

Universität Leipzig Institut für Informatik Automatische Sprachverarbeitung	Algorithmen und Datenstrukturen WS 2017/18 – Serie 3		
Prof. Dr. Uwe Quasthoff	Ausgabe am 21.11.2017	Abgabe am 28.11.2017	Seite 1/2

Algorithmen und Datenstrukturen

WS 2017/18– Serie 3

11 (6 Punkte) Quicksortieren

Gegeben sei folgende Zahlenfolge in einem 0-basierten Array A :

7 43 47 31 57 3 23

Sortieren Sie diese mit der in-situ Variante dieses Verfahrens laut Vorlesung 5, Folie 4. Listen Sie wie folgt alle Zwischenschritte auf: geben Sie für jeden Aufruf der Funktion $qsort(A, l, r)$, bei dem $l < r$, die Parameter l und r an, sowie das berechnete Pivotelement piv , den Inhalt von A (mit markierter Teilliste) nach Abarbeitung der äusseren do/while-Schleife und die rekursiven Aufrufe. Ein Aufruf $qsort(A, 2, 5)$ könnte also so beschrieben werden:

$l=2; r=5; piv=A[3]=31; 7\ 43\ \boxed{3\ 31\ 57\ 47}\ 23 ; qsort(A, 2, 2); qsort(A, 4, 5)$

Vergessen Sie keine rekursiven Aufrufe und listen Sie die Aufrufe in der korrekten Reihenfolge. Nutzen Sie *floor* als Rundungsfunktion.

12 (4 Punkte) Schlimmstes Quicksort

Ordnen Sie, wenn möglich, die Elemente $1, 2, \dots, 7$ in einem Beispieldatenfeld der Größe 7 duplikatfrei so an, dass die Sortierung mit Quicksort maximale Schritte benötigt. Betrachten Sie dabei die Fälle, dass als das Pivotelement

- stets ein zufälliges Element
- stets das mittlere Element (mit Abrunden des Indexes)
- stets das erste Element

gewählt wird. Begründen Sie Ihr Ergebnis.

13 (4 Punkte) Sortieren in Strömen

Gegeben sei ein Array A von Zahlen.

$A = [49\ 51\ 47\ 48\ 56\ 52\ 54\ 50\ 43\ 44\ 55\ 57\ 46\ 53\ 42\ 45]$

Sortieren Sie das Array A aufsteigend je einmal mit

- nicht-rekursivem Merge-Sort und
- natürlichem nicht-rekursivem Merge-Sort.

Universität Leipzig Institut für Informatik Automatische Sprachverarbeitung	Algorithmen und Datenstrukturen WS 2017/18 – Serie 3		
Prof. Dr. Uwe Quasthoff	Ausgabe am 21.11.2017	Abgabe am 28.11.2017	Seite 2/2

Folgen sie genau den in der Vorlesung gegebenen Algorithmen. Geben Sie (wie in den Beispielen der Vorlesung) nach jedem Durchlauf der äusseren Schleife den Arrayinhalt an und markieren Sie die Grenzen zwischen den jeweils bereits vom Algorithmus sortierten (bzw. als sortiert erkannten) Teillisten. **Achten Sie genau auf den korrekten Typ des Algorithmus und die Aufgabenstellung!**

14 (5 Punkte) Selection und Mergesort

a) Gegeben sei die folgende Namensliste:

Mauler , Gerhard
Schmidt , Heinz
Schmidt , Ernst
Hansen , Detlef
Schmidt , Antonia
Hansen , Frankie
Schmidt , Christiane
Mauler , Berta

Geben Sie die Liste nach einer lexikographischen Sortierung mit Selection-Sort und Merge-Sort wie in der Vorlesung beschrieben an. Schlüssel seien nur die Nachnamen, Vornamen werden bei Schlüsselvergleich ignoriert. Welcher entscheidende Unterschied der Algorithmen wird hier deutlich (und woran)?

b) Finden Sie einen möglichst einfachen Trick, durch den jedes auf paarweisen Schlüsselvergleichen beruhende Sortierverfahren stabil gemacht werden kann.

15 (6 Punkte) Reihenweise Haufen

Gegeben sei ein 1-basiertes Array

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

- a) Interpretieren Sie das Array als Darstellung eines Binärbaums und geben Sie diesen Binärbaum an. (1 Punkte)
- b) Stellen Sie für A die Heap Eigenschaft her. Geben Sie an, welche Schlüssel abgesenkt werden und wie der Binärbaum danach jeweils aussieht. (2 Punkte)
- c) Geben Sie, ausgehend von dem in b) hergestellten heap, die Schritte (Schlüsselvertauschungen und Knotenabsenkung) an, die der Heapsortalgorithmus während der ersten drei Iterationen durchführt (d.h. die Iterationen welche die zwei größten Elemente an die korrekte Position bringen). (2 Punkte)

Für b) und c) können Sie die Schritte des Algorithmus anhand der Binärbaumdarstellung oder des äquivalenten Arrays darstellen. Verwenden Sie die Algorithmen wie in Vorlesung 5 Folien 13-17 beschrieben.