



## Aliens

Nosso satélite acabou de descobrir uma civilização alienígena em um planeta remoto. Já obtivemos uma foto de baixa resolução de uma área quadrada do planeta. Essa foto mostra muitos sinais de vida inteligente. Nossos especialistas identificaram  $n$  pontos de interesse na foto. Os pontos são numerados de  $0$  a  $n - 1$ . Agora queremos tirar fotos de alta resolução que contenham todos esses  $n$  pontos.

Internamente, o satélite dividiu a área da foto de baixa resolução em um reticulado de células quadradas unitárias de  $m$  linhas por  $m$  colunas. As linhas e colunas do reticulado são numeradas consecutivamente de  $0$  a  $m - 1$  (a partir do topo e da esquerda, respectivamente). Usamos  $(s, t)$  para representar a célula na linha  $s$  e coluna  $t$ . O ponto de número  $i$  está localizado na célula  $(r_i, c_i)$ . Cada célula pode conter um número arbitrário desses pontos.

Nosso satélite está em uma órbita estável que passa diretamente sobre a diagonal *principal* do reticulado. A diagonal principal é o segmento de linha que conecta o vértice superior esquerdo ao vértice inferior direito do reticulado. O satélite pode tirar uma foto de alta resolução de qualquer área que satisfaça as seguintes restrições:

- o formato da área é um quadrado,
- dois vértices opostos do quadrado estão sobre a diagonal principal do reticulado,
- cada célula do reticulado ou está completamente fora ou completamente dentro da área fotografada.

O satélite é capaz de tirar no máximo  $k$  fotos de alta resolução.

Uma vez que o satélite termine de tirar fotos, ele transmitirá a foto de alta resolução de cada célula fotografada para a nossa base (independentemente de a célula conter ou não pontos de interesse). Os dados de cada célula fotografada serão transmitido *somente uma vez*, ainda que a célula seja fotografada múltiplas vezes.

Portanto, devemos escolher no máximo  $k$  áreas quadradas que serão fotografadas, assegurando que:

- cada célula que contenha pelo menos um ponto de interesse seja fotografada pelo menos uma vez, e
- o número de células fotografadas pelo menos uma vez seja o menor possível.

Sua tarefa é encontrar o menor número possível de células fotografadas.

### Detalhes da implementação

Você deve implementar a seguinte função (método):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`

- $n$ : o número de pontos de interesse,
- $m$ : o número de linhas (e colunas) do reticulado,
- $k$ : o número máximo de fotos que o satélite pode tirar,
- $r$  e  $c$ : dois vetores de tamanho  $n$  descrevendo as coordenadas das células do reticulado que contêm pontos de interesse. Para  $0 \leq i \leq n-1$ , o  $i$ -ésimo ponto de interesse está localizado na célula  $(r[i], c[i])$ ,
- a função deve retornar o menor número possível de células que serão fotografadas pelo menos uma vez (considerando que as fotografias cubram todos os pontos de interesse).

Por favor use os arquivos modelo fornecidos para detalhes da implementação na sua linguagem de programação.

## Exemplos

### Exemplo 1

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

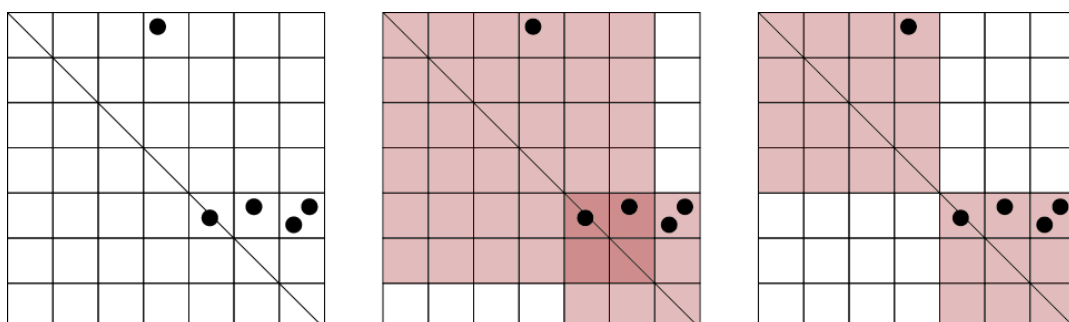
Neste exemplo temos um reticulado  $7 \times 7$  com 5 pontos de interesse. Os pontos de interesse estão localizados em quatro células diferentes:  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$  e  $(4, 6)$ . Você pode tirar no máximo 2 fotos de alta resolução.

Uma maneira de fotografar todos os cinco pontos de interesse é tirar duas fotos: uma foto do quadrado  $6 \times 6$  que contém as células  $(0, 0)$  e  $(5, 5)$ , e uma foto do quadrado  $3 \times 3$  que contém as células  $(4, 4)$  e  $(6, 6)$ . Se o satélite tirar essas duas fotos, ele transmitirá dados sobre 41 células. Essa quantidade não é ótima.

A melhor solução usa uma foto para capturar o quadrado  $4 \times 4$  que contém as células  $(0, 0)$  e  $(3, 3)$  e outra foto para capturar o quadrado  $3 \times 3$  que contém as células  $(4, 4)$  e  $(6, 6)$ . Isso resulta em somente 25 células fotografadas, que é o valor ótimo, portanto `take_photos` deve retornar 25.

Note que é suficiente fotografar a célula  $(4, 6)$  uma vez, mesmo que ela contenha dois pontos de interesse.

Este exemplo é mostrado nas imagens abaixo. A imagem à esquerda mostra o reticulado correspondente a este exemplo. A imagem central mostra a solução sub-ótima na qual 41 células são fotografadas. A imagem à direita mostra a solução ótima.

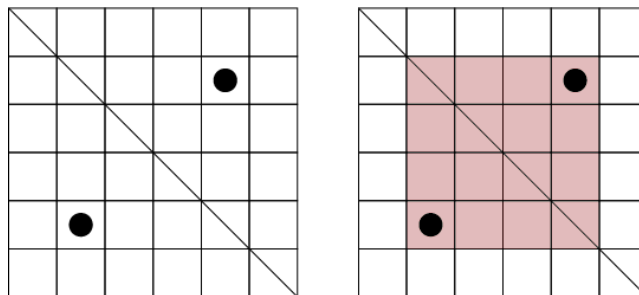


## Exemplo 2

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

Aqui nós temos 2 pontos de interesse localizados simetricamente: nas células (1, 4) e (4, 1). Qualquer foto que contenha um deles também contém o outro. Dessa forma, é suficiente usar uma única foto.

As imagens abaixo mostram este exemplo e sua solução ótima. Nesta solução o satélite captura uma única foto de 16 células.



## Subtarefas

Para todas as subtarefas,  $1 \leq k \leq n$ .

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

## Corretor exemplo

O corretor exemplo lê a entrada no seguinte formato:

- linha 1: inteiros  $n$ ,  $m$  e  $k$ ,
- linha  $2 + i$  ( $0 \leq i \leq n - 1$ ): inteiros  $r_i$  e  $c_i$ .