Vorsemesterkurs Informatik

Sommersemester 2014

Aufgabenblatt Nr. 6A

Aufgabe 1 (Vorbereitung der $\mathcal{O}$-Notation)

a) Entscheide für die folgenden Paare von Funktionen $f, g\colon \mathbb{N} \to \mathbb{R}$, ob es eine konstante Zahl $c \in \mathbb{R}$ gibt, so dass $f \leq c \cdot g$ gilt. Wenn ja, gib eine solche Zahl c an.

   i) $f(n) = 4n$ und $g(n) = 2n$
   ii) $f(n) = n$ und $g(n) = \frac{1}{2}n^2$
   iii) $f(n) = n^3$ und $g(n) = n^2$
   iv) $f(n) = 2n^3$ und $g(n) = \begin{cases} n^2 & \text{falls } n \text{ gerade,} \\ n^3 & \text{falls } n \text{ ungerade.} \end{cases}$
   v) $f(n) = e^n$ und $g(n) = n!$ (Erinnerung: Es gilt $0! = 1$)
   vi) $f(n) = \sqrt{n}$ und $g(n) = \frac{1}{2}n$
   vii) $f(n) = \sqrt{n}$ und $g(n) = \log n$ (hierbei sind $f, g\colon \mathbb{N} \setminus \{0\} \to \mathbb{R}$)

b) Entscheide für die folgenden Paare von Funktionen $f, g\colon \mathbb{N} \to \mathbb{R}$, ob es ein $n_0 \in \mathbb{N}$ gibt, so dass für alle $n \geq n_0$ gilt: $f(n) \leq g(n)$. Wenn ja, dann gib ein solches n_0 an.

   i) $f(n) = 4n$ und $g(n) = 2n$
   ii) $f(n) = n$ und $g(n) = \frac{1}{2}n^2$
   iii) $f(n) = n^2$ und $g(n) = 3n^2$
   iv) $f(n) = 4\log_2 n$ und $g(n) = n$ (hierbei sind $f, g\colon \mathbb{N} \setminus \{0\} \to \mathbb{R}$)
   v) $f(n) = 2n^3$ und $g(n) = \begin{cases} n^2 & \text{falls } n \text{ gerade,} \\ n^3 & \text{falls } n \text{ ungerade.} \end{cases}$
   vi) $f(n) = \sqrt{n}$ und $g(n) = \frac{1}{2}n$
   vii) $f(n) = e^n$ und $g(n) = n!$

Aufgabe 2 (Wachstum von Funktionen)

Ordnen Sie die folgenden Funktionen nach ihrer Größe, wenn n gegen ∞ geht. Unterstreichen Sie dabei in jeder Funktion den Summanden, der am größten wird.

a) $n \cdot e^n + n$

b) $n \cdot \ln(n)$

c) $n^2 + 3n + 1$

d) $n^2 + 2n - 5$

e) $\log_3(n) + 1000$

f) $n^{31} + e^n$

Aufgabe 3 (ausschlaggebende Terme)

Sortieren Sie die Terme innerhalb der folgenden Funktionen nach ihrer Größe, wenn n gegen ∞ geht. Geben Sie dann für jede der Funktionen $f(n)$, die kleinstmögliche Funktion $g(n)$ an, sodass $f = O(g)$ gilt.

a) $f(n) = n \cdot (e^n)^2 + n^2 \cdot e^n$

b) $f(n) = 100 \cdot \ln(n) + n \cdot \ln(n)$

c) $f(n) = n^2 + 20\ln(n)^2 + 4 \cdot n \cdot \ln(n)^2$

d) $f(n) = 3 \cdot n^{20} + 3n \cdot \ln(n)^{30} - 5$

e) $f(n) = n^{31} + e^{4 \cdot n} + e^n$

Aufgabe 4 (Pseudocode)

Betrachten Sie die folgenden Algorithmen in Pseudocode.

a) Versuchen Sie zu verstehen, welche Rechenvorschriften diese Algorithmen darstellen. (Auch wenn diese manchmal keinen tieferen Sinn haben)

b) Geben Sie die Laufzeit in der O-Notation an.

Algorithmus 1:

```
function algo_1(A[0..n]){
  zahl=A[0];
  for (int i = 1; i <= n; i++){
    if(A[i]<=zahl){
      zahl=A[i];
    }
  }
  return(zahl);
}
```

Algorithmus 2:

```
function algo_2(n){
  i = 1;
  for (int i = 1; i <= n; i++){
    for(int j = 3; j <= 3n; j++){
        for (int l = -3; l <= n/2; l++){
          print i + j - l;
        }
    }
  }
}
```

Algorithmus 3:

```
function algo_3(n){
  i = n;
  while (1 <= i){
    i = i / 3;
  }
}
```

Algorithmus 4:

```
function algo_4(n){
  for (int i = 1; i <= n*n; i = i * n){
    print i;
  }
}
```

Algorithmus 5:

```
function algo_5(A[0..n]){
  zahl=A[0];
  for(int i = 1; i<=n, i++){
    zahl=zahl+A[i]
  }
  return(zahl/n)
}
```