

Algorithmen und Datenstrukturen

Aufgabenblatt 4

Parallel zu diesem Aufgabenblatt gibt es auch Programmieraufgaben. Die entsprechende Aufgabenstellungen sind auf Moodle zu finden und dort zu bearbeiten.

Übung 1

_____/6 P.

- (a) Ist ein aufsteigend sortiertes Array ein min-Heap oder ein max-Heap? Begründen Sie Ihre Antwort. (1 Punkt)
- (b) Zeigen Sie, dass ein Heap mit n Elementen eine Höhe $\Theta(\log n)$ hat. (1 Punkt)
- (c) Zeigen Sie, dass ein Heap der Größe n maximal $\lceil \frac{n}{2^{h+1}} \rceil$ Knoten der Höhe h hat. (2 Punkte)
- (d) Zeigen Sie, dass in einem Heap die Anzahl der inneren Knoten um eins kleiner oder gleich der Anzahl an Blättern ist. (2 Punkte)

Übung 2

_____/4 P.

Zeigen Sie, dass die Laufzeit von *BuildHeap* aus der Vorlesung in $\mathcal{O}(n)$ liegt. Sie können voraussetzen, dass die Anzahl der Knoten auf Höhe h für einen Heap der Größe n durch $\lceil \frac{n}{2^{h+1}} \rceil$ beschränkt ist. (4 Punkte).

Übung 3

_____/3 P.

Die Wichtel müssen 5 mit Geschenken vollgestopfte Säcke dem Gewicht nach auf den Schlitten des Weihnachtsmanns verteilen. Dabei steht den Wichteln eine Waage zur Verfügung, mit der sie jeweils für zwei Säcke bestimmen können, ob der erste Sack leichter als der zweite Sack ist oder nicht. Da die Säcke sehr groß sind und die Zeit knapp ist, weigern die Wichtel sich die Waage mehr als 7 mal zu benutzen. Beschreiben Sie in Pseudocode einen Algorithmus, der für die Wichtel die Säcke nach Gewicht sortiert, sodass die Waage dabei nicht mehr als 7 mal benutzt werden muss. (3 Punkte)

Hinweis: Der Pseudocode darf beliebig viele Vergleichsoperationen enthalten, allerdings dürfen bei jedem Durchlauf des Algorithmus maximal 7 Vergleichsoperationen ausgeführt werden.

Übung 4

_____/5 P.

Betrachten Sie ein Array $A[1, \dots, n]$, dass aus n ganzen Zahlen besteht. Für jede Zahl im Array gilt $0 \leq A[i] \leq k$ für eine Zahl $k > 0$ und $i \in \{1, \dots, n\}$. Das Array A kann so verarbeitet werden, dass ein Algorithmus in konstanter Zeit für ein Intervall $[a..b]$ bestimmen kann, wie viele Zahlen aus A in dem Intervall enthalten sind.

- (a) Beschreiben Sie einen Algorithmus der A in $\Theta(n + k)$ verarbeitet und für eine verarbeitete Eingabe in $O(1)$ ausgibt, wie viele Zahlen aus A in ein Intervall $[a..b]$ fallen. (4 Punkte)
- (b) Zeigen Sie, dass Ihr Algorithmus die Laufzeitschranken einhält. (1 Punkt)

Übung 5

_____/3 P.

- (a) Erklären Sie, wie eine Queue durch zwei Stacks implementiert werden kann. (2 Punkte)
- (b) Analysieren Sie die Laufzeit der Queueoperationen aus (a) unter der Annahme, dass die Stackoperationen Zeit $O(1)$ benötigen. (1 Punkt)

Übung 6

_____/3 P.

Bonusaufgabe

Zeigen Sie, dass es keinen Algorithmus A geben kann, der die folgenden drei Eigenschaften gleichzeitig erfüllt.

- Der Algorithmus A benutzt wie ein Vergleichssortierer nur Vergleiche, Zuweisungen, Kopierungen, usw.
- Der Algorithmus A entfernt aus einem max-Heap das Maximum und stellt die max-Heap-Eigenschaft wieder her.
- Der Algorithmus A hat Laufzeit $O(1)$.

(3 Punkte)