24. We are given an undirected graph $G = (V, E)$ with edge weights $w_e \in \mathbb{R}$ for all $e \in E$ as well as an edge $\{k, \ell\} \in E$. The task is to find a spanning tree which contains the edge $\{k, \ell\}$ and has minimum sum weight among all such trees.
    (a) Provide an efficient algorithm for solving this problem and prove its correctness.
    (b) Generalize your approach to the case where one starts with a cyclefree set $E' \subset E$ of edges which all should be contained in the minimum weight spanning tree to be determined.
25. We are given an instance of the minimum spanning tree problem (graph $G = (V, E)$ and edge weights w_e).
    (a) Suppose that during an exam you realize after having determined an MST T^* that you have forgotten to take an edge from the set E into account. How can you fix this issue in a time-efficient way (i.e. without starting from scratch) and obtain an MST for the correct instance?
    (b) Which approach do you suggest for the case when someone by mistake included an edge into the MST T^* which is not in E.
26. (Sensitivity analysis) Let $T^* = (V, E_{T^*})$ be a minimum spanning tree and let e be an edge in E. We want to find the set of all values for the edge weight w_e such that the tree T^* remains an MST.
    (a) First, prove that this set is an interval I_e. (This interval is called tolerance interval of edge e.)
    (b) Devise an algorithm for the efficient determination of the interval I_e.
    (c) Provide an algorithm which determines the tolerance intervals I_e for alle $e \in E$ in one step. (This algorithm should be faster than the algorithm which results by applying the algorithm from (b) separately for each edge e.)
27. Consider the following special case of the minimum spanning tree problem: We are given a complete undirected graph G as well as n positive real numbers α_i, $i = 1, \ldots, n$. The task is to find a minimum spanning tree with respect to the edge weights $w_e = \alpha_i \alpha_j$ where $e = \{i, j\}$. Provide an algorithm that solves this problem as efficiently as possible. (Your approach should be faster than applying a general MST algorithm from the shelf.)
28. **(Minimum ratio spanning tree)** We are given an undirected graph $G = (V, E)$ along with two real edge weights w_e and v_e for all $e \in E$. The task is to find a spanning tree $T^* = (V, E_{T^*})$ which minimizes the following linear fractional objective function

    $$\frac{\sum_{e \in E_{T^*}} w_e}{\sum_{e \in E_{T^*}} v_e},$$

    where we assume that $\sum_{e \in E_T} v_e > 0$ holds for all spanning trees $T = (V, E_T)$.

    Devise a polynomial time algorithm for solving this problem. (Hint: Apply binary search.)
29. **(Balanced spanning tree problem)** We are given an undirected graph $G = (V, E)$ and real edge weights w_e, $e \in E$. The task is to find a so-called *balanced spanning tree*, i.e., a spanning tree $T = (V, E_T)$ which minimizes the difference between the largest and the smallest weight

    $$\max_{e \in E_T} w_e - \min_{e \in E_T} w_e.$$

    Devise an efficient algorithm to find a balanced spanning tree.
30. **Parametric spanning tree problems)** Let $G = (V, E)$ be a connected undirected graph with edge weights which are affine-linear functions of a real parameter λ, i.e., $w_e(\lambda) = u_e + \lambda v_e$, for $e \in E$. Let $(G, w(\lambda))$ denote the graph G together with the weights $w_e(\lambda)$. Let T^λ be a minimum spanning tree in $(G, w(\lambda))$.

(a) Prove or disprove the following statement: There exists a constant $k \in \mathbb{N}$ such that T^{λ} is a minimum spanning tree in G with respect to the weights v_e for all $\lambda \geq k$ and $T^{-\lambda}$ is a maximum spanning tree with respect to the weights v_e for all $\lambda \leq -k$.

(b) Consider a parameter value $\tilde{\lambda} \in \mathbb{R}$. Prove that there exist values $\underline{\lambda}$ and $\bar{\lambda}$ with $\underline{\lambda} < \bar{\lambda}$ such that $T^{\tilde{\lambda}}$ is a minimum spanning tree in $(G, w(\lambda))$ for all $\lambda \in [\underline{\lambda}, \bar{\lambda}]$.

(c) Devise an algorithm that determines a minimum spanning tree T^{λ} for all $\lambda \in \mathbb{R}$ (i.e., in dependance of the parameter λ.) Analyse its running time.

(d) Consider the special case where $v_e \in \{0, 1\}$ for all $e \in E$. How does this help in improving the running time achieved in (c)?

31. Provide an example for an independence system $(S, \mathcal{F})$ in which all bases (maximal independent sets) have the same cardinality, but which is not a matroid.

32. Prove that deleting elements does not destroy the matroid property. More specifically let $(S, \mathcal{F})$ be a matroid and let $U \subset S$. The (sub)matroid generated by U is given by $(U, \mathcal{F}_U)$ where

$$\mathcal{F}_U := \{ A \subseteq S : A \in \mathcal{F} \text{ and } A \subseteq U \}$$

Prove that this indeed is a matroid.

33. Let $(S, \mathcal{F})$ be a matroid and let $k \in \mathbb{N}$. Show that the so-called *truncated matroid*

$$\mathcal{F}_{\leq k} := \{ A \subseteq S : A \in \mathcal{F} \text{ and } |A| \leq k \}$$

is a matroid.

34. Let $G = (V, E)$ be an undirected graph. Consider the set system $(S, \mathcal{F})$ with $S = E$ and

$$\mathcal{F} := \{F \subseteq E : \text{each component of the graph induced by } F \text{ contains } \leq 1 \text{ cyle and no even cyle.}\}$$

Prove that this set system is a matroid.

35. What happens in the previous example if the condition "and no even cycle" is replaced by "and no odd cycle"? Do we still get a matroid?

36. We are given a matroid $(S, \mathcal{F})$ and a weight function $c : S \to \mathbb{R}$. Provide an algorithm which finds a (not necessarily maximal) independent set $F^* \in \mathcal{F}$ with maximum weight.

37. Prove the following statement: Let S be a finite set and let $\mathcal{B} \subseteq 2^S$. $\mathcal{B}$ is the set of bases (maximal independent sets) of a matroid if and only if the following two properties hold:

(a) $\mathcal{B} \neq \emptyset$.

(b) For all $B_1, B_2 \in \mathcal{B}$ and $x \in B_1 \setminus B_2$ there exists an element $y \in B_2 \setminus B_1$ such that $(B_1 \setminus \{x\}) \cup \{y\} \in \mathcal{B}$.

38. We are given a ground set S together with a partition $S = S_1 \cup S_2 \cup \cdots \cup S_k$ into k subsets and k numbers $b_1, \ldots, b_k \in \mathbb{N}$. Consider the set system $(S, \mathcal{F}$ where

$$\mathcal{F} := \{ A \subseteq S : |A \cap S_i| \leq b_i \text{ for } i = 1, \ldots, k \}$$

Prove that $(S, \mathcal{F})$ is a matroid (it is called partition matroid).

39. Provide an example which demonstrates that the solution obtained by the greedy algorithm for the knapsack problem (consider the items in the order obtained by $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \leq \frac{p_n}{w_n}$ and pack an items into the knapsack if it still fits) can get arbitrarily bad.

40. We are given the directed graph G with attached arc weights w_{ij} displayed in Figure 2.

    (a) Determine an aborescence with root 1 and minimum weight in the given graph.

    (b) How does the solution in (a) change if no root is provided?

    (c) Determine an aborescence with root 1 and maximum weight in the given graph.

    (d) How does the solution in (c) change if no root is provided?

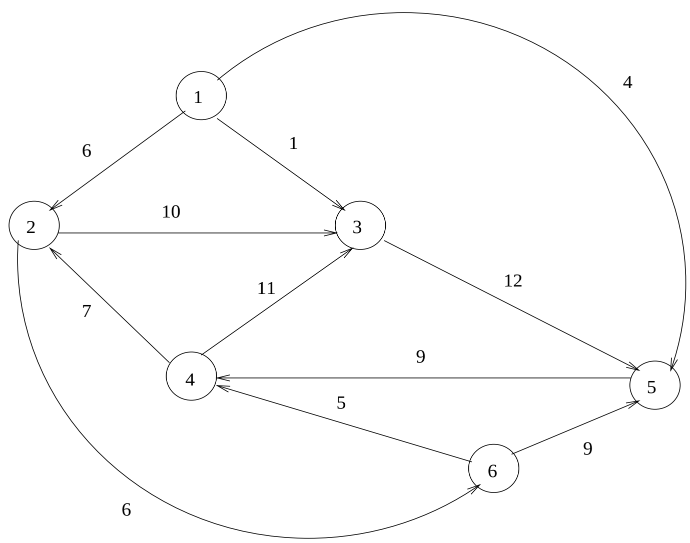

Figure 2

41. Determine a branching with maximum weight for the weighted graph displayed in Figure 3.

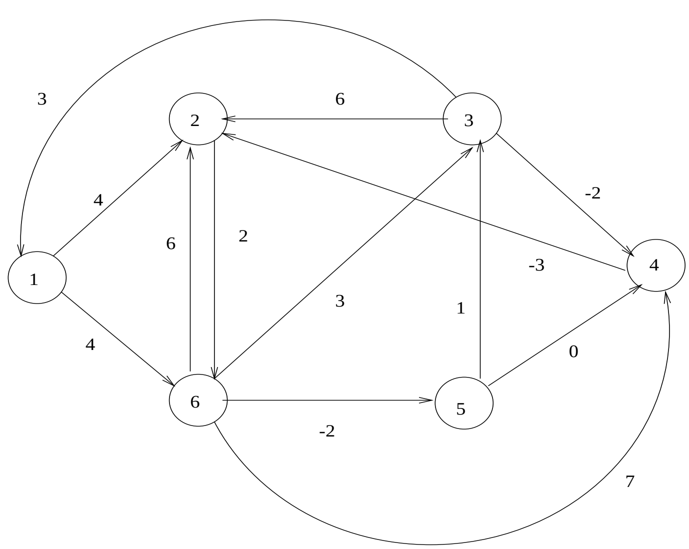

Figure 3

42. Devise an efficient algorithm for solving the following arborescence problem with bottleneck objective function: We are given a directed graph $G = (V, A)$, a root $r \in V$ and real arc weights w_{ij} for $(i, j) \in A$. The task is to find an arborescence $\mathcal{T} = (V, A_{\mathcal{T}})$ with root r and minimum bottleneck weight $w(\mathcal{T}) = \max_{a \in A_{\mathcal{T}}} w_a$.

43. In the class we have proved that the minimum weight rooted arborescence problem can be reduced to the maximum weight branching problem. However, the problem can also be solved directly by a modification of Edmond's branching algorithm. Show how.

44. We are given a directed graph $G = (V, A)$, a root $r \in V$ and real arc weights w_{ij} for $(i, j) \in A$. The task is to find an arborescence with root r and minimum weight.

    Formulate this problem as an integer linear programming problem (linear constraints and linear objective function, integrality constraints for all/some variables are allowed).

45. Use Dijkstra's algorithm to obtain a shortest path tree for the weighted graphs display in Figure 4. (The edge without a weight in the figure has weight 1.)

46. Find a directed graph $G = (V, A)$ with arc weights $w_{ij} \in \mathbb{R}$ such that G contains no negative cycle and Dijkstra's algorithm fails (i.e. there exists at least one vertex i for which the algorithm does not find a shortest path from the start vertex s to i).

47. Find a directed graph $G = (V, A)$ with arc weights $w_{ij} \in \mathbb{R}$ such that at least one arc weight is negative, no negative cycle exists and Dijkstra's algorithm works correctly (i.e. determines a shortest path tree).

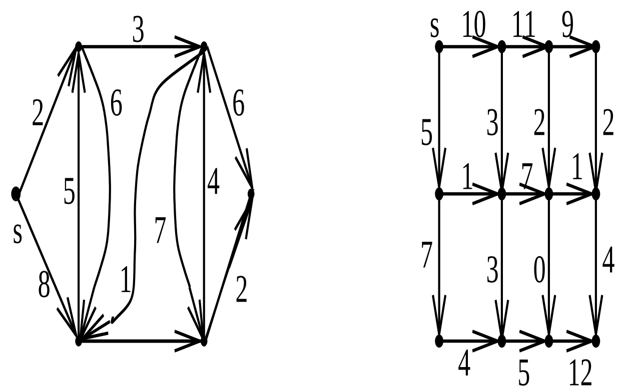

Figure 4

48. Which of the following statements are true and which are false? (Provide a proof for true statements and a counterexample for false ones).

    (a) If the arc weights w_{ij} are pairwise distinct, there exists a unique shortest path tree.
    (b) Adding a natural number k to the arc weights of all arcs w_{ij} in a shortest path problem, has no effect on the set of shortest paths.
    (c) Given a directed graph $G = (V, A)$, arc weights $w_{ij} \geq 0$ and a start vertex s, Dijkstra's algorithm (as presented in class) finds a shortest path from the start vertex s to all other vertices $t \in V \setminus \{s\}$ with the additional property that the number of arcs along the path is minimized.