

AlgoDat Tutorium Blatt 5

Alex B.

Mai 2024

1 Countingsort

- Vergleichende Sortieralgorithmen laufen im schlechtesten Fall in $\Omega(n * \log(n))$
- Um in linearer Zeit zu sortieren, brauchen wir einen nicht vergleichenden Algorithmus
- Countingsort: Zähle wie viele Elemente im Array kleiner als aktuell betrachtet sind und platziere Element hinter diesen.

2 Aufgaben

1. Zeige die Funktionsweise von Countingsort an folgenden Arrays
 - a) $\langle 2, 4, 1, 0, 2, 6, 5, 5, 3, 7 \rangle$
 - b) $\langle 7, 7, 5, 12, 2, 3 \rangle$
 - c) $\langle 21, 112, 44, 21, 27, 8 \rangle$
2. Wie müsste man Countingsort verändern, um von vorne beginnend einzusortieren?
3. Welches Problem hat Countingsort bei Probleminstanzen mit sehr großen Zahlen? Wie könnte man es lösen?
4. Kann man mit Countingsort auch nicht aufsteigend sortieren? Wenn ja, wie?

3 Radixsort

- Idee: Sortiere nach einem bestimmten Größemaß (z.B. Hunderterstelle zuerst, dann Zehnerstelle, etc.)
 - Benötigt ein stabiles Sortierverfahren
1. Zeige die Funktionsweise von Radixsort an folgenden Arrays

- a) $\langle 21, 112, 44, 21, 27, 8 \rangle$
 - b) $\langle 728, 205, 687, 140, 159, 666, 394, 392, 155 \rangle$
 - c) $\langle CBZ, TBD, BYE, AAB, XYZ, LMN, ACE, FSI \rangle$
2. Welche der bisher betrachteten Sortierverfahren waren stabil?
 3. Wie kann man jedes Sortierverfahren stabil machen?

4 Zusatzaufgabe Hoare-Kalkül

Schreibe ein Programm, das ein Array, in dem eine ganze, nichtnegative Zahl enthalten ist sowie eine zweite Zahl n erhält, wobei n größer gleich die Länge des Arrays ist. Das Programm soll ein neues Array ausgeben, welches die Zahl des ursprünglichen Arrays enthält, welche mit führenden Nullen auf die Länge n verlängert wurde. Beweise anschließend, dass dein Programm korrekt ist.