

Übungsblatt 4

Abgabe bis zum 14. Juni 2011

Aufgabe 19

mündlich

Entscheiden Sie für folgende Sortierverfahren, ob sie stabil sind (mit Begründung):

- (a) InsertionSort
- (b) BubbleSort
- (c) MergeSort
- (d) QuickSort
- (e) HeapSort
- (f) BucketSort
- (g) CountingSort
- (h) RadixSort

Aufgabe 20

mündlich

Implementieren Sie die Floyd-Strategie in Pseudocode.

Aufgabe 21

mündlich

Bestimmen Sie für $n = 1, \dots, 10$ die Fragekomplexität von **HeapSort** im besten, schlechtesten und durchschnittlichen Fall für eine Permutation der Zahlen $1, \dots, n$. Vergleichen Sie diese Werte jeweils mit den entsprechenden Werten für **QuickSort** aus ?? (d). Welche Werte ergeben sich für die Floyd-Strategie?

Aufgabe 22

mündlich, optional

Bestimmen Sie asymptotische Schranken für die Anzahl von Vergleichen, die **HeapSort** im besten Fall vornimmt, um eine Folge von n paarweise verschiedenen Zahlen zu sortieren. Welche Schranke ergibt sich, wenn die Folgenglieder auch mehrfach vorkommen können?

Aufgabe 23

2 Punkte

Sei $k \geq 2$. Ein k -Heap H ist wie ein (binärer) Heap definiert, nur dass jeder Knoten (bis zu) k Kinder hat. Bestimmen Sie (jeweils mit Begründung):

- (a) Die zur Speicherung von H in einem Feld benötigten Indexfunktionen **parent**(i) und **child**(i, j), $j = 1, \dots, k$, wobei **child**(i, j) den Index des j -ten Kindes des Knotens mit Index i angibt. *(mündlich)*
- (b) Die Tiefe eines k -Heaps mit n Knoten. *(2 Punkte)*

Aufgabe 24

mündlich

- (a) Schätzen Sie die erwartete Laufzeit von **BucketSort** für den Fall ab, dass eine Folge von n im Intervall $[a, b)$ unabhängig gleichverteilten Zufallszahlen auf m Buckets verteilt wird und die Buckets in quadratischer Zeit sortiert werden.
- (b) Wie groß ist die erwartete Laufzeit bei Verwendung von $m = \sqrt{n}$ Buckets?

Aufgabe 25

mündlich

Beweisen Sie die Korrektheit von **CountingSort**.

Aufgabe 26

mündlich

Implementieren Sie die Prozeduren **Max**, **Min**, **Prec**, **Succ** und **Remove** für (doppelt) verkettete Listen und binäre Suchbäume. Geben Sie jeweils asymptotische Schranken für die Laufzeit in Abhängigkeit von der Listenlänge bzw. der Suchbaumtiefe an.

Aufgabe 27

8 Punkte

Implementieren Sie eine *Prioritätswarteschlange* $P = (H, s)$ mit Hilfe eines in einem Feld H gespeicherten Heaps (s gibt die aktuelle Größe von P an). P soll bis zu n Schlüsselwerte aufnehmen können. Geben Sie Pseudocode für folgende Prozeduren an.

- (a) Die Prozedur **Insert**(H, s, k) fügt ein neues Element mit dem Wert k in den Heap ein und aktualisiert den Wert von s . *(mündlich)*
- (b) Die Funktion **Max**(H) gibt den größten in P gespeicherten Schlüsselwert zurück. *(2 Punkte)*
- (c) Die Funktion **RemoveMax**(H, s) liefert das Element mit dem maximalen in P gespeicherten Schlüsselwert zurück und entfernt dieses Element aus H . Zudem wird s aktualisiert und die Heap-Eigenschaft wieder hergestellt. *(2 Punkte)*
- (d) Die Prozedur **IncreaseKey**(H, i, k) erhöht den Wert von $H[i]$ auf das Maximum von $H[i]$ und k und stellt die Heap-Eigenschaft wieder her. *(2 Punkte)*
- (e) Die Prozedur **ChangeKey**(H, s, i, k) setzt den Wert von $H[i]$ auf k und stellt die Heap-Eigenschaft wieder her. *(2 Punkte)*

Geben Sie jeweils asymptotische Schranken für die Laufzeit der Prozeduren an.