

Einführung in die Algorithmik – Hausaufgabenserie 3

Henri Heyden, Nike Pulow
stu240825, stu239549

Aufgabe 3: Radixsort Zeitkomplexität

Radixsort hat normalerweise eine lineare Laufzeit in Bezug auf die Länge der zu sortierende Liste. Dem liegt zugrunde, dass nicht über Vergleiche, sondern über Partitionierung der Ziffern sortiert wird. Auch das Umwandeln der Integer-Werte in Strings ist linear.

Wenn allerdings die Zahlenwerte beliebig lang sind, also eine unbeschränkte Anzahl von Ziffern übergeben werden kann, dann kann ein zusätzlicher Faktor von $\log(n)$ in der Laufzeit von Radixsort entstehen, da Radixsort eine Schleife über die maximale Anzahl von Ziffern der Elemente ausführen muss, um sie zu sortieren. Wenn die Anzahl der Ziffern logarithmisch mit der Größe der Liste wächst, dann erschließt sich so der zusätzliche Faktor von $\log(n)$ bei der Laufzeit.

Wenn man die Liste `list(range(n))` sortiert, enthält sie nur ganzzahlige Werte von 0 bis $n-1$, die alle maximal $\log(n)$ Stellen haben. Allerdings hat das Sortieren der Liste `list(range(n))` dennoch Laufzeit $O(n * \log(n))$, da in jedem Durchlauf des Radixsorts die Partitionierung der n Zahlen durchgeführt werden muss. Da der Radixsort insgesamt $\log(n)$ Durchläufe benötigt, ergibt sich insgesamt eine Laufzeit von $O(n * \log(n))$.