technische universität dortmund

SoSe 2018

M. Buchin
A. Krivošija, D. Mezlaf
L. Pfahler, A. Rey

DAP2 – Heimübung 2

Ausgabedatum: 13.04.2018 — Abgabedatum: Mo. 23.04.2018 bis 12 Uhr

Abgabe:

Schreiben Sie unbedingt immer Ihren vollständigen Namen, Ihre Matrikelnummer und Ihre Gruppennummer auf Ihre Abgaben! Beweise sind nur dort notwendig, wo explizit danach gefragt wird. Eine Begründung der Antwort wird allerdings *immer* verlangt.

Scheine:

Für Studierende in den Bachelor-Studiengängen ist die Erbringung von Studienleistungen Voraussetzung für die Teilnahme an der Modulprüfung (Klausur)[1]. Die Studienleistung für die DAP2-Übungen wird erbracht durch

- Erreichen von mindestens 50 % der Punkte, die in den Heimarbeitsübungsaufgaben erreichbar sind, und
- Erreichen von 50 % der Punkte in mindestens einem der beiden Übungstests.

Die Heimübungen dürfen in Gruppen von maximal drei Studierenden abgegeben werden. Die gemeinsame Bearbeitung in solchen Gruppen ist ausdrücklich erwünscht.

Aufgabe 2.1 (5 Punkte): (Laufzeitanalyse)

Führen Sie eine exakte Worst-Case-Laufzeitanalyse für den unten gegebenen Algorithmus **BearbeiteArray** bei Eingabe eines Arrays der Länge n durch, d. h., finden Sie eine Funktion $f(n)$, deren Wert die Worst-Case-Laufzeit dieses Algorithmus ist.

```
BearbeiteArray(Array A):
n ← length[A]
for i ← 1 to n − 1 do
    max ← i
    for j ← i + 1 to n do
        if A[max] < A[j] then
            max ← j
    if max ≠ i then
        temp ← A[max]
        A[max] ← A[i]
        A[i] ← temp
```

Geben Sie außerdem Antworten auf folgenden Fragen:

[1]In anderen Studiengängen ist die Erbringung der Studienleistung möglicherweise ebenfalls Pflicht. Bitte überprüfen Sie Ihre jeweilige Prüfungsordnung oder das jeweilige Modulhandbuch.

a) Was macht dieser Algorithmus?

b) Führen Sie eine asymptotische Worst-Case-Laufzeitanalyse für diesen Algorithmus durch, d. h. finden Sie eine möglichst kleine Funktion $g(n)$, sodass $f(n) \in \mathrm{O}(g(n))$ gilt.

Lösung:

Behauptung: Es handelt sich hier um eine Variante des Selection-Sort-Algorithmus zur Sortierung eines Eingabearrays in absteigender Reihenfolge. Er benötigt bei einer Eingabe der Länge n im Worst Case $T(n) = \frac{3}{2}n^2 + \frac{11}{2}n - 5$ Rechenschritte.

Beweis (Laufzeitanalyse): Wir gehen Zeile für Zeile durch den Pseudocode und geben jeweils an, wie viele Rechenschritte die Zeile benötigt:

1. Eine Zuweisung benötigt einen Rechenschritt.
2. der Kopf der **for**-Schleife benötigt n Rechenschritte, weil nach $n-1$ Iterationen noch einmal die Abbruchbedingung geprüft werden muss.
3. Diese Zeile wird $n-1$ mal ausgeführt (für jeden Schleifenrumpfaufruf).
4. Der Kopf der inneren Schleife wird bei dem i-ten Schleifendurchlauf genau $n-i+1$ mal ausgeführt, also insgesamt $s_1 = \sum_{i=1}^{n-1}(n-i+1)$ mal.
5. In Iteration i der äußeren Schleife wird der Rumpf der inneren Schleife $n-i$ mal ausgeführt, also insgesamt $s_2 = \sum_{i=1}^{n-1}(n-i)$ mal.
6. Im Worst Case wird diese Zeile bei jeder Ausführung von Zeile 5 aufgerufen, also insgesamt s_2 mal.
7. Diese **if**-Abfrage wird in jedem Durchlauf der äußeren Schleife aufgerufen, also insgesamt $n-1$ mal.

8.–10. Im Worst Case werden diese Zeilen in jedem Durchlauf der äußeren Schleife aufgerufen, also insgesamt je $n-1$ mal.

Wir berechnen zunächst die Summen:

$$s_2 = \sum_{i=1}^{n-1}(n-i) \overset{\ell=n-i}{=} \sum_{\ell=1}^{n-1}\ell \overset{\text{Gauß}}{=} \frac{(n-1)(n-1+1)}{2} = \frac{(n-1)n}{2}$$

und

$$s_1 = s_2 + \sum_{i=1}^{n-1} 1 = \frac{(n-1)n}{2} + n - 1.$$

Nun summieren wir alle ermittelten Beiträge auf zu

$$\begin{aligned} T(n) &= 1 + n + (n-1) + s_1 + 2 \cdot s_2 + 4 \cdot (n-1) \\ &= 1 + n + 5 \cdot (n-1) + \frac{(n-1)n}{2} + n - 1 + 2 \cdot \frac{(n-1)n}{2} \\ &= 3 \cdot \frac{(n-1)n}{2} + 7n - 5 = \frac{3}{2}n^2 + \frac{11}{2}n - 5. \end{aligned}$$

Asymptotisch gilt, dass $\frac{3}{2}n^2 + \frac{11}{2}n - 5 \in O(n^2)$ ist. □

Aufgabe 2.2 (5 Punkte): (Landau–Notation)

Welche der folgenden Aussagen sind korrekt?

a) $2(n^2)^4 \in \mathcal{O}\left(4^n\right)$

b) $\left(3\ln n^3\right)^3 \in o\left(\frac{1}{3}\sqrt[3]{n}\right)$

c) $\sqrt{n^4} + \sqrt[4]{n^2} \in \Omega\left(4n\right)$

d) $5 - 4n + 3n^2 - 2n^3 + n^4 \in \omega\left(10n^4 + n^3\log^3 n\right)$

e) $2^{\frac{1}{2}\log(n^2)} \in \Theta\left(2n\right)$

Geben Sie in allen Fällen einen Beweis an, um Ihre Antwort zu begründen. Sie dürfen die Ergebnisse benutzen, die in der Vorlesung bewiesen wurden. Nutzen Sie dabei die Definitionen der Landau-Symbole aus der Vorlesung.

Lösung:

a) Korrekt. Zu zeigen: $\exists c' > 0, \exists n_0' > 0, \forall n \geq n_0'$: $2(n^2)^4 \leq c \cdot 4^n$.

In der Vorlesung ist gegeben, dass $n^\ell \in \mathcal{O}(2^n)$ für alle $\ell \geq 2$ ist, somit existieren für $\ell = 8$ ein $c > 0$ und ein $n_0 > 0$, sodass für alle $n \geq n_0$ gilt, dass $n^8 \leq c \cdot 2^n = \frac{2c}{2} \cdot 2^n$. Außerdem gilt $2 \leq 4 \Rightarrow 2^n \leq 4^n$ für alle $n \geq 0$. Deswegen gilt

$$2(n^2)^4 = 2n^8 \leq 2c \cdot 2^n \leq 2c \cdot 4^n$$

für das c und das n_0 von oben und für alle $n \geq n_0$. Für $c' = 2c > 0$ und $n_0' = n_0$ folgt die Behauptung.

b) Korrekt. Zu zeigen: $\forall c > 0, \exists n_0' > 0, \forall n \geq n_0'$: $\left(3\ln n^3\right)^3 < \frac{1}{3}\sqrt[3]{n}$.

Es gilt $\left(3\ln n^3\right)^3 = \left(3^2\ln n\right)^3 = 3^6\ln^3 n$. Aus der Hierarchie aus der Vorlesung ist bekannt, dass $\log n \in o\left(n^\ell\right)$ für $\ell > 0$ ist. D. h. für $\ell = \frac{1}{9}$ gilt: für alle $c > 0$ existiert ein $n_0 > 0$, sodass für alle $n \geq n_0$ gilt, dass $\log n < c \cdot n^{1/9}$ ist. Mit $\ln n = \frac{\log_2 n}{\log_2 e}$ folgt, dass

$$\left(3\ln n^3\right)^3 = 3^6\ln^3 n < 3^6 \cdot \left(c' \cdot n^{1/9}\right)^3 = c'' \cdot \frac{1}{3}n^{1/3}$$

für $c'' = 3^7 \cdot (c')^3 > 0$ und $c' = \frac{c}{\log_2 e} > 0$. Da es für jedes solche $c'' > 0$ ein $c > 0$ gibt, für das die erste Aussage gilt, folgt die Behauptung.

c) Korrekt. Zu zeigen: $\exists c' > 0, \exists n_0' > 0, \forall n \geq n_0'$: $\sqrt{n^4} + \sqrt[4]{n^2} \geq c \cdot 4n$.

Es gilt $\sqrt{n^4} + \sqrt[4]{n^2} = n^2 + \sqrt{n} \geq n^2$ für alle $n \geq 1$. Nach der Vorlesung ist $n^2 \in \Omega(n)$, d. h., es existiert ein $c > 0$ und ein $n_0 > 0$, sodass für alle $n \geq n_0$ gilt, dass $n^2 \geq c \cdot n$ ist. Für dieses c und dieses n_0 gilt also für alle $n \geq n_0$:

$$n^2 + \sqrt{n} \geq n^2 \geq c \cdot n \geq c' \cdot 4n$$

für $c' = \frac{c}{4} > 0$.

d) Nicht korrekt. Wenn diese Aussage gültig wäre, müsste gelten, dass $\forall c > 0, \exists n_0' > 0, \forall n \geq n_0$: $5 - 4n + 3n^2 - 2n^3 + n^4 > c \cdot \left(10n^4 + n^3 \log^3 n\right)$.

Zu zeigen: $\exists c > 0, \forall n_0' > 0, \exists n \geq n_0$: $5 - 4n + 3n^2 - 2n^3 + n^4 \leq c \cdot \left(10n^4 + n^3 \log^3 n\right)$.

Es gilt für alle $n \geq 1$:

$$5 - 4n + 3n^2 - 2n^3 + n^4 \leq 5 + 3n^2 + n^4 \leq 5n^4 + 3n^4 + n^4 = 9n^4 \leq 1 \cdot \left(10n^4 + n^3 \log^3 n\right).$$

Für $c = 1$ gilt die gesuchte Aussage also für alle $n \geq 1$, also gibt es auch zu jedem $n_0 > 0$ ein mindestens genauso großes n, das die Aussage erfüllt.

e) Korrekt. Zu zeigen: $2^{\frac{1}{2}\log(n^2)} \in \mathcal{O}(2n)$ und $2^{\frac{1}{2}\log(n^2)} \in \Omega(2n)$.

Es gilt: $2^{\frac{1}{2}\log(n^2)} = 2^{\log n} = n$. Für $c = 1$ und $n_0 = 1$ gilt für alle $n \geq n_0$, dass $n \leq c \cdot 2n$ ist. Für $c = 1/2$ und $n_0 = 1$ gilt für alle $n \geq n_0$, dass $n \geq c \cdot 2n$ ist. $\square$