

Übungsblatt 3

Aufgabe 1. Wahr oder falsch? Begründen Sie ihre Antwort.

- (a) Für jede Liste mit n Elementen benötigt Quicksort $O(n^2)$ Vergleiche.
- (b) Für jede Liste mit n Elementen benötigt Quicksort $\Theta(n^2)$ Vergleiche.
- (c) Es existiert eine Liste mit n Elementen, für die Quicksort $\Omega(n^2)$ Vergleiche benötigt.
- (d) Es existiert eine Liste mit n Elementen, für die Quicksort $\Theta(n \cdot \log n)$ Vergleiche benötigt.
- (e) Es existiert eine Liste mit n Elementen, für die Quicksort $O(n)$ Vergleiche benötigt.
- (f) Im Durchschnitt benötigt Quicksort $n \cdot \log n$ Vergleiche für Listen mit n Elementen.
- (g) Im Durchschnitt benötigt Quicksort $\Theta(n \cdot \log n)$ Vergleiche für Listen mit n Elementen.

Aufgabe 2. Sortieren Sie die Liste $[3, 5, 2, 1, 4]$ mit Standard-Heapsort und kodieren Sie die verwendeten Einsinkpfade nach dem Verfahren auf den Folien 74–75.

Aufgabe 3. Sortieren Sie die Liste

$[7, 3, 8, 1, 5, 2, 4, 6]$

mit Standard-Heapsort und mit Bottom-up-Heapsort. Geben Sie die Anzahl der Vergleiche an.

Aufgabe 4. Eine *Priority Queue* (dt.: *Vorrangwarteschlange*) ist eine Datenstruktur, die eine Menge von Elementen darstellt, wobei jedes Element einen *Schlüssel* besitzt. Die folgenden Operationen werden unterstützt:

- EXTRACT-MAX gibt das Element mit dem größten Schlüssel zurück und entfernt es.
- INSERT(x) fügt das Element x der Priority Queue hinzu.

Beschreiben Sie, wie mit Hilfe eines Heaps eine Priority Queue effizient implementiert werden kann. Geben Sie die Worst-Case Laufzeit für beide Operationen an.

Aufgabe 5. Welche der folgenden Sortieralgorithmen sind stabil: Mergesort, Quicksort, Heapsort. Wie kann ein beliebiger Sortieralgorithmus so modifiziert werden, dass er stabil läuft?

Aufgabe 6. Sortieren Sie die folgende Liste mit Radixsort.

[224, 421, 319, 121, 914, 314]