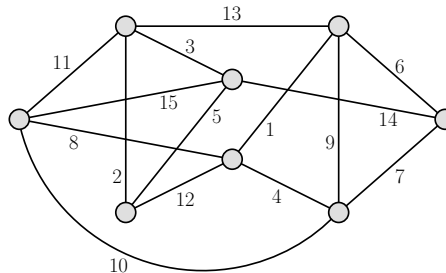


Übungsblatt 11

Aufgabe 11.1:

(4+2 Punkte)

- (a) Bestimmen Sie mithilfe des Algorithmus von Kruskal einen minimalen Spannbaum des folgenden Graphen.



- (b) Einige Probleme lassen sich mithilfe einer Divide-and-Conquer-Strategie lösen. Wir wollen diesen Ansatz für die Berechnung von minimalen Spannäumen untersuchen.

Gegeben seien ein zusammenhängender ungerichteter Graph $G = (V, E)$ mit einer Kantengewichtung $w: E \rightarrow \mathbb{R}_{>0}$ und eine Bipartition (V_1, V_2) der Knotenmenge V . Es seien E_1 und E_2 die Mengen der Kanten $\{u, v\}$ mit $u, v \in V_1$ bzw. $u, v \in V_2$. Wir nehmen an, dass die Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ zusammenhängend sind. Ein Divide-and-Conquer-Ansatz könnte wie folgt aussehen.

1. Bestimme minimale Spannäume von G_1 und G_2 .
2. Bestimme eine kostenminimale Kante zwischen Knotenmenge V_1 und Knotenmenge V_2 .
3. Vereinige die beiden Spannäume und die Kante zu einem Spannbaum von G .

Geben Sie ein Beispiel für G , V_1 und V_2 an, bei dem dieser Ansatz keinen minimalen Spannbaum liefert.

Aufgabe 11.2:

(2+4+2 Punkte)

Gegeben sei ein zusammenhängender ungerichteter Graph $G = (V, E)$ mit einer Kantengewichtung $w: E \rightarrow \mathbb{R}_{>0}$. Wir wollen in Zeit $O(|E| \log |E|)$ einen **maximalen** Spannbaum von G bestimmen, d. h. einen Spannbaum T mit maximalem Gewicht $w(T)$. Dazu wollen wir aus w eine Kantengewichtung w' konstruieren und den Algorithmus von Kruskal auf die Eingabe (G, w') anwenden. Damit diese Vorgehensweise korrekt ist, muss gelten:

$$T \text{ ist minimaler Spannbaum bzgl. } w' \implies T \text{ ist maximaler Spannbaum bzgl. } w \quad (1)$$

- (a) Wir betrachten die folgende Eigenschaft:

$$\text{Für alle Spannäume } T_1 \text{ und } T_2 \text{ gilt } w'(T_1) \leq w'(T_2) \implies w(T_1) \geq w(T_2) \quad (2)$$

Zeigen Sie, dass Eigenschaft (1) aus Eigenschaft (2) folgt.

- (b) Untersuchen Sie für $w'(e) = \frac{1}{w(e)}$ und für $w'(e) = W - w(e)$ mit $W = 1 + \max\{w(e') : e' \in E\}$, ob Eigenschaft (2) gilt. Beweisen Sie Ihre Vermutung.

Hinweis: Alle Spannäume von G haben genau $|V| - 1$ Kanten.

- (c) Wir betrachten diejenige Funktion w' aus Aufgabenteil (b), für die Eigenschaft (2) nicht erfüllt ist. Zeigen Sie, dass dennoch Eigenschaft (1) erfüllt ist. Gehen Sie der Einfachheit halber davon aus, dass der minimale Spannbaum bzgl. w' eindeutig ist.

Aufgabe 11.3:

(4 Punkte)

Seien $G = (V, E)$ ein zusammenhängender ungerichteter Graph mit einer Kantengewichtung $w: E \rightarrow \mathbb{R}_{>0}$, C ein Kreis in G und e eine teuerste Kante von C . Zeigen Sie, dass es einen minimalen Spannbaum von G gibt, der die Kante e nicht enthält.

Hinweis: Nutzen Sie Ideen aus dem Korrektheitsbeweis des Algorithmus von Kruskal.

Aufgabe 11.4:

(6 Punkte)

Implementieren Sie den Algorithmus von Kruskal. Wenden Sie Ihren Algorithmus auf die Instanzen

- (a) `Ue11_small.txt` mit $|V| = 100$,
- (b) `Ue11_medium.txt` mit $|V| = 500$ und
- (c) `Ue11_large.txt` mit $|V| = 1000$

an, die auf der Homepage heruntergeladen werden können. Das Dateiformat ist dort ebenfalls erklärt. Geben Sie für jede der Instanzen an, ob sie zusammenhängend ist. Falls das der Fall ist, dann bestimmen Sie einen minimalen Spannbaum T und geben Sie dessen Gewicht an. Geben Sie außerdem für jeden Knotengrad im Graphen $G_T = (V, T)$, der mindestens einmal vorkommt, an, wie viele Knoten in G_T diesen Grad besitzen.

Aufgabe 11.5:

(6 Zusatzpunkte)

Wir betrachten den folgenden Algorithmus, der als Eingabe einen nichtleeren ungerichteten Graphen $G = (V, E)$ und eine Kantengewichtung $w: E \rightarrow \mathbb{R}_{>0}$ erhält.

PRIM(G, w)

1. **if** (G ist nicht zusammenhängend) **then return** „ G ist nicht zusammenhängend.“
2. Wähle einen Knoten $v_0 \in V$.
3. Setze $V_T = \{v_0\}$ und $T = \emptyset$.
4. **for** $i = 1, \dots, |V| - 1$ **do**
5. Bestimme eine gewichtsminimale Kante $e_i = \{u_i, v_i\}$ mit $u_i \in V_T$ und $v_i \notin V_T$.
6. Setze $V_T = V_T \cup \{v_i\}$ und $T = T \cup \{e_i\}$.
7. **end for**
8. **return** T

Zeigen Sie, dass der Algorithmus **PRIM** einen minimalen Spannbaum von G bezüglich der Gewichtsfunktion w berechnet, sofern G zusammenhängend ist.