

## Aliens

Nuestro satélite recientemente ha descubierto una civilización alienígena en un planeta remoto. Hemos obtenido una foto en baja resolución de un área cuadrada del planeta que muestra varios signos de vida inteligente. Nuestros expertos han identificado  $n$  puntos de interés en la foto. Estos puntos están numerados de  $0$  a  $n - 1$ . Ahora queremos tomar fotos de alta resolución que en conjunto contengan todos los  $n$  puntos de interés.

Internamente, el satélite ha dividido el área de la foto en baja resolución en una cuadrícula de  $m$  por  $m$  celdas. Tanto los renglones como las columnas están numeradas consecutivamente de  $0$  a  $m - 1$  (iniciando arriba y en la izquierda). Denotamos  $(s, t)$  como la celda en el renglón  $s$  y la columna  $t$ . El  $i$ -ésimo punto de interés está ubicado en la celda  $(r_i, c_i)$ . Una celda puede tener un número arbitrario de puntos de interés.

Nuestro satélite se encuentra en una órbita estable que pasa directamente sobre la diagonal *principal* de la cuadrícula. La diagonal principal es el segmento de línea que conecta la esquina de arriba a la izquierda con la esquina de abajo a la derecha de la cuadrícula.

El satélite puede tomar fotos de alta resolución de cualquier área que satisfaga las siguientes restricciones:

- la forma del área siempre es cuadrada,
- las esquinas opuestas del cuadrado (arriba a la izquierda y abajo a la derecha) están contenidas en la diagonal principal de la cuadrícula,
- cada celda de la cuadrícula está completamente adentro o completamente afuera del área fotografiada.

El satélite puede tomar a lo más  $k$  fotos de alta resolución.

Una vez que el satélite termine de tomar fotos, transferirá a nuestra base una foto de alta resolución de cada celda fotografiada (independientemente de si la celda contiene puntos de interés). La información de cada celda será transmitida *sólo una* vez, incluso si la celda fue fotografiada varias veces.

Por lo tanto, tenemos que escoger a lo más  $k$  áreas cuadradas que serán fotografiadas, asegurando que:

- cada celda que contiene al menos un punto de interés es fotografiada al menos una vez, y
  - el número de celdas que son fotografiadas al menos una vez es minimizado. Tu tarea es encontrar este número.
- Tu tarea es encontrar el mínimo número posible de celdas fotografiadas.

## Detalles de implementación

Debes implementar la siguiente función (método):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: el número de puntos de interés,
  - `m`: el número de renglones (y de columnas) de la cuadrícula,
  - `k`: el máximo número de fotos que el satélite puede tomar,
  - `r` y `c`: dos arreglos de longitud `n` que describen las coordenadas de las celdas que contienen puntos de interés. Para  $0 \leq i \leq n - 1$ , el  $i$ -ésimo punto de interés indica sus coordenadas: `(r[i], c[i])`,
  - la función debe regresar la mínima cantidad de celdas que deben ser fotografiadas al menos una vez dado que se tienen que cubrir todos los puntos de interés.

Por favor usa los archivos plantilla proporcionados de tu lenguaje de programación.

## Ejemplos

### Ejemplo 1

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

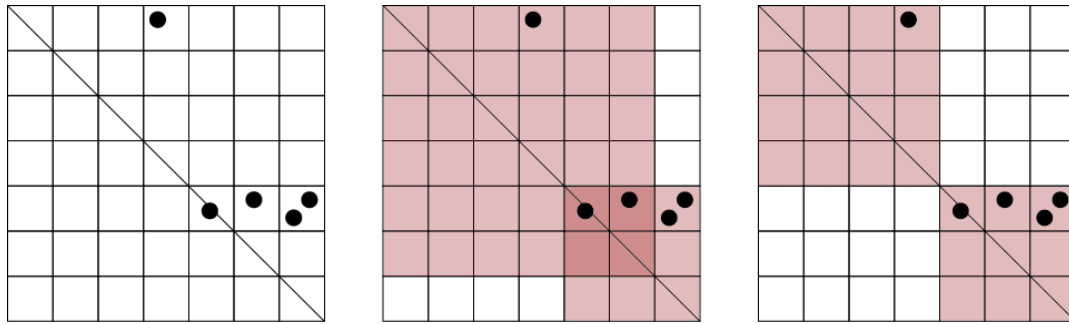
En este ejemplo tenemos una cuadrícula de  $7 \times 7$  con 5 puntos de interés. Los puntos de interés están localizados en 4 celdas distintas: `(0, 3)`, `(4, 4)`, `(4, 5)` y `(4, 6)`. Puedes tomar a lo más dos fotos de alta resolución.

Una forma de capturar todos los puntos de interés es tomando las siguientes dos fotos: una foto de un cuadrado de  $6 \times 6$  donde las esquinas opuestas son `(0, 0)` y `(5, 5)` y otra foto de un cuadrado de  $3 \times 3$  con esquinas opuestas `(4, 4)` y `(6, 6)`. Si tomamos estas dos fotos, el satélite transmitirá las fotos de 41 celdas. Esta cantidad no es óptima.

La solución óptima usa una foto para capturar el cuadrado de  $4 \times 4$  con esquinas opuestas `(0, 0)` y `(3, 3)`, y toma otra foto para capturar el cuadrado de  $3 \times 3$  con esquinas opuestas `(4, 4)` y `(6, 6)`. El resultado es 25 celdas fotografiadas que es la solución óptima, por lo tanto `take_photos` debe regresar 25.

Puedes notar que es suficiente fotografiar la celda `(4, 6)` una vez, incluso cuando contiene dos puntos de interés.

El ejemplo está ilustrado abajo. La figura de la izquierda muestra la cuadrícula que corresponde al ejemplo. La figura del centro muestra una solución que no es óptima, en la cual 41 celdas fueron fotografiadas. La figura de la derecha muestra la solución óptima donde 25 celdas fueron fotografiadas.



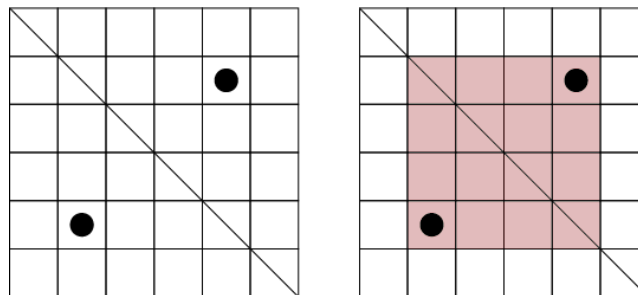
## Ejemplo 2

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

En este caso tenemos 2 puntos de interés en posiciones simétricas: en las celdas  $(1, 4)$  y  $(4, 1)$ .

Cualquier foto válida que contenga uno de los puntos de interés también abarcará el otro punto de interés. Por lo tanto, es suficiente con tomar sólo una foto.

En la figura de abajo se muestra el ejemplo y la solución óptima donde el satélite captura una sola foto con 16 celdas.



## Subtareas

Para todas las subtareas,  $1 \leq k \leq n$ .

1. (4 puntos)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 puntos)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , para toda  $i$  tal que  $0 \leq i \leq n - 1$ ,  $r_i = c_i$ ,
3. (9 puntos)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 puntos)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 puntos)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 puntos)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

## Evaluador de ejemplo

El evaluador de ejemplo lee la entrada de la siguiente forma:

- línea 1: enteros  $n$ ,  $m$  y  $k$ ,
- línea  $2 + i$  ( $0 \leq i \leq n - 1$ ): los enteros  $r_i$  y  $c_i$ .