

Зад. 1

Даден е масив $A[1..n][1..2] \in (\mathbb{N} \times \mathbb{N})^n$, който съдържа n на брой двойки от естествени числа със семантика на времеви интервали - съответно началата и краищата на интервалите.

а) Да се намери максимален брой непресичащи се интервали от A .

б) Да се намери броя на интервалите от масива A , които са строго съдържащи се в поне един интервал от масива A

Note: Интервалите (2, 3) и (3, 5) нямат сечение.. т.е всички интервали са отворени!

Task1a($A[1..n][1..2]$, n):

```
sortDecByColumn( $A[1..n][1..2]$ ,  $n$ , 2, column=1)
cnt ← 0
firstAviable ←  $A[1][2]$  // или  $+\infty$ 
for  $i \leftarrow 1$  to  $n$ 
    if  $A[i][2] \leq$  firstAviable then
        firstAviable ←  $A[i][1]$ 
        cnt ← cnt+1
return cnt
```

Task1b($A[1..n][1..2]$, n):

```
sortDecByColumn( $A[1..n][1..2]$ ,  $n$ , 2, column=2)
sortIncByColumn( $A[1..n][1..2]$ ,  $n$ , 2, column=1) //stable  $O(n \log n)$ 
cnt ← 0
currMax ←  $A[1][2]$ 
itsStart ←  $A[1][1]$ 
for  $i \leftarrow 2$  to  $n$ 
    if ( $A[i][2] \leq$  currMax and  $A[i][1] >$  itsStart) or ( $A[i][2] <$  currMax and  $A[i][1] \geq$  itsStart) then
        cnt ← cnt+1
    if  $A[i][2] >$  currMax then
        currMax ←  $A[i][2]$ 
        itsStart ←  $A[i][1]$ 
```

Зад. 2

Дадено е естествено число k и n на брой тръби с дължини, намиращи се в масива $A[1..n] \in \mathbb{N}^n$. Компания произвеждаща тръби има заявка за k на брой тръби с еднаква и при това максимална дължина. Помогнете на компанията да прецени търсената дължина.

$n=10$

$k=1$

7 1 1 1 1 1 1 1 1 1

1 - ok

2 - ok

3 - not ok

4 - not ok

- 5 - not ok
- 6 - not ok
- 7 - not ok
- 8 - obviously not

```

Check2(A[1..n][1..2], n, k, L):
  cnt ← 0
  for i ← 1 to n
    cnt ← cnt + ⌊  $\frac{A[i]}{L}$  ⌋
    if cnt ≥ k then
      return TRUE
  return FALSE

```

```

Task2(A[1..n], n, k):
  minL ← 1
  maxL ← max(A[1..n], n) + 1 // +1 е за ±1 сметките
  while minL < maxL do
    midL ← ⌊  $\frac{\text{minL} + \text{maxL}}{2}$  ⌋
    if Check2(A[1..n], n, k, midL) = FALSE then
      maxL ← midL
    else
      minL ← midL + 1
  return minL - 1

```

Зад. 3

Дадено е цяло число k и масив $A[1..n] \in \mathbb{Z}^n$. Да се намери $\max\{L \mid \exists i \in \{1, \dots, n-k+1\}: \forall j \in \{i, \dots, i+k-1\} \text{ е изп. } A[j] \geq L\}$.

```

Check3(A[1..n], n, k, L):
  cnt ← 0
  for i ← 1 to n
    if A[i] ≥ L then
      cnt ← cnt + 1
    if cnt ≥ k then
      return TRUE
    if A[i] < L then
      cnt ← 0
  return FALSE

```

Task3... същата като Task2, но minL ще бъде $\min(A[1..n], n)$

Зад. 4

Дадена е масив $A[1..n] \in \{0, \dots, U\}^n$ с уникални числа, където $n \ll U$, и много на брой заявки от вида:

Вход: $k \in \{0, \dots, U\}$

Изход: $\text{succ}_A(k) = \text{IF } \exists x \in A \ x \geq k \ \text{THEN } \min \{x \in A \mid x \geq k\} \ \text{ELSE } -1$

Да се напишат 2 алгоритъма: индексация - $\text{Index}(A[1..n])$ и заявка - $\text{Query}(k, \dots)$, където заявката максимално оптимална

Ограничение: $O(\text{Index}) \ll U$

A: 60 57 42 31 20 15 13 7 3

$U = 2^6$

60 → 111100

57 → 111001

42 → 101010

31 → 011111

20 → 010100

15 → 001111

13 → 001101

07 → 000111

03 → 000011

$a > b$

$\text{bin}(a) >_{\text{lex}} \text{bin}(b)$

Struct Node:

parent, left, right = NULL

mostLeft, mostRight = NULL

value = -1

Индексацията ще прави съответно:

1. За всяко ниво k на дървото ще си направим хеш (перфектен) S_k , който ще ни казва в конст. време дали даден префикс с дължина k се съдържа вътре.

$O(n \log U)$

2. Листата да ги направим на свързан списък.

$O(n \log U)$

3. Към всяко съст. ще си добавим най-ляво листо на поддървото му и аналогично за дясното

$O(n \log U)$

4. Към всяко съст. ще си добавим родителя му и двата му преки наследника и ще пазаим стойността в листата

$O(n \log U)$

5. Масив $\text{pos}[k] = 2^k$, $k \leq \lceil \log U \rceil$

$O(\log U)$

```

succ(root, k): //O(loglogU)
  leaf ← root.mostRight
  if leaf.value < k then
    return -1
  l ← 0
  r ← ⌈logU⌉
  if k ∈ Sr then
    return k
  while l+1 < r do
    m ← ⌊ $\frac{l+r}{2}$ ⌋
    if  $\frac{k}{\text{pow}[\lceil\log U\rceil - m]} \in S_m$  then
      l ← m
    else
      r ← m
  q ← Sm $\left(\frac{k}{\text{pow}[\lceil\log U\rceil - l]}\right)$ 
  if q.right ≠ NULL then
    return q.mostLeft.value
  return next(q.mostRight).value

```