

Übungen zur Vorlesung

Grundlagen der Programmierung II

Blatt 12

Aufgabe 1:

In einer zu sortierenden Folge bestehen die Elemente a_i im allgemeinen aus einer Schlüsselkomponente $a_i.key$ und einem beliebig großen Datensatz $a_i.rest$. Man ist daran interessiert, ob zwei Elemente a_i, a_j mit gleichen Schlüsseln ($a_i.key = a_j.key$) nach einer Sortierung in derselben Reihenfolge stehen wie in der Ausgangsfolge. Ein Sortierverfahren, das die Reihenfolge solcher Elemente nicht ändert, heißt *stabil*, ein Verfahren, das dies nicht garantiert *instabil*.

Erläutern Sie für die Sortierverfahren a) Bubblesort, b) Quicksort, c) Heapsort, d) Mergesort und e) Bucketsort, warum sie stabil bzw. instabil sind.

Aufgabe 2:

Die Schlüssel 1, 2, 3, ... sollen in aufsteigender Reihenfolge in einen zunächst leeren B-Baum der Ordnung 2 eingefügt werden. Die Einfügung welches Schlüssels bewirkt zum ersten Mal, daß die Blätter das Niveau 3 haben? (Die Wurzel habe Niveau 0).

Aufgabe 3:

Bestimmen Sie sechs Algorithmen, die zu N ganzen Zahlen und gegebenem $K, T \in \mathbb{N}$ berechnen, ob es unter den N Zahlen K Zahlen gibt, deren Summe höchstens T beträgt.

Die Algorithmen sollen folgende Laufzeiten besitzen:

$$O(N^K), O(N^2), O(N \cdot K), O(N \cdot \log N), O(N \cdot \log K), O(N + K \cdot \log N)$$

Aufgabe 4:

Für die folgende Schlüsselfolge

51 9 53 7 91 17 90 28 66 43 16 52 62 68 77 71

bestimme man den Ausgangsheap sowie die Heaps nach den ersten drei Schritten des Heapsort-Algorithmus. Man benutze die Darstellung durch Binärbäume.