

Algorithmen und Datenstrukturen

Übungsblatt 3

Ioan Oleksii Kelier

Florian Keppeler

Aufgabe 3

(a) Bucket sort

Sei $f(n, k)$ eine Funktion, die die benötigte Zeit zur Ausführung des Algorithmus beschreibt.

- (1) Zu Beginn des Algorithmus müssen wir ein Array erstellen, das $k + 1$ Warteschlangen enthält, was $O(k)$ Zeit benötigt.
- (2) Dann gehen wir durch alle Elemente des anfänglichen Arrays und fügen jedes Element auf der Grundlage des Werts des Schlüssels zur Warteschlange hinzu. Dafür benötigen wir $O(n)$ Zeit, da wir durch das Array mit n Elementen gehen und ein Element in die Warteschlange einfügen, was $O(1)$ kostet.
- (3) Am Ende wird eine Liste mit k Elementen durchlaufen, von denen jedes eine Warteschlange ist. Die Extraktion von Elementen aus der Warteschlange kostet $O(1)$. Da die meisten Warteschlangen leer sind oder nur wenige Elemente enthalten, benötigen wir $O(k)$ Zeit für die oben genannten Vorgänge.

Zusammengefasst benötigen wir $O(n + k)$ Zeit, da $f(n, k) = n + k + k = n + 2k \Rightarrow f(n) \in O(n + k)$ gilt.

(b) Radix sort

Da die Implementierung von RadixSort in dieser Aufgabe hauptsächlich auf der Verwendung der vorherigen Sortierung beruht, können wir davon ausgehen, dass wir $O(m \cdot n)$ Zeit benötigen, da wir m Aufrufe an BucketSort mit $k = b$ machen, um das ursprüngliche Array schrittweise Bit für Bit zu sortieren.