## Algorithmen und Programmieren 3, WS 1999/2000 — 1. Übungsblatt

Abgabe bis Dienstag, 2. November 1999

---

1. (2 Punkte) Die Computer werden immer schneller, die Speicherelemente immer billiger. Wird der Entwurf von effizienten Algorithmen angesichts dieser Entwicklung in Zukunft an Bedeutung verlieren? Bei welchen Computeranwendungen werden Effizienzfragen eine größere/kleinere Rolle spielen?

   Diskutieren Sie diese Fragen. (mindestens 5–10 Zeilen)

2. (0 Punkte) Das folgende Programm ist ein Versuch, Sortieren durch Einfügen zu implementieren.

```
     void insertionSort(int[] a)
     { int x;
       for (int i = 0; i<a.length; i++)
       { † x = a[i];
         // Sortiere x zwischen a[0..i-1] ein
         // Bestimme zunächst die richtige Position j:
(*)      int j=i-1; † while (j>=0 && a[j-1]>x) j--† ;
         // Nun verschiebe einen Teil des Feldes und füge x an dieser Stelle ein:
       † for (int k = i-1; k>=j; k--) a[k+1]=a[k];
         a[j]=x; †
       }
     }
```

   Stellen Sie dieses Programm richtig und fügen Sie an den durch † bezeichneten Stellen in Ihrem Programm Schleifeninvarianten in der Form vom Zusicherungen (assertions) ein, aus denen die Korrektheit des Programmes hervorgeht. (Ein formaler Beweis ist nicht erforderlich, aber die Zusicherungen müssen aussagekräftig sein und vor allem zutreffen.)

3. (6 Punkte) Kann man Sortieren durch Einfügen schneller machen, indem man die Einfügestelle j schneller findet als oben in Zeile (∗), zum Beispiel durch binäres Suchen? Wie ist es im besten und im schlechtesten Fall?

4. (0 Punkte) Zeigen Sie:

   (a) Wenn $f(n) = a_d n^d + a_{d-1} n^{d-1} + \cdots + a_1 n + a_0$ ein Polynom vom Grad $d$ mit positivem Leitkoeffizienten ($a_d > 0$) ist, dann ist
   $$f(n) = \Theta(n^d).$$

   (b) $\log_a n = \Theta(\log_b n)$ für $a, b > 1$.

   (c) Unter welchen Bedingungen folgt $g(n) = \Theta(f(n))$ aus $f(n) = O(g(n))$?

5. (7 Punkte) Beweisen Sie:

   (a) Wenn $f(n) = O(g(n))$ ist, dann ist $f(n) + g(n) = O(g(n))$.

   (b) $\max\{f(n), g(n)\} = \Theta(f(n) + g(n))$ für $f(n), g(n) > 0$.

   (c) Welche der folgenden Aussagen sind richtig (Begründen Sie Ihre Antworten):
   $n\sqrt{n} = O(n(\log n)^2)$; $n(\log n)^2 = O(n\sqrt{n})$; $n \log n \cdot (\log\log n) = O(n^2)$;
   $n^2 = O(n \log n \cdot (\log\log n))$; $n^2 \cdot 2^n = O(2^{n+2})$; $n^2 \cdot 2^n = O(3^n)$; $3^n = O(n^2 \cdot 2^n)$;

   (d) Finden Sie möglichst einfache Ausdrücke der Form $O(\cdot)$ für folgende Funktionen:
   $3n^2 - 4n + 32 + 27\, n \cdot \lceil \log_2 n \rceil / 2$; $\max\{n \lceil \log_2 n \rceil, (\lceil \log_2 n \rceil)^4\}$; $2^{2n + \lceil \log_2 n \rceil}$

6. (5 Punkte) Formulieren Sie *Sortieren durch Auswahl* in einer Programmiersprache Ihrer Wahl.