Tutorium 10, Blatt 3

Freitag, 29. April 2022 07:56

Tutoraufgabe 1 (Laufzeitanalyse):

Gegeben sei ein Algorithmus, der für ein Array von Integern überprüft, ob die Einträge aufsteigend sortiert sind:

```
1 def is_sorted(a):
2     i = 1
3     while i < len(a):
4         if a[i - 1] > a[i]:
5             return False
6         i += 1
7     return True
```

[1,2,3]

Bei Betrachtung der Laufzeit wird angenommen, dass die Vergleiche jeweils eine Zeiteinheit benötigen (siehe Zeilen 3 und 4). Die Laufzeit aller weiteren Operationen wird vernachlässigt.
Sei $n \in \mathbb{N}$ die Länge des Arrays a.

a) Bestimmen Sie in Abhängigkeit von n die Best-case Laufzeit $B(n)$. Begründen Sie Ihre Antwort. Geben Sie für jedes n ein Beispielarray an, bei dem diese Laufzeit erreicht wird.

b) Bestimmen Sie in Abhängigkeit von n die Worst-case Laufzeit $W(n)$. Begründen Sie Ihre Antwort. Geben Sie für jedes n ein Beispielarray an, bei dem diese Laufzeit erreicht wird.

c) Bestimmen Sie in Abhängigkeit von n die Average-case Laufzeit $A(n)$. Begründen Sie Ihre Antwort. Hierzu nehmen wir für $n \geq 2$ folgende Verteilung der Eingaben an:

- $\Pr(a[0] = 1) = 1$,
- $\Pr(a[n-1] = 0) = 1$ und
- für alle $i \in \{1, \ldots, n-2\}$ gilt: $\Pr(a[i] = 0) = \Pr(a[i] = 1) = 0.5$.

[1,1,1,0,..., 0]

Der erste Eintrag ist also immer 1 und der letzte Eintrag ist immer 0. Die übrigen Einträge sind mit gleicher Wahrscheinlichkeit entweder 0 oder 1. Insbesondere ist die Wahrscheinlichkeit, dass das Array aufsteigend sortiert ist immer 0. Für $n < 2$ ist die Verteilung der Eingaben nicht relevant.

Hinweise:

- Wir gehen davon aus, dass die Indizierung von Arrays mit 0 beginnt.
- Geben Sie die geforderten Laufzeiten in geschlossener Form (d.h. ohne Summenzeichen, $\Pr(\cdots)$ oder ähnliches) an.

a) [1,0]

$$B(n) = \begin{cases} 1, & \text{falls } n < 2 \\ 2, & \text{falls } n \geq 2 \end{cases}$$

b) [1,2,3]

$$W(n) = \begin{cases} 1, & \text{falls } n < 2 \\ 2n-1, & \text{falls } n \geq 2 \end{cases}$$

c) Für $n < 2 \rightarrow 1$

Fall: $k = n-1$ [1, 1, ..., 1, 0]

$$\underbrace{\Pr(a[0]=1)}_{1} \cdot \underbrace{\Pr(a[1]=1) \cdot \ldots \cdot \Pr(a[n-2]=1)}_{0,5 \cdot 0,5 \cdot \ldots = 0,5^{n-2}} \cdot \underbrace{\Pr(a[n-1]=0)}_{1} = 0,5^{n-2}$$

Fall: $k < n-1$

$$\underbrace{Pr(a[0]=1)}_{1} \cdot \underbrace{Pr(a[1]=1) \cdot \ldots \cdot Pr(a[k]=0)}_{0{,}5 \cdot 0{,}5 \ldots = 0{,}5^k} = 0{,}5^k$$

$$\begin{aligned} 0.5^{n-2} + \sum_{k=1}^{n-2} 0.5^k &= 0.5^{n-2} + \left(\sum_{k=0}^{n-2} 0.5^k\right) - 0.5^0 \\ &= 0.5^{n-2} + \frac{1-0.5^{n-1}}{1-0.5} - 1 \\ &= 0.5^{n-2} + 2 \cdot \left(1 - 0.5^{n-1}\right) - 1 \\ &= 0.5^{n-2} + 2 - 0.5^{n-2} - 1 = \mathbf{1} \end{aligned}$$

$$2 \cdot (n-1) \cdot 0{,}5^{n-2} + \sum_{k=1}^{n-2} 2 \cdot k \cdot 0{,}5^k$$

$$= 2 \cdot \left((n-1) \cdot 0{,}5^{n-2} + \underbrace{\sum_{k=0}^{n-2} k \cdot 0{,}5^k}\right)$$

$$= 2 \cdot \left((n-1) \cdot 0{,}5^{n-2} + \frac{(n-2) \cdot 0{,}5^n - (n-1) \cdot 0{,}5^{n-1} + 0{,}5}{(0{,}5-1)^2}\right)$$

$$= 2 \cdot \left((n-1) \cdot 0{,}5^{n-2} + (n-2) \cdot 0{,}5^{n-2} - (n-1) \cdot 0{,}5^{n-3} + 0{,}5^{-1}\right)$$

$$\begin{aligned} &= 2 \cdot \left((n-1) \cdot 0.5^{n-2} + (n-2) \cdot 0.5^{n-2} - (n-1) \cdot 0.5^{n-2} \cdot 2 + 2\right) \\ &= 2 \cdot \left(((n-1) + (n-2) - 2(n-1)) \cdot 0.5^{n-2} + 2\right) \\ &= 2 \cdot \left((n-1+n-2-2n+2) \cdot 0.5^{n-2} + 2\right) \end{aligned}$$

$$= 2 \cdot (-1 \cdot 0{,}5^{n-2} + 2)$$

$$= 4 - 0{,}5^{n-3}$$

$$A(n) = \begin{cases} 1, & \text{falls } n < 2 \\ 4 - 0{,}5^{n-3}, & \text{falls } n \geq 2 \end{cases}$$

Tutoraufgabe 2 (Rekursionsgleichung):

Finden Sie für die Rekursionsgleichung F, die wir als

$$F(1) = 1 \text{ und } F(n) = F(n-1) + n^2 + n \text{ für } n > 1$$

definieren, ein Polynom $g : \mathbf{R}_+ \to \mathbf{R}_+$ als obere Abschätzung an, welche nicht rekursiv definiert ist. Beweisen Sie anschließend per vollständiger Induktion, dass ihr Vorschlag tatsächlich eine obere Abschätzung ist, also dass $F(n) \leq g(n)$ ist. Geben Sie schließlich auch eine Abschätzung der Funktion g in O-Notation an.

$$\underbrace{n^2 + (n-1)^2 + \ldots + (n-n)^2}_{n}$$

$$n^2 \cdot n = n^3$$

$$g(n) = an^3 + bn^2 + cn + d$$

IA:

$$F(1) = 1 \leq a + b + c + d$$

IH: Die Aussage gelte für ein beliebiges aber festes n.

IS:

$$F(n+1) = F(n) + (n+1)^2 + (n+1)$$

$$\overset{IH}{\leq} an^3+bn^2+cn+d+(n+1)^2+(n+1)$$
$$= an^3+(b+1)n^2+(c+3)n+d+2$$

$$g(n+1) = a(n+1)^3+b(n+1)^2+c(n+1)+d$$
$$= an^3+(3a+b)n^2+(3a+2b+c)n+a+b+c+d$$

$1 \leq a+b+c+d$

$(a \leq a)$

$b+1 \leq 3a+b$

$c+3 \leq 3a+2b+c$

$d+2 \leq a+b+c+d$

$a = \frac{1}{3}$ $b = 1$ $c = \frac{2}{3}$ $d = -1$

$$g(n) = \frac{1}{3}n^3+n^2+\frac{2}{3}n-1$$

IA: $F(1) = 1 = \frac{1}{3}\cdot 1^3+1^2+\frac{2}{3}\cdot 1-1 = g(1)$

IH: Die Aussage gelte für ein festes aber beliebiges n

IS: $g(n+1) = \frac{1}{3}(n+1)^3+(n+1)^2+\frac{2}{3}(n+1)-1$

$$= \underbrace{\frac{1}{3}n^3+n^2+\frac{2}{3}n-1}+n^2+3n+2$$

$$\overset{IH}{\geq} (n) + n^2+3n+2$$

$$= F(n)+(n+1)^2+(n+1)$$

$$= F(n+1)$$

$g(n) \in O(n^3)$

Da g eine obere Schrank von F ist liegt $F(n)$ in $O(n^3)$

Tutoraufgabe 3 (Master Theorem):

Benutzen Sie das Master Theorem um für folgende Rekursionsgleichungen $T(n)$ eine möglichst gute O-Klasse anzugeben, in der $T(n)$ liegt.

a)

$T(1) = 1$ $\quad a = 16 \quad b = 2$

$T(n) = 16 \cdot T(\frac{n}{2}) + n^5 + \log_4(n)$ for $n > 1$

$f(n) = n^5 + \log_4(n) \in O(n^5) \quad 16 < 32 = 2^5$

$T(n) \in O(n^5)$

b)

$T(1) = 2$ $\quad a = 27 \quad b = 3$

$T(n) = 27 \cdot T(\frac{n}{3}) + n^3 + n^2$ for $n > 1$

$f(n) = n^3 + n^2 \in O(n^3) \quad 27 = 3^3 \quad T(n) \in O(n^3 \cdot \log(n))$

c)

$T(1) = 3$ $\quad a = 26 \quad b = 5$

$T(n) = 26 \cdot T(\frac{n}{5}) + n \cdot \log_2 n$ for $n > 1$

$f(n) = n \cdot \log_2 n \in O(n^2) \quad 26 > 25 = 5^2$

$T(n) \in O(n^{\log_5(26)})$

Für $T(1) = c$ und $T(n) = a \cdot T(\frac{n}{b}) + f(n)$ falls $n > 1$ gilt

Wenn	Dann
$f \in O(n^p)$ und $a < b^p$	$T(n) \in O(n^p)$
$f \in O(n^p)$ und $a = b^p$	$T(n) \in O(n^p \cdot \log(n))$
$f \in O(n^p)$ und $a > b^p$	$T(n) \in O(n^{\log_b(a)})$