Algorithmen & Datenstrukturen 2

Wintersemester 2022/23

Prof. Dr. Martin Hoefer
Tim Koglin, Marco Schmalhofer

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
Algorithmen und Komplexität

Übungsblatt 0

Ausgabe: 18.10.2022
Abgabe: keine

Bei diesem Übungsblatt handelt es sich um eine **Präsenzübung**. Eine Abgabe erfolgt nicht. Die Aufgaben werden in den ersten Tutorien im Zeitraum vom 31.10. bis zum 04.11. besprochen.

Um einer Übungsgruppe zugewiesen zu werden, und somit am Übungsbetrieb teilnehmen zu können, müssen Sie bis spätestens **Donnerstag, den 20. Oktober 2022** um 23:55 Uhr Ihre Terminpräferenzen im AUGE-System angegeben haben. Sie werden informiert, sobald die Zuweisung feststeht. Weitere Informationen zum Übungsbetrieb finden Sie auf der Webseite zur Veranstaltung.

Aufgabe 0.1 *Asymptotische Notation* (-)

Seien $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit zuweisen. Die asymptotische Notation ist folgendermaßen definiert:

$f = O(g) \iff$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass für alle $n \geq n_0$ gilt $f(n) \leq cg(n)$.

$f = \Omega(g) \iff g = O(f)$

$f = \Theta(g) \iff f = O(g),\ \ f = \Omega(g)$

a) Beweisen oder widerlegen Sie mit Hilfe der obigen Definition: $\sum_{i=1}^{n-1}(n-i) = \Theta(n^2)$

b) Beweisen oder widerlegen Sie: Für jede Funktion f gilt $f(n+1) = \Theta(f(n))$

c) Ordnen Sie die folgenden Funktionen $f_1, , ..., f_6$ gemäß ihres asymptotischen Wachstums beginnend mit der am langsamsten wachsenden Funktion.

$f_1(n) = \sqrt[5]{n^3}$

$f_2(n) = \sum_{i=1}^{n} \frac{1}{i}$

$f_3(n) = n \log_2(n)$

$f_4(n) = 2^{100} n$

$f_5(n) = (\sin^2(n) + \cos^2(n))^n$

$f_6(n) = n^{0.5} + 2^{\log_4 n}$

Aufgabe 0.2 *Rekursionsgleichungen* (-)

Bestimmen Sie eine Lösung der folgenden Rekursionsgleichungen in Θ-Notation. Gehen Sie davon aus, dass n eine Potenz von 3 ist.

a) $T(1) = 1, \quad T(n) = 27 \cdot T\left(\frac{n}{3}\right) + 8^{\log_2(n^3)}$

b) $T(0) = 0, \quad T(n) = 3 \cdot T(n-3) + 2$

Aufgabe 0.3 *Laufzeitanalyse und Pseudocode* (-)

Gegeben ist der folgende Algorithmus A für eine ganzzahlige Eingabe n:

Algorithmus A:

```
Sum = 0;
for (i = 1 ; i ≤ n ; i++){
    for (j = 1 ; j ≤ f(i); j++){
        Sum=Sum+1
    }
}
```

Welchen Wert hat die Variable *Sum* am Ende der Berechnung, wenn

a) $f(i) = n$?

b) $f(i) = n - i$?

c) $f(i) = \log(i)$? *Es genügt eine asymptotisch exakte Angabe in der Form* $\Theta(\cdot)$.

d) $f(i) = \sqrt{i}$? *Es genügt eine asymptotisch exakte Angabe in der Form* $\Theta(\cdot)$.

Aufgabe 0.4 *Randomisierte Algorithmen* (-)

Ein randomisierter Algorithmus A für ein Problem X habe die Laufzeit $a(n)$ auf allen Instanzen der Größe n, liefert jedoch mit Wahrscheinlichkeit $b(n)$ falsche Resultate (unabhängig von der konkreten Aufgabe). Zudem liegt ein Testalgorithmus ("Tester") C vor, mit dem die Korrektheit der Ausgaben von A überprüft werden können. Der Tester C benötigt $c(n)$ Schritte, um die Ausgabe von A zu einer Eingabe der Größe n zu prüfen.

a) Kombinieren Sie auf effiziente Weise den Algorithmus A mit dem Tester C zu einem randomisierten Algorithmus, der stets die korrekte Lösung liefert.

b) Was ist die erwartete Laufzeit dieses Algorithmus für allgemeine $a(n)$, $b(n)$ und $c(n)$?

c) Nehmen Sie an, dass $a(n) = n^2 \log n$, $b(n) = 0.7$ und $c(n) = n^5$ gilt.

   Was ist die (asymptotische) erwartete Laufzeit in Abhängigkeit von n?

   Zusatzfrage: Was geschieht, wenn $b(n) = \frac{n-1}{n}$?

Bei allgemeinen Anmerkungen zu den Übungsaufgaben oder Fragen zum Übungsbetrieb erreichen Sie uns unter der folgenden E-Mail-Adresse: algo222@cs.uni-frankfurt.de.