Datenstrukturen und Algorithmen: Übungsblatt #2

Abgabe am 19. April 2018

Finn Hess (378104), Jan Knichel (??????), Paul Orschau (381085)

22.04.2018

Aufgabe 1

Wir definieren die Quasiordnung $\sqsubseteq$ auf Funktionen als
$f \sqsubseteq g \Leftrightarrow f \in \mathcal{O}(g)$.

a) Beweisen Sie, dass $\sqsubseteq$ eine Quasiordnung ist, das heißt dass $\sqsubseteq$ reflexiv und transitiv ist.

b) Sortieren Sie einige Funktionen in aufsteigender Reihenfolge bezüglich der Quasiordnung $\sqsubseteq$.

Lösung

Teil a)

Teil b)

Aufgabe 2

Beweisen oder wiederlegen Sie folgende Aussagen:

a) $\frac{1}{4}n^3 - 7n + 17 \in \mathcal{O}(n^3)$

b) $n^4 \in \mathcal{O}(2n^4 + 3n^2 + 42)$

c) $\log(n) \in \mathcal{O}(n)$

d) $\forall \epsilon > 0 : \log(n) \in \mathcal{O}(n^\epsilon)$

e) $a^n \in \Theta(b^n)$, für zwei beliebige Konstanten $a, b > 1$

Lösung

Teil a)
Teil b)
Teil c)
Teil d)
Teil e)

Aufgabe 3

a) Zeigen oder wiederlegen Sie:

$$o(g(n)) \cap \Theta(g(n)) = \emptyset$$

b) Zeigen oder widerlegen Sie:

$$f(n) \in \Omega(g(n)) \wedge f(n) \in \mathcal{O}(h(n)) \implies g \in \Theta(h(n))$$

Lösung

Teil a)

Teil b)

Aufgabe 4

Gegeben sei ein Algorithmus, der für ein Array von Booleans überprüft, ob alle Einträge wahr sind:

```
int allTrue(bool [] E) {
  if (E.length < 1) {
    return -1;
  }
  int m = E.length;
  int i = 0;
  while (i<E.length) {
    if (E[i] == true) {
      m = m - 1;
      if (m == 0) {
        return 1;
      }
    }
    i = i + 1;
  }
  return 0;
}
```

Bei Betrachtung der Laufzeit wird angenommen, dass Vergleiche (z.B. $x < y$ oder $b == 0$) jeweils eine Zeiteinheit benötigen. Die Laufzeit aller anderen Operationen wird vernachlässigt.
Sei n die Länge des Arrays E.

a) Bestimmen Sie in Abhängigkeit von n die Best-case Laufzeit $B(n)$.

b) Bestimmen Sie in Abhängigkeit von n die Worst-case Laufzeit $W(n)$.

c) Bestimmen Sie in Abhängigkeit von n die Average-case Laufzeit $A(n)$. Hierzu nehmen wir eine uniforme Verteilung der möglichen Eingaben D_n an, d.h. jede beliebige Eingabe $E \in D_n$ tritt mit Wahrscheinlichkeit $1/|D_n|$ auf.

d) Geben Sie einen äquivalenten Algorithmus an, dessen Average-case Laufzeit ab einer gewissen Eingabelänge kleiner ist. Hierzu nehmen wir wieder eine uniforme Verteilung der möglichen Eingaben an. Begründen Sie Ihre Antwort kurz. Sie müssen nicht die Average-case Laufzeit ihres Algorithmus berechnen.

e) Gibt es einen äquivaltenen Algorithmus, dessen Worst-case Laufzeit in $o(n)$ liegt? Begründen Sie ihre Antwort.

Lösung

Teil a)
Teil b)
Teil c)
Teil d)
Teil e)