

Aliens

Nuestro satélite ha descubierto una civilización alien en un planeta remoto. Ya hemos obtenido una fotografía en baja resolución de un área de forma cuadrada del planeta. La fotografía muestra muchos signos de vida inteligente. Nuestros expertos han identificado n puntos de interés en la fotografía. Los puntos están numerados desde 0 a $n - 1$. Ahora queremos tomar fotografías en alta resolución que contengan todos esos n puntos.

Internamente, el satélite ha dividido el área de la fotografía en baja resolución en una grilla de m por m celdas cuadradas. Tanto las filas como las columnas de la grilla están numeradas consecutivamente de 0 a $m - 1$ (desde arriba y desde la izquierda, respectivamente). El par (s, t) denota la celda en la fila s y la columna t . El punto número i está ubicado 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* de la grilla. La diagonal principal es la línea que conecta la esquina superior izquierda con la esquina inferior derecha de la grilla. El satélite puede tomar una fotografía en alta resolución de cualquier área que satisfaga las siguientes restricciones:

- la forma del área es un cuadrado,
- dos esquinas opuestas del cuadrado se encuentran en la diagonal principal de la grilla,
- cada celda de la grilla está completamente contenida o completamente afuera del área fotografiada.

El satélite es capaz de tomar a lo más k fotografías en alta resolución.

Una vez que el satélite termine su labor, transmitirá a nuestra base la fotografía en alta resolución de cada celda que fue fotografiada (sin importar si la celda contiene puntos de interés o no). La información para cada celda fotografiada será transmitida solo *una vez*, incluso si fue fotografiada varias veces.

De esta manera, tenemos que escoger a lo más k áreas cuadradas para ser fotografiadas, asegurando que:

- cada celda que contiene al menos un punto de interés es fotografiada por lo menos una vez, y
- el número de celdas que fueron fotografiadas al menos una vez sea el mínimo posible.

Tu tarea es encontrar este número.

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 también columnas) en la grilla,
 - `k`: el máximo número de fotografías que el satélite puede tomar,
 - `r` y `c`: dos arreglos de tamaño `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 está ubicado en la celda $(r[i], c[i])$,
 - la función debe retornar el número que minimiza la cantidad de celdas que son fotografiadas al menos una vez, de modo que las fotografías cubran todos los puntos de interés.

Usar los archivos con las plantillas para obtener 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])
```

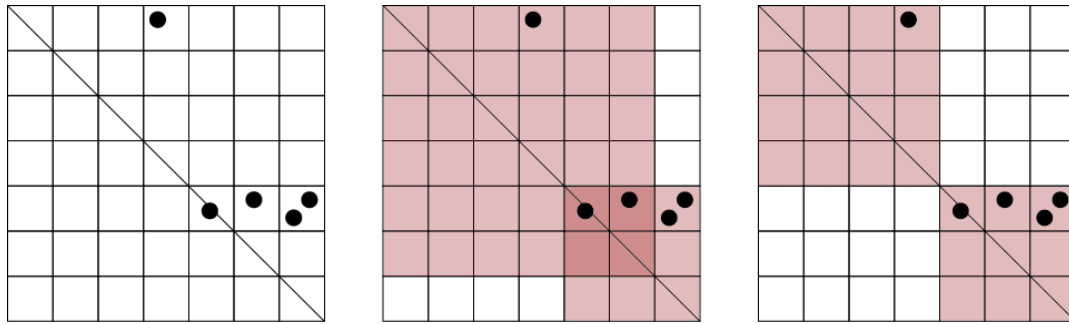
En este ejemplo tenemos una grilla de 7×7 con 5 puntos de interés. Los puntos de interés están ubicados en cuatro celdas diferentes: $(0, 3)$, $(4, 4)$, $(4, 5)$, y $(4, 6)$. Puedes tomar a lo más 2 fotografías en alta resolución.

Una forma de capturar los cinco puntos de interés es tomando dos fotografías: una con la celda $(0, 0)$ como esquina superior izquierda y la celda $(5, 5)$ como esquina inferior derecha, y la otra con la celda $(4, 4)$ como esquina superior izquierda y la celda $(6, 6)$ como esquina inferior derecha. Si tomamos estas dos fotografías, el satélite transmitirá las fotografías de 41 celdas. Este valor no es óptimo.

La solución óptima usa una fotografía para capturar el cuadrado de 4×4 contenido entre las celdas $(0, 0)$ y $(3, 3)$ y otra fotografía para capturar el cuadrado de 3×3 contenido entre las celdas $(4, 4)$ y $(6, 6)$. Esto resulta en solo 25 celdas fotografiadas, que es el óptimo, así que `take_photos` debe retornar 25.

Notar que basta con fotografiar la celda $(4, 6)$ solo una vez, pese a que contiene dos puntos de interés.

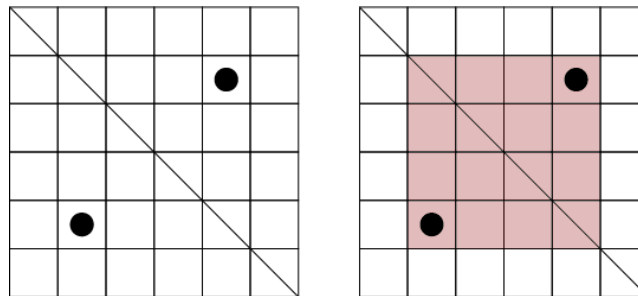
Este ejemplo es ilustrado en las figuras de más abajo. La figura de más a la izquierda muestra la grilla que corresponde a este ejemplo. La figura de al medio muestra la solución subóptima en la cual 41 celdas son fotografiadas. La figura de la derecha muestra la solución óptima.



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 fotografía válida que contenga a uno de ellos, contiene también al otro. Por lo tanto, basta con utilizar una única fotografía. La solución óptima (descrita a continuación) fotografía 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 todo 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) $1 \leq n \leq 100\,000$, $1 \leq m \leq 1\,000\,000$.

Grader de ejemplo

El grader de ejemplo lee el input 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 .