

Algorithmen und Datenstrukturen

Sommersemester 2012

3. Übungsblatt

Aufgabe 1: Stackverwaltung in rekursiven Methoden

Eine rekursive Lösung des Problems der Türme von Hanoi sei analog zur Vorlesung wie folgt gegeben:

```
1 public static void rekhanoi(int start, int ziel, int hilf, int hoehe){
2     /* Start von rekhanoi */
3     if (hoehe>0) {
4         rekhanoi(start, hilf, ziel, hoehe-1) /* nach 1. Rekursion */;
5         zug(start, ziel);
6         rekhanoi(hilf, ziel, start, hoehe-1) /* nach 2. Rekursion */;
7     }
8 }
9
10 public static void main(String[] args){
11     rekhanoi(1,2,3,2) /* nach Aufruf in main */;
12 }
```

Zeichnen Sie den gesamten Programmstack (vgl. Kap. 0.6 Folie 113) jeweils an der mit `start` von `rekhanoi` bezeichneten Programmstelle. Verwenden Sie zur Bezeichnung der Rückkehrpunkte die im obigen Programm angegebenen Kommentare. (9 Punkte)

Aufgabe 2: O- und Ω -Notation

Beweisen Sie folgende Behauptungen:

- a) $T_1(n) = n^3 + n - 100$ ist $O(n^3)$ (2 Punkte)
- b) $T_2(n) = n^4 - 100 * n^3$ ist nicht $O(n^3)$ (2 Punkte)
- c) $T_3(n) = 3^n$ ist nicht $\Omega(4^n)$ (2 Punkte)
- d) Für das Produkt zweier Laufzeiten $T_1(n)*T_2(n)$ gilt, dass es $O(f(n)*g(n))$ ist. (2 Punkte)

Aufgabe 3: Laufzeitabschätzung

In dieser Aufgabe soll die Laufzeit von zwei Algorithmen abgeschätzt werden. Allgemein kann angenommen werden, dass eine Wertzuweisung, Schleifeninitialisierung und Bedingungsüberprüfung jeweils einen Schritt benötigt.

- a) Schätzen Sie die Laufzeit für den folgenden einfachen Sortieralgorithmus in **O**-Notation ab und leiten Sie Ihre Abschätzung her. Nehmen Sie an, dass das Einfügen eines Elements an beliebiger Position in einer Liste eine Laufzeit **O(1)** erfordert. (4 Punkte)

```
1 public class Sorter<T extends Comparable<T>> {
2
3     public List<T> simpleSort(List<T> a) {
4         int n = a.size();
5         List<T> b = new ArrayList<T>();
6
7         for (int i = 0; i < n; i++) {
8             boolean inserted = false;
9             for (int j = 0; j < b.size(); j++) {
10                if ((b.get(j)).compareTo(a.get(i)) > 0) {
11                    b.add(j, a.get(i));
12                    inserted = true;
13                    break;
14                }
15            }
16            if (!inserted) {
17                b.add(a.get(i));
18            }
19        }
20        return b;
21    }
22 }
```

- b) Schätzen Sie die Laufzeit für den folgenden rekursiven Algorithmus in **O**-Notation ab und leiten Sie Ihre Abschätzung her, indem Sie eine Rekursionsgleichung aufstellen. Der Parameter n sei eine natürliche Zahl größer 0. (4 Punkte)

```
1 boolean istZweierpotenz(int n){
2     if(n==1)return true;
3     if(n%2!=0)return false;
4     return istZweierpotenz(n/2);
5 }
```

Beachten Sie die Hinweise auf dem ersten Übungsblatt.

Abgabetermin: Donnerstag, den 26. April bis 10 Uhr im Institut für Informationssysteme, 2. OG im Informatik-Neubau (Raum 2.071, Küche neben Sekretariat)