

**Algorithmen und Datenstrukturen**  
SS 2021

**Übungsblatt 5: Sortieren**

Tutorien: 24.05.-28.05.2021

**Aufgabe 5-1**      *Quicksort*

Gegeben sei folgendes Array:

37	16	9	42	51	17	71	56	89	39	6	3
----	----	---	----	----	----	----	----	----	----	---	---

- (a) Wenn sie ein Element des unsortierten Arrays finden wollen, wie lange müssen Sie maximal suchen? Begründen Sie ihre Lösung.
- (b) Sortieren Sie das Array mittels Quicksort, nach dem Algorithmus der Vorlesung. Verwenden Sie dabei das letzte Element als Pivot-Element.  
*Machen Sie die einzelnen Schritte bestmöglich nachvollziehbar. Jeweils eine Rekursionsschritt für alle Teilfolgen des Arrays dürfen in einer Zeile gemacht werden. Kennzeichnen Sie das jeweils aktuelle Pivot-Element und die bereits an die richtige Position getauschten Pivot-Elemente.*
- (c) Nun da das Array sortiert ist, wie lange brauchen sie im schlechtesten Fall um ein Element zu finden?
- (d) Zahlt es sich laufzeittechnisch aus zu sortieren, welche Voraussetzungen braucht es, damit sich dies auszahlt?
- (e) Ist dieser Sortieralgorithmus stabil? Begründen Sie.
- (f) Der Algorithmus sei nun an seine obere Laufzeit gestoßen ( $O(n^2)$ ). Unter welchen Umständen kommt es zu diesem Worst Case?
- (g) Was ist die optimale Laufzeit des Sortieralgorithmus und wie sieht das Array aus, wenn diese auftritt?
- (h) Was ist im Allgemeinen eine gute Wahl für das Pivot-Elements?

**Aufgabe 5-2**      *Pferderennen*

Gegeben sind 25 Pferde, von denen die drei Schnellsten ermittelt werden sollen. Dabei gilt:

- Jedes Pferd galoppiert stets konstant mit seiner maximalen Geschwindigkeit.
- Kein Pferd ist genauso schnell wie ein anderes.
- Es gibt eine Rennbahn mit genau 5 Spuren.
- Es gibt keine Stoppuhren oder ähnliche Hilfsmittel.

Wie viele Rennen sind nötig, um die drei schnellsten Pferde zu bestimmen?

**Aufgabe 5-3**      *Heapsort*

Sei  $B = [42, 17, 12, 15, 4, 11, 31, 14]$ .

- (a) Welches Element/Welche Elemente verletzen die Heap-Eigenschaft? Ersetzen Sie diese durch geeignete natürliche Zahlen, sodass die Heap-Eigenschaft erfüllt ist. Nehmen sie möglichst wenige Ersetzungen vor.
- (b) Sortieren Sie das unveränderte Array  $B$  mit Heapsort. Geben Sie die Zwischenschritte als Binärbäume an und machen Sie deutlich, welche Elemente vertauscht werden.
- (c) Angenommen, Sie bekommen täglich eine neue natürliche Zahl mitgeteilt. Außerdem sollen Sie zu jeder Zeit eine sortierte Liste mit allen zuvor mitgeteilten Zahlen abgeben können. Beschreiben Sie einen auf Heapsort basierenden Algorithmus. Eine konkrete Implementierung ist nicht nötig. Geben Sie die obere Schranke des täglichen Aufwands an.

**Aufgabe 5-4**      *3-MergeSort*

Wir definieren 3-MergeSort fast analog zu MergeSort: Statt Halbieren wird eine Liste stets gedrittelt. Beim Zusammenführen werden dann statt aus zwei Teillisten drei Teillisten verglichen. Bestimmen Sie die Komplexität vom 3-MergeSort mit dem Mastertheorem. Es hilft, den Algorithmus vorher an einer kleinen Liste exemplarisch durchzuführen.