

## Вонземјани

Нашиот сателит само што откри вонземјанска цивилизација на една далечна планета. Веќе имаме добиено фотографија со ниска резолуција на една квадратна област од планетата. Фотографијата покажува многу знаци на интелигентен жив свет. Нашите експерти идентификуваа  $n$  точки од интерес во фотографијата. Точките се нумерирани со целите броеви од  $0$  до  $n - 1$ . Сега би сакале да направиме фотографии со висока резолуција кои ќе ги содржат сите  $n$  точки.

Интерно, сателитот ја има поделено областа од фотографијата со ниска резолуција на  $m$  по  $m$  квадратна мрежа (анг. grid) од единечни квадратни ќелии. Редиците и колоните на мрежата се последователно нумерирани со целите броеви од  $0$  до  $m - 1$  (одгоре-надолу и одлево-надесно, соодветно). Со  $(s, t)$  ќе ја означуваме ќелијата во редицата  $s$  и колоната  $t$ . Точката од интерес со број  $i$  се наоѓа во ќелијата  $(r_i, c_i)$ . Секоја ќелија може да содржи произволен број на точки од интерес.

Нашиот сателит е на стабилна орбита која што поминува директно над *главната* дијагонала на квадратната мрежа. Главна дијагонала на мрежата е отсечката која што го поврзува горното лево ќоше со долното десно ќоше на мрежата. Сателитот може да направи фотографија со висока резолуција на која било област која што ги задоволува следниве ограничувања:

- обликот на областа е квадрат,
- две спротивни ќошиња на квадратот лежат на главната дијагонала на квадратната мрежа,
- секоја ќелија на квадратната мрежа е или целосно во фотографираната област или пак целосно надвор од неа (не е можно фотографираната област да содржи само дел од некоја ќелија).

Сателитот може да направи најмногу  $k$  фотографии со висока резолуција.

Откако сателитот ќе заврши со правење на фотографии, тој ќе ја испрати фотографијата со висока резолуција на секоја од фотографираниите ќелии кон нашата база (независно од тоа дали таа ќелија содржи или не содржи точки од интерес). Податоците за секоја фотографирана ќелија ќе бидат испратени само *еднаш*, дури и ако истата била фотографирана повеќе пати.

Според тоа, треба да избереме најмногу  $k$  квадратни области што ќе бидат фотографирани, при што ќе важи дека:

- секоја ќелија која што содржи барем една точка од интерес е фотографирана барем еднаш, и
- бројот на ќелии што се фотографирани барем еднаш е минимизиран.

Ваша задача е да го најдете најмалиот можен вкупен број на фотографирани ќелии.

## Детали за имплементација

Треба да ја имплементирате следнава функција (метод):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: бројот на точки од интерес,
  - `m`: бројот на редици (еднаков со бројот на колони) во квадратната мрежа (анг. grid),
  - `k`: максималниот број на фотографии што може да ги направи сателитот,
  - `r` и `c`: две низи со должина `n` кои што ги опишуваат координатите на ќелиите од мрежата што содржат точки од интерес. За секое `i`,  $0 \leq i \leq n - 1$ , `i`-тата точка од интерес се наоѓа во ќелијата `(r[i], c[i])`,
  - функцијата треба да го враќа најмалиот можен вкупен број на ќелии кои што се фотографирани барем еднаш (знаејќи дека фотографиите треба да ги покријат сите точки од интерес).  
Ве молиме да ги користите дадените темплејт датотеки за детали околу имплементацијата во вашиот програмски јазик.

## Примери

### Пример 1

```
take_photos(5, 7, 2, [0, 4, 4, 4, 4], [3, 4, 6, 5, 6])
```

Во овој пример имаме  $7 \times 7$  квадратна мрежа со 5 точки од интерес. Точките од интерес се наоѓаат во четири различни ќелии:  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$  и  $(4, 6)$ . Може да направите најмногу 2 фотографии со висока резолуција.

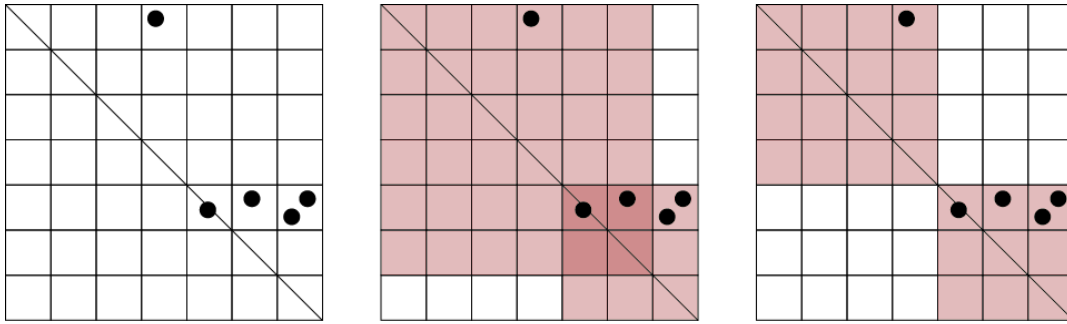
Еден начин да се опфатат сите пет точки од интерес е да се направат две фотографии: една фотографија од  $6 \times 6$  квадратот што ги содржи ќелиите  $(0, 0)$  и  $(5, 5)$ , и уште една фотографија од  $3 \times 3$  квадратот што ги содржи ќелиите  $(4, 4)$  и  $(6, 6)$ . Ако сателитот ги направи овие две фотографии, истиот ќе испрати податоци за вкупно 41 фотографирани ќелија. Овој број не е оптимален.

Оптималното решение користи една фотографија за да го опфати  $4 \times 4$  квадратот што ги содржи ќелиите  $(0, 0)$  и  $(3, 3)$ , и уште една фотографија за да го опфати  $3 \times 3$  квадратот што ги содржи ќелиите  $(4, 4)$  и  $(6, 6)$ . Ова ќе резултира со само 25 фотографирани ќелии, што е оптимално, па функцијата `take_photos` треба да врати 25.

Да забележиме дека е доволно да се фотографира ќелијата  $(4, 6)$  само еднаш, иако таа содржи две точки од интерес.

Овој пример е прикажан на сликите подолу. Левата слика ја прикажува квадратната мрежа што одговара на овој пример. Централната слика го

прикажува неоптималното решение во кое е фотографирана 41 ќелија. Десната слика го прикажува оптималното решение.

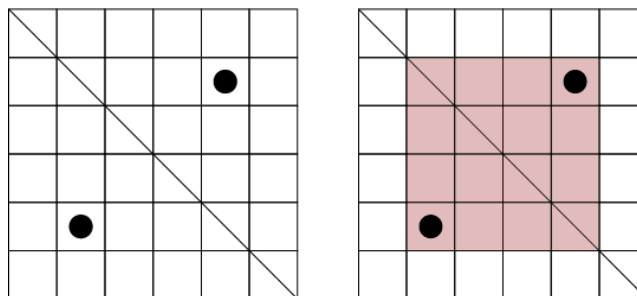


## Пример 2

`take_photos(2, 6, 2, [1, 4], [4, 1])`

Овде имаме 2 точки од интерес лоцирани симетрично: во ќелиите  $(1, 4)$  и  $(4, 1)$ . Која било валидна фотографија која што содржи една од овие точки - ја содржи, исто така, и другата точка. Според тоа, доволно е да се искористи една фотографија.

На сликите подолу е прикажан овој пример, како и неговото оптимално решение. Во ова решение сателитот прави една фотографија од 16 ќелии.



## Подзадачи

Кај сите подзадачи,  $1 \leq k \leq n$ .

1. (4 поени)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 поени)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , за секое  $i$  така што  $0 \leq i \leq n - 1$ ,  $r_i = c_i$ ,
3. (9 поени)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 поени)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 поени)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 поени)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

## Оценувач

Оценувачот чита податоци од стандардниот влез во следниот формат:

- линија 1: целите броеви  $n$ ,  $m$  и  $k$ ,
- линија  $2 + i$  ( $0 \leq i \leq n - 1$ ): целите броеви  $r_i$  и  $c_i$ .