

Übung 10: Sortieren

Abgabetermin: 7.6.2011

Name: _____ Matrikelnummer: _____

Gruppe: G1 Di 10:15 G2 Di 11:00 G3 Di 12:45

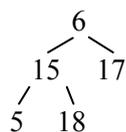
Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	8	<input type="checkbox"/>	Zeichnungen	-	<input type="checkbox"/>	
Aufgabe 2	16	<input type="checkbox"/>	Java-Programm Testergebnisse	Java-Programm	<input type="checkbox"/>	

Aufgabe 1: Heapsort

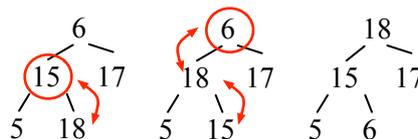
Wenden Sie den Heapsort-Algorithmus auf die Zahlenfolge 5, 11, 4, 2, 15, 16, 2, 3, 18, 1 an. Übertragen Sie die Zahlen in einen Binärbaum (a) und dokumentieren Sie wie die Heapordnung mit *downHeap* hergestellt wird (b). Zeigen Sie danach den Sortiervorgang durch Abspalten des jeweils letzten Elements im Wechsel mit *downHeap* (c).

Beispiel: 6 15 17 5 18

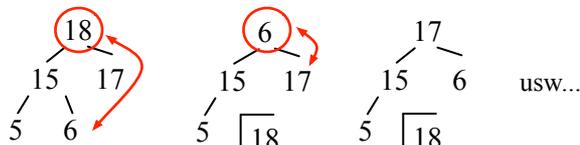
a) Binärbaum



b) Heapordnung herstellen



c) Sortieren



Abzugeben ist: Zeichnungen

Aufgabe 2: HeapSort, MergeSort

Implementieren Sie die Sortierverfahren Heap-Sortieren und Mischsortieren.

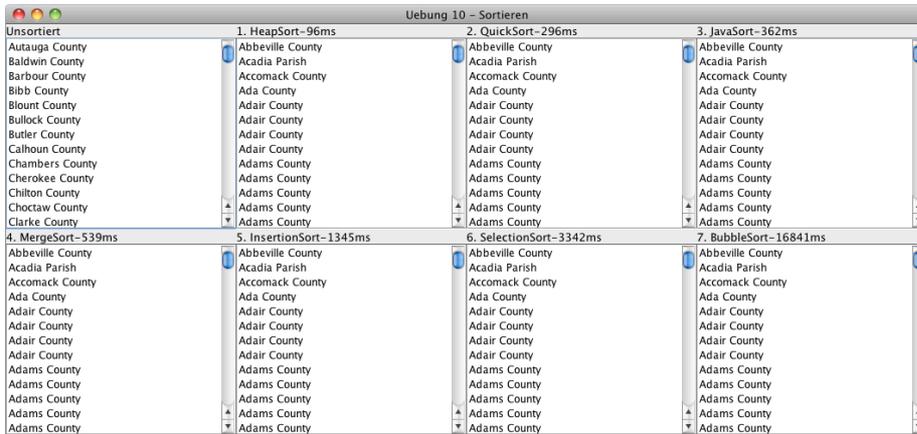
```
Sorter heapSort = new HeapSorter();
Integer[] values = new Integer { 10, 3, 1, 7, 6, 5, 4, 2, 9, 8 };
heapSort.sort(values);
for (int x: values) {
    Out.print(" " + x);
} // Ausgabe: 1 2 3 4 5 6 7 8 9 10
```

Implementieren Sie die Klassen *HeapSorter* und *MergeSorter* mit folgenden Schnittstellen (alles *public*):

```
class HeapSorter implements Sorter {
    void sort(Comparable[] items) { ... }
}
class MergeSorter implements Sorter {
    void sort(Comparable[] items) { ... }
}
interface Sorter {
    void sort(Comparable[] items);
}
```

Die *sort*-Methode sortiert die *Comparable*-Objekte im übergebenen Array.

Testen Sie Ihre Implementierung mit dem vorgegebenen Programm *SortTest.java*. Dieses Programm sortiert Strings, misst die benötigte Zeit und vergleicht Ihren *HeapSorter* und *MergeSorter* mit fünf weiteren Sortierverfahren.



Abzugeben ist: Java-Programm, Testergebnisse