



## Người ngoài hành tinh (Aliens)

Vệ tinh của chúng ta vừa phát hiện ra một nền văn minh trên một hành tinh xa xôi. Chúng ta đã thu nhận được một bức ảnh có độ phân giải thấp của một vùng hình vuông của hành tinh. Bức ảnh cho thấy nhiều dấu hiệu của cuộc sống văn minh. Các chuyên gia của chúng ta đã xác định  $n$  điểm quan tâm trong bức ảnh. Các điểm được đánh số từ  $0$  đến  $n - 1$ . Bây giờ chúng ta muốn chụp các ảnh có độ phân giải cao chứa tất cả  $n$  điểm này.

Vệ tinh đã chia vùng của bức ảnh có độ phân giải thấp thành một lưới ô vuông đơn vị kích thước  $m \times m$ . Cả hàng và cột của lưới được đánh số liên tiếp từ  $0$  đến  $m - 1$  (tương ứng bắt đầu từ trên và từ trái). Chúng ta sử dụng  $(s, t)$  để biểu thị ô trong hàng  $s$  và cột  $t$ . Điểm thứ  $i$  nằm trong ô  $(r_i, c_i)$ . Mỗi ô có thể chứa một số lượng tùy ý những điểm này.

Vệ tinh của chúng ta có một quỹ đạo ổn định, là di chuyển trực tiếp trên đường chéo *chính* của lưới. Đường chéo chính là đoạn thẳng nối góc trái trên và góc phải dưới của lưới. Vệ tinh có thể chụp ảnh độ phân giải cao của bất kỳ vùng nào thỏa mãn các ràng buộc sau:

- Hình dạng của vùng này là hình vuông,
- Hai góc đối diện của hình vuông đều nằm trên đường chéo chính của lưới,
- Mỗi ô của lưới nằm hoàn toàn bên trong hoặc nằm hoàn toàn bên ngoài vùng được chụp.

Vệ tinh có thể chụp nhiều nhất  $k$  bức ảnh có độ phân giải cao.

Khi thực hiện chụp ảnh xong, vệ tinh sẽ truyền bức ảnh độ phân giải cao của mỗi ô chụp được về căn cứ của chúng ta (không phân biệt ô có chứa điểm quan tâm hay không). Dữ liệu cho mỗi ô được chụp sẽ chỉ được truyền *một lần*, ngay cả khi ô được chụp ảnh nhiều lần.

Như vậy, chúng ta phải chọn nhiều nhất  $k$  vùng hình vuông để chụp ảnh, đảm bảo rằng:

- Mỗi ô có chứa ít nhất một điểm quan tâm được chụp ít nhất một lần, và
- Số lượng ô được chụp ít nhất một lần là nhỏ nhất.

Nhiệm vụ của bạn là tìm ra số lượng nhỏ nhất có thể các ô được chụp.

### Chi tiết cài đặt

Bạn cần cài đặt hàm (thủ tục) sau:

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - $n$ : số lượng điểm quan tâm,
  - $m$ : số lượng hàng (và cũng là số lượng cột) của lưới,

- $k$ : số lượng bức ảnh tối đa vệ tinh có thể chụp,
- $r$  và  $c$ : hai mảng độ dài  $n$  mô tả tọa độ các ô của lưới chứa các điểm quan tâm. Điểm quan tâm thứ  $i$  nằm ở ô  $(r[i], c[i])$ , với  $0 \leq i \leq n-1$ ,
- hàm cần trả lại số lượng nhỏ nhất có thể các ô được chụp ít nhất một lần (bảo đảm rằng các bức ảnh phải phủ tất cả các điểm quan tâm).

Hãy sử dụng các file mẫu cho trước trong phần cài đặt theo ngôn ngữ bạn lựa chọn.

## Ví dụ

### Ví dụ 1

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

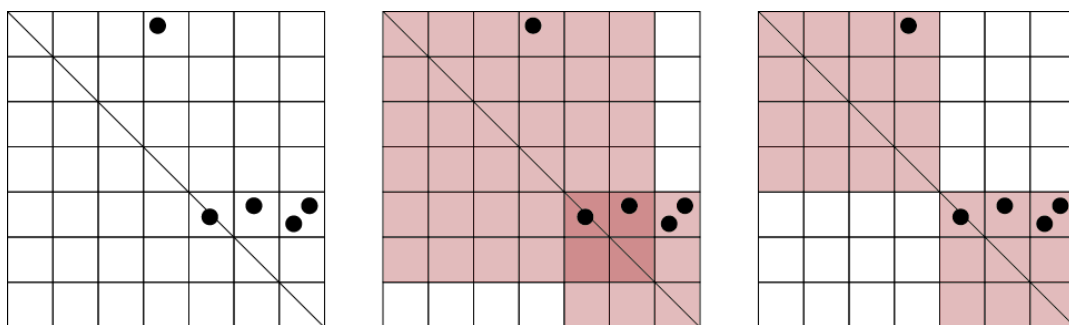
Trong ví dụ này, chúng ta có một lưới kích thước  $7 \times 7$  với 5 điểm quan tâm. Các điểm quan tâm được đặt tại bốn ô khác nhau:  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$  và  $(4, 6)$ . Bạn có thể chụp nhiều nhất 2 ảnh có độ phân giải cao.

Một cách để chụp tất cả năm điểm quan tâm là chụp hai bức ảnh: một bức ảnh kích thước  $6 \times 6$  ô chứa các ô  $(0, 0)$  và  $(5, 5)$ , và một bức ảnh kích thước  $3 \times 3$  ô chứa các ô  $(4, 4)$  và  $(6, 6)$ . Nếu vệ tinh chụp hai bức ảnh này, nó sẽ truyền dữ liệu của 41 ô. Số lượng này không phải là tối ưu.

Phương án tối ưu chụp một bức ảnh kích thước  $4 \times 4$  ô chứa các ô  $(0, 0)$  và  $(3, 3)$  và bức ảnh khác kích thước  $3 \times 3$  ô chứa các ô  $(4, 4)$  và  $(6, 6)$ . Kết quả là chỉ 25 ô được chụp, đây là tối ưu, do đó `take_photos` phải trả về 25.

Chú ý rằng chụp ô  $(4, 6)$  một lần là đủ, mặc dù nó có chứa hai điểm quan tâm.

Ví dụ này được hiển thị trong hình dưới đây. Hình bên trái sẽ hiển thị lưới tương ứng với ví dụ này. Hình ở giữa hiển thị phương án gần tối ưu trong đó 41 ô được chụp. Hình bên phải hiển thị phương án tối ưu.

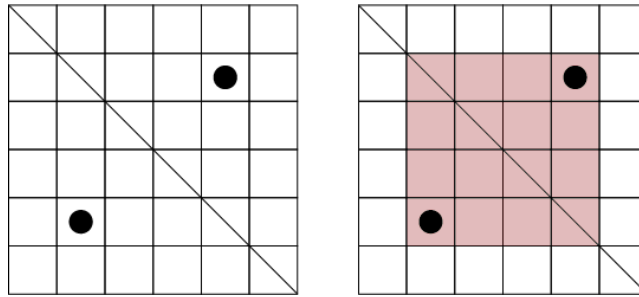


### Example 2

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

Ở đây chúng ta có 2 điểm quan tâm nằm đối xứng: trong ô  $(1, 4)$  và  $(4, 1)$ . Bất kỳ bức ảnh hợp lệ chứa một trong hai đều chứa điểm còn lại. Do đó, chỉ chụp một bức ảnh là đủ.

Hình dưới đây hiển thị ví dụ này và phương án tối ưu của nó. Trong phương án này, vệ tinh chụp một bức ảnh duy nhất gồm 16 ô.



### Subtasks

Với tất cả các subtask,  $1 \leq k \leq n$ .

1. (4 điểm)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 điểm)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , với mọi  $i$  sao cho  $0 \leq i \leq n - 1$ ,  
 $r_i = c_i$ ,
3. (9 điểm)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 điểm)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 điểm)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 điểm)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

### Sample grader

Chương trình grader mẫu đọc dữ liệu đầu vào theo định dạng sau:

- dòng 1: số nguyên  $n$ ,  $m$  và  $k$ ,
- dòng  $2 + i$  ( $0 \leq i \leq n - 1$ ): số nguyên  $r_i$  và  $c_i$ .