

Algorithmen und Datenstrukturen
SS 2019

Übungsblatt 5: Sortieren

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? Bilden Sie aus dem sortierten Array einen binären Baum mit 39 als Wurzel.
- (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 *MergeSort*

Gegeben sei folgende Liste von Zahlen: 24, 15, 76, 6, 43, 9, 41, 68

- (a) Sortieren Sie die Zahlenfolge mit dem MergeSort-Algorithmus, und geben Sie dabei alle Zwischenschritte an. Verwenden sie dazu die unten gegebene Vorlage.
- (b) Wie viele Tauschoperationen wurden beim Sortieren der Liste ausgeführt?

