Übungsblatt 18 Java

1. Insertion Sort
Ein Array von Ganzzahlen soll in aufsteigender Reihenfolge sortiert werden.

Das Vorgehen ist mit der Sortierung eines Spielkartenblatts vergleichbar. Am Anfang liegen die Karten des Blatts verdeckt auf dem Tisch. Die Karten werden nacheinander aufgedeckt und an der korrekten Position in das Blatt, das in der Hand gehalten wird, eingefügt. Um die Einfügestelle für eine neue Karte zu finden wird diese sukzessive (von rechts nach links) mit den bereits einsortierten Karten des Blattes verglichen. Zu jedem Zeitpunkt sind die Karten in der Hand sortiert und bestehen aus den zuerst vom Tisch entnommenen Karten.

```
int a[] = { 5, 3, 2, 6, 4, 1, 3, 7 };
[5,3,2,6,4,1,3,7]
[3,5,2,6,4,1,3,7]
[2,3,5,6,4,1,3,7]
[2,3,5,6,4,1,3,7]
[2,3,4,5,6,1,3,7]
[1,2,3,4,5,6,3,7]
[1,2,3,3,4,5,6,7]
```

2. Merge Sort
Ein Array von Buchstaben soll in aufsteigender Reihenfolge sortiert werden.
Das Array wird fortlaufend halbiert bis nur noch Teilstücke der Länge 1 vorhanden sind. Die Teilstücke werden nach und nach wieder zusammengeführt (merge).

3. Quick Sort
Ein Array von Ganzzahlen soll in aufsteigender Reihenfolge sortiert werden.

```
int a[] = { 5, 3, 2, 6, 4, 1, 3, 7 };

Start:
(0,7) [5,3,2,6,4,1,3,7]
(0,7) [3,3,2,6,4,1,5,7]
(0,7) [3,3,2,1,4,6,5,7]
(0,4) [1,3,2,3,4]
(0,4) [1,2,3,3,4]
(2,4) [3,3,4]
(5,7) [5,6,7]
Ende:
(0,7) [1,2,3,3,4,5,6,7].
```

Zunächst wird die zu sortierende Liste in zwei Teillisten („linke“ und „rechte“ Teilliste) getrennt. Dazu wählt Quicksort ein sogenanntes Pivotelement aus der Liste aus (Zur Vereinfachung das erste Arrayelement). Alle Elemente, die kleiner als das Pivotelement sind, kommen in die linke Teilliste, und alle, die größer sind, in die rechte Teilliste. Die Elemente, die gleich dem Pivotelement sind, können sich beliebig auf die Teillisten verteilen. Nach der Aufteilung sind die Elemente der linken Liste kleiner oder gleich den Elementen der rechten Liste.
Anschließend muss man also nur noch jede Teilliste in sich sortieren, um die Sortierung zu vollenden. Dazu wird der Quicksort-Algorithmus jeweils auf der linken und auf der rechten Teilliste ausgeführt. Jede Teilliste wird dann wieder in zwei Teillisten aufgeteilt und auf diese jeweils wieder der Quicksort-Algorithmus angewandt, und so fort. Diese Selbstaufrufe

werden als Rekursion bezeichnet. Wenn eine Teilliste der Länge eins oder null auftritt, so ist diese bereits sortiert und es erfolgt der Abbruch der Rekursion, d. h. für diese Teilliste wird Quicksort nicht mehr aufgerufen.

4. Lineare Suche

In einem Array von 20 zufälligen Ganzzahlen soll geprüft werden ob eine vom Benutzer eingegebene Zahl enthalten ist. Falls die Zahl vorhanden ist wird der Index der Zahl ausgegeben, andernfalls eine -1.

```
51 63 75 59 22 95 18 24 86 34 33 66 58 71 15 32 76 94 34 55
Suche (oder -1 für Ende): 22
Gefunden: 4
Suche (oder -1 für Ende): 3
Gefunden: -1
```

5. Binäre Suche

In einem sortierten Array von Ganzzahlen soll eine Zahl mit der Methode der Intervallhalbierung gefunden werden. Falls die gefunden wurde soll der Index ausgegeben werden. Wird sie nicht gefunden so soll der Index k berechnet werden an der sie eingefügt werden müsste. In diesem Fall sei die Rückgabe -k-1.

```
1 4 9 13 17 21 39 43 54 68 70
Suche (oder -1 für Ende): 17
Gefunden: 4
Suche (oder -1 für Ende): 5
Gefunden: -3
```

Finden Sie eine rekursive und eine iterative Lösung.
Wie viele Suchschritte sind maximal in einem Array der Größe 1000000 nötig?