Prof. G. Zachmann
David Mainzer

TU Clausthal
Institut für Informatik
12. Mai 2010

Sommersemester 2010

Übungen zu Informatik II - Blatt 6

Abgabe in der Übung am 18. 05 / 19. 05. 2010

Bitte beachten Sie, dass die Programmieraufgaben von Ihnen in der Übung vorgeführt und erklärt werden müssen. Zusätzlich senden Sie die Lösung, unter Angabe ihres Namens, an **dm@tu-clausthal.de**.

Aufgabe 1 (Heaps als Priority-Queue, *2+1+1+1.5 Punkte*)

Aus Informatik I kennen sie Queues. Eine p-Queue ist eine Queue, in der beim Herausnehmen eines Elementes aus der Queue immer das Element mit höchstem Wert gewählt wird (= Operation extract-max). Mithilfe eines Heaps lässt sich solch eine Datenstruktur effizient implementieren. Beim Einfügen eines Elementes in den Heap muss ein `Upheap()` durchgeführt werden. Beim Herausnehmen eines Elementes ein `Downheap()`. Laden Sie sich das Programm *heaps2.py* von der Vorlesungshomepage herunter. Ergänzen Sie das Programm um:

a) Die Routine `Upheap()`, welche ein in den Heap eingefügtes Element an die richtige Position innerhalb des Heaps bringt.
b) Die Routine `put(v)`, welche ein Element in die p-Queue einfügt.
c) Die Routine `get()`, welche ein Element aus der p-Queue heraus nimmt.
d) Geben Sie den Aufwand für die Routinen `put(v)` und `get()` an. Wie hoch wäre der Aufwand wenn anstelle der p-Queue ein unsortiertes Array bzw. ein sortiertes Array verwendet wird. Verwenden Sie für Ihre Aufwandsangaben die $\mathcal{O}$-Notation.

Bitte testen Sie Ihre Implementierung!

Aufgabe 2 (Merge-Schritte, *2 Punkte*)

Wir betrachten den Merge-Schritt bei Mergesort. Gegeben seien zwei sortierte Teilfolgen:

$$x_1 \leq x_2 \leq \ldots \leq x_{n/2} \text{ und } y_1 \leq y_2 \leq \ldots \leq y_{n/2}.$$

Diese werden zu einer sortierten Folge $z_1 \leq z_2 \leq \ldots \leq z_n$ gemergt.

Geben Sie einen Fall an, wo der Merge-Schritt genau $\frac{n}{2}$ Vergleiche benötigt, und einen Fall, wo genau $n-1$ Vergleiche benötigt werden.

Aufgabe 3 (Sortieren mit Mergesort, *1 Punkte*)

Sortieren Sie die Zahlenfolge 5,4,9,7,1,3,0,8,4 mit Mergesort.

Geben Sie die Zahlenfolgen nach jedem Rekursionsschritt an.

Aufgabe 4 (Listenaufteilung, *1 Punkte*)

Trainer K möchte das durchschnittliche Niveau seiner Mannschaft, die aus $2n$ Spielern besteht, erhöhen. Dazu teilt er die $2n$ Spieler in 2 Teams zu je n ein, die gegeneinander spielen sollen, wobei er die Aufteilung so *unfair* wie möglich machen möchte, d.h., er möchte die Teams so *unbalanciert* bzgl. Fähigkeiten machen wie möglich.

Jeder Spieler hat einen definierten "Fähigkeits-Score", gegeben in Form einer Float-Zahl auf der nach oben offenen "Kaiser-Skala". Zeigen Sie, wie Trainer K die beiden Teams in Zeit $\mathcal{O}(n \log n)$ zusammenstellen kann.

Aufgabe 5 (Summe zweier Elemente, *4 Punkte*)

Gegeben seien zwei Mengen A und B, jede der Größe n, und eine Zahl x. Beschreiben Sie einen Algorithmus, mit dem man in Zeit $\mathcal{O}(n \log n)$ herausfinden kann, ob zwei Elemente $a \in A$ und $b \in B$ existieren, so daß $a + b = x$.

Aufgabe 6 (Heapsort, *2+1 Punkte*)

Angenommen, man könnte einen Heap (a.k.a p-Queue) so implementieren, daß die Operationen *insert* und *extract-max* beide in der Zeit $\mathcal{O}(1)$ im worst-case ausgeführt werden können.

a) Welche Laufzeit hätte Heapsort dann?

b) Kann es solch einen Heap geben? Begründen Sie ihre Entscheidung.