Combinatorial optimization 1
Winter term 2016/2017

1st working sheet

1. Prove that there are n^{n-2} labelled spanning trees in a complete graph on n-vertices by using a double counting argument. To this end you can consider so-called *labelled rooted trees* (LRT); these are trees with a distinguished vertex as a root and labelled edges directed towards the root. What can be stated about the out-degree of the vertices in an LRT?

   Count LRTs in two different ways: a) by considering how many pairwise different LRTs can be obtained from a single spanning tree, and b) by considering how many LRTs can be constructed starting with an empty graph on n vertices and adding one edge at a time until an LRT is obtained.

2. Show that the number $\tau(K_{n,m})$ of labelled spanning trees in a complete bipartite graph $K_{n,m}$ is given as $\tau(K_{n,m}) = m^{n-1}n^{m-1}$, for any $n, m \in \mathbb{N}$. Recall that the complete bipartite graph $K_{n,m}$ is defined as a graph with vertex set $V(K_{n,m}) := \{v_1, v_2, \ldots, v_n\} \cup \{w_1, w_2, \ldots, w_m\}$ and edge set $E(K_{n,m}) := \{(v_i, w_j) : i \in \{1, 2, \ldots, n\}, j \in \{1, 2, \ldots, m\}\}$, for any $n, m \in \mathbb{N}$.

3. Consider the problem of determining a connected spanning subgraph H of minimum weigth $c(H) := \sum_{e \in E(H)} c(e)$ of a given graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$. Can you give a polynomial time algorithm to solve this problem?

4. Consider a graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$. Let $E_1 \subseteq E$ be the set of edges e in $E(G)$ such that there exists a minimum spanning tree T_e of G with $e \in E(T_e)$. Propose an algorithm to determine the set E_1 for a given graph G with edge weights $c\colon E \to \mathbb{R}$ and analyze its complexity. Try to design an algorithm which is as efficient as possible.

5. Apply Kruskal's algorithm to determine a minimum spanning tree in the graph depicted in Figure 1. Use the branching data structure to check for existence of cycles (cf. lecture).

6. Apply Prim's algorithm to determine a minimum spanning tree in the graph depicted in Figure 1. Choose vertex 1 to initialize T (cf. lecture).

7. *The minimum bottleneck spanning tree problem (MBSTP).*
   Consider an input graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$. The MBSTP consists in determining a spanning tree T^* in G with minimum weight of its heaviest edge, i.e. a spanning tree T^* which fulfills $b(T^*) = \min\{b(T) : T \text{ is a spanning tree in } G\}$, where $b(T)$ is defined as $b(T) := \max\{c(e) : e \in E(T)\}$ for every spanning tree T in G. Propose an algorithm to solve this problem and analyze its complexity. Try to design an algorithm which is as efficient as possible.

8. Let $G = (V, E)$ be a directed graph (digraph) with n vertices and let r be some distinguished vertex in V, $r \in V$. Show that the following statements are equivalent:

   (1) G is an arborescence with root r.

   (2) G is a branching with $n - 1$ edges and $\deg^-(r) = 0$.

   (3) G has $n - 1$ edges and every vertex is reachable from r, i.e. there exists a directed path from r to each vertex.

   (4) Every vertex is reachable from r, but the removal of a single arbitrary edge from G destroys this property.

   (5) $\delta^+(X) \neq \emptyset$ holds for all $X \subset V$ with $r \in X$, but the removal of a single arbitrary edge from G destroys this property.

   (6) $\deg^-(r) = 0$ and for every $v \in V \setminus \{r\}$ there is a unique directed r-v-path (i.e. a directed path from r to v).

   (7) $\deg^-(r) = 0$, $\deg^-(v) = 1$, for all $v \in V \setminus \{r\}$, and G is cycle-free.

(8) $\deg^-(r) = 0$, $\deg^-(v) \leq 1$, for all $v \in V \setminus \{r\}$, and every vertex $v \in V$ is reachable from r.

9. Show that the three problems MWAP (*Minimum Weighted Arborescence Problem*), MWRAP *Minimum Weighted Rooted Arborescence Problem*) und MWBP (*Maximum Weighted Branching Problem*) are equivalent. Recall that:

   (a) The MWAP consists in finding a spanning arborescence with minimum weight in an input digraph with edge weights, if an arborescence exists.

   (b) The MWRAP consists in finding a spanning arborescence with root r and minimum weight in an input digraph G with edge weights and some $r \in V(G)$, if such an arborescence exists.

   (c) The MWBP consists in finding a branching with maximum weight in an input digraph with edge weights.

10. Can it be decided in linear time whether a digraph has a spanning arborescence? Argue your answer carefully.

    Hint: Observe that a candidate-root vertex can be determined by starting at an arbitrary vertex and moving backwards as long as possible along edges ending at the current vertex. The cycles which are closed during this procedure should be contracted.

11. Use Edmond's Branching algorithm to determine a maximum branching in the digraph depicted in Figure 2.

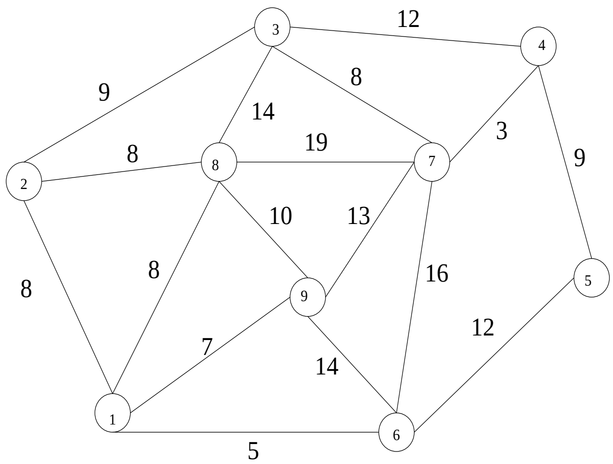

Figure 1: Input graph for Exercises 5 and 6

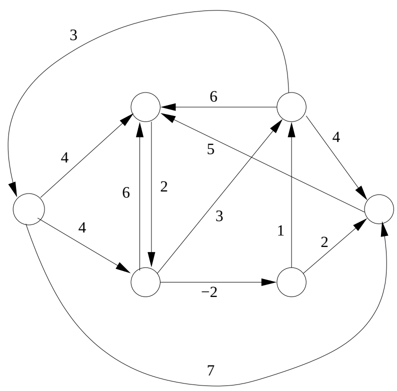

Figure 2: Input digraph for Exercise 11