Übungsblatt 12

Aufgabe 1 (Rekursives Mergesort, 5 Punkte)

Betrachten Sie folgende partielle Implementierung von Mergesort aus der Vorlesung:

```
public void mergesort(int [] a, int links, int rechts) {
   if (links < rechts) {
     int mitte = (links + rechts) / 2;
     mergesort(a, links, mitte );
     mergesort(a, mitte + 1, rechts);
     reisverschluss(a, links, mitte, rechts);
   }
}
```

Vervollständigen Sie das Programm mit einer Implementierung der noch fehlenden Methode `reisverschluss(int a[], int links, int mitte, int rechts)`. Der Gesamtaufwand der Implementierung soll $O(n \log n)$ nicht übersteigen. Ihre Implementierung darf keine Ein- oder Ausgabeanweisungen enthalten. Die Methode reisverschluss kopiert die im Rekursionsschritt sortierten Elemente von `a[links]` bis `a[mitte]` sowie diejenigen von `a[mitte+1]` bis `a[rechts]` nach dem Reisverschlussprinzip in ein weiteres Feld `int [] b`, so dass danach alle Elemente von `a[links]` bis `a[rechts]` in Feld `b` aufsteigend sortiert sind. Danach müssen innerhalb der Methode die Elemente von `b` nach `a[links]` bis `a[rechts]` kopieren werden. Testen Sie Ihre Implementierung mit einer JUnit-Testklasse.

Aufgabe 2 (Bottom Up Mergesort, 5 Punkte)

Mergesort lässt sich auch ohne Rekursion implementieren.

In einem ersten Durchgang werden alle zwei nebeneinander stehenden Werte verschmolzen, dann alle zwei nebeneinander stehenden Zweiergruppen, dann Viergruppen und so weiter.

Das folgende Beispiel zeigt alle drei Durchläufe: die zu verschmelzenden Bereiche sind mit eine senkrechten Strich getrennt, die linke und rechte Teilfolge mit einem Punkt.

```
8.7 | 6.5 | 4.3 | 2.1
7 8 . 5 6 | 3 4 . 2 1
5 6   7 8 . 1 2   3 4
1 2   3 4   5 6   7 9
```

Implementieren Sie bottom-up-Mergesort.