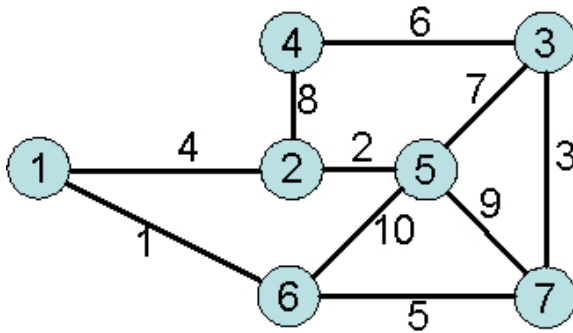


6. Übungsblatt

Ausgabe: 25.11.2005 **Abgabe:** 02.12.2005, 10 Uhr
Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Aufgabe 17:

9 Punkte



Finden Sie zum oben abgebildeten Graphen G einen aufspannenden Baum minimalen Gewichts. Benutzen Sie:

- (a) die Färbungsmethode von Tarjan,
- (b) den Algorithmus von Kruskal,
- (c) den Algorithmus von Prim (mit hellgrünen Kanten).

Geben sie jeweils bei jedem Schritt die zu färbende Kante in der Form

(Knoten k_1 , Knoten k_2), die Farbe und die neuen roten, grünen bzw. hellgrünen Mengen an.

Aufgabe 18:**6 Punkte**

- (a) Zeigen Sie: Wenn in einem Graphen G mit reellen Kantengewichten für jeden Schnitt die den Schnitt kreuzende Kante minimalen Gewichts eindeutig ist, dann hat G einen eindeutigen aufspannenden Baum minimalen Gewichts.
- (b) Gilt die Umkehrung von (a)?
- (c) Zeigen oder widerlegen Sie: Wenn in einem Graphen G mit reellen Kantengewichten alle Kanten paarweise verschiedene Gewichte haben, dann hat G einen eindeutigen aufspannenden Baum minimalen Gewichts.
- (d) Gilt die Umkehrung von (c)?

Aufgabe 19:**5 Punkte**

Gegeben sei ein vollständiger Graph $G = (V, E)$ mit reellen positiven Kantengewichten $c: E \rightarrow \mathbb{R}_{>0}$, die die Dreiecksungleichung erfüllen, d. h. für je drei Knoten $u, v, w \in V$ gilt $c(\{u, w\}) \leq c(\{u, v\}) + c(\{v, w\})$.

Eine *Rundreise* in G ist ein Kreis $C = v_1, \dots, v_n, v_{n+1} = v_1$ der alle Knoten aus V enthält. Die *Länge* $l(C)$ einer Rundreise C ist die Summe der Gewichte aller n Kanten in C .

- (a) Zeigen Sie, dass das Gewicht eines MST eine untere Schranke für die Länge jeder Rundreise ist.
- (b) Gegeben sei ein MST T für G . Geben Sie einen effizienten (d. h. mit polynomieller Laufzeit) Algorithmus an, der ausgehend von T eine Rundreise C konstruiert, so dass $l(C) \leq 2 \cdot c(T)$ gilt. (Hier ist kein Pseudocode nötig - eine Beschreibung genügt.)

Tipp: Konstruieren Sie zuerst einen geschlossenen Weg P der jede Kante von T genau zweimal, keine weiteren Kanten und jeden Knoten aus V mindestens einmal enthält. Wandeln Sie P Schritt für Schritt in die gewünschte Rundreise C um indem Sie doppelt vorkommende Knoten umgehen. Beweisen Sie dass $l(C) \leq 2 \cdot c(T)$ gilt.

- (c) Geben Sie einen effizienten Algorithmus an (wiederum kein Pseudocode), der eine Rundreise für G berechnet, deren Länge höchstens doppelt so groß ist wie die Länge einer kürzesten Rundreise für G .

Tipp: Verwenden Sie (a) und (b).