

Übung 5

Tutoraufgabe 1 (Quicksort):

Sortieren Sie das folgende Array mithilfe von Quicksort. Geben Sie dazu das Array nach jeder Partition-Operation an und markieren Sie das jeweils verwendete Pivot-Element.

2	3	9	6	7	4	1	5	8

Tutoraufgabe 2 (Mergesort):

Sortieren Sie das folgende Array mithilfe von Mergesort. Geben Sie dazu das Array nach jeder Merge-Operation an.

2	3	9	6	7	4	1	5	8

Tutoraufgabe 4 (Sortieren von Polarkoordinaten):

In dieser Aufgabe möchten wir Punkte $(x, y) \in \mathbb{Z}^2$ im 2-dimensionalen Raum nach verschiedenen Kriterien (Schlüsseln) sortieren.

- Beschreiben Sie, wie man die Sortierverfahren aus der Vorlesung abändern kann, um ein gegebenes Array von Punkten aufsteigend nach dem Euklidischen Abstand zum Ursprung zu sortieren.
- Beschreiben Sie, wie man die Sortierverfahren aus der Vorlesung abändern kann, um ein gegebenes Array von Punkten aufsteigend nach dem Winkel zwischen der x -Achse und der Linie die den Ursprung mit einem Punkt verbindet zu sortieren.
- Beschreiben Sie einen Algorithmus der als Eingabe ein Array a von n Punkten bekommt und dieses Array so sortiert, dass wir jeweils die Punkte $a[i]$ und $a[i+1]$ (für $i \in \{0, \dots, n-2\}$ sowie $a[n-1]$ und $a[0]$) mit geraden Linien verbinden können, **ohne** dass sich die Linien kreuzen bzw. überlappen.

Wir gehen davon aus, dass sich in jedem Quadranten des Koordinatensystems mindestens ein Punkt befindet. Begründen Sie kurz, warum ihr Algorithmus korrekt ist.

Beispiel. In der unten stehenden Abbildung sehen wir eine mögliche Eingabe (oben links), eine korrekte Ausgabesortierung mit eingezeichneten Linien (oben rechts), sowie zwei falsche Ausgabesortierungen bei der sich die Linien sowohl kreuzen (unten links) als auch überlappen (unten rechts).

