

## Aliens

Satelitul nostru a descoperit recent o civilizație extraterestră pe o planetă îndepărtată. Am obținut deja o fotografie în rezoluție joasă a unei zone pătrate a suprafeței planetei. Pe fotografie pot fi observate mai multe semne de prezență a unei civilizații. Expertii au identificat  $n$  puncte de interes pe fotografie. Punctele sunt numerotate de la  $0$  la  $n - 1$ . Acum dorim să obținem fotografii de rezoluție înaltă care să conțină toate aceste  $n$  puncte.

Satelitul a divizat aria fotografiei de rezoluție joasă într-un caroiaj de  $m$  pe  $m$  celule pătrate cu laturi unitare. Atât liniile, cât și coloanele caroiajului sunt numerotate consecutiv de la  $0$  la  $m - 1$  (pornind din stânga, respectiv -sus). Vom folosi  $(s, t)$  pentru a marca celula din linia  $s$  și coloana  $t$ . Punctul cu numărul  $i$  se află în celula  $(r_i, c_i)$ . Fiecare celulă poate conține un număr arbitrar de asemenea puncte.

Satelitul nostru are o orbită stabilă care trece direct pe diagonala *principală* a caroiajului. Diagonala principală este segmentul de linie care unește colțul de stînga-sus cu colțul de dreapta-jos al caroiajului. Satelitul poate face fotografii de rezoluție înaltă pe orice suprafață care satisface următoarelor condiții:

- conturul suprafeței este un pătrat,
- două colțuri diagonal opuse ale pătratului aparțin diagonalei principale a caroiajului,
- Oricare celulă a caroiajului este în întregime sau în interiorul suprafeței fotografiate, sau în afara ei.

Satelitul poate face cel mult  $k$  fotografii de rezoluție înaltă.

După ce satelitul finalizează procesul de fotografiere, urmează transmiterea către bază a imaginii de înaltă rezoluție pentru fiecare din celulele fotografiate (indiferent dacă celula conține sau nu puncte de interes). Datele pentru fiecare celulă fotografiată vor fi transmise *o singură dată*, chiar dacă celula a fost fotografiată de mai multe ori.

Așadar, urmează să fie selectate pentru fotografiere cel mult  $k$  suprafețe pătrate, astfel încât:

- fiecare celulă care conține cel puțin un punct de interes va fi fotografiată cel puțin o dată, și
- numărul celulelor care vor fi fotografiate cel puțin o dată va fi minimal.

Sarcina ta este să găsești cel mai mic număr total posibil al celulelor fotografiate.

### Detalii de implementare

Trebuie să implementezi următoarea funcție (metodă):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: numărul punctelor de interes,
  - `m`: numărul de linii (și coloane) în caroiaj,
  - `k`: numărul maximal de fotografii care pot fi făcute de satelit,
  - `r` și `c`: două array-uri de lungime `n` descriind coordonatele celulelor din caroiaj care conțin puncte de interes. Pentru  $0 \leq i \leq n-1$ , punctul `i` de interes este localizat în celula `(r[i], c[i])`,
  - funcția va returna cel mai mic număr total posibil de celule care sunt fotografiate cel puțin o dată (în condiția în care fotografiile trebuie să conțină toate punctele de interes).

Te rugăm să folosești fișierele-template furnizate pentru detalii de implementare în limbajul de programare pe care îl utilizezi.

## Exemple

### Exemplul 1

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

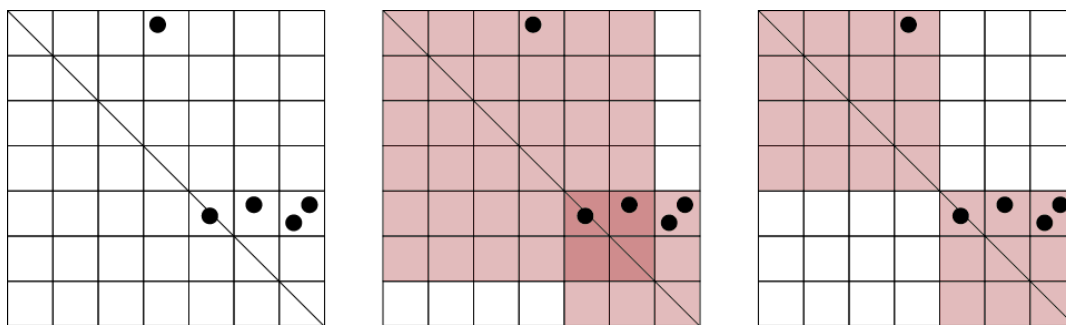
În acest exemplu avem un caroiaj  $7 \times 7$  cu 5 puncte de interes. Punctele de interes sunt localizate în patru celule disticte: `(0, 3)`, `(4, 4)`, `(4, 5)` și `(4, 6)`. Poți face cel mult 2 fotografii de rezoluție înaltă.

O soluție pentru a captura toate cele cinci puncte de interes este de a face două fotografii: prima, un pătrat de dimensiunea  $6 \times 6$  conținând celulele `(0, 0)` și `(5, 5)`, și a doua, un pătrat de dimensiunea  $3 \times 3$  conținând celulele `(4, 4)` și `(6, 6)`. Dacă satelitul va face aceste două fotografii, vor fi transmise date despre 41 de celule. Această soluție nu este optimă.

Soluția optimă folosește o fotografie pentru a captura imaginea unui pătrat  $4 \times 4$  care va conține celulele `(0, 0)` și `(3, 3)` și o a doua fotografie pentru a captura imaginea unui pătrat  $3 \times 3$  care va conține celulele `(4, 4)` și `(6, 6)`. Acest rezultat conține doar 25 de celule fotografiate, rezultat optim. Astfel `take_photos` va returna 25.

De remarcat că este suficient ca celula `(4, 6)` să fie fotografiată o singură dată, chiar dacă conține două puncte de interes.

Acest exemplu este prezentat în figurile care urmează. Figura din stânga prezintă caroiajul care corespunde exemplului. Figura din mijloc prezintă soluția suboptimală, în care sunt fotografiate 41 celule. Figura din dreapta prezintă soluția optimă.

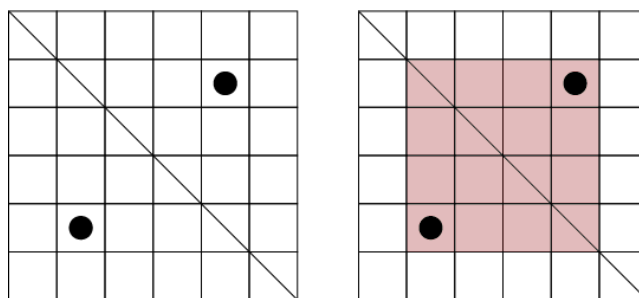


## Exemplul 2

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

Aici avem **2** puncte de interes localizate simetric în celulele  $(1, 4)$  și  $(4, 1)$ . Oricare fotografie validă care conține unul dintre puncte, îl va conține și pe celălalt. Prin urmare, este suficient să fie făcută o singură fotografie.

Figura care urmează prezintă acest exemplu și soluția lui optimă. În această soluție satelitul captează o singură fotografie cu imagini a **16** celule.



## Subtask-uri

Pentru toate subtask-urile,  $1 \leq k \leq n$ .

1. (4 puncte)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 puncte)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , pentru toți  $i$  astfel încât  $0 \leq i \leq n - 1$ ,  $r_i = c_i$ ,
3. (9 puncte)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 puncte)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 puncte)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 puncte)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

## Sample grader

Sample grader-ul citește inputul în următorul format:

- linia 1: întregii  $n$ ,  $m$  și  $k$ ,
- linia  $2 + i$  ( $0 \leq i \leq n - 1$ ): întregii  $r_i$  și  $c_i$ .