Universität des Saarlandes
FR 6.2 Informatik

Prof. Dr. Kurt Mehlhorn
Dr. Antonios Antoniadis
André Nusser

WiSe 2017/18

Übungen zu Ideen der Informatik

http://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter17/ideen/

Blatt 2 Abgabeschluss: 6.11.2017

Aufgabe 1 (*10 Punkte*) Betrachten Sie folgendes Programm:

$s \leftarrow 0$; $i \leftarrow 1$;
while $i \leq 8$
  $s \leftarrow s + i$; $i \leftarrow i + 1$;
  if i ist gerade
    drucke s
  else
    $i \leftarrow i + 1$
drucke s;

Fragen:

a) Führen Sie das Programm aus.

b) Wieviele Zahlen werden gedruckt?

c) Was ist der Endwert von s?

d) Was ist der Endwert von i?

Lösung:

a) $s \leftarrow 0$;
$i \leftarrow 1$;
$i \leq 8$ ist wahr; daher betreten wir die Schleife.
$s \leftarrow s + i = 1$;
$i \leftarrow i + 1 = 2$;
i is gerade ist wahr; daher fuühren wir den then-Fall aus
drucke s druckt 1.
$i \leq 8$ ist wahr, da i den Wert zwei hat; daher betreten wir die Schleife.
$s \leftarrow s + i = 1 + 2 = 3$;
$i \leftarrow i + 1 = 3$;
i is gerade ist falsch; daher führen wir den else-Fall aus.

$i \leftarrow i + 1 = 4;$
$i \leq 8$ ist wahr; daher betreten wir die Schleife.
$s \leftarrow s + i = 3 + 4 = 7; i \leftarrow i + 1 = 5;$ i is gerade ist falsch; daher führen wir den else-Fall aus.
$i = i + 1 = 6;$
$i \leq 8$ ist wahr; daher betreten wir die Schleife.
$s \leftarrow s + i = 7 + 6 = 13; i \leftarrow i + 1 = 7;$
i is gerade ist falsch; daher führen wir den else-Fall aus.
$i = i + 1 = 8;$
$i \leq 8$ ist wahr; daher betreten wir die Schleife.
$s \leftarrow s + i = 13 + 8 = 21; i \leftarrow i + 1 = 9;$
i is gerade ist falsch; daher führen wir den else-Fall aus.
$i = i + 1 = 10;$
$i \leq 8$ ist falsch; daher betreten wir die Schleife nicht.
drucke s druckt 21.

b) Es werden zwei Zahlen gedruckt, 1 und 21.

c) Der Endwert von s ist 21.

d) Der Endwert von i ist 10.

Aufgabe 2 (*10 Punkte*) Wir haben vier Programme für die gleiche Aufgabe. Die Programme nehmen einen Eingabewert n und rechnen dann für $T_1(n) = 1000n$, $T_2(n) = 100n \log_{10}(100n)$, $T_3(n) = 5n^2$, bzw. $T_4(n) = 2^n$ Sekunden.

a) Was sind die Laufzeiten der vier Programme für $n = 1$, 10, 100, 1000, 10000?

b) Bestimmen Sie $T_i(2n)/T_i(n)$ als Funktion von n für alle vier Programme.

c) Sei $T(n) = c \cdot n^k$ für zwei Konstanten c und k. Was ist $T(2n)/T(n)$? Erklären sie das Ergebnis mit Worten.

d) Für welche Bereiche von n ist welches Programm am schnellsten?

e) Sie haben 10^6 Sekunden Rechenzeit (etwas mehr als ein Tag) zur Verfügung. Wie großdarf n sein, dass Sie das Problem mit dem i-ten Programm innerhalb dieser Zeit lösen können? Sei n_i dieser Wert. Wie lange läuft das Programm auf der Eingabe $2n_i$?

Lösung:

a)

	T_1	T_2	T_3	T_4
1	10^3	$2 \cdot 10^2$	5	2
10	10^4	$3 \cdot 10^3$	$5 \cdot 10^2$	2^{10}
100	10^5	$4 \cdot 10^4$	$5 \cdot 10^4$	2^{100}
1000	10^6	$5 \cdot 10^5$	$5 \cdot 10^6$	2^{1000}
10000	10^7	$6 \cdot 10^6$	$5 \cdot 10^8$	2^{10000}

b)
- $\frac{T_1(2n)}{T_1(n)} = \frac{2000n}{1000n} = 2.$

- $\frac{T_2(2n)}{T_2(n)} = \frac{200n \log_{10}(200n)}{100n \log_{10}(100n)} = \frac{2 \log_{10}(200n)}{\log_{10}(100n)}$.
- $\frac{T_3(2n)}{T_3(n)} = \frac{5(2n)^2}{5n^2} = 4$.
- $\frac{T_4(2n)}{T_4(n)} = \frac{2^{2n}}{2^n} = 2^n$.

c) $\frac{T(2n)}{T(n)} = \frac{c \cdot (2n)^k}{c \cdot n^k} = 2^k$. Wenn sich die Eingabegröße um einen Faktor 2 vergrössert, vergrössert sich die Laufzeit um einen Faktor 2^k. Der Exponent ist der Grad des Polynoms.

d) Für $n \in [1, 8]$ ist T_4 am schnellsten, für $n \in [9, 77]$ T_3, und für $n \in [78, 10^8]$ ist T_2 am schnellsten. Ab $n = 10^8$ ist dann T_1 am schnellsten.

e)
- $n_1 = 10^3$. Bei $2n_1$ braucht man $2 \cdot 10^6$ Sekunden.
- $n_2 = 1894$. Bei $2n_2$ braucht man etwas mehr als $2 \cdot 10^6$ Sekunden.
- $n_3 = \sqrt{2 \cdot 10^5} \approx 4.5 \cdot 10^2$. Bei $2n_3$ braucht man $4 \cdot 10^6$ viele Sekunden.
- $n_4 \approx 20$. Bei $2n_4$ braucht man $2^{40} \approx 10^{12}$ Sekunden.

Aufgabe 3 (*10 Punkte*) (Landausymbol für asymptotisches Wachstum) Seien f und g Funktionen von den natürlichen Zahlen in die positiven reellen Zahlen. Wir schreiben $f = O(g)$, falls es eine konstante C gibt, so dass $f(n)/g(n) \leq C$ für alle $n \geq 1$. Im Folgenden schreiben wir $n^2 + 5n$ für die Funktion $n \mapsto n^2 + 5n$. Es gilt etwa $2n^2 + 5n = O(n^2)$, da $(2n^2 + 5n)/n^2 = 2 + 5/n \leq 7$ für $n \geq 1$. Sind die folgenden Aussagen wahr oder falsch?

a) $2n^2 = O(n)$?

b) $n^{17} + n^4 = O(2^n)$?

c) $n^{17} + n^4 = O(n^{18})$?

Lösung:

a) Falsch, da $\frac{2n^2}{n} = 2n \to \infty$ für $n \to \infty$.

b) Richtig, da $\frac{n^{17}+n^4}{2^n} \to 0$, für $n \to \infty$. Also es existiert ein $c < +\infty$ so dass $f(n)/g(n) \leq c$.

c) Richtig, da $\frac{n^{17}+n^4}{n^{18}} = \frac{1}{n} + \frac{1}{n^{14}} \to 0$ wenn $n \to \infty$. Wie vorhin, es existiert $c < \infty$ so dass $f(n)/g(n) \leq c$.

Aufgabe 4 (*10 Punkte*) Um einen Turing-Award zu gewinnen erfindet Kurt Mehlhorn an einem Vormittag ein neues Rechnermodell. Er behauptet, dass sein Modell, genauso wie das Von-Neumann Modell, universell sei. Um seine Behauptung zu etablieren, beschliesst Kurt eine der beiden folgenden Aussagen zu beweisen:

a) Kann man ein Programm auf der Von-Neumann Maschine ausführen, so kann man es auch auf der Mehlhorn Maschine ausführen.

b) Kann man ein Programm auf der Mehlhorn Maschine ausführen, so kann man es auch auf der Von-Neumann Maschine ausführen.

Diskutieren Sie die Implikationen beider Varianten und erklären Sie welchen Beweis Kurt führen sollte.

Rechner war	spannend ☐ okay☐ langweilig ☐
	schwierig ☐ okay ☐ einfach ☐

Lösung: Die Von-Neumann Maschine ist universell. Also wenn (a) gilt, dann kann man jedes Programm auch auf der Mehlrhorn Maschine ausführen, und ist daher auch universell. Wenn (b) gilt, gibt uns das keinerlei Informationen: Es könnte sein, dass die Mehlhorn Maschine nur eins oder sogar garkein Programm ausführen kann, und (b) würde trotzdem gelten. Also, sollte Kurt Mehlhorn den ersten Beweis führen.