



Aliens

Nuestro satélite acaba de descubrir una civilización alienígena en un planeta remoto. Hemos obtenido una foto de baja resolución de un área cuadrada del planeta. La foto muestra diversos signos de vida inteligente. Nuestros expertos han identificado n puntos de interés en la foto. Los puntos están enumerados del 0 al $n - 1$. Ahora queremos tomar fotos de alta resolución que contengan todos los n puntos.

Internamente, el satélite ha dividido el área de la foto de baja resolución en un grid de m por m celdas cuadradas de lado uno. Tanto las filas como las columnas del grid están consecutivamente enumeradas del 0 al $m - 1$ (desde arriba e izquierda, respectivamente). Utilizamos (s, t) para referirnos a la celda ubicada en la fila s y columna t . El i -ésimo punto se ubica en la celda (r_i, c_i) . Cada celda puede contener un número arbitrario de estos puntos.

Nuestro satélite se encuentra en una órbita estable que pasa directamente sobre la diagonal *principal* del grid. La diagonal principal es el segmento de línea que conecta las esquinas Norte-Oeste y Sur-Este del grid. El satélite puede tomar una foto de alta resolución de cualquier área que satisfaga las siguientes restricciones:

- la forma del área es un cuadrado,
- dos esquinas opuestas del cuadrado están ubicadas en la diagonal principal,
- cada celda del grid se encuentra completamente dentro o completamente fuera del área fotografiada.

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

Una vez el satélite haya tomado las fotos, este transmitirá la foto de alta resolución de cada una de las celdas fotografiadas a nuestra base (sin importar que la celda contenga o no puntos de interés). La data para cada celda fotografiada será transmitida *una sola vez*, aun si la celda haya sido fotografiada múltiples veces.

Por lo tanto, debemos escoger a lo sumo k áreas cuadradas que serán fotografiadas, asegurándonos que:

- cada celda que contenga al menos un punto de interés es fotografiada al menos una vez, y
- el número de celdas fotografiadas al menos una vez es minimizado.

Tu tarea es encontrar el número total más pequeño 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 filas (y columnas) en el grid,
- 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 del grid que contienen puntos de interés. Para $0 \leq i \leq n - 1$, el i -ésimo punto de interés se encuentra en la celda $(r[i], c[i])$,
- la función debe retornar el número más pequeño posible de celdas que son fotografiadas al menos una vez (dado que las fotos deben cubrir todos los puntos de interés).

Por favor utiliza los archivos plantilla provistos para conocer los detalles de implementación en tu lenguaje de programación.

Ejemplos

Ejemplo 1

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

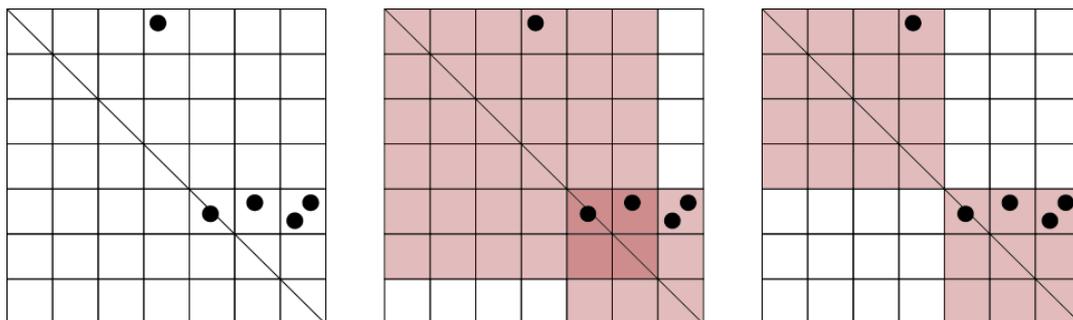
En este ejemplo tenemos un grid de 7×7 con 5 puntos de interés. Los puntos de interés se ubican en cuatro celdas distintas: $(0, 3)$, $(4, 4)$, $(4, 5)$ y $(4, 6)$. Puedes tomar a lo sumo 2 fotos de alta resolución.

Una manera de capturar todos los cinco puntos de interés es tomando dos fotos: una del cuadrado 6×6 con las celdas $(0, 0)$ y $(5, 5)$, y la otra del cuadrado 3×3 con las celdas $(4, 4)$ y $(6, 6)$. Si el satélite toma estas dos fotos, este transmitirá la data de 41 celdas. Este número no es óptimo.

La solución óptima utiliza una foto para capturar el cuadrado 4×4 que contiene las celdas $(0, 0)$ y $(3, 3)$ y otra foto para capturar el cuadrado 3×3 que contiene las celdas $(4, 4)$ y $(6, 6)$. De esta forma, solo 25 celdas son fotografiadas, lo cual es óptimo, así que `take_photos` retornaría 25.

Fíjate que es suficiente fotografiar la celda $(4, 6)$ una sola vez, aunque esta contenga dos puntos de interés..

Este ejemplo se muestra en las imágenes debajo. La figura de la izquierda muestra el grid que corresponde a este ejemplo. La figura del medio muestra la solución sub-óptima en la cual 41 celdas son fotografiadas. La figura de la derecha muestra la solución óptima.

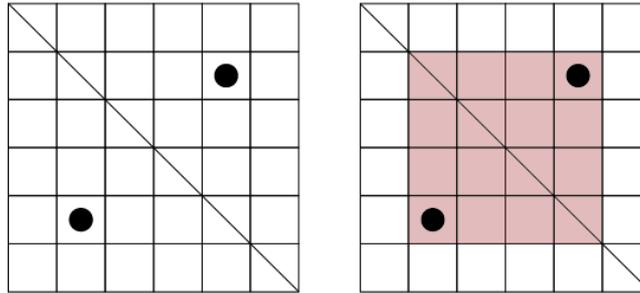


Ejemplo 2

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

Aquí tenemos 2 puntos de interés ubicados simétricamente: en las celdas (1,4) y (4,1). Cualquier foto válida que contenga uno de ellos lo contiene al otro también. Por tanto, es suficiente utilizar una sola foto.

Las figuras debajo muestran este ejemplo y su solución óptima. En esta solución el satélite captura una única foto con 16 celdas.



Sub-Tareas

En todas las sub-tareas, $1 \leq k \leq n$.

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) $1 \leq n \leq 2000$, $1 \leq m \leq 1\,000\,000$,
5. (12 puntos) $1 \leq n \leq 20\,000$, $1 \leq k \leq 100$, $1 \leq m \leq 1\,000\,000$,
6. (7 puntos) $1 \leq n \leq 20\,000$, $1 \leq k \leq 1000$, $1 \leq m \leq 1\,000\,000$,
7. (40 puntos) $1 \leq n \leq 200\,000$, $1 \leq m \leq 1\,000\,000$.

Grader de Ejemplo

El grader de ejemplo lee la entrada en el siguiente formato:

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