

## Übungsblatt 02

Zu bearbeiten bis 13.11.2008, 24h

### Programmieraufgabe 2

Schreiben Sie eine Funktion

```
public static int[] mergearray(int[] a, int[] b),
```

die 2 aufsteigend sortierte Felder in aufsteigender Folge zusammenführt und das zusammengeführte Feld als Ergebnis zurückgibt. Die Felder können unterschiedlich lang sein.

Geben Sie diese Funktion in einem Testprogramm ab. Rufen Sie die Funktion mit „guten“ Testbeispielen auf! (Was sind gute Testbeispiele?)

### Hausaufgabe 2

Gegeben sei das folgende Zahlenfeld von Integer-Werten:

12, 8, 4, 6, 1, 5, 3, 14, 16, 10, 7, 9, 2, 11, 13, 15

Sortieren Sie die Folge mit Hilfe von

- a) Insertion-Sort
- b) Bubble-Sort
- c) Selection-Sort
- d) Merge-Sort

Geben Sie den Inhalt des Zahlenfeldes nach jedem Durchlauf der äußeren Schleife an. Geben Sie für Merge-Sort, wie in der Vorlesung gezeigt, zusätzlich die Zerlegungs- und Zusammenfassungsschritte an.

Bitte als PDF-Datei einchecken.

### Präsenzaufgabe 2

- a) Die Binomialkoeffizienten nicht negativer Zahlen  $0 \leq k \leq n$  lassen sich wie folgt rekursiv definieren:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \text{ mit } \binom{n}{0} = \binom{n}{n} = 1$$

Schreiben Sie eine rekursive Funktion

```
public static long rbiko(int n, int k)
```

zur Berechnung des Binomialkoeffizienten.

b) Schreiben Sie einen rekursiven Sudoku-Löser.

Gehen Sie dazu wie folgt vor: Ein Feld `int[][] s = new int[9][9]` enthalte das Rätsel. Die noch zu füllenden Felder enthalten den Wert 0. Beginnend beim Feld `s[0][0]` wird probiert: Ist das Feld besetzt, geht es zum nächsten Feld. Ist das Feld noch 0, so werden nacheinander alle Zahlen 1-9 eingesetzt und geprüft, ob diese Lösung zulässig ist. Wenn nein, geht es mit der nächsten Zahl weiter. Wenn ja, mit dem nächsten Feld. Eine unzulässig gesetzte Zahl muss wieder entfernt werden. Um die Zulässigkeit der an der Position `s[zeile][spalte]` gesetzten Zahl zu prüfen, implementieren Sie boolesche Funktionen `pruefeZeile(int z)`, `pruefeSpalte(int s)`, `pruefeDreierblock(int z, int s)`.

Abzugeben bis Donnerstag, 13.11.2008, 24h in Gruppenverzeichnis auf <https://svn.uni-koblenz.de/oopm0809/students>