



## Aliens

인공위성이 어떤 외계 행성의 궤도를 돌다가 외계 문명을 발견했다. 그 행성의 어떤 정사각형 영역의 저해상도 사진을 확보했다. 사진에는 지능이 있는 생명체의 흔적이 많이 나타난다. 전문가들이 사진에서  $n$  개의 흥미로운 위치를 찾아내었다. 흥미로운 위치들은  $0$  부터  $n - 1$  까지 번호가 붙어 있다. 흥미로운  $n$  개의 위치를 모두 포함하는 고해상도 사진을 찍으려고 한다.

인공위성은 저해상도 사진을  $m \times m$  격자로 나누었다. 이렇게 나누었을 때, 정사각형인 하나의 칸을 격자칸이라고 부른다. 격자의 행과 열들은  $0$  부터  $m - 1$  까지 번호가 붙어 있다 (위에서 아래로, 왼쪽에서 오른쪽으로). 우리는  $s$  번째 행,  $t$  번째 열에 있는 격자칸을  $(s, t)$  로 표시한다. 흥미로운 위치  $i$  번은 격자칸  $(r_i, c_i)$  에 포함된다. 하나의 격자칸은 여러 개의 흥미로운 위치를 포함할 수 있다.

인공위성은 행성의 안정적인 궤도를 돌고 있는데, 그 궤도는 격자칸의 **중심 대각선** 위를 지난다. 중심 대각선이라는 것은 격자의 왼쪽 위 구석과 오른쪽 아래 구석을 잇는 선분을 말한다. 인공위성은 다음 조건을 만족하는 임의의 영역에 대해서 고해상도 사진을 찍을 수 있다.

- 영역의 모양이 정사각형이다.
- 대각 꼭지점 한 쌍이 모두 중심 대각선 위에 있다.
- 각각의 격자칸은 영역에 완전히 포함되거나 전혀 포함되지 않아야 한다.

인공위성은 최대  $k$  개의 고해상도 사진을 찍을 수 있다.

인공위성이 사진을 모두 찍고 나면, 인공위성은 사진에 포함된 모든 (흥미로운 위치가 있는지에 무관하게) 격자칸들에 대한 고해상도 사진을 기지로 전송할 것이다. 사진에 포함된 한 격자칸의 데이터는 단 **한 번**만 전송된다. 즉, 한 격자칸이 여러번 사진에 찍혔더라도 단 한 번만 전송된다는 것이다.

따라서, 다음의 조건을 만족하도록 최대  $k$  개의 정사각형 영역을 골라야 한다.

- 흥미로운 위치를 포함하는 격자칸은 적어도 하나의 사진에 포함된다.
- 적어도 한 번 이상 사진에 찍힌 격자칸의 개수가 최소라야 한다.

당신이 할 일은 위의 조건을 만족하는 격자칸의 최소 개수를 찾는 것이다.

## Implementation details

다음 함수를 구현하여야 한다.

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - $n$ : 흥미로운 위치의 개수,
  - $m$ : 격자칸의 행(열)의 개수,
  - $k$ : 인공위성이 찍을 수 있는 사진의 최대 개수,
  - $r$ 과  $c$ : 흥미로운 위치를 포함하는 격자칸들의 좌표를 저장한 크기  $n$  인 2개의 배열. 각  $i$  에 대해  $i$  번째 흥미로운 위치는 격자칸  $(r[i], c[i])$  에 있다 ( $0 \leq i \leq n - 1$ ).
  - 이 함수는 한 번 이상 사진에 찍히는 격자칸의 최소 개수를 리턴해야 한다. 물론, 사진들이 모든 흥미로운 위치를 포함해야 한다.

사용하는 언어 별로 제공되는 `template`를 참고하여 구현의 디테일을 확인하라.

## Examples

### Example 1

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

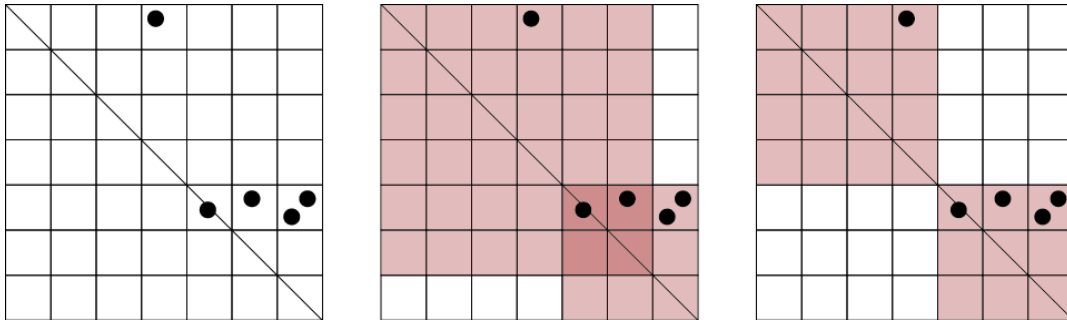
이 예에서는  $7 \times 7$  격자칸에 5 개의 흥미로운 위치가 있다. 흥미로운 위치들은 다음 4 개의 격자칸에 포함된다:  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$ ,  $(4, 6)$ . 고해상도 사진은 최대 2 개를 찍을 수 있다.

모든 5 개의 흥미로운 위치를 찍는 방법들 중 하나는 다음의 두 개의 사진을 찍는 것이다: 하나는 격자칸  $(0, 0)$  과  $(5, 5)$  를 포함하는  $6 \times 6$  격자칸의 사진이며, 다른 하나는 격자칸  $(4, 4)$  와  $(6, 6)$  을 포함하는  $3 \times 3$  격자칸의 사진이다. 이 두 개의 사진을 찍으면 인공위성은 41 개 격자칸의 데이터를 전송해야 하는데, 이는 최적이지 않다.

최적인 방법은 격자칸  $(0, 0)$  과  $(3, 3)$  를 포함하는  $4 \times 4$  격자칸의 사진과 격자칸  $(4, 4)$  와  $(6, 6)$  를 포함하는  $3 \times 3$  격자칸의 사진을 찍는 것이다. 이 경우 25 개 격자칸이 사진에 포함된다. 이 방법이 최적이므로 `take_photos` 함수는 25 를 리턴해야 한다.

격자칸  $(4, 6)$  은 두 개의 흥미로운 위치를 포함하지만 단 하나의 사진에만 포함되어도 충분함에 주의하라.

아래 그림에 현재의 예를 보였다. 제일 왼쪽 그림은 입력 상황을 표현한 것이다. 중간 그림은 41 개의 격자칸의 데이터를 전송하는 비최적 방법이다. 제일 오른쪽 그림이 최적 방법이다.

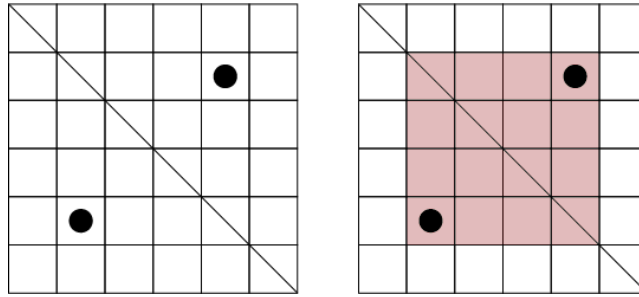


### Example 2

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

이 예에는 2 개의 흥미로운 위치가 대칭인 장소, 즉, 격자칸  $(1, 4)$  와  $(4, 1)$  에 존재한다. 문제에서 사진을 찍을 수 있는 조건 때문에, 흥미로운 위치 중 하나를 포함하는 사진은 다른 흥미로운 위치도 반드시 포함한다. 따라서 1 개의 사진을 찍는 것으로 충분하다.

다음 그림에 현재 예와 그 최적 방법을 보였다. 최적 방법은 16 개의 격자칸을 포함하는 하나의 사진을 찍는다.



## Subtasks

모든 Subtask에서,  $1 \leq k \leq n$ .

1. (4 points)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 points)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$  이고,  $0 \leq i \leq n - 1$  인 모든  $i$  에 대해  $r_i = c_i$  임,
3. (9 points)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 points)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 points)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 points)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

## Sample grader

Sample grader는 다음의 형식으로 입력을 읽어들이는다:

- line 1: 정수  $n$ ,  $m$ ,  $k$ ,
- line  $2 + i$  ( $0 \leq i \leq n - 1$ ): 정수  $r_i$  와  $c_i$ .