

Übungen zur Vorlesung
Datenstrukturen, Algorithmen und Programmierung 2 (DAP2)
Sommersemester 2007

Blatt 10

Aufgabe 10.1 (5 Punkte)

Das Sortierverfahren **Heapsort** benötigt einen (**Min-**)Heap als Datenstruktur. Betrachte die folgenden Operationen auf einem **Minheap** der Größe n und beschreibe, wie sie realisiert werden können. Gib zusätzlich zu jeder Operation die benötigte Laufzeit in O -Notation an. Als Grundlage wähle die Definition des **Heaps** als Array aus dem Skript (Definition 4.5.1, S. 103).

- **getMin()**: liefert das kleinste Element des Heaps zurück
- **deleteMin()**: löscht das kleinste Element aus dem Heap
- **insert(x)**: fügt das Element x in den Heap ein
- **delete(i)**: löscht das Element an Position i aus dem Heap
- **getMax()**: liefert das größte Element des Heaps zurück

Aufgabe 10.2 (5 Punkte)

Betrachte **Bucketsort**, wobei die zu sortierenden Wörter die Eigenschaft haben, dass auf ein Leerzeichen nur weitere Leerzeichen folgen können. Wenn also das j -te Wort sein erstes Leerzeichen an Position $l_j + 1$ hat, ist seine Länge genau $l_j \leq l$. Können wir in diesem Fall **Bucketsort** so modifizieren, dass er wesentlich effizienter ist, wenn viele l_i kleiner als l sind?

Beschreibe den modifizierten Algorithmus und gib seine Rechenzeit bzgl. der Parameter M, n, l und $l^* := l_1 + \dots + l_n$ an.

Aufgabe 10.3 (5 Punkte)

Zeige, dass der Greedy-Algorithmus zur Lösung des Münzwechselproblems für Euros stets die optimale Lösung findet. (Dabei beschränken wir uns nur auf Münzen als Wechselgeld, zulässig als Wechselgeld sind daher also nur 1-, 2-, 5-, 10-, 20-, 50- Eurocentmünzen sowie 1- und 2- Euromünzen.)

Aufgabe 10.4 (5 Punkte)

Das folgende Rucksackproblem mit 8 Objekten und einer Rucksackgröße von $G = 16$ sei gegeben:

Objekt-Nr.	1	2	3	4	5	6	7	8
Gewicht	1	4	5	3	7	3	2	1
Nutzen	20	28	10	12	21	9	3	1

Löse das Problem mit dem Greedy-Algorithmus aus der Vorlesung, in dessen Lösung die Objekte entweder nur vollständig oder überhaupt nicht berücksichtigt werden (Skript S. 120 oben). Kannst du eine Aussage darüber treffen, inwieweit die Lösung des Algorithmus für dieses Problem schlechter ist als die optimale Lösung für dieses Problem?