ДАА семинар 5

Зад 1 Да се докаже, че BubbleSort е коректен сортиращ алгоритъм:

```
BubbleSort(A[1..n]:int)
1  for i ← 1 to n
2      for j ← 1 to n−i
3          if A[j] > A[j+1]
4              Swap(A[j], A[j+1])
```

Решение: За да докажем коректността на алгоритъма трябва да използваме две твърдения за двата цикъла.

// Може последователно или един в друг.

(*) Инвариант за вътрешния цикъл: При всяко достигане на ред 2 $A[j]$ съдържа най-голям елемент на $A[1..j]$

Док. на (*):

– База: При първото достигане на ред 2 $j=1$, тоест $A[1..j]=A[1]$. Тривиално $A[1]=A[j]$ е най-голям елемент.

– Поддръжка: Нека инвариантът е изпълнен за някое достигане на ред 2, което не е последно.
Разглеждаме 2 случая за условието на ред 3:

1сл $A[j] > A[j+1]$: От инв. имаме, че $A[j]$ е най-голям за $A[1..j]$
Тогава $A[j]$ е най-голям за $A[1..j+1]$.
На ред 4 $A[j]$ и $A[j+1]$ си разменят стойностите. Следователно $A[j+1]$ е най-голям за $A[1..j+1]$.

При следващо достигане за $j'=j+1$ инвариантът е в сила.

2сл $A[j] \leq A[j+1]$. Заедно с ИП получаваме, че $A[j+1]$ е най-голям за $A[1..j+1]$.
Ред 4 не се изпълнява. В термините на новото j при следващо достигане на ред 2, инв. е изпълнен.

– Терминация: При последното достигане на ред 2 имаме, че $j=n-i+1$ и $A[j]=A[n-i+1]$ е най-голям за $A[1..n-i+1]$

□ (*)

(**) Инвариант за външния цикъл: При всяко достигане на ред 1 е вярно, че $A[n-i+2..n]$ съдържа $i-1$ най-големи елемента на входния $A[1..n]$ в сортиран вид.

Док. на (**):

– База: При първото достигане на ред 1 $i=1$, т.е. $A[n-i+2..n] = A[n+1..n]$ – празен масив, който съдържа $0=i-1$ макс. ел-та на A.
$\Rightarrow$ Инв. е в сила.

– Поддръжка: Нека инв. е изпълнен за някое достигане на ред 1, което не е последно.

1) От ИП имаме, че $A[n-i+2..n]$ са максимални за $A[1..n]$ и са в сортиран вид.

2) От (*) имаме, че $A[n-i+1]$ е най-голям за $A[1..n-i+1]$

Дали $A[n-i+1] \le A[n-i+2]$? Да. Ако допуснем противното получаваме противоречие с ИП.
Тогава $A[n-i+1 \ldots n]$ съдържа i най-големи елементи на $A[1..n]$ в сортиран вид.

При следващо достигане на ред 1 за $i' = i+1$ е изп. инвариантът.

– Терминация: При последното дост. на ред 1 $i=n+1$. Тогава $A[n-i+2..n] = A[n-(n+1)+2..n] = A[1..n]$ съдържа $n+1-1=n$ най-големи елементи на $A[1..n]$ в сортиран вид.

$\square$ (**)

От (**) следва че BubbleSort е коректен сортиращ алгоритъм.

Зад 2 Какво връща алгоритъмът? Докажете формално.

AlgX(a, n) $a \in \mathbb{R}$, $n \in \mathbb{N}$

```
1   if n=0
2       return 1
3   if n ≡ 0 (mod 2)
4       return AlgX(a*a, n/2)
5   return a * AlgX(a, n-1)
```

Решение: Твърдим, че $AlgX(a,n) = a^n$ за $a \in \mathbb{R}$, $n \in \mathbb{N}$.
Ще го докажем с пълна индукция по n.

– База: $n=0$. От ред 1 и 2 имаме, че $AlgX(a,0)=1$

– ИП: Нека $(\forall m \le n)(AlgX(a,m)=a^m)$

– Стъпка: Разгл. $n+1>0$. Условието на ред 1 не е изпълнено.

1сл. $n+1 \equiv 0 \pmod 2$. От това и $n+1>0 \Rightarrow n+1 \geq 2 \Rightarrow \frac{n+1}{2} > 0$

Условието на ред 3 е вярно, т.е. на ред 4 алг. връща

$$AlgX(a*a, \underbrace{\tfrac{n+1}{2}}_{>0}) \stackrel{ИП}{=} (a*a)^{\frac{n+1}{2}} = (a^2)^{\frac{n+1}{2}} = a^{n+1} \quad ✓$$

2сл. $n+1 \equiv 1 \pmod 2$. Изпълнението пропуска ред 4 и отива на ред 5, където връщаме:

$$a * AlgX(a,(n+1)-1) = a * AlgX(a,n) \stackrel{ИП}{=} a * a^n = a^{n+1} \quad ✓$$

Зад 3 Да дефинираме „силон" като максималният по включване подмасив, в който елементите са ненамаляващи, който не е празен.

Докажете, че следният алгоритъм връща броят силонове в масива $A[1..n]$.

//Не остана време за тази

```
NumSlopes(A[1..n] - непразен)
1.  res ← 1
2.  for i ← 2 to n
3.     if A[i-1] > A[i]
4.        res ← res + 1
5.  return res
```

Решение: Инвариант: При всяко достигане на ред 2 променливата res съдържа броят силонове в $A[1..i-1]$.

- База: При първото достигане на ред 2 $i=2 \Rightarrow A[1..i-1] = A[1]$. В него има само един елемент и съответно един силон.

  От ред 1 $res=1 \Rightarrow$ инв. е верен

- Поддръжка: Нека инв. е изпълнен за някое срещане на ред 2, което не е последно.

I сл $A[i-1] \leq A[i]$

По дефиницията за силон имаме, че $A[i]$ участва в силона на $A[1..i-1]$. Тоест броят силонове не се променя.

Ред 4 не се изпълнява и $res' = res \stackrel{ИП}{=}$ брой силонове $A[1..i-1]$ = бр.сил. в $A[1..i]$

При следващо достигане, в термините на новото i, инв. е верен.

II сл $A[i-1] > A[i]$.

В този случай A[i] не участва в съседния силон и ще започне нов.

Тоест $\text{др. } A[1..i] = 1 + \text{др. } A[1..i-1]$

От ред 4 $res' = res + 1 \overset{\text{ИП}}{=} \text{др. } A[1..i-1] + 1 = \text{др. } A[1..i]$

При следващото достигане за $i' = i+1$ инв. се запазва.

- Терминация: При последното достигане на ред 2 $i = n+1$. От инв. имаме, че $res = \text{др. } A[1..n]$, което връщаме на ред 5.

Рекурентни уравнения

- Ще ги използваме за да намираме сложността на рекурсивни алг.

Методи за решаване:
- развиване
- дърво на рекурсията

} неформални; дават ни интуиция

- индукция → първо трябва да знаем какво доказваме
- Мастър теорема
- Akra-Bazzi и др.

Зад $T(n) = 2T\left(\frac{n}{2}\right) + 1$

Решение: Ще използваме развиване:

неформална част

$$T(n) = 2T\left(\tfrac{n}{2}\right) + 1 = 2\left(2T\left(\tfrac{n}{4}\right) + 1\right) + 1 = 4T\left(\tfrac{n}{4}\right) + 2 + 1 =$$
$$= 4\left(2T\left(\tfrac{n}{8}\right) + 1\right) + 2 + 1 = 8T\left(\tfrac{n}{8}\right) + 4 + 2 + 1 = 16T\left(\tfrac{n}{16}\right) + 8 + 4 + 2 + 1 = \ldots =$$
$$= n \cdot T\left(\tfrac{n}{n}\right) + \tfrac{n}{2} + \tfrac{n}{4} + \ldots + 8 + 4 + 2 + 1 = nT(1) + \tfrac{n}{2} + \tfrac{n}{4} + \ldots + 8 + 4 + 2 + 1 =$$
$$= n \cdot \left(\tfrac{1}{1} + \tfrac{1}{2} + \tfrac{1}{4} + \ldots + \tfrac{1}{n}\right) = n \sum_{i=0}^{\lg n} \frac{1}{2^i}$$

(1) $T(n) \leq n \cdot \sum_{i=0}^{\infty} \frac{1}{2^i} = n \cdot \underbrace{\sum_{i=0}^{\infty} a_i}_{}$

константа, защото редът е сходящ:

$$\lim_{i\to\infty} \frac{a_{i+1}}{a_i} = \lim_{i\to\infty} \frac{\left(\frac{1}{2}\right)^{i+1}}{\left(\frac{1}{2}\right)^i} = \lim_{i\to\infty} \frac{\cancel{\left(\frac{1}{2}\right)^i} \cdot \frac{1}{2}}{\cancel{\left(\frac{1}{2}\right)^i}} = \lim_{i\to\infty} \frac{1}{2} = \frac{1}{2} < 1$$

(2) От друга страна: $n \cdot 1 \leq T(n)$

Тогава от (1) и (2) $\Rightarrow T(n) \asymp n$

Това не е формално доказателство обаче. Ще използваме индукция, за да докажем, че:

1) $T(n) = O(n)$
2) $T(n) = \Omega(n)$

Док:
1) Трябва да покажем, че съществуват константи $c>0, n_0 \geq 0$, т.че

$\forall n \geq n_0 : T(n) \leq c \cdot n$

// Няма база, защото не се интересуваме от началните условия

ИП: $T(\frac{n}{2}) \leq c \cdot \frac{n}{2}$

Стъпка: $T(n) = 2T(\frac{n}{2}) + 1 \overset{\text{ИП}}{\leq} 2(c \cdot \frac{n}{2}) + 1 = c.n + 1 \overset{?}{\leq} c.n$

Няма такава константа $c>0$, за която да е изпълнено горното неравенство.

Ще използваме засилване на твърдението:

Търсим $c>0, b>0$, така че $T(n) \leq c.n - b$ за дост. големи n.

ИП: $T(\frac{n}{2}) \leq c \frac{n}{2} - b$

Стъпка: $T(n) = 2T(\frac{n}{2}) + 1 \overset{\text{ИП}}{\leq} 2.(c.\frac{n}{2} - b) + 1 = cn - 2b + 1$

Искаме $cn - 2b + 1 \leq cn - b$

$\Updownarrow$

$cn - b + 1 - b \leq cn - b$

$\Updownarrow$

$1 - b \leq 0 \iff b \geq 1$

Тогава за $\forall b \geq 1$ $\forall c > 0$ е вярно, че $T(n) \leq cn - b$

$\Rightarrow T(n) = O(n)$

2) $T(n) = \Omega(n)$

Търсим константа $d>0$, т.че за достатъчно големи n:

$T(n) \geq d.n$

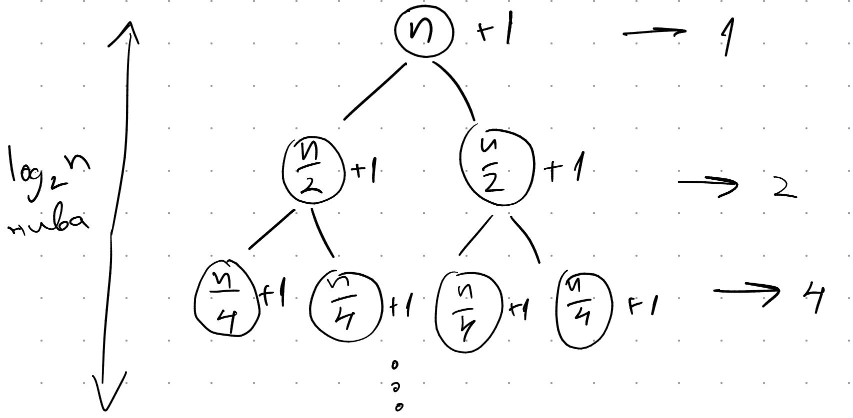
ИП: $T(\frac{n}{2}) \geq d.\frac{n}{2}$

Стъпка: $T(n) = 2T(\frac{n}{2}) + 1 \overset{\text{ИП}}{\geq} 2d.\frac{n}{2} + 1 = dn + 1 \geq d.n$ за $\forall d > 0$

$\Rightarrow T(n) = \Omega(n)$

Следователно, $T(n) = \Theta(n)$

• За тази задача можем да разгледаме и дърво на рекурсията:



Общо тегло:

$$\sum_{i=0}^{\lg n} 2^i = \Theta(n)$$

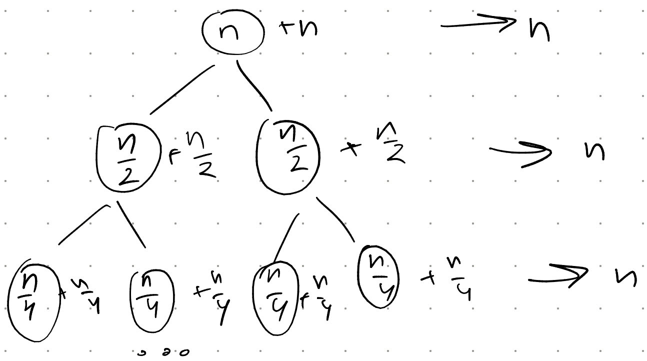
// геом. прогресия

$$S = \frac{1-q^n}{1-q} = \frac{1-2^{\lg n}}{1-2} = \frac{1-n}{-1} = n-1 = \Theta(n)$$

Този метод отново не е формален!

Зад 2 Намерете асимптотично решение на $T(n) = 2T(\frac{n}{2}) + n$ чрез дърво на рекурсията

Решение:



Дървото има $\log_2 n$ нива $\Rightarrow T(n) = n \lg n = \Theta(n \lg n)$

ДР Докажете по индукция

Зад 3 Каква е асимптотиката на $T(n) = T(n-1) + n$?

Решение: $T(n) = T(n-1) + n = T(n-2) + (n-1) + n = T(n-3) + (n-2) + (n-1) + n =$

$$= \ldots = T(0) + 1 + 2 + \ldots + (n-1) + n = T(0) + \sum_{i=1}^{n} i = T(0) + \underbrace{\frac{n(n+1)}{2}}_{\Theta(n^2)} = \Theta(n^2)$$

Отново трябва да го докажем формално:

1) $T(n) = O(n^2)$

Търсим константа $c > 0$, т.че за достатъчно големи n:

$$T(n) \leq c \cdot n^2$$

ИП: $T(n-1) \le c(n-1)^2$

Стъпка: $T(n) = T(n-1) + n \overset{ИП}{\le} c(n-1)^2 + n = cn^2 - 2cn + c + n$

$$cn^2 - 2cn + c + n \le cn^2 \iff$$
$$-2cn + c + n \le 0 \iff$$
$$n(1-2c) + c \le 0 \iff$$
$$1 - 2c \le -c$$
$$1 \le c \quad и \ \forall n \ge 1$$

2) $T(n) = \Omega(n^2)$

Търсим конст. $d > 0$, т.че за дост големи n: $T(n) \ge d.n^2$

ИП: $T(n-1) \ge d(n-1)^2$

Стъпка: $T(n) = T(n-1) + n \overset{ИП}{\ge} d(n-1)^2 + n = dn^2 - 2dn + d + n$

$$dn^2 - 2dn + d + n \overset{?}{\ge} dn^2$$
$$-2dn + d + n \ge 0$$
$$n(1-2d) + d \ge 0$$
$$d > 0$$
$$\Rightarrow 1 - 2d \ge 0$$
$$d \le \frac{1}{2} \quad и \ \forall n \ge 0$$

ДР $\quad T(n) = 2T(n-1) + n \asymp ?$

Зад 4 Решете уравнението $T(n) = nT(n-1) + 1$

Решение: $T(n) = nT(n-1) + 1 = n((n-1)T(n-2) + 1) + 1 =$

$= n(n-1)T(n-2) + n + 1 = n(n-1)((n-2)T(n-3) + 1) + n + 1 =$

$= n(n-1)(n-2)T(n-3) + n(n-1) + n + 1 =$
$= n(n-1)(n-2)(n-3)T(n-4) + n(n-1)(n-2) + n(n-1) + n + 1 = \dots$

$= n(n-1)(n-2)\dots 2.\underset{\overset{\shortparallel}{1}}{T(1)} + n(n-1)\dots 4.3 + n(n-1)\dots 5.4 + \dots + n(n-1) + n + 1 =$

$= n! + \frac{n!}{2} + \frac{n!}{3} + \dots + \frac{n!}{(n-2)!} + \frac{n!}{(n-1)!} + \frac{n!}{n!} = n!\sum_{i=1}^{n}\frac{1}{i!} = n!\sum_{i=1} a_i$

$$\lim_{i\to\infty}\frac{a_{i+1}}{a_i}=\lim_{i\to\infty}\frac{\frac{1}{(i+1)!}}{\frac{1}{i!}}=\lim_{i\to\infty}\frac{1}{i+1}=0 \Rightarrow \exists c\text{ - конст.: } \sum_{i=0}^{\infty} a_i \le c$$

$$\Rightarrow \left.\begin{array}{l} T(n)\le n!\,c \\ \text{и } T(n)\ge n!.1 \end{array}\right\} \Rightarrow T(n)=\Theta(n!)$$

// Може и с интуицията, че $\sum_{i=1}^{n}\frac{1}{i!}$ много бързо става много малко

Ще докажем по индукция:

1) $T(n)=O(n!)$

Директно ще използваме засилено предположение:

Търсим конст. $b>0, c>0$, такива че за достатъчно големи n е вярно:

$$T(n)\le cn!-b \le c.n!$$

↑ за да махнем $+1$

ИП: $T(n-1)\le c.(n-1)!-b$

$$T(n)=nT(n-1)+1 \overset{\text{ИП}}{\le} n(c.(n-1)!-b)+1=cn!-bn+1$$

$$cn!-bn+1 \overset{?}{\le} cn!-b$$

$$-bn\le -b-1$$

$$bn\ge b+1$$

$$n\ge\frac{b+1}{b}$$ - да, за дост. големи n при $b>0$

2) $T(n)=\Omega(n!)$

Тривиално се проверява, че $\exists d>0: T(n)\ge d.n!$ за дост. големи n.

$$\Rightarrow T(n)=n!$$