Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Komplexität

186.813 Algorithmen und Datenstrukturen 1 VU 6.0

Übungsblatt 3

für die Übung am Montag den 25. bzw. Dienstag den 26. April 2016.

Geben Sie bis **spätestens Sonntag, 24.4.2016, 23:59 Uhr** über TUWEL an, welche Beispiele Sie bearbeitet und gelöst haben. Gehen Sie dabei folgendermaßen vor:

- TUWEL (`https://tuwel.tuwien.ac.at`)
  Kurs *186.813 Algorithmen und Datenstrukturen 1 (VU 6.0)*
- Übungsblätter
- Bearbeitete Beispiele ankreuzen **und** abgeben
  - Link *Übungsblatt 3 - Details & Bewertung*
    Button *Abgabe bearbeiten*
    Bearbeitete Beispiele anhaken und *Änderungen speichern.*
  - Link *Hochladen Lösungen Übungsblatt 3*
    Button *Abgabe hinzufügen*
    PDF-Datei mit Lösungen hochladen und *Änderungen sichern.*

Bitte beachten Sie:

- Sie können **vor** der Deadline beliebig oft ihre Auswahl an Beispielen und das zugehörige Lösungs-PDF verändern, aber **nach** der Deadline gibt es **keine** Veränderung ihrer angekreuzten Beispiele **und** der PDF-Datei!
- Sie können Ihre Lösungen entweder direkt in einem Textverarbeitungsprogramm erstellen, oder aber auch gut leserliche Scans bzw. Fotos von handschriftlichen Ausarbeitungen einreichen.
- Bitte geben Sie Ihren Namen, Matrikelnummer und E-Mail-Adresse in den Ausarbeitungen an.
- Wenn Sie zur Präsentation Ihrer Lösung eines von Ihnen angekreuzten Beispiels ausgewählt werden und dieses aber nicht bearbeitet haben oder dieses Beispiel nicht in der PDF-Datei vorhanden ist, verlieren Sie **alle** Punkte dieser Übungseinheit!

  (Zusätzlich werden stichprobenartig die abgegebenen PDF-Dateien auf Übereinstimmung mit der entsprechenden Kreuzerlliste überprüft.)

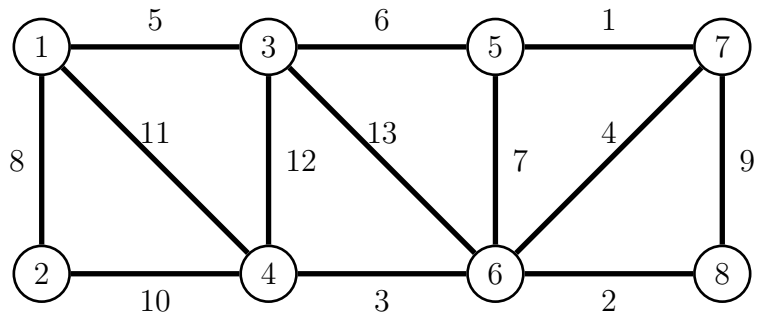
Aufgabe 17 Gegeben sind n Programme $P_1, \ldots, P_n$, welche auf einer Festplatte mit einer Kapazität von D Megabyte gespeichert werden sollen. Der Platz, der für ein Programm P_i benötigt wird, ist s_i Megabyte. Nehmen Sie an, dass es nicht möglich ist, alle Programme gleichzeitig auf der Festplatte zu speichern ($D < \sum_{i=1}^{n} s_i$).

(a) Maximiert ein Greedy-Algorithmus, der die Programme in aufsteigender, nach Größe geordneter, Reihenfolge auswählt, die Anzahl der zu speichernden Programme? Geben Sie einen Beweis oder ein Gegenbeispiel an.

(b) Maximiert ein Greedy-Algorithmus, der die Programme in absteigender, nach Grösse geordneter, Reihenfolge auswählt, die Auslastung der Festplatte? Geben Sie einen Beweis oder ein Gegenbeispiel an.

Aufgabe 18 Ist der Pfad zwischen zwei Knoten in einem minimalen Spannbaum notwendigerweise ein kürzester Pfad zwischen denselben Knoten im ursprünglichen Graphen? Geben Sie einen Beweis oder ein Gegenbeispiel an.

Aufgabe 19 Gegeben sei der folgende ungerichtete Graph mit Kantengewichten.



Führen Sie den Algorithmus von Kruskal auf diesem Graphen aus. Was ist der resultierende minimale Spannbaum? In welcher Reihenfolge werden die Kanten ausgewählt?

Aufgabe 20 Führen Sie den Algorithmus von Prim auf dem Graphen aus Aufgabe 19 beginnend mit dem Knoten 2 aus. Was ist der resultierende minimale Spannbaum? In welcher Reihenfolge werden die Kanten ausgewählt?

Aufgabe 21 Nehmen Sie an, man will n Zahlen entweder in einem Max-Heap oder in einem sortierten Array speichern. Geben Sie für jeden der vier folgenden Anwendungsfälle an, welche der beiden Datenstrukturen besser geeignet ist bzw. ob beide gleich gut geeignet sind. Erklären Sie ihre Antworten.
Hinweis: Bei einem Max-Heap sind die Werte im linken und rechten Kindknoten immer kleiner oder gleich dem Wert im Vorgängerknoten und das Maximum befindet sich im Wurzelknoten.

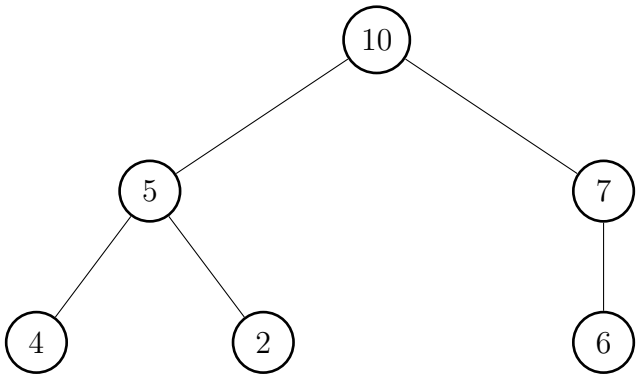
(a) Schnelles Finden eines maximalen Elements.

(b) Schnelles Löschen eines gegebenen Elements.

(c) Schnelles Aufbauen der Datenstruktur aus einer Menge unsortierter Elemente.

(d) Schnelles Finden eines minimalen Elements.

Aufgabe 22 Gegeben sei der folgende Max-Heap:



Hinweis: Bei einem Max-Heap sind die Werte im linken und rechten Kindknoten immer kleiner oder gleich dem Wert im Vorgängerknoten und das Maximum befindet sich im Wurzelknoten. Daher müssen auch die Heap-Operationen entsprechend angepasst werden.

(a) Fügen Sie das Element 8 in den Heap ein. Geben Sie die einzelnen Schritte und den resultierenden Heap an.

(b) Löschen Sie danach das größte Element (10) sowie auch 2 in dieser Reihenfolge aus dem resultierendem Heap von (a). Geben Sie wiederum die einzelnen Schritte und den neuen Heap an.

Aufgabe 23 Sie haben in der Vorlesung den Algorithmus von Dijkstra kennengelernt. Dieser Algorithmus ist nicht für negative Kantengewichte ausgelegt worden. Lösen Sie dazu folgende Aufgaben:

- Geben Sie einen kantengewichteten Graphen (sowie einen Startknoten) an, der mindestens eine Kante mit negativem Gewicht enthält, sodass Dijkstras Algorithmus die richtige Lösung ausgibt.
- Geben Sie einen kantengewichteten Graphen (sowie einen Startknoten) an, der mindestens eine Kante mit negativem Gewicht enthält, sodass Dijkstras Algorithmus eine falsche Lösung ausgibt.

Aufgabe 24 Führen Sie Mergesort auf dem Array $[51, 13, 28, 4, 60, 9]$ aus und geben Sie den Ablauf des Aufrufes in Form eines Ablaufdiagrammes entsprechend der Folie 10 im Foliensatz über Divide-and-Conquer an.
