

QuickSort

Das QuickSort-Verfahren basiert darauf, zuerst ein Pivot-Element auszuwählen, anschließend die Liste anhand diesem aufzuteilen und die Teillisten wieder per QuickSort rekursiv zu sortieren. Anschließend werden die Teillisten wieder zusammengesetzt.

```
public static int [] quicksort(int [] unsortiert) {  
    // Abbruchbedingung wenn unsortiert.length <= 1  
  
    // Pivot-Element auswählen (erstes Element des Arrays)  
  
    // linke und rechte Teilliste anlegen  
  
  
  
  
    // Liste anhand des Pivot-Elements aufteilen  
  
  
  
  
  
  
    // linke und rechte Teilliste "kürzen"  
  
  
  
    // linke und rechte Teilliste rekursiv sortieren  
  
  
  
    // Gesamtliste sortiert zusammenfügen  
  
    }  
}
```

Listing 1: Überblick QuickSort

Liste „kürzen“ bzw. zusammenfügen

Ein Array in JAVA hat immer eine konstante Länge. Um ein kürzeres Array zu erhalten, müssen wir dieses neu anlegen und die Elemente kopieren. Hierzu gibt es die Methode

```
System.arraycopy(Quelle, Start-Index, Ziele, Start-Index, Länge);
```

Listing 2: Inhalt eines Arrays kopieren

1. Programmierung

Fülle zunächst oben stehenden Code aus. Programmiere anschließend die Methode.

2. Test

Lasse anschließend ein Array mit 10 (100, 1000,...) Einträgen zufällig befüllen und ausgeben. Rufe dann deine Methode auf und lasse die sortierte Liste wieder ausgeben.

3. Laufzeit

Um die Laufzeit des Verfahrens zu messen, können wir die Methode `System.currentTimeMillis();` verwenden. Diese gibt die vergangenen Millisekunden seit 1. Januar 1970 an. (Datentyp `long`)

Lassen wir diese vor und nach der Sortierung ausgeben können wir daraus die benötigte Zeit berechnen. Erzeuge (nacheinander) verschieden große Arrays und messe damit die Laufzeiten. Vergleiche die verschieden großen Arrays.