Algorithmen und Programmierung II

SS 07
M. Al Saad

Aufgabenblatt 6

Besprechungstermin: 11.-14.06.2007

Aufgabe 1:

In der Kombinatorik heißt eine Teilmenge der Mächtigkeit k (von einer Grundmenge der Mächtigkeit n) eine Kombination k-ter Klasse. Beispiel: Grundmenge $G = \{a, b, c, d\}$. Kombinationen 2-ter Klasse von G sind: $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, $\{c, d\}$.

Für die Anzahl C_n^k der Kombinationen k-ter Klasse gilt:

$$C_n^k = \binom{n}{k} = \frac{n!}{(n-k)! \cdot k!}$$

Schreiben Sie eine Java-Funktion zur Berechnung von C_n^k.

Aufgabe 2:

Schreiben Sie eine Methode `Lineare Suche`, die angewendet auf ein Feld `double[]a` und einen Wert `double w` den kleinsten Index `i` ausgibt für den gilt `a[i]=w`, falls es ein solches `i` gibt und sonst `-1` ausgibt.

Hinweis: Eine elegante Lösung besteht zunächst aus der Programmierung einer verallgemeinerten Aufgabe (lineare Suche innerhalb gewisser Grenzen).

Aufgabe 3:

Schreiben Sie eine Methode zur Berechnung der laufenden Summen, die folgender Spezifikation genügt:

```
/**  Erzeuge ein Feld b mit gleicher Länge wie das Feld a und trage in b[i] den
     Wert summe a[0..i] ein. Gib b als Wert zurück.
     Pre: a!= null
 */
int [ ] laufsum (int [ ] a)
```

Aufgabe 4:

Schreiben Sie eine Methode `einfügen`, die folgender Spezifikation genügt:

```
/**  Wenn k in a[0..r] vorkommt, gib r zurück, sonst füge k an die richtige Stelle
     ein und gib r + 1 als Wert zurück. Pre: a!= null, a[0..r] ist sortiert und
     r < a.length-1
 */
int einfügen (int [ ] a, int r, int k)
```