Fachbereich Informatik
Dr. Michael Haupt
Visar Januzaj
Johannes Kinder
Sommersemester 2010

TECHNISCHE UNIVERSITÄT DARMSTADT

Grundlagen der Informatik 2

Übungsblatt 2

Abgabe: 07.05.2010, 15:00

T 2.1 Komplexitätsquiz

Seien $k \geq 1$ und $c > 1$ Konstanten. Kreuzen Sie in der Tabelle an, ob $f(n) \in \mathcal{O}(g(n))$, $f(n) \in \Omega(g(n))$ oder $f(n) \in \Theta(g(n))$.

$\mathbf{f(n)}$	$\mathbf{g(n)}$	$\mathcal{O}$	Ω	Θ	$\mathbf{f(n)}$	$\mathbf{g(n)}$	$\mathcal{O}$	Ω	Θ
n^2+n+2	n^2				$n^{\log c}$	$c^{\log n}$			
$n^3+n^2+\log_4 n$	n^4				n^k	c^n			
2^n	$2^{n/2}$				$\log_2 n$	$\log_{10} n$			
$n \log n$	n^2				2^{n+k}	2^n			

T 2.2 Potenzen

Gegeben sei folgendes Programm zur Berechnung von ganzzahligen Potenzen. Zeitkritische Operation sei der Funktionsaufruf.

```
public static int potenz(int basis, int exponent) {
  if (exponent==1) return basis;
  if (exponent mod 2 == 0) {
    return potenz(basis,exponent/2)*potenz(basis,exponent/2);
  } else {
    return potenz(basis,exponent/2)*potenz(basis,exponent/2)*basis;
  }
}
```

a) Definieren Sie eine geeignete Problemgröße.

b) Erstellen Sie die Rekurrenzgleichungen.

c) Bestimmen sie die asymptotische Laufzeit in Θ-Notation und zeigen deren Korrektheit mittels vollständiger Induktion.

d) Wie müssen Sie den Algorithmus verändern, um eine asymptotisch bessere Laufzeit zu erreichen? Welche Laufzeit können Sie so erreichen?

T 2.3 Rekurrenzgleichungen

Bestimmen Sie (wenn möglich mit Hilfe des Master-Theorems) in welcher Komplexitätsklasse sich die folgenden Rekurrenzgleichungen befinden. Seien Sie so exakt wie möglich und argumentieren Sie nachvollziehbar, wie Sie auf Ihre Lösungen gekommen sind. Annahme: $T(i)$ sei jeweils konstant für $i \leq 2$.

- $T(v) = 2 \cdot T(\lfloor\sqrt{v}\rfloor) + \lg v$
- $T(w) = T(9 \cdot w/10) + w$
- $T(x) = 16 \cdot T(x/4) + x^2$
- $T(y) = 7 \cdot T(y/2) + y^2$
- $T(z) = T(z-1) + z$

H 2.4 Nach asymptotischem Wachstum ordnen (2P)

Ordnen Sie folgende Funktionen nach ihrem asymptotischen Wachstum. Gesucht ist die Reihenfolge $g_1, g_2, \ldots, g_9$, so dass $g_1 \in \mathcal{O}(g_2), g_2 \in \mathcal{O}(g_3), \ldots, g_8 \in \mathcal{O}(g_9)$.

$2^n, 1.1^{n^2}, \sqrt{n}, \sqrt{\log n}, 10^{10}, e^{\log\log n^2}, \log\log n, (\log n)^{\log\log n}, (\log n)^{10}$.

H 2.5 Rekurrenzgleichungen (4P)

Lösen Sie, analog zur Tutoraufgabe **T.2.3**, folgende Rekurrenzen.
(Die Annahme, dass $T(i)$ für $i \leq 2$ konstant sei, gilt hier auch.)

- $T(k) = T(\sqrt{k}) + 1$
- $T(l) = 7 \cdot T(l/3) + l^2$
- $T(m) = 2 \cdot T(m/4) + \sqrt{m}$
- $T(n) = 2 \cdot T(n/2) + n^3$

H 2.6 Schleifen (4P)

Gegeben ist das nebenstehende Codefragment. Wie oft wird der innere Schleifenrumpf durchlaufen? Geben Sie die exakte Anzahl von Durchläufen an und schätzen Sie dann in Θ-Notation ab.

```
for(int i = 0; i < n; i++) {
  for (int j = 0; j < i; j++) {
    for (int k = j; k < i; k++) {
      //Schleifenrumpf
    }
  }
}
```