

Universität Potsdam
Institut für Informatik
Algorithmen und Datenstrukturen

07. Aufgabenblatt

Lernziele (zum Abhaken): Die Student:innen können. . .

für einen gegebenen Algorithmus eine Eingabe konstruieren, sodass der best- sowie worst-case in Bezug auf die Laufzeitkomplexität eintritt.

zwei gegebene Suchalgorithmen anhand der Anzahl der Vergleichsoperationen miteinander vergleichen.

für einen gegebenen Algorithmus die Rekursionsgleichung aufstellen und diese mithilfe des Mastertheorems lösen.

einen gegebenen Sortieralgorithmus auf eine gegebene Zahlensequenz anwenden.

11 Suchen

1. Geben Sie jeweils eine Folge von 12 Zahlen und eine zu suchende Zahl an, die für den Algorithmus Binäre Suche eine Eingabe

- (a) im schlechtesten Fall,
- (b) im besten Fall

darstellen.

2. Geben Sie für die Algorithmen

- (a) `linSearch`,
- (b) `binSearch`

jeweils an, wie oft sie höchstens einen Wert der zu durchsuchenden Sequenz mit dem zu suchenden Wert vergleichen, falls die Sequenz n Elemente enthält. Begründen Sie!

3. Zeigen Sie durch Anwendung des Mastertheorems, dass für die Laufzeit $t(n)$ des Algorithmus Binäre Suche $t(n) \in O(\log n)$ gilt.

12 Sortieren

Stellen Sie im Folgenden immer alle Zwischenschritte dar.

1. Sortieren Sie die Folge (23, 10, 79, 16, 5, 66) nach der Methode von `insertionSort` als in-place Algorithmus.
2. Sortieren Sie die Folge (11, 2, 0, 19, 3, 7, 78, 2, 38) mit Quicksort (nach dem in-place Algorithmus aus der Vorlesung), wobei als Pivotelement das Element in der Mitte der Sequenz zu wählen ist, also an `(low + high)//2`.
3. Geben Sie jeweils zwei Folgen der Länge 5 an, die für den in-place Algorithmus Quicksort (mit dem Pivotelement in der Mitte der Sequenz) eine Eingabe

- (a) im schlechtesten Fall,
- (b) im besten Fall

darstellen.