

Einsendeaufgabe 4

1- Gegeben sei das Zahlenfeld $A = (11, 15, 13, 8, 1, 10, 16, 12)$. A ist nach dem Algorithmus Quicksort aufsteigend zu sortieren.

(a) Führen Sie die erste Teilung nach dem im Skript gegebenen Verfahren durch.

i. Geben Sie die dabei auftretenden Vertauschungen der Reihe nach an.

ii. Welches ist der erste Teilungsindex?

(b) Stellen Sie den gesamten Ablauf in einem Binärbaum dar, so dass insbesondere das rekursive Teilen des Zahlenfeldes deutlich wird. Markieren Sie darin auch die jeweiligen Pivotelemente.

Für die Frage (a) erwarte ich etwas ähnliches wie $\text{Partition}(A, 1, 6)$: im Skript Seite 55 und für Frage (b) etwas ähnliches wie Abbildung 4.8 Rekursives Teilen des Feldes Seite 56.

2- Implementieren Sie in der Klasse `MeineArrays` den Algorithmus Quicksort Seite 54 im Skript mit der folgenden Änderung: die Methode muss die Anzahl der **Vergleiche** zurück geben (Achtung, nicht die Anzahl der Aufrufe). *Sie sollen nicht jeden Vergleich zählen. Der Algorithmus Partition macht immer $r-p$ Vergleiche. Sie brauchen nur die $r-p$ Vergleiche zu addieren.*

Sie implementieren zwei Methoden mit folgender Signatur:

```
public static long quickSort(int[] ar)
```

```
private static long quickSort(int[] ar, int p, int r)
```

Die erste Methode ist für Benutzer-innen des Algorithmus gedacht und ruft die zweite Methode `quickSort` auf. Die zweite Methode ist eine Hilfsmethode, ist rekursiv und implementiert den Algorithmus.

Der Algorithmus Partition wird ebenfalls als Hilfsmethode implementiert.

Testen Sie vollständig die Methode `quickSort` in der Klasse `MeineArraysTest`. Mit vollständig ist das Folgende gemeint: das Array ist leer, das Array hat ein Element, das Array hat mehrere Elemente. Prüfen Sie mit dem Array und Ihrer Antwort der Frage 1, dass Ihre Methode die Anzahl der Vergleiche richtig ermittelt.

Es ist WICHTIG, dass Sie die Algorithmen vom Skript genau übernehmen. Es gibt in der Literatur verschiedene Varianten von Partition. Eine andere Variante kann zu einer unterschiedlichen Anzahl der Vergleiche führen.

3- Testen Sie Ihre Methode mit den Zahlen in der Datei `QuickSort.txt` Copyright *Tim*

Roughgarden, Stanford University. Dafür benutzen Sie die Klasse `UtilitiesArraysFiles`.

Wie viele Vergleiche bekommen Sie? Passt das Ergebnis mit dem theoretischen Ergebnis im besten Fall $n \log n$? (Log sollte mit Basis 2 berechnet werden)

4- Stellen Sie die Rekursionsgleichung zur Bestimmung der Zeitkomplexität vom Quicksort im idealen Fall auf: das Array wird immer in zwei geteilt.

5- Lösen Sie die Rekursionsgleichung mit Hilfe des Master-Theorems und geben sie die Zeit-Komplexität in asymptotischer Notation.

6- Das letzte Element als Pivotelement ist meistens keine gute Wahl, und kann zu einer Zeit-Komplexität von $O(n^2)$ führen. Implementieren Sie Quicksort so, dass es leicht möglich ist, eine andere Wahl für das Pivotelement zu treffen, z.B. durch Zufall. Dafür benutzen Sie das Strategy Design Pattern. Schreiben Sie ein funktionales Interface namens `IQuickSortStrategy`, welche die einzige folgende Instanz-Methode enthält:

```
int findPivot(int l, int r);
```

Diese Methode findet den Index eines Pivotes zwischen den Positionen l und r und gibt den Index zurück. Schreiben Sie eine Klasse namens `PivotRandom`, welche die Schnittstelle

IQuickSortStrategy so implementiert, dass der Index zufällig gewählt wird, oder auch benutzen Sie ein Lambda-Ausdruck in der Testklasse. Überladen Sie die Methode quicksort wie folgt:

```
public static long quickSort(int[] ar, IQuickSortStrategy st)
private static long quickSort(int[] ar, int p, int r, IQuickSortStrategy st)
```

Die rekursive Methode wird so verändert, dass zuerst der Index des Pivotelements ermittelt wird, und das Pivotelement mit dem letzten Element getauscht wird. *Die Partitionsmethode bleibt unverändert.* Achten Sie darauf, dass der Index des Pivotelementes in jedem Aufruf mit der Strategie gefunden sein muss. Dies Vorgehen entspricht Algorithmus 4.8 Seite 58 im Skript. Wie viele Vergleichen bekommen Sie für Frage 3, wenn das Pivot durch Zufall gewählt wird?

Senden Sie in einer .zip Datei NamenESA4. zip ein Dokument mit Ihren Antworten und Ihre .java Dateien.