

## Задача ОТРЕЗИ

### Пояснение към решенията

#### Бавно решение (segments\_31p.cpp)

Елементите на дадената редица се прочитат в масива  $a[i]$ . Използва се тройно вложен цикъл, в който за всеки отрез с краища  $i$  и  $j$  се пресмята сумата от елементите му и всички така пресметнати суми се натрупват в променливата  $s$ .

```
long long s=0;
for(int i=1;i<=n;i++)
for(int j=i;j<=n;j++)
for(int k=i;k<=j;k++)
    s=s+a[k];
```

#### По-бързо решение (segments\_69p.cpp)

Може да напишем програма с двойно вложен цикъл по следни начин: Започваме в цикъл с индекс  $i$  да пробягваме левите краища на отрезите и за всеки такъв отрез пробягваме всички негови възможни дължини с вътрешен цикъл с индекс  $d$ . При всяка следваща стойност на  $d$ , за да пресметнем сумата на елементите на отреза с краища  $i$  и  $d$ , използваме, че вече в  $s$  имаме пресметнатата сумата на елементите на отреза с краища  $i$  и  $d-1$ :

```
for(int i=1;i<=n;i++)
{
    long long s=0;
    for(int d=0;d<=n-i;d++)
    {
        s = s+a[i+d];
        t = t+s;
    }
}
```

В променливата  $t$  се натрупва общата сума, която се отпечатва накрая.

#### Бързо решение (segments\_100p.cpp)

За да напишем програма без вложени цикли, а само с един цикъл, правим следното наблюдение, за пример, когато  $n = 5$ . Работим с масив, индексирани от 1. Написваме индексите на елементите, които участват в отрезите, започващи с индекс 1.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Това е първата група индекси, които разглеждаме. Втората група индекси е за отрезите, започващи с индекс 2:

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

Продължаваме да образуваме групи от индекси, отнасящи се за отрезите, които започват съответно с индекс 3, 4 и 5:

```
3
3 4
3 4 5
```

```
4
3 5
```

```
5
```

Така разглеждаме 5 групи от редове с индекси, като броят на редовете в групите е съответно 5, 4, 3, 2 и 1.

Индекс 1 се среща 5 пъти в една група – първата

Индекс 2 се среща по 4 пъти в две групи, а именно в първата и втората групи

Индекс 3 се среща по 3 пъти в три групи, а именно в първата, втората и третата групи

и т.н.

Да запишем в масива  $c[i]$  колко пъти се среща индекса  $i$  във всичките групи (в случая групите са 5):

```
c[1]=5*1
c[2]=4*2
c[3]=3*3
c[4]=2*4
c[5]=1*5
```

В общия случай може да запълним масива  $c[]$  със следния цикъл:

```
int k=n;
for(int i=1;i<=n;i++) {c[i]=k*i; k--;}
```

Вместо цикъл, може да използваме формулата  $c[i] = (n-i+1) * i$

Така в програмата всъщност може въобще да не се ползва този масив и всичките пресмятания да се извършат само с един цикъл:

```
long long int s=0;
for(int i=1;i<=n;i++) s = s + (long long int)a[i]*i*(n-i+1);
cout << s << endl;
```

*Зорница Дженкова*