

Aufgabe 02 - Sortieren

Sortieren mittels Quicksort

Implementieren Sie den Divide-And-Conquer Sortieralgorithmus *Quicksort* auf rekursive Art. Dabei wird ein Element, das *Pivot-Element*, verwendet, um die noch nicht sortierte Liste in zwei Teile aufzuteilen - ein Teil in dem die Elemente kleiner als das Pivot-Element und ein Teil in dem die Elemente größer als das Pivot-Element sind. Diese Teillisten werden dann anschließend in den rekursiven Aufrufen erneut geteilt, bis nur noch ein Element übrig ist (oder der Teil der Liste bereits sortiert ist). Das Prinzip ist in Abbildung 1 dargestellt. Anschließend werden die sortierten Ergebnisse zusammengefügt und als Ergebnis die gesamte, sortierte Liste zurückgegeben.

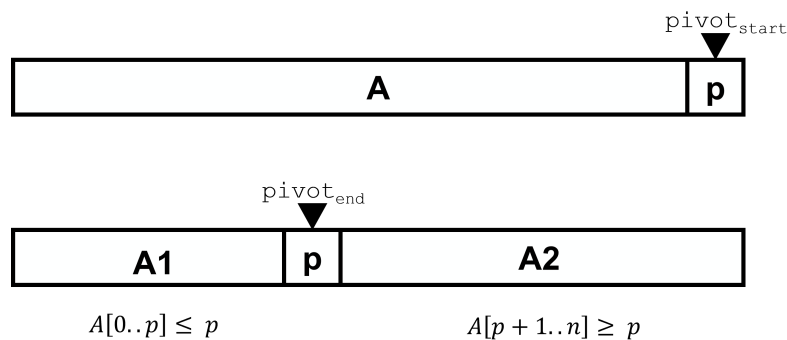


Abbildung 1: Divide-And-Conquer sortierverfahren Quicksort

Für Ihre Implementierung können Sie Arrays, Listen... verwenden. Testen Sie Ihre Algorithmen mit verschiedenen Inputs.

Die Position des Pivot-Elements kann die Durchlaufzeiten des Algorithmus beeinflussen. Überlegen Sie sich eine Variante des Algorithmus wo das Pivot-Element zufällig gewählt werden kann. Abbildung 2 zeigt ein Beispiel für die Positionierung des Pivot-Elements in der Mitte der Liste. Vergleichen Sie diese Variante mit der Ausführung Ihrer zuvor implementierten Variante aus der Vorlesung. Sie sollten für dieselben Folgen von Zahlen immer dasselbe (korrekte) Ergebnis (sortierte Liste) nach Ausführung erhalten, aber die dafür notwendigen Schritte sollten sich unterscheiden bzw. können einmal mehr und einmal weniger Schritte notwendig sein.

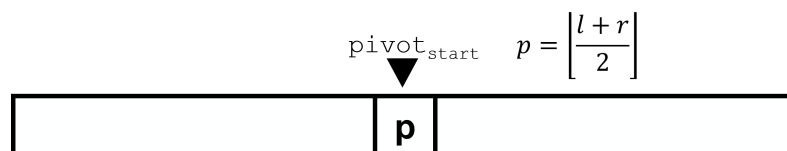


Abbildung 2: Quicksort mit Pivot-Element in der Mitte der Liste

Vergleichen Sie den QuickSort-Algorithmus mit einem Algorithmus (Merge-Sort, https://en.wikipedia.org/wiki/Merge_sort), der auf demselben Prinzip, Divide-And-Conquer, beruht. Vergleichen Sie die notwendigen Schritte die durchgeführt werden mit einer Ihrer Implementierung von QuickSort.