Algorithmen und Datenstrukturen I

WS 2009/10, **1. Aufgabenblatt**, Abgabe 28.10.2009

Aufgabe 1

10 Punkte

Gegeben sind die folgenden 6 Funktionen

a) $n \mapsto n\sqrt{n} + 2$

b) $n \mapsto 3^{n/2}$

c) $n \mapsto \frac{n^3}{\log(n+19)}$

d) $n \mapsto 19n^2 + n\cos(\pi n)$

e) $n \mapsto 5\sqrt[4]{n} + 2^{\log(3/(n+1))}$

f) $n \mapsto 10^7 \frac{n}{n+19}$

Bringen Sie die Funktionen in eine Reihenfolge $f_1, f_2, f_3, f_4, f_5, f_6$, so dass $f_1 \in O(f_2)$, $f_2 \in O(f_3)$, $f_3 \in O(f_4)$ usw. Begründen sie kurz jede dieser 5 Beziehungen anhand der Definition des O-Symbols.

Aufgabe 2

10 Punkte

Gegeben seien die folgenden Rekursionsgleichungen jeweils für eine Funktion T.

(a) $T(n) = 9T(n/3) + 2n^2 + n$

(b) $T(n) = T(n/2) + n^5 + 1$

(c) $T(n) = 27T(n/3) + 100n^2 + \sqrt{n}$

(d) $T(n) = nT(n-1)$

(e) $T(n) = T(n/2) + \log_2 n$

Bestimmen Sie jeweils eine exakte Schranke für das Verhalten von T. Finden Sie also eine Funktion f, so daß $T \in \Omega(f)$. In den Fällen, wo die polynomiale Version des Master-Theorems anwendbar ist, benutzen Sie dieses zum Finden von f und geben a, b und k an.

Aufgabe 3 10 Punkte

In der Vorlesung haben sie Verfahren zum effizienten Suchen in sortierten Listen kennen gelernt. Die Binärsuche lässt sich u.a. rekursiv implementieren, in Java für ganze Zahlen beispielsweise so:

```
public class Search {
  public static int binSearch (int[] f, int x, int l, int r) {
    int p=(l+r)/2;
    int c=(f[p]<=x ? f[p]=x ? 0 : 1 : -1);
    if (c==0) return(p);
    if (l==r) return(-1);
    if (c<0) {
      if (p>l) return(binSearch(f,x,l,p-1));
      else return(-1);
    }
    else {
      if (p<r) return(binSearch(f,x,p+1,r));
      else return(-1);
    }
  }
}
```

Geben Sie für `f[] = [1, 2, 4, 5, 8, 11, 14, 17, 19, 23, 25, 31, 37, 41, 43, 47];` an, wie die Methode `binSearch` auf f arbeitet, wenn nach folgenden Elementen gesucht wird:

(a) x=14

(b) x=45

Geben Sie für beide Teilaufgaben, die initialen Funktionsaufrufe von f an, d.h. überlegen Sie sich wie l und r anfangs zu belegen sind und protokollieren Sie für jeden Rekursionsaufruf die Werte der Variablen l, r und p, sowie den Rückgabe Wert von f.

Aufgabe 4 5 Punkte

Eine lineare Liste $L[]$ der Länge n sei wie folgt *teilsortiert*: Für alle Indizes $k, l \in \{1, \ldots, n\}$ gilt

$$a[k] > a[l] \;\Rightarrow\; k \geq l - 1$$

Wenn also zwei Elemente in der falschen Reihenfolge stehen, dann steht das Größere direkt vor dem Kleineren.

Geben Sie eine möglichst einfache Modifikation der Binärsuche aus der Vorlesung an, die $L[]$ auf das Vorhandensein eines gegebenen Schlüssels überprüft. Die maximalen Kosten (Laufzeit) sollen in $O(\log n)$ liegen.