

Übungsblatt 5

Abgabe bis zum 5. Juni 2012

Aufgabe 18

5 Punkte

- (a) Wie viele Vergleiche benötigt **HeapSort**, um die Eingabefolge (4, 2, 5, 1, 3) zu sortieren? (2 Punkte)
- (b) Geben Sie den Fragebaum von **HeapSort** für die Sortierung von 4 Zahlen an. (3 Punkte)

Aufgabe 19

mündlich

Entscheiden Sie für folgende Sortiervverfahren, 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 Aufgabe 14 (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**.