зад ② При подаден непразен стринг Str да се намери дължината на най-дълга палиндромна поредица – стринг, който се състои от елементи на Str, не непременно преки съседи, но взети в реда, в който се появяват в Str.

// пример "bbbab" – отговор 4 – "bbbb"

"cbbd" – отговор 2 – "bb"

Решение:

Създаване 2D таблица – dp[i][j] ще съдържа стойността на най-дълга палиндромна поредица в подстринга str[i...j]

За всеки подниз Str[i...j] разглеждаме str[i] и str[j], ако str[i] == str[j], то

$$dp[i][j] = dp[i+1][j-1] + 2$$

ако $str[i] \neq str[j]$, то

$$dp[i][j] = \cancel{dp}\ \max(dp[i+1][j],\ dp[i][j-1])$$

↳ отговорът ще бъде стойността на dp[1][N]

Рекурсивна дефиниция:

$$dp[i][j] := \begin{cases} 0, & i > j \\ 1, & i == j \\ dp[i+1][j-1] + 2, & j > i \ \&\& \ str[i] == str[j] \\ \max(dp[i+1][j],\ dp[i][j-1]), & j > i \ \&\& \ str[i] \neq str[j] \end{cases}$$

```
PalindromeSubseq(Str[1..N])
dp[N][N]
for i ← 1 to N
      dp[i][i] ← 1
          for j ← 1 to N
              for i ← j+1 to N
                  dp[i][j] ← 0
for i ← N-1 to 1
    for j ← i+1 to N
        if Str[i] = Str[j]
            dp[i][j] ← dp[i+1][j-1] + 2
        else
            dp[i][j] ← max(dp[i+1][j], dp[i][j-1])
return dp[1][N]
```

③ По дадено неотрицателно число N. Да се разбие на поне две положителни числа, сумата на които е N, а произведението им е максимално.

// Пример: $N=2$ – отговор 1 – $2=1+1$: $1*1=1$

$N=10$ – отговор 36 – $10=3+3+4$: $3*3*4=36$

Решение:

Правим таблица $dp[1..N]$ като $dp[i]$ съдържа максималното произведение, което може да получим от разбиването на числото i.

Рекурсивна дефиниция:

$$dp[i] := \begin{cases} 0, & i=1 \,||\, i=0 \\ 1, & i=2 \\ \max(j*(i-j),\ j*dp[i-j] \mid 1 \le j \le i-1) \end{cases}$$

```
MaxProduct(N)
        //if N=0 || N=1 return 0
dp[1..N]
dp[1] ← 0; dp[2] ← 1
for i ← 3 to N
    dp[i] ← 0
    for j ← 1 to i-1
        dp[i] ← max(dp[i], max(j*(i-j), j*dp[i-j]))
return dp[N]
```

зад ④ Дадени са два масива – Arr1 и Arr2 от положителни числа.

Числата от Arr1 и Arr2 се записват на два успоредни хоризонтални реда. Свързваща линия между елемент Arr1[i] и Arr2[j] може да се нарисува, ако Arr1[i] == Arr2[j] и линията не пресича никоя друга свързваща такава (както не може от едно число да излиза повече от една линия). Да се намери максималният брой от свързващи линии, които могат да се нарисуват.

// Пример

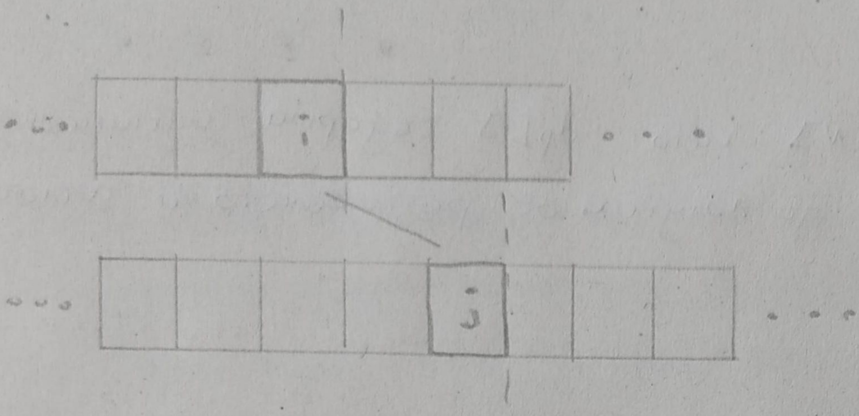
1 4 2

1 2 4

– отговор 2

Решение:

Правим двумерна таблица $dp[N+1][M+1]$, където $|Arr1| = N$ и $|Arr2| = M$. Клетка $dp[i][j]$ ще съдържа стойност - максималния брой непресичащи се свързващи линии, които могат да се построят м/у $Arr1[1..i]$ и $Arr2[1..j]$. Търсеният отговор ще се съдържа в клетка $dp[N][M]$.



За всяка наредена двойка (i,j) ще проверяваме:

1сл) $Arr1[i] = Arr2[j]$ е в сила

Тогава $dp[i][j] = dp[i-1][j-1] + 1$

максималният брой непресичащи се свързващи линии, които не включват елементи на позиция i от $Arr1$ и j от $Arr2$

по този начин избягваме да пресечем някоя съществуваща линия, прокарвайки линията (i,j)

2сл) $Arr1[i] \neq Arr2[j]$ е в сила

Тогава $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

Рекурсивна дефиниция:

$$dp[i][j] := \begin{cases} 0, & i=0 \,\|\, j=0 \\ dp[i-1][j-1]+1, & \overset{(\Delta)}{i>0 \,\&\&\, j>0} \,(\&\& \overset{(*)}{Arr1[i] = Arr2[j]}) \\ \max(dp[i-1][j], dp[i][j-1]), & (\Delta) \,\&\&\, \neg(*) \end{cases}$$

```
MaxUncrossedLinesCount(Arr1[1..N], Arr2[1..M])
dp[N+1][M+1]
dp[0][1..M+1] ← 0 // for нулата
dp[1..N+1][0] ← 0 // for нулата
for i ← 1 to N
    for j ← 1 to M
        if Arr1[i] = Arr2[j]
            dp[i][j] ← dp[i-1][j-1] + 1
        else
            dp[i][j] ← max(dp[i-1][j], dp[i][j-1])
return dp[N][M]
```

④