ДАА

упр.№5, 23 април 2020г.

Hello, world!

Док. че алг. връща второто по големина число в A

SecLargest(A[1..n]), където n > 2

Нека вс. числа са различни

```
1.  p ← max{A[1], A[2]}
2.  q ← min{A[1], A[2]}
3.  for i ← 3 to n
4.    if A[i] > p
5.      q ← p
6.      p ← A[i]
7.    else if A[i] > q
8.      q ← A[i]
9.  return q
```

Инв: При вс. дост. на ред 3 $p = \max(A[1..i-1])$
q = второто по големина в стария подмасив

База: При първо дост. $i=3 \Rightarrow A[1..i-1]$ ще съдържа два ел-та $\Leftrightarrow$ по-малкият от тях е втори по големина $\Rightarrow$ след ред 1 и ред 2
Инв. е изпълнена.

Подд.: Нека инв. е изп. на някое дост. на ред 3, което не е посл. Относно $A[i]$ имаме 3 сл.:

1 сл. $A[i] > p$. От инв. знаем, че p е най-голям в $A[1..i-1]$ и $A[i] > p \Rightarrow$ за $A[1..i]$ $A[i]$ е най-големият. Също така в $A[1..i]$ има точно 1 ел-т $> p \Rightarrow p$ е втори в $A[1..i]$

SecLargest(A[1..n]), където n > 2

```
1.  p ← max{A[1], A[2]}
2.  q ← min{A[1], A[2]}
3.  for i ← 3 to n
4.    if A[i] > p
5.      q ← p
6.      p ← A[i]
7.    else if A[i] > q
8.      q ← A[i]
9.  return q
```

След ред 5 новото $q' = p =$ второ в $A[1..i]$

и след ред 6 новото $p' =$ първо по гол. в $A[1..i]$

$\Rightarrow$ при след. дост. на ред 3 новото $i' = i+1$

$\Rightarrow$ спрямо новото i' Инв. е изпълнена

II сл. $p > A[i] > q \Rightarrow p$, който по Инв. е

най-голям в $A[1..i-1]$, е също най-голям в

$A[1..i] \Rightarrow$ при след. достигане на ред 3 тази

от-т няма да бъде променена.

За $q \rightarrow$ от Инв. в $A[1..i-1]$ има точно 1 по-голямо

$\Rightarrow$ в $A[1..i]$ има точно две такива (p и $A[i]$)

$\Rightarrow$ за $A[1..i]$ числото $A[i]$ е второ по големина

$\Rightarrow$ след ред 8 $q' = A[i]$

III сл.: $p > q > A[i]$ ⇒ и от Инв. следва, че p е макс. в $A[1..i]$ и q е второ по гол. в $A[1..i]$ ⇒ Нашият алг. няма да ги промени

⇒ при следв. дост. на ред 3 спрямо $i' = i+1$ .... Инв. е изпълн.

Терм.: Т.к. пром. на цикъла нараства на $\forall$ итерация, а горната му граница не се променя ⇒ ще има краен брой итерации и при посл. дост. на ред 3 $i = n+1$ ⇒ според Инв. p е макс. в $A[1..i-1]$, което е $A[1..n]$ и q е второ по големина в същия подмасив

⇒ след цикъла алг. връща точно тази ст-т

⇒ Тв. за алг. е вярно

$$T(n) = \Theta(1) + \sum_{i=3}^{n} \Theta(1) = \Theta(n)$$

$$M(n) = \Theta(1)$$

Док. Тв: Алг. връща произв. на $A[1..n]$

Product(A[1..n])

Инв: При всяко дост. на ред 2 $p = \prod_{k=1}^{i-1} A[k] * \prod_{k=j+1}^{n} A[k]$

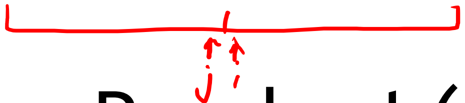
$A[1..i-1] * A[j+1..n]$

```
1.  i ← 1, j ← n, p ← 1
2.  while i < j
3.    p ← p*A[i]*A[j]
4.    i ← i + 1, j ← j - 1
5.  if n is odd
6.    p ← p*A[(n+1)/2]
7.  return p
```

База: .... $i=1$; $j=n$ ⇒ подмасивите $A[1..i-1]$ и $A[j+1..n]$ ⇒ имат произв. 1 и това е стойността на p (от ред 1) ⇒ Инв. е вярна.

Подд.: Нека Инв е изп. при някое дост., което не е ~~пр~~ последно ⇒ от Инв. $p = \prod A[1..i-1] * \prod A[j+1..n]$ ⇒ след ред 3 новото $p' = p * A[i] * A[j] \overset{\text{Инв.}}{=} \prod A[1..i-1] * \prod A[j+1..n] * A[i] * A[j]$

$\Rightarrow p' = \prod A[1..i] * \prod A[j..n]$.



Product(A[1..n])

След ред 4: $i' = i+1;\ j' = j-1 \Rightarrow$ съответно
$p' = \prod A[1..i'-1] * \prod A[j'+1..n] \Rightarrow$ при следв. дост. на ред 2 Инв. е вярна.

1. i ← 1, j ← n, p ← 1
2. while i < j
3. p ← p*A[i]*A[j]
4. i ← i + 1, j ← j - 1
5. if n is odd
6. p ← p*A[(n+1)/2]
7. return p

<u>Терм!</u> Т.к. на всяка итерация i нараства, а j намалява, то след краен брой итерации i ще стане не по-малко от j

<u>Инв2:</u> $i-1 = n-j$

$i+1-1 \overset{?}{=} n-j+1$

$\Rightarrow i'-1 = n-j'$

// т.е. подмасивите, чието произв. съдържа p, са едн. големи

$p = \prod A[1..n]$ ще върне алг.

$\Rightarrow$ I сл. n - нечетно $\Rightarrow$ при посл. дост. на ред 2
Инв.2 $i = j \Rightarrow i-1 = n-i \Leftrightarrow i = \frac{n+1}{2}$

Инв.1 $\Rightarrow p = \prod A[1..i-1] * \prod A[i+1..n]$

$\Rightarrow$ след ред 6 p ще се умн. по $A\left[\frac{n+1}{2}\right] \Rightarrow$ точно $A[i]$ $\Rightarrow$ при дост. на ред 7

II сл. n е четно $\Rightarrow$ при посл. дост. на ред 2 $i = j+1$

Ч4б.1 $\Rightarrow$ $p = \prod A[1..i-1] * \prod A[i..n] = \prod A[1..n]$

$\Rightarrow$ тази ст-т няма да бъде променена $\Rightarrow$ алг. връща нея

$\Rightarrow$ твърдението е вярно

---

for $i \leftarrow 1$ to n

for $j \leftarrow i$ to n

sum $\leftarrow 0$

for $k \leftarrow i$ to j

sum $\leftarrow$ sum $+ A[k]$

$$T(n) = \Theta(n) + \sum_{i=1}^{n} \sum_{j=i}^{n} 1 = \Theta(n^2)$$

$$M(n) = \Theta(n)$$

$$\sum A[i..j] = \sum A[1..j] - \sum A[1..i-1]$$

$$= B[j] - B[i-1]$$

$A[1...n]$

Префиксни суми: B $A[1]$, $A[1]+A[2]$, ...

$$\boxed{B[i] = \sum A[1..i]}$$

$$= B[i-1] + A[i]$$

Суфиксни суми C

$$\boxed{C[i] = \sum A[i..n]}$$

Тв: Алг. връща макс сума на подмасив на A[1..n]

Kadane(A[1..n]), където n > 0 и A[1] ≥ 0

```
1.  curr ← A[1], best ← A[1]
2.  for i ← 2 to n
3.    curr ← curr + A[i]
4.    if curr < 0
5.      curr ← 0
6.    if curr > best
7.      best ← curr
8.  return best
```

1, 2, 3, -10 ...

Инв: При ∀ дост. на ред 2 curr = макс. сума на суфикс на A[1..i-1], а best = макс. сума на подмасив на A[1..i-1]

База: ... i=2, т.е. A[1..i-1] има 1 ел-т. ⇒ два суфикса (с 0 или 1 ел-т) ⇒ т.к. A[1] ≥ 0, то суфиксът A[1..1] е този с макс. сума ⇒ тази сума е A[1] и това е стойността на curr (от ред 1). Аналог. за best ⇒ Инв.

Подд: Нека Инв е изп. ....

за A[1..i] непразните суфикси са {празния → 0; A[i] + суфикс на A[1..i-1]} ⇒ след ред 3 curr = макс. сума на тези

Kadane(A[1..n]), където n > 0 и A[1] ≥ 0

```
1.  curr ← A[1], best ← A[1]
2.  for i ← 2 to n
3.    curr ← curr + A[i]
4.    if curr < 0
5.      curr ← 0
6.    if curr > best
7.      best ← curr
8.  return best
```

⇒ след ред 4 и 5 curr' = макс. сума измежду ∀ суфикси на A[1..i]

curr

Аналог.: Подм. на A[1..i] = $\begin{cases} \text{суф. на } A[1..i] \\ \text{подмасиви } A[1..i-1] \end{cases}$

Инв best

⇒ след редове 6 и 7 best' = max{curr', best}

⇒ best' = макс. сума на подмасив на A[1..i]

⇒ .... Инв. е изпълнена

Терм.: ~~Поддр.:~~ .... i = n+1 ⇒ от Инв best = ... A[1..n] и алг. връща него ⇒ т.е. вярно

FastExp(x, n)

```
if n = 0
  return 1
if n = 1
  return x
k ← n/2 // целочислено деление
if n is even
  return sqr(FastExp(x, k)) // sqr(x) връща x*x
else
  return x * sqr(FastExp(x, k))
```