

Übungsblatt 11 **Ausgabe: Mi, 19.01.00** **Abgabe: Di, 25.01.00, 18 Uhr**

Aufgabe 1: Rekurrenzrelationen [4 Punkte]

Bestimmen Sie die Rekurrenzrelation der Kostenfunktion $T(n, a)$ der folgenden Java-Funktion und geben Sie mit Hilfe des O-Kalküls das asymptotische Verhalten der Kostenfunktion an. Bitte ausführlich begründen!

```
int f (int n, int a) {  
    if (n <= 1)  
        return 0;  
    else  
        return (f(n-1,a) + f(n-1, a/2));  
}
```

Aufgabe 2: Komplexitätsberechnung [6 Punkte]

Den drei unten gegebenen Programmstücken gehe folgender Vereinbarungsteil voraus:

```
const int c = 5;  
int s, i, j, n;
```

Bestimmen Sie die Zeitkomplexität dieser Programmstücke (die Additionsoperationen seien in konstanter Zeit durchführbar).

1.

```
s = 0;  
for (i = 1; i <= n; i++)  
    for (j = 1; j <= 2*n; j++)  
        s = s+i+j;
```
2.

```
s = 0;  
for (i = 1; i <= n; i++)  
    for (j = i; j <= 2*n; j++)  
        s = s+i+j;
```
3.

```
s = 0;  
for (i = 1; i <= n; i++)  
    for (j = 1; j <= c; j++)  
        s = s+i+j;
```

Aufgabe 3: B-Bäume (Abgabe per abox) [10 Punkte]

Entwickeln Sie eine Java Applikation zur Verwaltung von B-Bäume beliebiger Ordnung! Die Applikation soll in form einer Klasse BTree zur Sortierung natürlicher Zahlen \mathbf{N}^+ realisiert werden, die folgende Methoden implementiert:

```
public class BTree  
{  
    // Konstruktor erzeugt einen B-Baum n-ter Ordnung (mit n>0)  
    public BTree(int n);  
  
    // Sortiert eine Zahl z in den Baum ein, sofern diese noch nicht im Baum  
    // enthalten ist.  
    public void insert(int z);  
}
```

```

// Prueft, ob eine Zahl z im Baum enthalten ist.
public boolean contains(int z);

// Gibt alle im Baum enthaltenen Zahlen in sortierter Folge als String
// aus. Einzelne Zahlen werden durch Leerzeichen getrennt.
public String toString();

// Berechnet die Höhe des Baumes.
public int height();
}

```

Schreiben Sie eine statische main-Methode, die eine rudimentäre Kommandosprache zum Zugriff auf die Zahlen im Baum ermöglicht. In dieser Sprache gibt es die folgenden Befehle (s. auch Übungsblatt 7):

```

create zahl Initialisiert einen B-Baum entsprechender Ordnung.
add zahl    Einfügen eines neuen Eintrages, Duplikate sind nicht erlaubt.
            Keine Ausgabe.
search zahl Ausgabe "true", falls Zahl im Baum sonst "false"
height      Ausgabe der Höhe des Baumes
print       aufsteigend sortierte Ausgabe aller Einträge.

```

Das Programm beginnt per default mit einem Baum der Ordnung zwei. Wird der Befehl create ausgeführt wird der vorherige Baum verworfen.

Anmerkung: Das Programm muß als Kommandozeilenparameter den Namen einer Datei erwarten, die eine Folge von Befehlen der Kommandosprache enthält. Ein Beispiel für eine solche Kommandodatei finden Sie im Web¹.

Hier ein Beispiel für einen Kommandozeilenaufruf und der Ergebnisausgabe des von Ihnen zu entwickelnden Programmes:

```

kommandos.txt:                                     > java BTree kommandos.txt
create 2                                           1 2 3
add 1                                             true
add 2                                             2
add 3                                             false
print
search 1
height
search 4

```

¹<http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt11/kommandos.txt>