

---

**Westfälische Wilhelms-Universität Münster**  
**Übungen zur Vorlesung „Datenstrukturen und Algorithmen“ im SoSe 2017**

Prof. Dr. Klaus Hinrichs  
Aaron Scherzinger

Blatt 9

Abgabe im Learnweb bis zum 06.07.2017 um 10 Uhr

---

**Aufgabe 31:**  $((2+2+2+2)+6+10=24$  Punkte) Betrachten Sie die Datenstruktur Heap und bearbeiten Sie die folgenden Aufgaben:

(a) Geben Sie für einen Heap mit  $n$  Elementen die asymptotische Laufzeit für die Suche nach

- (i) dem kleinsten      (ii) dem größten      (iii) einem beliebigen      (iv) einem nicht

im Heap gespeicherten Element an. Gehen Sie davon aus, dass alle Aufgaben auf demselben Heap durchgeführt werden. Begründen Sie Ihre Antwort.

(b) Beschreiben Sie, wie der ADT **Stack** mit Hilfe eines Heaps implementiert werden kann. Sie dürfen hierbei für jedes abzuspeichernde Element  $\mathcal{O}(1)$  zusätzlichen Speicherplatz verwenden. Welche (technischen) Probleme können gegebenenfalls auftreten? Mit welcher asymptotischen Laufzeit können die Operationen auf einem **Stack** in der von Ihnen beschriebenen Implementation durchgeführt werden?

(c) Betrachten Sie die in einem Array `int[] a` gespeicherte Folge  $a=[18,19,33,10,24,23,41,36,35]$ . Sortieren Sie „von Hand“ diese Folge aufsteigend mit dem Heapsort-Verfahren. Fassen Sie dazu das Array `a` als Darstellung eines in ein Array eingebetteten fast vollständigen binären Baums auf, den Sie zuerst in einen Heap überführen müssen. Anschließend können Sie das aus der Vorlesung bekannte Heapsort-Verfahren anwenden. Geben Sie nach jeder Veränderung des Wurzelknotens des Heaps sowohl den Zustand des Arrays als auch den dadurch repräsentierten Heap an. Beachten Sie in diesem Zusammenhang, dass bei der Abarbeitung des Algorithmus die Anzahl der Array-Elemente konstant bleibt, obwohl der Heap kleiner wird.

**Aufgabe 32:**  $(2+6=8$  Punkte) Die Flugline Tamarindo Airlines belohnt ihre treuen Kunden nach folgendem Schema: Die  $\lfloor \log n \rfloor$  Kunden mit den meisten Flügen erhalten einen Gutschein für einen einwöchigen Urlaub in der Karibik; die  $\lceil \sqrt{n} \rceil$  Kunden mit den meisten Flügen erhalten einen Korb mit Bananen und Kokosnüssen. Da der Computer im Büro von Tamarindo Airlines recht langsam ist, soll ein möglichst effizienter Algorithmus angegeben werden.

(a) Geben Sie einen Algorithmus mit  $\mathcal{O}(n \log n)$  Laufzeit an, der die zu belohnenden Kunden bestimmt. Begründen Sie, warum Ihr Algorithmus die geforderte Komplexität hat.

(b) Geben Sie einen Algorithmus mit  $\mathcal{O}(n)$  Laufzeit an, der die zu belohnenden Kunden bestimmt. Begründen Sie, warum Ihr Algorithmus die geforderte Komplexität hat.

---

**Aufgabe 33:** (4+4+6+8+4=26 Punkte) Eine *double-ended queue* erlaubt Einfügen und Löschen von Elementen sowohl am Anfang als auch am Ende der Queue.

- (a) Geben Sie eine Signatur für den ADT *double-ended queue* an.
- (b) Schreiben Sie ein **JAVA**-Interface, das eine *double-ended queue* für beliebige Typen beschreibt.
- (c) Implementieren Sie in **JAVA** eine *double-ended queue* mit Hilfe eines zirkulären Puffers. Achten Sie dabei auf eine Kommentierung gemäß **JAVADOC** und geeignete Testläufe.
- (d) Implementieren Sie in **JAVA** eine *double-ended queue* für beliebige Typen mit Hilfe einer doppelt verketteten Liste.
- (e) Testen Sie Ihre Implementierungen an Hand geeigneter Testläufe auf Objekten des Typs **Long**.

**Hinweis:** Verwenden Sie für diese Aufgabe **keine** in **JAVA** bereitgestellten Container-Klassen wie z.B. **ArrayList**. Implementieren Sie insbesondere in Aufgabenteil (d) Ihre eigene doppelt verkettete Liste.

**Aufgabe 34:** (6 Punkte) Betrachten Sie eine Realisierung des ADT **PriorityQueue**, die folgende Eigenschaften habe:

- Die Datenstruktur verwendet intern nur binäre Vergleiche.
- Die Datenstruktur lässt sich (für eine Menge von  $n$  Elementen einer total geordneten Menge) in  $\mathcal{O}(n \log_2 \log_2 n)$  Zeit aufbauen, das Einfügen eines einzelnen Objektes kostet  $\mathcal{O}(\log_2 \log_2 n)$  Zeit.
- Die Operation **min** kann in  $\mathcal{O}(1)$  Zeit durchgeführt werden.

Welche asymptotische Laufzeit muss die Operation **delete** mindestens haben? Geben Sie eine möglichst gute Schranke an und begründen Sie Ihre Antwort.

**Bonusaufgabe**<sup>(\*)</sup>:

**Aufgabe 35:** (10+5+5=20 Punkte) Ergänzen Sie Ihre Lösung zu Aufgabe 19 oder die im Learnweb zur Verfügung gestellte Beispiellösung um die folgenden Punkte:

- (a) Realisierung (mit Animation) des Verfahrens *Heapsort* mit Speicherung des Heaps in einem Array in der aus der Vorlesung bekannten Art und Weise.
- (b) Realisierung eines Mechanismus zum Zählen der bei einer Sortierung durchgeführten Vergleiche und Vertauschungen.
- (c) Realisierung einer Testfunktion zum automatischen Ermitteln des „besten“ Sortierverfahrens für einen vorgegebenen Datensatz. Das Gütekriterium sei hierbei die Anzahl der Vergleiche und Vertauschungen, wobei die Vertauschungen doppelt gezählt werden sollen.

<sup>(\*)</sup>**Hinweis:** Bei dieser Aufgabe handelt es sich um eine Bonusaufgabe, d.h. Sie können mit dieser Aufgabe zusätzliche Punkte in der Übung erwerben, es ist aber möglich, ohne diese Aufgabe 100% der Übungspunkte zu erreichen. Wir empfehlen allen Teilnehmerinnen und Teilnehmern die Bearbeitung der Aufgabe.

---