Petra Mutzel
Nicola Beume, Christian Bockermann, Christian Horoba,
Ingo Schulz, Dirk Sudholt, Christine Zarges

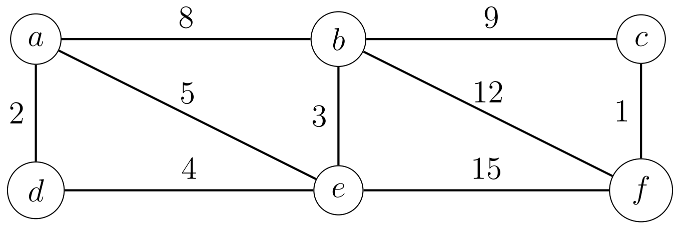
Sommersemester 2009

DAP2 Übung – Blatt 12

Ausgabe: 2. Juli, **Abgabe:** 9. Juli, 14:00 Uhr, **Block:** D

Aufgabe 12.1 (4 Punkte)

(a) Markiere die Kanten des minimalen Spannbaums, den der Algorithmus von Kruskal berechnet und gib dessen Gewicht an.



(b) Erstelle für den Ablauf des Algorithmus von Kruskal auf dem Graphen von (a) eine Tabelle mit folgenden Spalten und fülle sie für jede Iteration der While-Schleife (Skript, Listing 6.8) aus: (1) Zähler i, (2) aktuelle Kante, (3) Find-Aufrufe und Ergebnisse (Bsp. Find(p)=q), (4) Union-Aufruf, (5) Partitionen des Graphen nach der Union-Operation.

(c) Betrachte die durch gewurzelte Bäume realisierte UnionFind-Datenstruktur mit gewichteter Vereinigungsregel und Pfadkomprimierung. Bei Union-Operationen soll bei Bäumen gleicher Höhe der Baum mit dem alphabetisch größeren Repräsentanten in den Baum mit alphabetisch kleinerem Repräsentanten eingehängt werden. Zeichne für die Schritte in (b) nach jeder Union-Operation sowie nach jeder Pfadkomprimierung die Bäume.

Aufgabe 12.2 (4 Punkte)

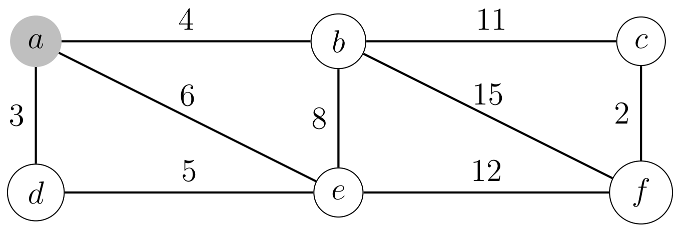
Sei $G = (V, E)$ ein zusammenhängender Graph mit Gewichtsfunktion $w : E \to \mathbb{R}$ und T ein minimaler Spannbaum für G. Das Gewicht einer teuersten Kanten auf dem eindeutigen Weg zwischen den Knoten u und v in T bezeichnen wir mit $w_{\max}(u, v)$. Beschreibe in Worten, was die folgenden Aussagen bedeuten, und beweise sie durch einen Widerspruchsbeweis.

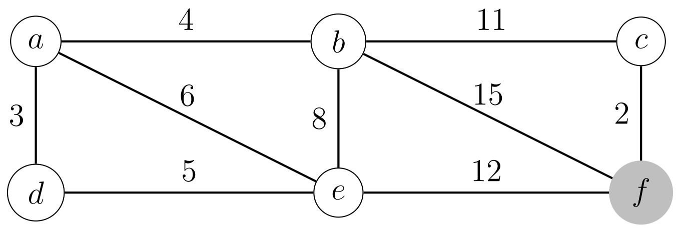
(a) T ist eindeutig $\quad\Rightarrow\quad \forall (u,v) \in (E \setminus T): \ w(u,v) > w_{\max}(u,v)$,

(b) $\forall (u,v) \in (E \setminus T): \ w(u,v) > w_{\max}(u,v) \quad\Rightarrow\quad T$ ist eindeutig.

Aufgabe 12.3 (4 Punkte)

(a) Nummeriere in den folgenden Graphen die Kanten in der Reihenfolge, in der sie vom Algorithmus von Prim in einen minimalen Spannbaum aufgenommen werden. Der markierte Knoten ist dabei jeweils der Startknoten. Gib das Gewicht der berechneten Spannbäume an.





(b) Erstelle für den Ablauf des Algorithmus von Prim auf dem ersten Graphen von (a) eine Tabelle mit folgenden Spalten und fülle sie für jede Iteration der While-Schleife (Skript, Listing 6.9) aus: (1) Nummer der Iteration, (2) Rückgabe von ExtractMin(), (3) PriorityQueue Q, (4) Feld π, (5) Knoten in Untergraph S, der zum Spannbaum wird.

Präsenzaufgabe 12.4

Entscheide für folgende Algorithmen, ob sie einen minimalen Spannbaum berechnen oder nicht. Gib jeweils eine Beweisidee oder ein Gegenbeispiel an. Wie können die Algorithmen implementiert werden und welche Laufzeit haben sie?

Algorithmus 1 Vielleicht-MST-1

procedure VIELLEICHT-MST-1(G, w)
    $T \leftarrow \emptyset$
    for jede Kante e (in beliebiger Reihenfolge) **do**
        if $(T \cup \{e\})$ enthält keinen Kreis **then**
            $T \leftarrow (T \cup \{e\})$
        end if
    end for
    return T
end procedure

Algorithmus 2 Vielleicht-MST-2

procedure VIELLEICHT-MST-2(G, w)
    Sortiere die Kanten absteigend nach Gewicht
    $T \leftarrow E$
    for jede Kante e (in sortierter Reihenfolge) **do**
        if $(T \setminus \{e\})$ ist ein zusammenhängender Graph **then**
            $T \leftarrow (T \setminus \{e\})$
        end if
    end for
    return T
end procedure