Prof. Dr. Leif Kobbelt
Thomas Ströder, Fabian Emmes
Sven Middelberg, Michael Kremer

Informatik 8
RWTH Aachen

Datenstrukturen und Algorithmen (SS 2013)

Übungsblatt 3

Abgabe: Montag, **06.05.2013**, 14:00 Uhr

- Die Übungen sollen in Gruppen von zwei bis drei Personen bearbeitet werden.
- Schreiben Sie die Namen jedes Gruppenmitglieds sowie alle Matrikelnummern auf die abgegebenen Lösungen.
- Schreiben Sie die Namen jedes Gruppenmitglieds sowie alle Matrikelnummern auch in die Quellcode-Dateien.
- Geben Sie Ihre Lösungen am **Anfang** der Globalübung, montags, 14:00 Uhr, ab.
- Schicken Sie den jeweiligen Quellcode bitte per **E-Mail** direkt an Ihre/n Tutor/in.
- Geben Sie außerdem den ausgedruckten Quellcode zusammen mit den schriftlichen Lösungen ab.
- Zu spät abgegebene Lösungen werden nicht bewertet.
- Sofern nicht anders gefordert, müssen alle Lösungen und Zwischenschritte kommentiert werden.

Prof. Dr. Leif Kobbelt
Thomas Ströder, Fabian Emmes
Sven Middelberg, Michael Kremer

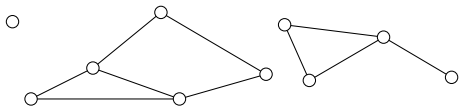
Informatik 8
RWTH Aachen

Aufgabe 1 (Graphen [**10 Punkte**])

1. Stellen Sie die folgende Adjazenzliste als Adjazenzmatrix A sowie als Diagramm dar. Berechnen Sie $(A+I)^2$ und $(A+I)^3$ wobei I die Einheitsmatrix ist. [4 Punkte]

   1: 3, 4
   2: 3
   3: 1, 2
   4: 1, 5
   5: 4

2. Gegeben sei ein ungerichteter, planarer Graph mit v Knoten, e Kanten, f Facetten und c Zusammenhangskomponenten. Die Zahl der Facetten eines planaren Graphen ist die Zahl der von Kanten begrenzten Gebiete in einer beliebigen planaren Darstellung des Graphen, die keine anderen Kanten enthalten. Das außerhalb des Graphen liegende Gebiet zählt hierbei nicht als Facette. Für den untenstehenden Graph ist z. B. $v = 10$, $e = 10$, $f = 3$, $c = 3$.

   Für solche Graphen gilt die Euler-Formel

   $$v - e + f = c.$$

   Diese Formel soll im Folgenden hergeleitet werden.

   (a) Zeigen Sie zunächst: Die Euler-Formel gilt für Bäume (hier: Baum = zusammenhängender, ungerichteter, zyklenfreier Graph). [2 Punkte]

   (b) Beweisen Sie die allgemeine Euler-Formel für zusammenhängende Graphen und $e \geq 1$ durch Induktion über e. Verwenden sie im Induktionsschritt Teilaufgabe (a). [2 Punkte]

   (c) Beweisen Sie die Behauptung unter Verwendung von Teil (b) für nichtzusammenhängende Graphen. [2 Punkte]

Aufgabe 2 (Programmierung: Heap [**10 Punkte**])

1. Implementieren Sie die Datenstruktur Heap (Warteschlange, die die Heap-Bedingung erfüllt) in Java. Verwenden Sie dazu die effiziente Implementierung auf einem Array (Vorlesung 1.4, Folien 11 ff.) Die Implementierung soll insbesondere die Funktionen `Enq` zum Hinzufügen eines Elements zum Heap, `Deq` zum Entfernen des obersten Elements und `Get` zum Zugriff auf das erste Element enthalten. In dem bereitgestellten Code ist die Warteschlange bereits als Integer-Array

   ```
   private int[] S;
   ```

   als Member-Variable der Klasse `Heap` deklariert. Diese wird im Konstruktor mit der festen Größe von 10 Elementen angelegt:

   ```
   this.S = new int[10];
   ```

   Vervollständigen Sie die im Code deklarierten Methoden `Enq`, `Deq`, sowie `Get`. Die zu bearbeitenden Stellen sind bereits im Code markiert. Sie dürfen alle existierenden Methoden in der Klasse `Heap` benutzen. [8 Punkte]

2. In der von uns bereitgestellten `main()`-Methode wird eine Instanz der Klasse `Heap` erzeugt und nacheinander einige Testwerte der Warteschlange hinzugefügt und wieder herausgenommen.

   Vervollständigen Sie die Funktion `printQueue` in der Klasse `Heap` so, dass sie die Prioritätsschlange mit Elementen x_0 bis x_{n-1} in der folgenden Form ausgibt:

   $$[x_0, x_1, x_2, \ldots, x_{n-1}]$$

   Beim Ausführen der `main()`-Methode erscheint nun jeweils das Zwischenergebnis jeder Operation. Verifizieren Sie, dass Ihre Implementierung stets das richtige Ergebnis produziert und die ausgegebene Prioritätsschlange zu jeder Zeit die Heap-Bedingung erfüllt. Führen Sie hierzu die von uns bereitgestellte `main()`-Methode aus. Editieren Sie die Klasse `MainClass` *nicht*! Sie brauchen zu diesem Aufgabenteil nichts aufschreiben. Drucken Sie lediglich die produzierte Ausgabe aus und geben Sie den Ausdruck zusammen mit dem Ausdruck des Quellcodes der Klasse `Heap` ab. [2 Punkte]