Домашна работа 1, DAA, Михаил Зарков, фн: 45655 — DAA, гр. 1

Зад. 1 $T(n) = n^2 T(\frac{n}{2}) + 1$, $T(n) \overset{?}{\asymp} n^{1+\lg n}$

Реш: Ще докажа $T(n) \asymp n^{1+\lg n}$ с включване в двете посоки:

$T(n) \preceq n^{1+\lg n}$ и $T(n) \succeq n^{1+\lg n}$.

$| T(n) \overset{?}{\preceq} n^{1+\lg n} |$

По деф: $T(n) \preceq n^{1+\lg n} \iff \exists c > 0: \ c\,T(n) \le n^{1+\lg n}$.

Доказвам с индукция по n:

1. База: Няма база, защото игнорирам началните условия.
   Т.е каквито и да са началните условия $T(k)$, мога да избера такава константа c, че $c.T(k) \le k^{1+\lg k}$.

2. ИП: допускам, че: $\exists c > 0$:
   $c.T(s) \le s^{1+\lg s}, \ \forall s < n$

3. Доказвам $c\,T(n) \le n^{1+\lg n}$

$$c.T(n) \overset{def.}{=} c.\left(n^2 T(\tfrac{n}{2}) + 1\right) \overset{ИП}{\le} c.n^2 \left(\tfrac{n}{2}\right)^{1+\lg \frac{n}{2}} + c =$$

$$= c.n^2.\frac{n^{\lg n}}{n} + c = c.n^{1+\lg n} + c$$

Нека $c < \frac{1}{3}$. Тогава $\forall n \ge 1$, $1 \le n^{1+\lg n}$.

Иначе: $$c.n^{1+\lg n} + c < \frac{1}{3} n^{1+\lg n} + \frac{1}{3} \underset{\forall n \ge 1}{\le} \frac{1}{3}.n^{1+\lg n} + \frac{1}{3} n^{1+\lg n} = \frac{2}{3} n^{1+\lg n} < n^{1+\lg n}$$

Получих $c \cdot T(n) \le n^{1+\lg n}$ за $c < \frac{1}{3}$, следователно $T(n) \precsim n^{1+\lg n}$

$\left| \, n^{1+\lg n} \overset{?}{\precsim} T(n) \, \right|$

$n^{1+\lg n} \precsim T(n) \overset{def.}{\iff} \exists c > 0: \; c \cdot n^{1+\lg n} \le T(n).$

Д-во с индукция по n:

1. База: очевидна, защото ~~...~~ игнорирам началните условия.

2. ИП: допускам, че: ~~...~~

$c \cdot k^{1+\lg k} \le T(k), \; \forall k < n$

3. Доказвам за $k = n$, $c \cdot n^{1+\lg n} \le T(n)$.

$$T(n) \overset{def.}{=} n^2 T\left(\frac{n}{2}\right) + 1 \overset{ИП}{\ge} n^2 c \left(\frac{n}{2}\right)^{1+\lg\left(\frac{n}{2}\right)} + 1 = c \cdot n^2 \frac{n^{\lg n}}{n} + 1 =$$

$$= c \cdot n^{1+\lg n} + 1 > c \cdot n^{1+\lg n} \Rightarrow \text{вярно за } n$$

$$\Rightarrow n^{1+\lg n} \precsim T(n) \Rightarrow n^{1+\lg n} \asymp T(n) \; \square$$

3. Стъпка: доказваме, че е вярно $(n+1)^{\lg 2(n+1)} \le T(n+1)$

$$T(n+1) \overset{def.}{=} (n+1)^2 \underbrace{T\left(\frac{n+1}{2}\right)} + 1 \ge \text{ИП за } \frac{n+1}{2}$$

$$\ge (n+1)^2 \left(\frac{n+1}{2}\right)^{\lg 2\cdot\left(\frac{n+1}{2}\right)} + 1 = (n+1)^2 \frac{(n+1)^{\lg(n+1)}}{n+1} + 1 =$$

$$= (n+1)(n+1)^{\lg(n+1)} + 1 > (n+1)(n+1)^{\lg(n+1)} = (n+1)^{1+\lg(n+1)} = (n+1)^{\lg 2(n+1)}$$

Получих $T(n+1) > (n+1)^{\lg 2(n+1)}$ — вярно за $n+1$

$\Rightarrow$ $\cancel{T(n)} \;$ $T(n) > n^{\lg 2n}$, $\forall n \ge 1$, $n \in \mathbb{N}$ $\Rightarrow$ $n^{\lg 2n} \preceq T(n)$ (2)

От (1) и (2) следва, че $n^{\lg 2n} \asymp T(n)$ □

Зад. 2 Условие:

```
int foo(int a){
    int i, x=6, y=1, z=0;

    for(i=0; i<a; ++i){
        z += y;
        y += x;
        x += 6;
    }

    return z;
}
```

$a \ge 1$, $a \in \mathbb{N}$

Какво връща foo?

Докажете го.

Реш. Функцията ще връща a^3.

Д-во

Инварианта за x:

При k-тото достигане на проверката за край на цикъла на ред 4, x има стойност $k \cdot 6$.

Инварианта за y:

При k-тото достигане на проверката за край на цикъла на ред 4, y има стойност $\frac{(k-1) \cdot k}{2} \cdot 6 + 1$.

Инварианта за z:

При k-тото достигане на проверката за край на цикъла на ред 4, z има стойност $(k-1)^3$.

Д-во на горните инварианти с индукция по k:

1. База: $k = 1$.

На ред 2 имаме инициализациите: $x = 6$, $y = 1$, $z = 0$.

Стигаме до проверката за край на цикъла на ред 4; x, y, z остават непроменени

За x: ~~(1-1)~~ $1 \cdot 6 = 6$ V

За y: $\frac{(1-1) \cdot 1}{2} \cdot 6 + 1 = 1$ V

За z: $(1-1)^3 = 0$ V

=> инвариантите са верни за $k = 1$.

2. <u>Поддръжка:</u>

Нека инвариантите са верни за някое k-то достигане на проверката за край на цикъла и k-тото достигане не е последно. Т.е допускам, че е достигнат ред 4 за k-ти път (не е последен) и $x = k.6$, $y = \frac{(k-1).k}{2}.6 + 1$, $z = (k-1)^3$.

На ред 5 имаме $z += y$, т.е z приема стойността:

$$(k-1)^3 + \frac{(k-1).k.6}{2} + 1 = k^3 - 3k^2 + 3k - 1 + 3k^2 - 3k + 1 = \underline{k^3}$$

На ред 6 имаме $y += x$, т.е y приема стойността:

$$\frac{(k-1).k.6}{2} + 1 + k.6 = \frac{(k-1)k.6 + 2k.6}{2} + 1 = \underline{\frac{k(k+1).6}{2} + 1}$$

На ред 7: $x += 6$; x приема стойността: $k.6 + 6 = \underline{(k+1)6}$

На ред 8 свършва тялото на цикъла и отиваме на ред 4 (k-тото достигане не беше последно). Тогава сме достигнали ред 4 за $k+1$-ви път и стойностите на x, y, z са:

$$\left.\begin{aligned} x &= k^3 \\ y &= \frac{k(k+1).6}{2} + 1 \\ z &= (k+1).6 \end{aligned}\right| \Rightarrow \text{инвариантите са верни и за } k+1\text{-вото достигане на ред 4}$$

3. <u>Терминация:</u>

При $a+1$-вото достигане на ред 4 променливата i има стойност a, следователно не влизаме в тялото на цикъла и отиваме на ред 10, където връщаме z, която според поддръжката има стойност $(a+1-1)^3 = a^3$. Значи наистина алгоритъмът връща входа си на трета степен. □

DAA, гр. 1

… $p_2 \in \{l, \dots, h-?\}$ и след изпълнението

Зад. 3

Забележка: Когато казвам, че масивът е сортиран или в предела, че е сортиран във възходящ ред.

а) При начално викане SomeAlg(A[1...n], 1, n), алгоритъмът сортира масива A.

б) Приемам, че входът на алгоритъма е валиден, т.е. $1 \le l, h \le n$

Доказвам, че при начално викане SomeAlg(A[1...n], l, h), алгоритъмът сортира подмасива A[l...h].

I сл. $h < l$ => дадения подмасив е празният. Считам, че празният масив е сортиран. ✓

if-ът на ред 1 не сработва. Не променяме A.

II сл. $l \le h$

Ще докажа с индукция по брой на елементите в A[l...h], че

* виж стр. 8 доп.

SomeAlg(A[1...n], l, h) сортира A[l...h]*. ($k = h-l+1$ − брой на ел. в A[l...h])

1. База: $k=1$; if-ът ~~не~~ на ред 1 не сработва, защото $l=h$. ~~...~~ ~~...~~ Излизаме от ф-ята SomeAlg без да променяме A.

$k=2$ => $h-l+1=2$, ~~...~~ => $h > l$ => if-ът на ред 1 сработва.

Ако A[l,h] не е сортиран, то if-ът на ред 2 сработва и на ред 3 сортираме A[l,h] (това става само със swap).

Ако A[l,h] е сортиран, то не е вярно, че A[h] < A[l], следователно ред 3 не се изпълнява (не променяме A).

$h - l + 1 = k = 2 \Rightarrow t = \lfloor \frac{2}{3} \rfloor = 0 \Rightarrow$ if-ът на ред 5 не сработва

и излизаме от ф-ята, като $A[l,h]$ е сортиран и не сме променили

елементите извън подмасива $A[l,h]$.

2. ИП: нека за някое $k \in \mathbb{N}$, $k \geq 2$ е изпълнено, че:

$\forall s = h-l+1$, $s \in \mathbb{N}$, $2 \leq s \leq k$, SomeAlg($A[1 \dots n], l, h$)

сортира $A[l \dots h]$ и swap-ва само елементи от $A[l \dots h]$.

3. Стъпка: Зб. $3 = 2+1 \leq k+1 \Rightarrow t \geq 1$ (ред 4).

Доказваме, че за $h-l+1 = k+1$, SomeAlg($A[1 \dots n], l, h$) сортира $A[l \dots h]$ и прави swap-ове само в $A[l \dots h]$.

~~Допускаме, че след изпълнението на SomeAlg подмасивът $A[l \dots h]$ не е сортиран, което е т.с.т.к. $\exists p, q \in \{l \dots h\}: p < q$ и $A[p] > A[q]$.~~

~~I сл. $p, q \in \{l, \dots, h-t\}$~~

~~От и Вървим отзад напред през алгоритъма.~~

~~От ИП знаем, че на ред 8 SomeAlg($A[1 \dots n], l, h-t$) сортира $A[l \dots h]$ и swap-ва само елементи в $A[l \dots h]$. Следователно след изпълнението на ред 7 елементите, които ще отидат в p и q, не може да са извън индексите $\{l \dots h-t\}$, защото SomeAlg($A, l, h-t$) не променя елементите извън тези индекси.~~

~~Щом елементите, които отиват в p и q са с индекси $p_1, q_1 \in \{l \dots h$~~

*(от стр 7) ... и SomeAlg прави swap-ове на елементи само в масива $A[l \dots h]$.

I сл. ~~Разглеждам числата~~

Имаме $l<h$; if-ът на ред 1 сработва.

На редове 2, 3 или swap-ваме елементи само в масива $A[l..h]$ или не правим нищо.

Понеже $h-l+1 = k+1 = 2+1 = 3$, то на ред 4 $t = \lfloor \frac{3}{3} \rfloor = 1$, следователно и if-ът на ред 5 сработва.

Нека разгледаме числата α, β, които са на позиции съответно p_1, q_1 веднага след изпълнението на ред 5.

Разглеждам различни случаи за първоначалните индекси p_1, q_1 (на α, β) и доказвам, че финалните индекси p_α, p_β т.е. след изпълнението на ред 8 са такива, че α и β са сортирани. Щом това е вярно за произволни две числа от масива $A[l..h]$, то е вярно за всички $\Rightarrow A[l..h]$ ще е сортиран.

I сл. след ред 5 $p_1, q_1 \in \{l, \dots, h-t\}$

По ИХ след изпълнението на ред 6 α и β биват сортирани с нови индекси p_2, q_2.

Ред 7 и ред 8 не могат да направят така, че α и β да не са сортирани, защото по ИХ, ~~ако~~ SomeAlg$(A, l+t, h)$ и SomeAlg$(A, l, h-t)$ ще сортират съответно $A[l+t, h]$ и $A[l, h-t]$ и също "не пипат" елементите извън тези масиви.

II сл. След изпълнението на ред 5 $p_1, q_1 \in \{h-t+1, h\}$

~~Ако α, β са сортирани след ред 5, то по аналогични причини като~~

~~I сл. те ще останат сортирани и след ред 8.~~

По аналогични причини като в I сл. α и β ще останат сортирани.

III сл. Нека след изпълнението на ред 5 $p_1 \in \{l, h-t\}$, $q_1 \in \{h-t+1, h\}$

Ако α и β са сортирани (аналогично на I, II) те ще останат сортирани.

Нека БОО, $\alpha > \beta$ (Зб: началният индекс на α е p_1, а на β е q_1).

3.1 След изпълнението на ред 6 α от p_1 отива в $p_2 \in \{l+t, h-t\}$.

Но тогава $p_2 \leq q_1 = q_2 \in \{l+t, h\}$, и ред 7 ги сортира, и

те остават сортирани до край.

3.2 След ред 6 α от $p_1 \to p_2 \in \{l, l+t-1\}$

Тъй като ред 6 сортира $A[l..h-t]$, това значи, че

$\forall i = l+t, \dots, h-t$, $A[i] \geq \alpha$. Числата в масива $A[l+t..h-t]$ са

$h-t-(l+t)+1 = h-l-2t+1$ на брой.

След изпълнението на ред 7 β отива в индекс q_2.

Тъй като $\beta < \alpha \leq A[l+t], \dots, A[h-t]$, то q_2 може да е най-много

$h-(h-l-2t+1)$ (ред 7 сортира $A[l+t..h]$ и има

поне $(h-l-2t+1)$ на брой по-големи от β).

Тогава q_2 ще е $q_2 \leq h-t$, проверка, че е така:

$$h-t-(h-(h-l-2t+1)) = h-t-(l+2t-1) = h-l+1-3t \geq$$
$$h-l+1-3\cdot\frac{h-l+1}{3} = 0 \Rightarrow h-t-q_2 \geq 0, \quad h-t \geq q_2$$

. Знам имаме $q_2, p_2 \in \{l, \dots, h-t\}$ и след изпълнението

на ред 8 ~~...~~ α и β ще се сортират.

Всички случаи са изчерпани $\Rightarrow A[l..h]$ е сортиран $\square$.

Г) Този въпрос ми се струва некоректно зададен, защото в условието не е указано какво очакване от алгоритъма, за да твърдим, че при промяна той остава коректен.

Приемам, че, ако масивът A при SomeAlg с и без първия if е различен, то SomeAlg без if-а е некоректен.

Разглеждам входа:

$A[5,3]$, $n=2$, $l=2$, $h=1$

Очевидно изходният масив на оригиналния SomeAlg е този $A[5,3]$.

Нека махнем if-а на ред 1. Тогава if-ът на ред 2 сработва.

На ред 3 swap-ваме $A[2]$ с $A[1]$ и получаваме $A[3,5]$.

$t = \left\lfloor \frac{1-2+2}{3} \right\rfloor = 0 \Rightarrow$ излизаме от ф-ята с изходен масив

$A[3,5] \neq A[5,3] \Rightarrow$ ~~...~~ промяната на проф. Дълбоков

промени изхода на алгоритъма $\Rightarrow$ е некоректен $\square$.