

Задача НАЙ-МАЛКО

Пояснение към решението

Бавно решение (сортиране)

Прочитаме в масива $a[]$ дадената редица. След това сортираме масива в ненамаляващ ред. Решението на задачата ще търсим измежду числата по-големи от m . Затова първо проверяваме дали $m+1 < a[0]$. Ако е така, отпечатваме $m+1$ и излизаме от програмата.

Ако не е така, започваме да обхождаме с цикъл `for` сортирания масив по реда на индексите му. В променливата p се записва предишният посетен елемент. В процеса на обхождането, когато за първи път се окаже, че $p+1 < a[i]$, това означава че стойността на $p+1$ не е от “разглежданите” и при това, е най-малката такава, и освен това е по-голяма от m . В този случай отпечатваме $p+1$ и излизаме от програмата. Ако не сме излезли от цикъла по описания начин, тогава трябва да отпечатаме $p+1$, чиято стойност след излизане от цикъла е равна на едно плюс стойността на най-големия елемент от дадената редица.

Бързо решение (с масив за маркиране)

Използваме

```
vector<char> v(n, false);
```

в който маркираме с `true` индексите i на вектора, които съответстват на стойностите a от дадената редица, такива че $a > m$. Маркирането става чрез $v[a-m]=true$, т.е. чрез изместване на индексите с m , за да се гарантира, че стойността $a-m$ принадлежи на дефиниционната област на вектора v , защото $a-m > 0$, поради оператора `if(a>m)` и $a-m \leq n$, поради ограниченията на задачата.

След маркирането обхождаме v и търсим първия немаркиран елемент. При намирането му отпечатваме $i+m$ и излизаме от програмата:

```
for (int i = 1; i <= n; i++)
{
    if (!v[i]) {cout << i+m << endl; return 0;}
}
```

Ако не сме излезли от цикъла по описания начин, това означава, че всичките елементи $v[1], \dots, v[n]$ са маркирани и те са n на брой. Тогава трябва да

отпечатаме $n+1+m$, чиято стойност след излизане от цикъла е равна на едно плюс стойността на най-големия елемент от дадената редица.

Бързо решение с малко памет (булев вектор за маркиране)

В бързото решение по-горе използваме вектор с елементи, които заемат по един байт от паметта. За да намалим паметта може да използваме вектор с елементи, които заемат по един бит. Това е реализирано чрез специалният за STL вектор `vector<bool>`.

Емил Келеведжиев