

Übungen zu Informatik II Sommersemester 2007

6. Aufgabenblatt

Präsenzaufgabe 1 (Quick Sort)

- In der Vorlesung wurde definiert, was ein In-place-Sortierverfahren ist, und behauptet, dass Quick Sort ein solches ist. Betrachten Sie den in der Vorlesung angegebenen Algorithmus QuickSort und überlegen Sie sich, warum dieser im strikten Sinn kein solches In-place-Sortierverfahren ist.
- Für die Abschätzung der durchschnittlichen Laufzeit von Simple Quick Sort wurden die Funktionen f_l und f_r eingeführt. Bestimmen Sie diese Funktionen für das Feld

$$a = [3, 7, 2, 5, 8, 9, 10, 4, 1, 6, 0] \text{ .}$$

Hausaufgabe 1 (Laufzeitanalyse Quick Sort) In der Vorlesung wurde erklärt, dass eine konkrete Quick Sort-Variante durch die Strategie zur Wahl des Pivotelementes bestimmt wird. Bei Simple Quick Sort wird zum Beispiel das am weitesten links stehende Feldelement gewählt. Wir definieren nun zwei weitere:

- Bei *Middle Quick Sort* wird das mittlere Feldelement gewählt. Falls das Feldsegment eine gerade Anzahl von Elementen enthält, wird das linke der beiden mittleren Feldelemente gewählt.
- Bei *Median Quick Sort* wird der Median aus den folgenden drei Feldelementen gewählt:
 - das am weitesten links stehende Feldelement,
 - das am weitesten rechts stehende Element,
 - das mittlere Feldelement (siehe oben).

In der Vorlesung wurde die Anzahl der von Simple Quick Sort durchschnittlich durchgeführten Feldvergleiche bestimmt. Überlegen Sie sich, ob die Analyse aus der Vorlesung auf *Middle Quick Sort* und *Median Quick Sort* einfach übertragen werden kann. Begründen Sie Ihre Aussage.

Hausaufgabe 2 (optimales Sortieren) Überlegen Sie sich ein vergleichsbasiertes Sortierverfahren für Felder aus S_4 , von dem Sie glauben, dass es möglichst wenige Vergleiche durchführt (durchschnittliche Anzahl der durchgeführten Feldvergleiche). Begründen Sie Ihre Vermutung.

Programmieraufgabe 1 (Bewertung Sortierverfahren) Modifizieren Sie die bisher von Ihnen erstellten Sortierverfahren Merge Sort, Simple Quick Sort und Heap Sort derart, dass die Anzahl der durchgeführten Feldvergleiche ermittelt wird, etwa durch schrittweises Erhöhen einer statischen Variablen.

Schreiben Sie dann ein Programm, das für wachsende Feldgrößen jeweils die durchschnittliche Anzahl der durchgeführten Feldvergleiche ausweist, indem es für jedes $n = 2, 3, 4, \dots$, für jedes Sortierverfahren und für jedes $a \in S_n$ die Anzahl der durchgeführten Feldvergleiche ermittelt und geeignet verrechnet.

Schreiben Sie abschließend ein Programm, das für wachsende Feldgrößen jeweils die durchschnittliche Anzahl der durchgeführten Feldvergleiche ausweist, indem es für jedes $n = 2, 3, 4, \dots$, für jedes Sortierverfahren und für eine ausreichende Anzahl von zufällig belegten Feldern mit Zahlen aus der Menge $\{0, \dots, n - 1\}$ jeweils eine Schätzung der Anzahl der durchgeführten Feldvergleiche ermittelt und geeignet verrechnet. Verwenden Sie für die Erzeugung der zufällig belegten Felder die Klasse `RandomArray` wie sie in Serie 3 eingeführt wurde.

Ausgabe: 12.5.2007, **Abgabe:** bis 18.5.2007, 12 Uhr, im Schrein