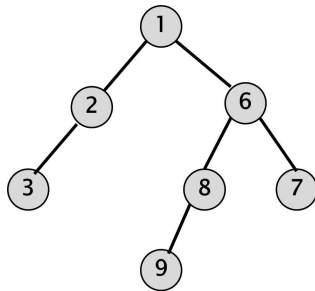


Aufgaben zum Thema Datenstrukturen

- (1) (a) Geben sie für $n \in \{0, 1, 2, 3, 4\}$ alle Möglichkeiten an, die Zahlen $1, \dots, n$ in einen Min-Heap einzuordnen. Der Heap soll minimale Höhe haben und ein rechter Unterbaum darf nicht höher sein als ein linker Unterbaum desselben Knotens.
- (b) Welche Heaps aus (a) haben in der Arraydarstellung keine Lücken?
- (c) Stellen Sie den folgenden Heap als Array dar. Verwenden Sie \perp , um Lücken zu kennzeichnen:



- (2) Gegeben sei das folgende Feld:

12	21	19	26	99	30	24	32	18	101	128
----	----	----	----	----	----	----	----	----	-----	-----

- (a) Erläutern Sie, warum das Feld keinen gültigen Min-Heap darstellt.
- (b) Reparieren Sie den Min-Heap mit dem aus der Vorlesung bekannten Verfahren. Geben Sie dabei den Heap nach jeder Änderung an (als Feld oder als Baum).
- (3) Gesucht ist eine Datenstruktur, in der Einfügen in konstanter Zeit ebenso möglich ist wie die Rückgabe des minimalen Werts.
- (a) Beschreiben Sie kurz eine mögliche Datenstruktur.
- (b) Schreiben Sie Pseudocode für das Einfügen. Es genügt, den für die spätere Rückgabe des Minimums relevanten Teil zu betrachten.
- (c) Schreiben Sie Pseudocode, der das Minimum zurückgibt.
- (d) Beschreiben Sie kurz, was beim Entfernen eines Elements passieren muss. Hat dies Auswirkung auf die Laufzeit des Löschens?
- (4) Für die folgenden Anwendungsfälle soll jeweils eine geeignete Datenstruktur ausgewählt werden. Geben Sie jeweils eine passende Datenstruktur an und begründen Sie Ihre Wahl:
- (a) In einer Musiksammlung sollen häufig neue Musikstücke hinzugefügt, gelöscht und gesucht werden. Über die zukünftige Größe der Sammlung kann beim Anlegen noch keine Aussage gemacht werden.

- (b) An einen Datenbankserver werden zu manchen Zeitpunkten so viele Anfragen geschickt, dass er sie nicht sofort bearbeiten kann. Daher soll er die Möglichkeit bekommen, die Anfragen zwischenspeichern zu können, bis er wieder genügend freie Ressourcen besitzt.
- (c) Ein Prozess-Scheduler in einem Betriebssystem arbeitet mit unterschiedlich hohen Prioritäten. Bei jedem Aufruf soll jeweils der Prozess mit der höchsten Priorität ausgeführt werden, wobei die Auswahlgeschwindigkeit für die Leistungsfähigkeit des Betriebssystems eine entscheidende Rolle spielt.
- (5) Zeigen Sie, wie eine Warteschlange mit einer **einfach verketteten** Liste sowie zwei Zeigern K (ältestes Element) und E (neuestes Element) implementiert werden kann. Hierbei sollen sowohl $\text{ENQUEUE}(X)$ als auch $\text{DEQUEUE}()$ die Laufzeit $O(1)$ besitzen.
- Gehen Sie dafür folgendermaßen vor:

- Stellen Sie mit einer Skizze dar, wie die beiden Zeiger in der Liste positioniert werden müssen.
- Geben Sie den Pseudocode für beide Warteschlangenoperationen an. Stellen Sie dabei sicher, dass Ihre Implementierung auch korrekt mit einer leeren Warteschlange umgehen kann.