1. Tutorium - Algorithmen I

Nina Zimbel

22.04.2015

1. Organisatorisches
2. Pseudocode
3. Laufzeitanalyse
4. Schleifeninvarianten

Pseudocode

```
Function do (a:Array) : Array
    Number p, Array b
    for j:= 0 to a.length do
        p = a[j]
        for i := j to a.length do
            if a[i] < p then p = a[i]
            i++
        b[j] = p, j++
    return b
```

Zweite Vereinfachung: Asymptotik

$$O(f(n)) = \{g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

„höchstens"

$$\Omega(f(n)) = \{g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

„mindestens"

$$\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$$

„genau"

$$o(f(n)) = \{g(n) : \forall c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

„weniger"

$$\omega(f(n)) = \{g(n) : \forall c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

„mehr"

Schleifeninvarianten

- **Vorbedingung**: Bedingung für korrektes funktionieren der Schleife
- **Invariante**: gilt vor und nach jedem Schleifendurchlauf
- **Nachbedingung**: Leisungsgarantie der Schleife

- wird für Korrektheitsbeweise verwendet

Aufgaben

- Erklären Sie warum die Aussage
  "Die Laufzeit eines Algorithmus A beträgt mindestens O(n^2)."
  bedeutungslos ist.
- Gilt $2^{n+1} = O(2^n)$?
- Gilt $2^{2n} = O(2^n)$?
- Beweisen Sie: o(g(n)) $\cap\ \omega(g(n)) = \emptyset$
- Beweisen Sie:
  Die Laufzeit eines Algorithmus A beträgt $\Theta(g(n)) \Leftrightarrow$
  Die Laufzeit von A ist im schlechtesten Fall O(g(n)) und im günstigsten Fall $\Omega(g(n))$