Algorithmische Mathematik:
Graphen & Anwendungen

Frühlingssemester 2018
P. Zaspel und I. Kalmykov

Universität Basel

Übungsblatt 5.

zu bearbeiten bis **Dienstag, 3.4.2018, 14:00 Uhr**.

Aufgabe 1. (Binäre Bäume)

Sei $B = (V, E)$ ein Binärbaum, d.h. B ist ein gerichteter Baum mit Wurzel $s \in V$ und $|\mathrm{post}(v)| \in \{0, 1, 2\}$ für alle $v \in V \setminus \{s\}$. Entsprechend der Vorlesung definieren wir die Tiefe eines Knotens $v \in V$ in B als

$$T_B(v) := \mathrm{dist}_B(s, v),$$

und die Höhe eines (Binär-)Baums B als

$$H_B := \max\left\{\mathrm{T}_B(v) : v \in V, v \text{ ist ein Blatt}\right\}.$$

Zusätzlich sei die mittlere Weglänge in B definiert durch

$$P_B := \frac{1}{n} \sum_{v \in V} (T_B(v) + 1),$$

wobei $n = |V|$. Beweisen Sie folgende Aussagen

a) $H_B \geq \lceil \log_2(n + 1) - 1 \rceil$,

b) $P_B \geq \lfloor \log_2(n + 1) \rfloor - 1$.

(4 Punkte)

Aufgabe 2. (Optimale aufspannende Bäume)

Sei $G = (V, E)$ ein zusammenhängender ungerichteter Graph und $\gamma \colon E \to \mathbb{R}$ eine Gewichtsfunktion. Wir ordnen einer Teilmenge der Kanten $Q \subset E$ mit

$$\gamma(Q) = \sum_{e \in Q} \gamma(e)$$

das **Gewicht** von Q zu.
Ein **aufspannender Baum** $B = (V, R)$ für G ist ein aufspannender Teilgraph von G, der ein Baum ist. Ein **minimaler** aufspannender Baum ist ein aufspannender Baum $B = (V, T)$ in G, so dass für einen beliebigen weiteren aufspannenden Baum $\hat{B} = (V, Q)$ in G gilt: $\gamma(T) \leq \gamma(Q)$. Ferner bezeichnen wir eine Kantenmenge $F \subset E$ als **gut**, wenn es einen minimalen aufspannenden Baum $B = (V, T)$ gibt mit $F \subseteq T$.
Der folgende Algorithmus 1 von Kruskal bestimmt für einen zusammenhängenden ungerichteten Graphen G einen minimalen aufspannenden Baum.

Algorithmus 1 (Kruskals Algorithmus)

Input: zusammenhängender ungerichteter Graph $G = (V, E)$,
Kantengewichte $\gamma \colon E(G) \to \mathbb{R}$

Output: minimaler aufspannender Baum $B = (V, T)$ in G.

Sortiere $E = \{e_1, \ldots, e_m\}$, so dass $\gamma(e_1) \leq \gamma(e_2) \leq \ldots \leq \gamma(e_m)$.
$T \leftarrow \emptyset$
for $i \leftarrow 1$ to m **do**
  if $(V, T \cup \{e_i\})$ ist kreisfrei **then**
    $T \leftarrow T \cup \{e_i\}$
  end if
end for
return $B = (V, T)$

Zeigen Sie die partielle Korrektheit von Algorithmus 1. Gehen Sie dabei wie folgt vor:

a) Sei $F \subset E$ gut, $e \in E \setminus F$. Zeigen Sie:

$$\exists X \subset V \colon \delta_G(X) \cap F = \emptyset,\ e \in \delta_G(X) \text{ und } \forall f \in \delta_G(X) \colon \gamma(e) \leq \gamma(f) \\ \Longrightarrow F \cup \{e\} \text{ ist gut.} \tag{$\star$}$$

Dabei bezeichnet δ_G analog zum Blatt 4 die Menge der von X ausgehenden Kanten in G:

$$\delta_G(X) := \{(x, w) \in E : x \in X, w \in V \setminus X\}.$$

b) Benutzen Sie Teil a) um die partielle Korrektheit von Algorithmus 1 zu zeigen.

Bemerkung. Die minimalen aufspannenden Bäume besitzen zahlreiche Anwendungen. Ein Beispiel ist eine Telefongesellschaft, die ein Teil eines vorhandenen Kabelnetzes mieten möchte, wobei jedes Kabel zwei Städte miteinander verbindet. Dabei sollen die Kosten minimal gehalten werden. Wir können die Städte als Knoten (V) und die Kabelverbindungen als Kanten (E) in einem Graph $G = (V, E)$ darstellen, und die Mietkosten als Gewichtsfunktion γ interpretieren. Der mit dem Kruskals Algorithmus berechnete minimale aufspannende Baum in G liefert ein Kabelnetz mit minimalen Mietkosten.

(4 Punkte)