

Übungen zur Vorlesung
Algorithmen und Datenstrukturen
WiSe 2018/19
Blatt 4

Wichtige Hinweise:

- > Falls Sie bei der Bearbeitung einer Aufgabe größere Schwierigkeiten hatten und deswegen die Bearbeitung abgebrochen haben, so versuchen Sie bitte Ihre Schwierigkeiten in Form von Fragen festzuhalten. Bringen Sie Ihre Fragen einfach zur Vorlesung oder zur Übung mit!
- > Kursraum: <https://elearning.uni-regensburg.de/course/view.php?id=9228>

Aufgabe 1:

Modifizieren Sie die in der Vorlesung vorgestellten Sortieralgorithmen wie folgt:

1. Drehen Sie die Reihenfolge bei InsertionSort: Statt über das Maximum einzusortieren, soll über das Minimum einsortiert werden.
2. Lassen Sie bei BubbleSort das Maximum nach hinten wandern, statt das Minimum nach vorne wandern zu lassen.
3. Setzen Sie bei SelectionSort anstelle des Minimums jeweils das Maximum an die richtige Stelle.
4. Wählen Sie bei QuickSort als Pivot-Element ein zufälliges Element.

Erläutern Sie jeweils Laufzeit und Korrektheit Ihrer Algorithmen.

Aufgabe 2:

Generieren Sie jeweils gleichverteilt zufällige ganze Zahlenfolgen mit n Zahlen und ermitteln Sie auf Ihrem konkreten Rechner, wie viele Zahlen Sie innerhalb von einer Minute mit einem einfachen und mit einem fortgeschrittenen Sortiervorgang Ihrer Wahl sortieren können. Überlegen Sie sich eine geeignete Speicherung der Zahlen!

Aufgabe 3:

Demonstrieren Sie die Funktionsweise von Merge Sort und Heap Sort anhand des Feldes $a[] = \{-5, 13, -32, 7, -3, 17, 23, 12, -35, 19\}$. Führen Sie die Demonstration von Merge Sort anhand eines Rekursionsbaums durch, indem Sie den Baum in der Phase „Teilen“ von oben nach unten aufbauen und anschließend in der Phase „Mischen“ von unten nach oben durchlaufen. Nennen Sie jeweils die entstandenen Teilfolgen. Führen Sie bei der Demonstration von Heap Sort sowohl entstehende Bäume als auch entstehende Felder mit. Überprüfen Sie Ihre Ausführungen mit Hilfe eines C, C++, C# oder Java-Programms, das die beiden Sortiervorgänge implementiert und die wesentlichen Informationen ausgibt.

Aufgabe 4:

Wandeln Sie den iterativen InsertionSort Algorithmus aus der Vorlesung in einen rekursiven und den rekursiven MergeSort Algorithmus aus der Vorlesung in einen iterativen um. Analysieren Sie jeweils die Laufzeit Ihrer Algorithmen.