trees

7.1. What Is a Tree?

The theory of trees originated in the study of hydrocarbons and isomers.[1]



methane



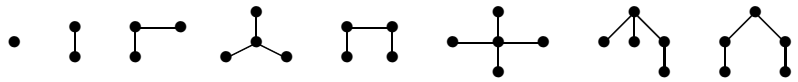
isobutane

Toward the end of the nineteenth century, Cayley posed the question of determining the number of possible isomers of a given compound. This led to the theory of graph enumeration (see Chapter 4 and the suggestions for further reading at the end of Part 1). Trees are the fundamental building blocks for the construction of graphs. However, they are of interest not only in graph theory, for they also provide the optimal data structure for many discrete problems, in particular for searching and sorting problems, to which we shall return in Chapter 9.

Definition 7.1. A graph is called a **tree** if it is connected and contains no circuits. A graph all of whose components are trees is called a **forest**.

[1]Two chemical compounds with the identical composition but different arrangements of their atoms are called *isomers*.

The trees with five or fewer vertices are depicted below:



Let $G = (V, E)$ be a connected graph. A subgraph T that is a tree of order $n = |V|$ (i.e., contains all the vertices of G) is called a **spanning tree**. Clearly, every connected graph G contains at least one spanning tree. Either G is already a tree, or else G possesses a circuit C. If we remove an edge e from C, then $G_1 := (V, E \setminus \{e\})$ is again connected. Either G_1 is a spanning tree or else G_1 possesses a circuit C_1. We remove an edge e_1 from C_1, and so on. After a finite number of steps we obtain a spanning tree.

Theorem 7.2. *The following are equivalent:*

a. $G = (V, E)$ *is a tree.*
b. *Every pair of vertices in* G *are joined by exactly one path.*
c. G *is connected, and* $|E| = |V| - 1$.

Proof. a $\Rightarrow$ b: If u and v were joined by two paths, these paths would yield a circuit.

b $\Rightarrow$ a: If C is a circuit, then two vertices of C are connected by more than one path.

a $\Rightarrow$ c: A tree contains one or more vertices of degree 1. Namely, let $P = u, u_1, u_2, \ldots, v$ be a longest path in G. Then all neighbors of u are in P, that is, $d(u) = 1$ (and also $d(v) = 1$), since G has no circuits. We now remove u and the incident edge uu_1, thereby obtaining a tree $G_1 = (V_1, E_1)$ on $n - 1$ vertices with $|V_1| - |E_1| = |V| - |E|$. After $n - 2$ steps we obtain a tree G_{n-2} on two vertices. That is, $G_{n-2} = K_2$, and we have $|V| - |E| = |V_{n-2}| - |E_{n-2}| = 1$.

c $\Rightarrow$ a: Let T be a spanning tree of G. From what we have just established,

$$1 = |V(G)| - |E(G)| \leq |V(T)| - |E(T)| = 1\,,$$

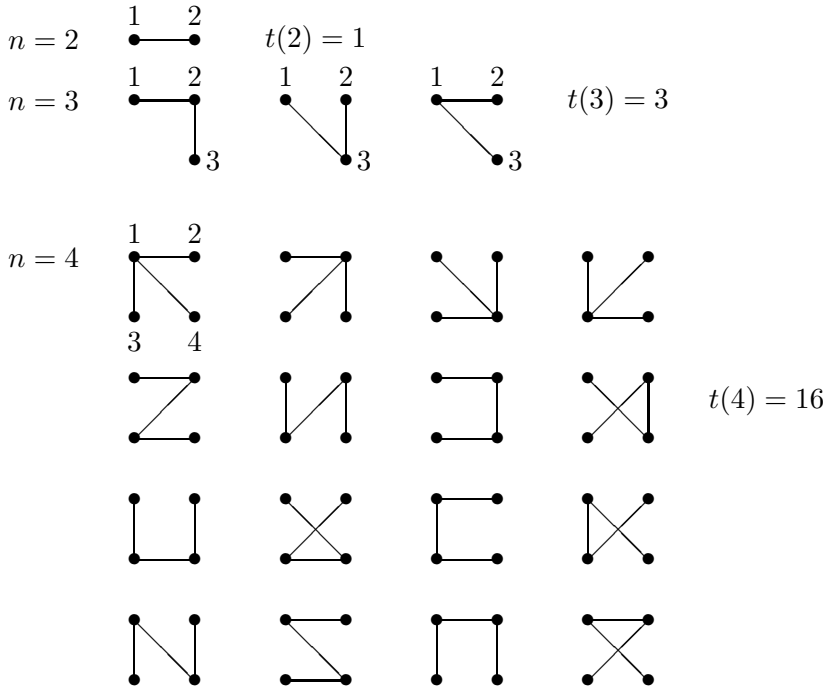
and so $E(G) = E(T)$, that is, $G = T$. □

If a graph $G = (V, E)$ consists of t components, application of Theorem 7.2 to the individual components yields the result that every spanning forest possesses $|V| - t$ edges. There are some additional characterizations of trees. For example, G is a tree if and only if G is connected and every edge is a cut edge (proof?). Additionally, from Theorem 6.1 we immediately obtain the following corollary:

Corollary 7.3. *If T is a tree of order $n \geq 2$, and $(d_1, d_2, \ldots, d_n)$ is the degree sequence, then*

$$\sum_{i=1}^{n} d_i = 2n - 2\,.$$

How many spanning trees does a graph G possess? In general, this is a difficult problem. For complete graphs, however, we can easily provide an answer. Let K_n denote the complete graph on $\{1, 2, \ldots, n\}$. Let $t(n)$ denote the number of spanning trees. Let us examine the situation for small values of n:



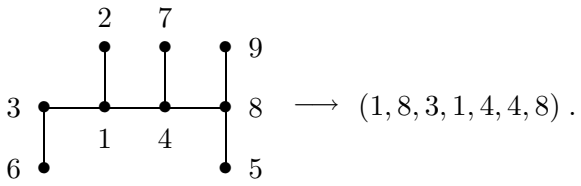
For $t(5)$ one may calculate $t(5) = 125$. Observe that the trees are not all distinct (in the sense of isomorphism). The following formula, suggested by the first few values, is one of the most astounding results in combinatorics.

Theorem 7.4. *The number of spanning trees in the complete graph K_n is given by $t(n) = n^{n-2}$.*

Proof. The expression n^{n-2} suggests that we use the rule of equality. Let $V = \{1, \ldots, n\}$ be the set of vertices. We begin by constructing a bijection from the set of all trees to the set of all sequences $(a_1, \ldots, a_{n-2})$ with $1 \leq a_i \leq n$, the number of which, as we know, is n^{n-2}. The mapping $T \to (a_1, a_2, \ldots, a_{n-2})$ is constructed as follows:

(1) From among all vertices of degree 1, find the one with minimal number v. The number of v's neighbor is a_1.
(2) Delete v and the incident edge. This yields a tree on $n-1$ vertices. Go to step (1) and execute the instructions $(n-2)$ times. This yields the sequence $a_1, a_2, \ldots, a_{n-2}$.

Example 7.5.



$\longrightarrow\ (1,8,3,1,4,4,8)\,.$

We must now show that conversely, for every sequence $(a_1, a_2, \ldots, a_{n-2})$ there exists precisely one tree T. What does the sequence tell us about the associated tree? Let d_i be the degree of the vertex i. Suppose the number i appears f_i times in the sequence. Since every time i is added to the sequence a neighboring vertex of i is deleted, we have $f_i \leq d_i - 1$ for all i. Note that $f_i \leq d_i - 1$, since i continues to reside in the part of the tree that remains. Therefore, its degree is at least 1. From Corollary 7.3 we conclude that

$$n-2 = \sum_{i=1}^{n} f_i \ \leq\ \sum_{i=1}^{n}(d_i - 1) = 2n - 2 - n = n - 2\,.$$

Thus $f_i = d_i - 1$ for all i. In particular, the numbers that never appear in the sequence are precisely those of the vertices of degree 1.

We thus obtain the inverse mapping:

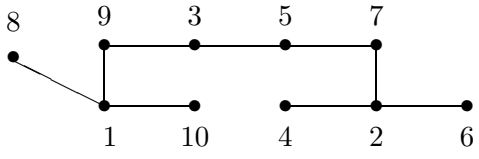
(1) Find the minimal b_1 that does not appear in the sequence $(a_1, \ldots, a_{n-2})$; this yields the edge $b_1 a_1$.
(2) Find the minimal $b_2 \neq b_1$ that does not appear in the sequence $(a_2, \ldots, a_{n-2})$, and so on.

Observe that the last edge arises automatically from the above condition on the degree.

Example 7.6.

(a) 2 2 7 5 3 9 1 1
(b) 4 6 2 7 5 3 8 9

$\longrightarrow$



□

7.2. Breadth-First and Depth-First Search

How does one find a spanning tree, or more generally a spanning forest, in a graph G given by its adjacency matrix (or equivalently, by its neighborhood lists)? How can one tell whether G is connected?

Example 7.7. Let G be given by the following neighborhood lists:

a	b	c	d	e	f	g	h
b	a	b	a	b	g	c	a
d	c	d	b			f	g
h	d	g	c			h	
	e						

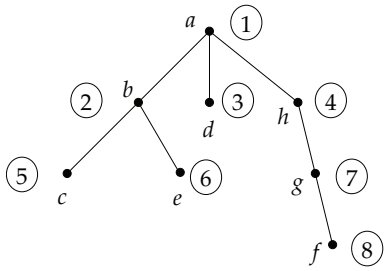
Is G connected? The following algorithm, **breadth-first search** (BFS), constructs a spanning tree (if one exists). The algorithm searches all the vertices neighboring a given vertex before proceeding, that is, the search goes first through the breadth of all neighboring vertices, whence the name of the algorithm.

Algorithm 7.8 (Breadth-first search)**.**

(1) Select a start vertex and give it the number 1. Vertex 1 is now the *current vertex*.

(2) Suppose the current vertex is number i and that numbers $1, \ldots, r$ have been assigned to vertices. If $r = n$, stop: The spanning tree is complete. Otherwise, give the neighbors of i the numbers $r+1, r+2, \ldots$ and add the edges $i(r+1), i(r+2), \ldots$ to the spanning tree. If the number $i+1$ has not been assigned, stop: The graph G is not connected (and so there is no spanning tree). Otherwise, make vertex $i+1$ the current vertex and repeat step (2).

In our example, we obtain the following:



And indeed G is connected.

We would now like to prove the correctness of our algorithm. We will prove that if G is connected, then Algorithm 7.8 indeed creates a spanning tree and asserts that the graph is connected. A similar proof shows that if

G is not connected, the algorithm halts with the assertion that the graph is not connected.

Suppose, then, that G is connected. Since a numbered vertex is a neighbor of at most one vertex with a smaller number, the algorithm never creates a circuit, and since edges are always added to the graph that has been created thus far, the resulting graph T is a tree.

Suppose $v \notin V(T)$, with u the start vertex of the algorithm. Since G is connected, there is a path $u = v_0, v_1, \ldots, v$, and therefore an index i with $v_i \in V(T),\ v_{i+1} \notin V(T)$. At some point in the algorithm, v_i was the current vertex. By step (2), all not-yet-numbered neighbors of v_i are added to T, and therefore v_{i+1} is after all in $V(T)$, a contradiction.

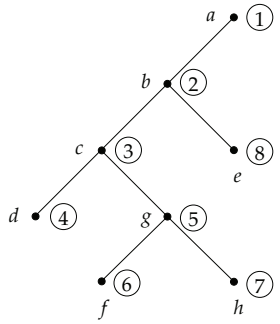
In a certain sense, the dual of breadth-first search is **depth-first search**, DFS. We move downward through the graph, edge by edge, until we can go no further. Then we take one step back and start again downward along another path. Here is the algorithm for depth-first search:

Algorithm 7.9 (Depth-first search)**.**

(1) Choose a start vertex and assign it the number 1. This is now the current vertex, and vertex 1 is the *predecessor* vertex.

(2) Suppose the current vertex has number i, and the numbers $1, \ldots, r$ have been assigned. If $r = n$, stop: The graph is connected and the spanning tree is complete. Otherwise, choose a not-yet-numbered neighbor of i, give it the number $r + 1$, and add the edge $i(r + 1)$ to the graph under construction. The current vertex is now $r + 1$, and i is the predecessor vertex. If there is no unnumbered neighbor of i, go to the predecessor vertex of i if $i > 1$. This is now the current vertex. Repeat step (2). If $i = 1$ and there is no unnumbered neighbor, then G is not connected: stop.

In our example we obtain



The proof of correctness of Algorithm 7.9 is similar to that of the previous algorithm.

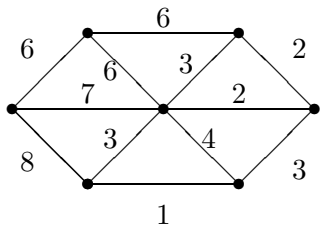
We would like to know how long it takes our algorithms, say Algorithm 7.8, to run. Step (2) searches all unnumbered neighbors of the current vertex and therefore requires $O(\sum_{u \in V} d(u)) = O(|E|)$ steps. Since the algorithm also clearly requires $\Omega(|E|)$ operations, we have the running time $\Theta(|E|)$. The analysis of depth-first search is analogous.

7.3. Minimal Spanning Trees

Suppose we have created a communication network with a number of switches (these will be the vertices) and connections between the switches (the edges). To establish a connection between switches u and v costs $w(uv)$ units. We would like to arrange the switches in such a way that each switch can communicate with every other switch at minimal cost. A similar problem is the construction of a network of roads connecting a number of points with minimal cost of travel between points.

We model this problem with a **weighted graph**. Let there be given a weighted graph $G = (V, E)$ together with a weight function $w : E \longrightarrow \mathbb{R}$. We would like to find a spanning tree T with minimal weight $w(T) = \sum_{k \in E(T)} w(k)$.

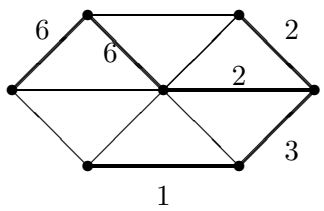
Example 7.10.



G

A naive approach is the following: Choose an edge of minimal weight. If one has already determined j edges, one chooses the next edge as that of minimal weight that does not lead to the creation of a circuit. After $n - 1$ steps a tree has been constructed.

For our graph, we obtain, for example,



$w(T) = 20$

Is this tree optimal? In fact, our algorithm always produces an optimal tree, as we shall see shortly. Since our method always chooses the best edge available, we say that we have a **greedy algorithm**.

The structural elements of our algorithmic problem are the forests contained in G. Before we prove the optimality of the greedy algorithm, we would like to analyze the set-theoretic properties of forests and thereby develop the basic ideas of the algorithm.

Definition 7.11. Let S be a finite set and $\mathcal{U} \subseteq \mathcal{B}(S)$ a family of subsets of S. The pair $\mathcal{M} = (S,\mathcal{U})$ is called a **matroid** and $\mathcal{U}$ the family of **independent sets** of $\mathcal{M}$ if the following hold:

1. $\varnothing \in \mathcal{U}$,
2. $A \in \mathcal{U},\ B \subseteq A \Rightarrow B \in \mathcal{U}$,
3. $A, B \in \mathcal{U},\ |B| = |A| + 1 \Rightarrow \exists\, v \in B \setminus A$ with $A \cup \{v\} \in \mathcal{U}$.

A maximal independent set is called a **basis** of the matroid. From axiom 3 above, it follows that every basis of $\mathcal{M}$ has the same number of elements as any other basis (clear?). This number is called the **rank** $r(\mathcal{M})$ of the matroid. More precisely, axiom 3 ensures that every independent set can be extended to a basis by the inclusion of additional elements.

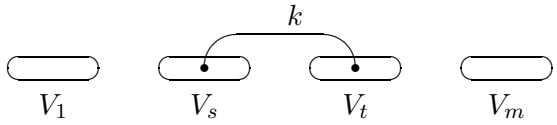
The term "matroid" suggests that we are dealing with a generalization of matrices. Consider n vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ in a vector space of dimension m over a field, for example, the real numbers $\mathbb{R}$. We may write the vectors $\boldsymbol{a}_j$ as columns $(a_{1j}, \ldots, a_{mj})^T$ and thereby obtain an $m \times n$ matrix. In this case, S is the set $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ and $A \subseteq S$ is independent if the vectors in A form a *linearly* independent set, where we declare that the empty set $\varnothing$ is linearly independent. Axiom 2 is clear, and axiom 3 is the Steinitz exchange theorem from linear algebra. The rank in this case is of course the dimension of the subspace spanned by the $\boldsymbol{a}_j$'s.

And now back to graphs. We consider all subgraphs $H = (V, A)$ on the complete set of vertices V and may therefore identify $H = (V, A)$ with the edge set $A \subseteq E$. The family of these subgraphs thus corresponds precisely to the family $\mathcal{B}(E)$ of all subsets of E. Now let $\mathcal{W} \subseteq \mathcal{B}(E)$ denote the family of edge sets of all *forests* of G.

Theorem 7.12. *If $G = (V, E)$ is a graph, then $\mathcal{M} = (E, \mathcal{W})$ is a matroid.*

Proof. Axioms 1 and 2 clearly are satisfied. Let $W = (V, A)$ and $W' = (V, B)$ be two forests with $|B| = |A| + 1$. Moreover, let $T_1, \ldots, T_m$ be the components of $W = (V, A)$ with vertex sets $V_1, \ldots, V_m$ and edge sets $A_1, \ldots, A_m$. From Theorem 7.2 we have $|A_i| = |V_i| - 1$, $i = 1, \ldots, m$, $V =$

$V_1 + \cdots + V_m$, $A = A_1 + \cdots + A_m$:



Since every forest on V_i has at most $|V_i| - 1$ edges, on account of $|B| > |A|$ there must be an edge $k \in B$ that connects two distinct sets V_s and V_t. But then $W'' = (V, A \cup \{k\})$ is a forest, and axiom 3 is satisfied. □

We see that the bases of $\mathcal{M} = (E, \mathcal{W})$ are the spanning forests, and the rank of the matroid is $|V| - t$, where t is the number of components of G.

We now know that the forests form a matroid. But we would like to solve our minimal tree problem. The following theorem states that the greedy algorithm yields an optimal basis in *every* weighted matroid $\mathcal{M} = (S,\mathcal{U})$. Furthermore, this holds as well for forests, or in the case of connected graphs, for trees. The meaning of this theorem is now clear: Whenever we can prove a matroid structure in an optimization problem with weight function, the greedy algorithm works! And further, in Exercise 7.30 we shall see that in general, the greedy algorithm yields the optimum for every weight function *precisely* in the case of matroids.

Theorem 7.13. *Let $\mathcal{M} = (S,\mathcal{U})$ be a matroid with weight function $w : S \longrightarrow \mathbb{R}$. The following algorithm produces a basis of minimal weight:*

(1) *Let $A_0 = \varnothing \in \mathcal{U}$.*
(2) *If $A_i = \{a_1, \ldots, a_i\} \subseteq S$, then let $X_i = \{x \in S \setminus A_i : A_i \cup \{x\} \in \mathcal{U}\}$. If $X_i = \varnothing$, then A_i is the desired basis. Otherwise, choose some $a_{i+1} \in X_i$ of minimal weight and set $A_{i+1} = A_i \cup \{a_{i+1}\}$. Repeat step (2).*

Proof. Let $A = \{a_1, \ldots, a_r\}$ be the obtained set. That A is a basis follows at once from axiom 3. Because of the greedy construction, we see with the help of axiom 2 that $w(a_1) \leq w(a_2) \leq \cdots \leq w(a_r)$ must hold. For $w(a_1) \leq w(a_2)$ this is clear because of step (1). Let $2 \leq i \leq r-1$. Since $\{a_1, \ldots, a_r\} \in \mathcal{U}$, it follows that $\{a_1, \ldots, a_{i-1}\} \in \mathcal{U}$ holds as well. Therefore, $a_i, a_{i+1} \in X_{i-1}$, and because of step (2), we have $w(a_i) \leq w(a_{i+1})$. Suppose $B = \{b_1, \ldots, b_r\}$ were a basis with $w(B) < w(A)$, where we assume $w(b_1) \leq \cdots \leq w(b_r)$. There would then be a smallest index i with $w(b_i) < w(a_i)$, and on account of step (1), we would have $i \geq 2$. We consider the independent sets $A_{i-1} = \{a_1, \ldots, a_{i-1}\}, B_i = \{b_1, \ldots, b_i\}$. By axiom 3, there would then exist $b_j \in B_i \setminus A_{i-1}$ with $A_{i-1} \cup \{b_j\} \in \mathcal{U}$. Since now $w(b_j) \leq w(b_i) < w(a_i)$, in the ith step, the greedy algorithm would have chosen b_j instead of a_i, which is a contradiction. □

The specialization of the greedy algorithm to matroids on graphs was discovered by Kruskal and is therefore called Kruskal's algorithm for the

MST (minimal spanning tree) problem. How many computational steps does Kruskal's algorithm require? First we must order the edges k_i by weight: $w(k_1) \leq w(k_2) \leq \cdots \leq w(k_q)$, $q = |E|$. In other words, we must sort the q weights $w(k_i)$. We shall study how to do this effectively in Chapter 9, where we shall prove that $O(q \lg q)$ comparisons are necessary. Step (2) constructs the tree iteratively. After i steps we have a forest with $n - i$ components $V_1, V_2, \ldots, V_{n-i}$. Suppose k_h was the most recently added edge. On account of $w(k_1) \leq \cdots \leq w(k_h) \leq w(k_{h+1})$, we take the next edge $k_{h+1} = uv$ and test whether it is admissible, that is, whether a circuit is created by its addition. We have

$$k_{h+1} \text{ is admissible} \iff u, v \text{ are in different } V_j\text{'s.}$$

We determine in which sets V_u, V_v the vertices u, v are located. If $V_u \neq V_v$, we add k_{h+1} and merge $V_u \cup V_v \cup \{uv\}$ into one component. If $V_u = V_v$, we test the next edge. We must therefore execute at most n comparisons for each of u and v, and the total number of operations in step (2) is $O(nq) = O(q^2)$. Therefore, altogether our algorithm needs $O(q \lg q) + O(q^2) = O(q^2)$ operations, and the reader may determine that with a suitable data structure we need only $O(q \lg q)$ operations for step (2), and so altogether we require $O(|E| \lg |E|)$.

If we exchange "minimal" with "maximal" and $\leq$ with $\geq$, then the greedy algorithm gives a basis with maximal weight, or for graphs, a tree of maximal weight.

7.4. The Shortest Path in a Graph

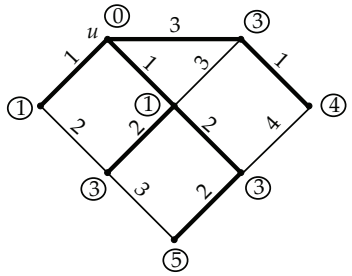
Here is another optimization problem on weighted graphs. Suppose we have a street plan in front of us and find ourselves at location u. We would like to get from u to v in the shortest possible time. The roads k (the edges of the graph) have weight $w(k) \geq 0$, which gives the minimal time required to traverse the edge k (on foot, by car, tram, etc.).

Modeled as a graph, this means that we are given a connected graph $G = (V, E)$ and a weight function $w : E \to \mathbb{R}^+ = \{x \in \mathbb{R} : x \geq 0\}$. Let $u \in V$. For a path $P = P(u, v)$ from u to v, we denote by $\ell(P) = \sum_{k \in E(P)} w(k)$ the (weighted) *length* of P. We are seeking a *shortest* u, v-path for which $\ell(P)$ is minimal. The **distance** $d(u, v)$ is defined as the length of a shortest path. In the special case $w(k) = 1$ for all $k \in E$, it follows that $\ell(P)$ is precisely the previously defined length (the number of edges), and $d(u, v)$ the previously defined distance.

Let u be fixed. The following famous algorithm of Dijkstra constructs a spanning tree whose unique path from u to v is always the shortest, for *all* $v \in V$:

(1) Let $u_0 = u, V_0 = \{u_0\},\ E_0 = \varnothing,\ \ell(u_0) = 0$.
(2) Suppose $V_i = \{u_0, u_1, \ldots, u_i\},\ E_i = \{k_1, \ldots, k_i\}$. If $i = n-1$, we are done. Otherwise, consider for all edges $k = vw$, $v \in V_i$, $w \in V \setminus V_i$, the expression $f(k) = \ell(v) + w(k)$ and choose $\overline{k}$ with $f(\overline{k}) = \min f(k)$. Let $\overline{k} = \overline{v}\,\overline{w}$. Then set $u_{i+1} = \overline{w},\ k_{i+1} = \overline{k},\ V_{i+1} = V_i \cup \{u_{i+1}\}$, $E_{i+1} = E_i \cup \{k_{i+1}\},\ \ell(u_{i+1}) = f(\overline{k})$. Repeat step (2).

Example 7.14.



The circled numbers are the $\ell(v)$.

Theorem 7.15. *Let $G = (V, E)$ be a connected graph with weight function $w : E \longrightarrow \mathbb{R}^+$, $u \in V$. Dijkstra's algorithm gives a spanning tree T with the property that the unique path from u to v is always a minimal u, v-path in G with $d(u, v) = \ell(v)$ for all v.*

Proof. The algorithm certainly constructs a spanning tree. In the first step an edge of minimal weight $u = u_0$ to a neighbor is chosen; thus $k_1 = u_0u_1$ is a minimal u_0, u_1-path with $\ell(u_1) = w(k_1) = d(u_0, u_1)$. Suppose the partial tree $T_i = (V_i, E_i)$ has the desired properties, and step (2) constructs $\overline{k} = \overline{v}\,\overline{w}$. We must show that $\ell(\overline{w}) = f(\overline{k}) = \ell(\overline{v}) + w(\overline{k})$ is equal to the (weighted) distance $d(u_0, \overline{w})$. For the $u_0, \overline{w}$-path P_0 just constructed, we have $\ell(P_0) = d(u_0, \overline{v}) + w(\overline{k}) = \ell(\overline{v}) + w(\overline{k}) = \ell(\overline{w})$. Let P be a shortest $u_0, \overline{w}$-path and v the last vertex of V_i in P, with $w \in V \setminus V_i$ as successor, $k = vw$. Then the partial paths $P(u_0, v)$, $P(w, \overline{w})$ are also shortest paths, and we obtain

$$\begin{aligned} d(u_0, \overline{w}) &= \ell(P(u_0, v)) + w(k) + \ell(P(w, \overline{w})) \\ &= (\ell(v) + w(k)) + \ell(P(w, \overline{w})) \\ &= f(k) + \ell(P(w, \overline{w})) \\ &\geq f(\overline{k}) = \ell(\overline{w}) = \ell(P_0)\,. \end{aligned}$$

Therefore, P_0 is a shortest path. □

We see that our algorithm always constructs a shortest prolongation of the partial tree. We are therefore again dealing with a greedy algorithm.

Several variants of the shortest-path problem come at once to mind. Suppose we would like to determine a shortest path between only two given vertices u and v. We can use Dijkstra's algorithm with u as source, and this gives a shortest path from u to v. No algorithm is known that is asymptotically faster than Dijkstra's algorithm. Or perhaps we would like to find shortest paths for *all* pairs of vertices u, v. We can solve this problem by applying our algorithm to every vertex u as source, but there are generally faster algorithms available. See the cited literature at the end of this part, after Chapter 10.

Finally, we remark that it should be clear how the procedure may be modified for directed graphs. In this case, we seek shortest *directed* paths from u to all other vertices.

Exercises for Chapter 7

7.1 Prove the following characterization of trees: Let G be a graph on n vertices and q edges. Then G is a tree if and only if the following conditions are satisfied: (a) G has no circuits and $q = n - 1$. (b) G has no circuits and if any pair of nonneighboring vertices are joined by an edge, then the resulting graph has precisely one circuit. (c) G is connected ($G \neq K_n$ if $n \geq 3$), and if any two nonneighboring vertices are joined by an edge, the resulting graph has exactly one circuit.

▷ **7.2** Show that a connected graph with an even number of vertices always has a spanning subgraph in which all vertices have odd degree. Does this hold for disconnected graphs?

7.3 Let G be a connected graph. For $u \in V$ we set $r(u) = \max(d(u, v) : v \neq u)$. The parameter $r(G) = \min(r(u) : u \in V)$ is called the *radius* of G, and $Z(G) = \{u \in V : r(u) = r(G)\}$ is the *center* of G. Show that the center of a tree consists of either one vertex or two neighboring vertices.

7.4 Let $d_1 \geq \cdots \geq d_n > 0$ be a sequence of natural numbers. Show that $(d_1, \ldots, d_n)$ is the degree sequence of a tree if and only if $\sum_{i=1}^{n} d_i = 2n - 2$.

▷ **7.5** Determine among all trees with n vertices those for which the sum $\sum_{u \neq v \in V} d(u, v)$ is minimal and those for which it is maximal.

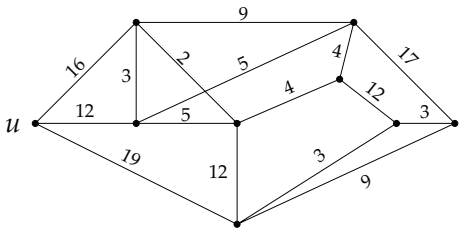
7.6 Carry out the correctness proof of depth-first search.

7.7 Generalize BFS or DFS by constructing an algorithm that determines the connected components of a graph.

7.8 Verify precisely that all bases of a matroid have the same size.

▷ **7.9** Suppose a connected graph G has a different weight on each edge. Show that G possesses a unique minimal spanning tree.

7.10 Consider the following graph G with cost function. Determine using Dijkstra's algorithm the shortest paths from u to all other vertices:



▷ **7.11** Show that a tree has at least as many vertices of degree 1 as the maximal degree. When does equality hold?

▷ **7.12** This exercise gives a recursive proof of the formula $t(n) = n^{n-2}$ for the number of spanning trees in K_n. Number the vertices $1, 2, \ldots, n$. Let $C(n,k)$ be the number of spanning trees in which the vertex n has degree k. Prove the recurrence $C(n,k) = \frac{k(n-1)}{n-1-k} C(n,k+1)$ and thereby conclude the desired formula.

▷ **7.13** Let B be the incidence matrix of a graph G. In every column we change arbitrarily one of the two 1's into -1 (this corresponds to an orientation of G) and call the new matrix C. Let $M = CC^T$. Prove that the number of spanning trees of G is given by $t(G) = \det M_{ii}$, where M_{ii} results from deleting the ith row and ith column from M (and this holds for every i). Hint: Let P be an $r \times s$ matrix and Q an $s \times r$ matrix, $r \leq s$. Then a theorem from linear algebra states that $\det(PQ)$ is equal to the sum of the products of determinants of the corresponding $r \times r$ submatrices.

7.14 Again verify, now using the previous exercise, $t(K_n) = n^{n-2}$.

7.15 Calculate $t(K_{m,n})$. Answer: $m^{n-1}n^{m-1}$.

7.16 Consider the complete graph K_n on $\{1, \ldots, n\}$ and let $d_1, \ldots, d_n$ be a sequence of natural numbers each greater than or equal to 1 with $\sum_{i=1}^{n} d_i = 2n - 2$. Show that the number of spanning trees in which the vertex i has degree d_i is equal to $\frac{(n-2)!}{(d_1-1)!\cdots(d_n-1)!}$, and derive from this an additional proof of the formula $t(K_n) = n^{n-2}$.

▷ **7.17** Let $G = K_n \setminus \{k\}$ be the complete graph with an edge removed. Without using Exercise 7.13, calculate the number of spanning trees in G.

▷ **7.18** The lattice graph $G(2,n)$ consists of two paths of n vertices (let them be numbered 1 to n) whose vertices with the same number are joined by an edge. For example, $G(2,2) \cong C_4$. Determine the number of spanning trees of $G(2,n)$ using generating functions.

7.19 Show that every automorphism of a tree leaves at least one vertex or one edge fixed. Hint: Exercise 7.3.

7.20 For n employees, each of whom drives his or her own car to work, there are n parking places available in the company parking lot. Each driver has a spot that

he or she prefers, and in fact, driver i prefers $g(i)$, $1 \le g(i) \le n$. The drivers arrive at the parking lot one after the other: first 1, followed by 2, and so on. The ith driver parks in space $g(i)$ if it is free. Otherwise, he or she takes the next number $k > g(i)$ that is available if the space is empty. Otherwise, the driver goes home and never returns to work. Here is an example with $n = 4$:

	1	2	3	4
g	3	2	2	1

, then $1 \to 3,\ 2 \to 2,\ 3 \to 4,\ 4 \to 1,$

but

	1	2	3	4
g	2	3	3	2

, $1 \to 2,\ 2 \to 3,\ 3 \to 4,\ 4 \to$ quits

Let $p(n)$ be the number of functions g that allow each employee to obtain a parking space. Determine $p(n)$. Hint: $p(2) = 3$, $p(3) = 16$, $p(4) = 125$.

▷ **7.21** Let $G = (V, E)$ be a connected graph and let $w : E \to \mathbb{R}^+$ be a cost function. Show that the following algorithm constructs a minimal spanning tree: (1) Select an edge uv of minimal weight and set $S = \{u, v\},\ T = V \setminus S$. (2) If $T = \varnothing$, stop. Otherwise, choose from among the edges between S and T an edge $k = \overline{u}\,\overline{v}$ of minimal weight, $\overline{u} \in S$, $\overline{v} \in T$, and set $S \leftarrow S \cup \{\overline{v}\}$, $T \leftarrow T \setminus \{\overline{v}\}$. Repeat step (2).

7.22 Estimate the running time of the algorithm in the previous exercise in terms of the number of vertices n.

7.23 Consider K_n on $\{1, \ldots, n\}$ with the cost function $w(ij) = i+j$. (a) Construct an MST tree T. (b) What is $w(T)$? (c) Is T uniquely determined?

▷ **7.24** Let $\mathcal{M} = (S,\mathcal{U})$ be a matroid, and $\mathcal{B}$ the family of bases. Show that $\mathcal{B}$ satisfies the following conditions: (a) $A \ne B \in \mathcal{B} \Rightarrow A \not\subseteq B, B \not\subseteq A$. (b) Let $A \ne B \in \mathcal{B}$. Then for each $x \in A$ there exists $y \in B$ with $(A \setminus \{x\}) \cup \{y\} \in \mathcal{B}$. Show conversely that a family of sets $\mathcal{B}$ that satisfies conditions (a) and (b) is the family of bases of a matroid.

7.25 Let $\mathcal{M} = (S,\mathcal{U})$ be a matroid and $\mathcal{B}$ the family of bases. Show that the family $\mathcal{B}^* = \{S \setminus B : B \in \mathcal{B}\}$ defines a matroid $\mathcal{M}^*$. The matroid $\mathcal{M}^*$ is called the *dual* of the matroid $\mathcal{M}$. What is the rank of $\mathcal{M}^*$?

7.26 Let $\mathcal{M} = (V, \mathcal{W})$ be the usual matroid induced by the graph $G = (V, E)$. Give a graph-theoretic description of the dual matroid $\mathcal{M}^*$. That is, what sets of edges are independent in $\mathcal{M}^*$?

▷ **7.27** As in the previous exercise, let $\mathcal{M} = (V, \mathcal{W})$ be given. A set of edges $A \subseteq V$ is said to be *minimally dependent* if $A \notin \mathcal{W}$ but $A' \in \mathcal{W}$ for every proper subset A' of A. Describe the minimally dependent sets in the graph and also the minimally dependent sets in $\mathcal{M}^*$. Hint: The minimally dependent sets in $\mathcal{M}^*$ are *minimal cut sets* A; that is, $G = (V, E \setminus A)$ has one component more than G and is minimal with respect to this property.

7.28 Show that circuits and minimal cut sets always have an even number of edges in common.

7.29 Let B be the incidence matrix of $G = (V, E)$, interpreted as a matrix over the field $\{0, 1\}$. Show that $A \subseteq E$ is independent in the matroid $(E, \mathcal{W})$ if and only if the associated set of columns is linearly independent.

▷ **7.30** Let $(S, \mathcal{U})$ be a collection of sets satisfying axioms 1 and 2 of a matroid. Show that $(S, \mathcal{U})$ is a matroid (thus satisfies axiom 3 as well) if and only if the greedy algorithm yields the optimum for *every* weight function $w : S \to \mathbb{R}$.

7.31 Develop a Dijkstra algorithm for directed graphs.

7.32 Determine the shortest path from 1 to all i in the following directed graph, given by a weight matrix. A missing entry means that the corresponding edge is not present in the graph:

	1	2	3	4	5	6	7
1			4	10	3		
2			1	3	2	11	
3		9		8	3	2	1
4		4	5		8	6	3
5	1		1	2		3	1
6		1	1	3	2		
7	2	4	3			2	

▷ **7.33** Dijkstra's algorithm functions only for nonnegative edge weights $w(u, v) \in \mathbb{R}^+$. Suppose that there also exist negative weights on a graph. If directed circuits with negative total weight exist, then no shortest distance can exist (why?). Suppose, then, that we have a directed graph $\vec{G} = (V, E)$ with source $u \in V$, and a weight function $w : E \to \mathbb{R}$ without negative circuits, where we assume that all vertices $x \neq u$ from u can be reached by a directed path. Show that the following algorithm of Bellman–Ford creates a tree with shortest directed path from u: (1) Number $E = \{k_1, \ldots, k_q\}$; set $\ell(u) = 0$, $\ell(x) = \infty$ for $x \neq u, B = \varnothing$. (2) Run through the edges of E. Let $k_i = (x, y)$. If $\ell(x) + w(x, y) < \ell(y)$, set $\ell(y) \leftarrow \ell(x) + w(x, y)$ and $B \leftarrow (B \cup \{k_i\}) \setminus \{k\}$, where k is the previous edge in B with end vertex y. Repeat $(|V| - 1)$ times. Then $\vec{G} = (V, B)$ is a "shortest" tree. Hint: Let $u, v_1, \ldots, v_k = v$ be a shortest path in $\vec{G}$. Show by induction that $\ell(v_i) = d(u, v_i)$ after the ith iteration.

7.34 Determine shortest paths from 1 in the following graph, given by its length matrix:

	1	2	3	4	5
1		6	5		
2			7	3	−2
3				−4	8
4		−1			
5	2			7	

▷ **7.35** Think about how the algorithm of Bellman–Ford should be extended to return the output "no solution" in the case of negative circuits.

7.36 A minimax or bottleneck tree is a spanning tree in which the maximum of the edge weights is as small as possible. Show that every MST tree is also a minimax tree.

Index