Kombinatorische Optimierung WS 2010-2011
1. Übungsblatt

1. Für einen gegebenen ungerichteten Graphen $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{R}$ soll ein zusammenhängender aufspannender Teilgraph minimalen Gewichtes bestimmt werden. Kann dieses Problem effizient gelöst werden? Wie?

2. *Das maximale Wald-Problem (MaxWP)*
   Sei $G = (V, E)$ ein zusammenhängender ungerichteter Graph mit Kantengewichten $c\colon E \to \mathbb{R}$. Beim maximalen Wald-Problem muss ein kreisfreier Teilgraph von G (also ein *Wald*) $W = (V', E')$ mit $V' \subseteq V$ und $E' \subseteq E$ und maximalem Gewicht $c(W) := \sum_{e\in E'} c(e)$ bestimmt werden. Zeigen Sie, dass dieses Problem äquivalent zum minimalen Spannbaumproblem (MST-Problem) ist, d.h. die Lösung des MaxWP-Problems kann als Lösung eines MST-Problems in einem Graphen $\tilde{G} = (\tilde{V}, \tilde{E})$ mit den Kantengewichten $\tilde{c}\colon \tilde{E} \to \mathbb{R}$ ermittelt werden, wobie die Inputgröße dieses MST-Problems als Polynom der Inputgröße des ursprünglichen MaxWP-Problems gesehen werden kann.

3. Gesucht ist die Menge derjenigen Kanten e in einem ungerichteten Graphen $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{R}$, für die es einen enthaltenden spannenden Baum minimalen Gewichtes in G gibt (oder anders ausgedruckt, man möchte die Vereinigung aller aufspannenden Bäume minimalen Gewichtes in G bestimmen). Geben Sie einen möglichst effizienten Algorithmus zur Lösung dieses Problems an und analysieren Sie die Zeitkomplexität ihres Algorithmus.

4. Zeigen Sie, dass für den vollständig bipartiten Graphen $\tau(K_{n,m}) = m^{n-1}n^{m-1}$ gilt.

5. Bestimmen Sie mit dem Verfahren von Kruskal einen minimalen spannenden Baum für den Graphen aus Abbildung 1.

6. Bestimmen Sie mit dem Verfahren von Prim einen minimalen spannenden Baum für den Graphen aus Abbildung 1 (wählen Sie den Knoten 1 als Startknoten).

7. *Das Engpass-Spannbaum-Problem (bottleneck MST)*
   Für einen gegebenen ungerichteten Graphen $G = (V, E)$ mit Kantengewichten $c\colon E \to \mathbb{R}$ soll ein spannender Baum T gefunden werden, mit der Eigenschaft, dass das Gewicht der schwersten Kante in T so gering wie möglich ist. Geben Sie einen möglichst effizienten Algorithmus zur Lösung dieses Problems an und analysieren Sie die Zeitkomplexität ihres Algorithmus.

8. Das Voronoï-Diagramm einer endlichen Menge $V \subset \mathbb{R}^2$ besteht aus den Gebieten
   $$P_u = \{x \in \mathbb{R}^2\colon ||\mathrm{x} - \mathrm{v}||_2 = \min_{\mathrm{w}\in\mathrm{V}} ||\mathrm{x} - \mathrm{w}||_2\}$$
   für alle $v \in V$. Die Delaunay-Triangulierung von V ist der Graph
   $$(V, \{(v, w)\colon v, w \in V, v \neq w, |P_u \cap P_v| > 1\}).$$
   Ein minimaler Spannbaum für V ist ein Baum $T = (V(T), E(T))$ mit Knotenmenge $V(T) = V$ und minimaler Länge $\sum_{(v,w)\in E(T)} ||v - w||_2$. Beweisen Sie, dass jeder minimaler Spannbaum von V ein Teilgraph der Delaunay-Triangulierung ist.

9. Für einen gegebenen ungerichteten Graphen $G = (V, E)$ mit den Kantengewichten $c\colon E \to \mathbb{R}$ und einen Knoten $v \in V$ soll ein aufspannender Baum T minimalen Gewichtes in G bestimmt werden, in dem v kein Blatt ist. Kann man dieses Problem in polynomieller Zeit lösen? Begründen Sie ihre Antwort!

10. Kann man in linearer Zeit entscheiden ob ein Graph eine aufspannende Arboreszenz hat? Begründen Sie ihre Antwort.

    Hinweis: Um eine mögliche Wurzel zu bestimmen kann man bei einem beliebigen Knoten anfangen und so lange wie möglich Kanten rückwärts entlang gehen. Wenn man einen Kreis erreicht, dann wird der Kreis kontrahiert.

11. Beweisen Sie, dass das Spannbaumpolytop eines ungerichteten Graphen $G = (V, E)$ mit n Knoten, $n := |V|$, im allgemeinen eine echte Teilmenge des folgenden Polytops ist

$$\{x \in [0,1]^E\colon \sum_{e \in E} x_e = n - 1, \sum_{e \in \delta(X)} x_e \geq 1 \text{ für } \emptyset \subset X \subset V\}.$$

Hinweis: Um zu beweisen, dass das obige Polytop nicht ganzzahlig ist, betrachten Sie den Graphen in Abbildung 2, wobei die Zahlen die Kantengewichte sind.

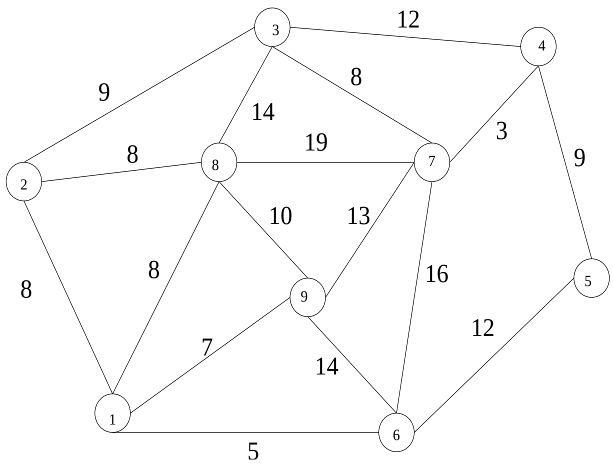

Abbildung 1: Input für die Beispiele und 6

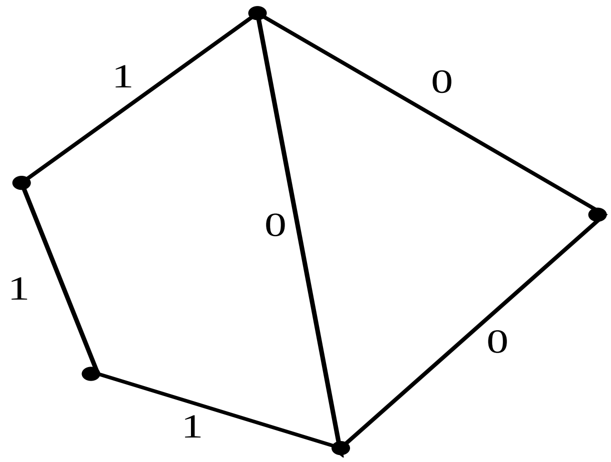

Abbildung 2: Zu verwenden in Bsp. 11