

Зад. 1 (Зад. 4 от миналия път)

Дадена е матрица $A[1..m][1..n] \in \{0, 1\}^{m \times n}$. Да се намери най-големия квадрат от единици и да се изведе от колко единици е съставен.

Пример:

```
0 1 0 1 1 0
1 1 1 1 1 1
1 0 0 1 1 1
1 1 0 1 1 1
1 1 1 1 1 1
```

Изход:

9

task1($A[1..m][1..n]$, m, n):

```
  d[1..m][1..n] //init matrix
  ans ← 0
  for i ← 1 to n
    d[1][i] ← A[1][i]
    if A[1][i] = 1 then
      ans ← 1
  for i ← 2 to m
    for j ← 1 to n
      if A[i][j] = 0 then
        d[i][j] ← 0
      else
        d[i][j] ← 1 + min(d[i-1][j], d[i-1][j-1], d[i][j-1])
        ans ← max(ans, d[i][j])
  return ans*ans
```

Зад. 2

Дадена е матрица $A[1..m][1..n] \in \{0, 1\}^{m \times n}$. Да се намери най-големия правоъгълник от единици и да се изведе от колко единици е съставен.

Пример:

```
0 1 1 0 0 1
1 1 0 1 1 1
1 1 1 1 0 1
0 0 1 1 1 1
1 1 1 1 1 1
1 1 1 0 1 1
```

Изход:

8

```
// 2 3 4 3 2 5 4 4 2 3 2 4
```

```
maxRectAreaInHist(A[1..n], n): //not okay
  maxArea ← 0
  s ← Stack.Init() //stack of indexes
  Нека си разширим A[0..n+1] = 0 + A[1..n] + 0
  for i ← 0 to n+1
    while s.isEmpty()=FALSE and A[i] ≤ A[s.top()] do
      area ← (i-s.top())*A[s.top()]
      maxArea ← max(maxArea, area)
      s.pop()
    idx ← s.top()
    s.push(i)
    area ← A[i]*(i-idx)
    maxArea ← max(maxArea, area)
  return maxArea

maxRectAreaInHist2(A[1..n], n): //seems okay.. ще го разгледаме другия път
  maxArea ← 0
  s ← Stack.Init() //stack of indexes
  Нека си разширим A[0..n+1] = 0 + A[1..n] + 0
  for i ← 0 to n+1
    while s.isEmpty()=FALSE and A[i] ≤ A[s.top()] do
      idx ← s.pop()
      area ← (i-s.top()-1)*A[idx]
      maxArea ← max(maxArea, area)
    if s.isEmpty()=TRUE or A[i] > A[s.top()] then
      s.push(i)
  return maxArea

maxRectArea(A[1..m][1..n], m, n):
  maxArea ← 0
  arr[1..n] ← [0, ..., 0]
  for i ← 1 to m
    for j ← 1 to n
      if A[i][j]=1 then
        arr[j] ← arr[j]+1
      else
        arr[j] ← 0
  maxArea ← max(maxArea, maxRectAreaInHist(arr, n))
  return maxArea
```

Зад. 3

Даден е низ $s[1..n]$ съставен от малки латински букви. Да се намери най-дългия палиндрим на подмасив

а) от последователни елементи

б) със (строго) нарастващи индекси на елементите

а)

Пример:

Вход:

асабсба

Изход:

5 //абсба

остана за другия път...

б)

Пример:

Вход:

adbсба

Изход:

5 //абсба

остана за другия път

Зад. 4

Даден е масив $A[1..n] \in \mathbb{N}_0^n$. Двама играча играят последователно ходове, като на всеки ход взимат или най-левия или най-десния елемент на масива (и масива намаля с 1-ица съответно). Печели този, чиято сума на взетите числа е най-голяма. Да се определи максималната сума за 2та играча, при положение, че и 2та играча играят оптимално.

Пример:

Вход:

3 9 1 2

Изход:

(11, 4)

//p1: 2+9

//p2: 3+1

task4(A[1..n], n):

 d[1..n][1..n] //matrix of (int, int) couples

 for i←1 to n

 d[i][i]←(A[i], 0)

 for i←1 to n-1

 for j←i+1 to n

 if A[i]+d[i+1][j].second > A[j]+d[i][j-1].second then

 d[i][j].first←A[i]+d[i+1][j].second

 d[i][j].second←d[i+1][j].first

```
else  
    d[i][j].first ← A[j] + d[i][j-1].second  
    d[i][j].second ← d[i][j-1].first  
return d[1][n]
```