Grundzüge der Informatik 1

Vorlesung 3 – flipped classroom

Pseudocodebefehle und Laufzeiten - Vorüberlegungen

Rechenmodell

- Idee: Ignoriere rechnerabhängige Konstanten
- Eine Pseudocodeoperation benötigt einen Zeitschritt
- Wird eine Instruktion im Laufe des Algorithmus k-mal ausgeführt (z.B. in Schleifen), so benötigt sie insgesamt k Zeitschritte

Formales Modell

- Random Access Machines (RAM Modell)
- Details unterscheiden sich von unserem Modell

Pseudocode und Laufzeiten

Zusammenfassung

- Einzelne Pseudocode Befehle brauchen einen Rechenschritt
- Befehle, die k-mal ausgeführt werden, benötigen auch k Rechenschritte
- Tritt eine Schleife k mal in den Schleifenrumpf ein, so wird das Schleifenkonstrukt (k+1)-mal aufgerufen und der Schleifenrumpf wird k-mal ausgeführt
- Daraus ergibt sich die Anzahl der benötigten Rechenschritte
- Beim Aufruf von Prozeduren werden die übergebenen Elemente kopiert
- Pseudocode kann von diesen Definitionen abweichen, wenn es die Beschreibung des Algorithmus erleichtert

Laufzeitanalyse

Herangehensweise

- Beschreibe Laufzeit als Funktion der Eingabegröße n
- Ziel: Finde obere Schranken (Garantien) für die Laufzeit eines Algorithmus

Definition (Worst Case Analyse)

- Für jedes n definiere die Worst-Case Laufzeit T(n) durch
- T(n) = maximale Laufzeit über alle Eingaben der Größe n

Definition (Average Case Analyse)

- Für jedes n definiere die Average-Case Laufzeit T(n) durch
- T(n) = durchschnittliche Laufzeit über alle Eingaben der Größe n
- Benötigt Definition von „durchschnittlich“

Pseudocode und Laufzeiten

```
Algorithmus4(n)
i=n
j=0
while i>0 do
    j=j+i
    i=i-1
output << j
```

Aufgabe 1

- Was ist die (Worst-Case) Laufzeit von Beispiel4(n)?

Laufzeitanalyse

Algorithmus5(A, n)

```
for i=1 to n do
    if A[i] ist gerade then
        for j=1 to n do
            k=k+1
```

Aufgabe 2

- Was ist die Worst-Case Laufzeit von Algorithmus5(A, n)?

Landau Notation

O-Notation

- $O(g(n)) = \{f(n)$: Es gibt positive Konstanten c und n_0, so dass für alle $n \geq n_0$ gilt: $0 \leq f(n) \leq c \cdot g(n)\}$

Interpretation

- $f(n) \in O(g(n))$ bedeutet, dass $f(n)$ höchstens so stark wächst wie $g(n)$ für $n \to \infty$
- Dabei ignorieren wir beim Wachstum konstante Faktoren
- Die O-Notation liefert eine obere Schranke
- Die Funktionen $f(n)$ und $g(n)$ müssen asymptotisch nicht-negativ sein (für n groß genug, sind die Funktionen nicht-negativ)

Landau Notation

Aufgabe 3

- Beweisen oder wiederlegen Sie:
- $5n^2 + 10\,n \in O(n^2)$

Landau Notation

Aufgabe 4

- Beweisen oder wiederlegen Sie:
- $5n \log n + 10\, n \in O(n)$

Landau Notation

Aufgabe 5

- Die Laufzeit von InsertionSort war $T(n) = (3n^2+7n-8)/2$. Was ist eine möglichst gute Abschätzung in der O-Notation?
- Welche der folgenden Aussagen sind wahr?
- $n^3 \in O(n^2)$
- $\log^5 n \in O(\sqrt{n})$
- Finden Sie eine möglichst gute Abschätzung in der O-Notation für:
- $130n + 2n^2 + 15 n \log n + 12 \log n$
- $150n^{10} + 20 \log n + 21 n^{10} \log n$