Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

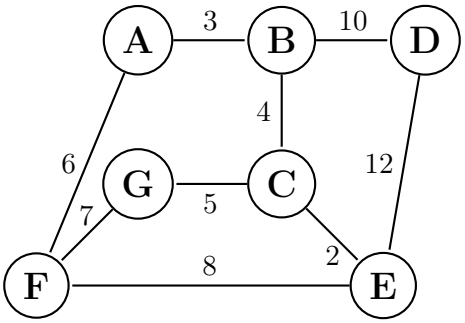
Institut für Theoretische Informatik
Peter Widmayer
Thomas Tschager
Antonis Thomas

27. April 2016

Datenstrukturen & Algorithmen Blatt 9 FS 16

Aufgabe 9.1 *Minimale Spannbäume.*

Gegeben sei folgender Graph $G = (V, E)$.



a) Zeichnen Sie einen minimalen Spannbaum, den der Algorithmus von Kruskal berechnet.

b) Ein Zyklus in einem Graphen ist ein Weg, der an seinem Ausgangsknoten endet. Ein Graph ist *zyklenfrei*, wenn er keinen Zyklus enthält. Zeigen Sie: Jeder ungerichtete zyklenfreier Graph $G = (V, E)$ hat höchstens $|V| - 1$ Kanten.

c) Zeigen Sie: Kommt in einem Graphen kein Kantengewicht doppelt vor, dann ist der minimale Spannbaum eindeutig bestimmt.

Aufgabe 9.2 *Union-Find Strukturen.*

In Union-Find Strukturen wird eine Menge durch einen Baum repräsentiert. Wir betrachten im Folgenden das Verfahren der "Vereinigung nach Grösse" (siehe Kapitel 6.2.2). Es soll ein Baum der Höhe $h \in \mathbb{N}$ erzeugt werden. Beschreiben Sie, wie solch ein Baum mit einer Folge von UNION-Operationen generiert werden kann. Wie viele UNION-Operationen sind dazu mindestens nötig, und wie viele Knoten hat der entstehende Baum?

Aufgabe 9.3 *Längste aufsteigende Teilfolge.*

Der folgende Java-Code enthält eine unvollständige Implementierung eines Dynamischen Programms zur Berechnung der Länge einer längsten aufsteigenden Teilfolge. Gegeben sei eine Zahlenfolge $A = A[0], \ldots, A[n-1]$ mit paarweise verschiedenen Zahlen $A[i] \in \mathbb{N}^+$ für $i = 0, \ldots, n-1$. Das Programm berechnet eine eindimensionale Tabelle T mit $n+1$ Einträgen. Der Eintrag $T[i]$ für $i > 0$ enthält das kleinste Element $A[j]$ für $j \in \{0, \ldots, n-1\}$, sodass $A[j]$ am Ende einer aufsteigenden Teilfolge der Länge i steht. Anfangs wird $T[0]$ mit $-\infty$ und alle anderen Einträge der Tabelle mit ∞ initialisiert. Im i-ten Berechnungsschritt wird der grösste Index $j < i$ mit $T[j] < A[i]$ gesucht. Da die Einträge in T zu jedem Zeitpunkt eine monotone Folge bilden, kann dieser Eintrag effizient durch eine binäre Suche gefunden werden. Falls kein solcher Index existiert, setzen wir $j = 0$. Schliesslich setzen wir $T[j+1] = \min(T[j+1], A[i])$. Nach n Berechnungsschritten entspricht die Länge der längsten aufsteigenden Teilfolge dem grössten Index k

mit $T[k] \neq \infty$.

Vervollständigen Sie den untenstehenden Code (Zeilen 4, 5, 17 und 18).

```
1  static int binarySearch(int A[], int l, int r, int key) {
2      while (l < r) {
3          int m = l + (r - l + 1)/2;
4          if (A[m] >= key) ____________;
5          else ____________;
6      }
7      return l;
8  }
9
10 static int computeTable(int A[], int size) {
11     int[] T = new int[size+1];       // DP-Tableau
12     for (int i = 1; i <= size; i++)  // Initialisierung
13         T[i] = Integer.MAX_VALUE;
14     T[0] = Integer.MIN_VALUE;
15     int l = 0;
16
17     for (int i = _____; i _______; i = _____) {
18         int j = binarySearch(____, ____, ____, ____);
19         if ( A[i] < T[j+1] ) T[j+1] = A[i];
20         if (l < j+1) l = j+1;
21     }
22     return l;
23 }
```

Aufgabe 9.4 *Minimale Spannbäume (**Programmieraufgabe**).*

In dieser Aufgabe soll Kruskals Algorithmus zur Berechnung eines *minimalen Spannbaums* implementiert werden. Das Problem ist wie folgt definiert: Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit der Kostenfunktion $c : E \to \mathbb{Q}^+$, und gesucht wird eine zyklenfreie zusammenhängende Teilmenge $T \subseteq E$ mit $|T| = |V| - 1$ (d.h. ein Baum (V, T)), dessen Kosten $\sum_{e \in T} c(e)$ minimal unter allen möglichen Bäumen sind.

Eingabe Die erste Zeile der Eingabe enthält lediglich die Anzahl t der Testinstanzen. Danach folgt genau eine Zeile pro Testinstanz. Sie enthält die Beschreibung des Graphs $G = (V, E)$ im Format $n, m, u_1, v_1, c_1, \ldots, u_m, v_m, c_m$. Es sind $1 \leq n, m \leq 10000$ mit $V = \{1, \ldots, n\}$ und $|E| = m$. Für alle i, $1 \leq i \leq m$, definieren $u_i, v_i \in \{1, \ldots, n\}$ die Kante $\{u_i, v_i\} \in E$ mit einem Kantengewicht von c_i, $1 \leq c_i \leq 1000$.

Ausgabe Für jede Testinstanz soll lediglich eine Zeile ausgegeben werden. Sie enthält die Kosten eines minimalen Spannbaums.

Beispiel

Eingabe:

```
1
10 15 1 2 1 2 3 2 1 4 7 2 5 2 3 6 3 3 7 5 4 5 4 5 6 3 6 7 2 4 8 1 5 9 3 6 9 4 7 10 6 8 9 5 9 10 3
```

Ausgabe:

```
21
```

Hinweis Wir stellen eine Code-Vorlage zur Verfügung, in der das Einlesen der Eingabe bereits implementiert wurde. Implementieren Sie die Klasse `UnionFind` und den Algorithmus von Kruskal. Für diese Aufgabe gibt es lediglich ein Testset.

Abgabe: Am Mittwoch, den 4. Mai 2016 in Ihrer Übungsgruppe.