

## Aliens

Notre satellite vient tout juste de découvrir une civilisation extraterrestre sur une planète lointaine. Nous avons déjà obtenu une photo basse résolution d'une zone carrée de la planète. La photo montre un grand nombre de signes de vie intelligente. Nos experts ont identifié  $n$  points d'intérêt sur la photo. Les points sont numérotés de  $0$  à  $n - 1$ . Nous voulons désormais prendre des photos de haute résolution qui contiennent tous ces  $n$  points.

En interne, le satellite a découpé la surface de la photo de basse résolution en une grille de taille  $m$  par  $m$ , de cellules unitaires carrées. Les lignes et les colonnes de la grille sont numérotées consécutivement de  $0$  à  $m - 1$  (depuis le haut et la gauche, respectivement). On note  $(s, t)$  la cellule à la ligne  $s$  et à la colonne  $t$ . Le point d'intérêt numéro  $i$  est situé dans la cellule  $(r_i, c_i)$ . Chaque cellule peut contenir un nombre arbitraire de ces points.

Notre satellite est sur une orbite stable qui passe directement par-dessus la diagonale *principale* de la grille. La diagonale principale est le segment de droite qui relie les coins haut-gauche et bas-droite de la grille. Le satellite peut prendre une photo haute résolution de n'importe quelle zone qui satisfait les contraintes suivantes :

- la forme de la zone est un carré,
- deux coins opposés du carré se situent sur la diagonale principale de la grille,
- chaque cellule de la grille est soit complètement à l'intérieur de la zone photographiée, soit complètement à l'extérieur.

Le satellite n'est capable de prendre qu'au plus  $k$  photos haute résolution.

Une fois que le satellite a fini de prendre les photos, il transmettra la photo haute résolution de chaque cellule photographiée à notre base terrestre (et ce, indépendamment du fait que cette cellule contienne ou non des points d'intérêt). Les données de chaque cellule photographiée seront transmises une *unique* fois, même si la cellule a été photographiée plusieurs fois.

Ainsi, nous devons choisir au plus  $k$  zones carrées qui seront photographiées, assurant que :

- chaque cellule contenant au moins un point d'intérêt a été photographiée au moins une fois, et
- le nombre de cellules qui ont été photographiées au moins une fois est minimisé.

Votre tâche est de trouver le plus petit nombre total possible de cellules photographiées.

### Détails de l'implémentation

Vous devez implémenter la fonction (méthode) suivante :

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n` : le nombre de points d'intérêt,
  - `m` : le nombre de lignes (et aussi de colonnes) de la grille,
  - `k` : le nombre maximum de photos que le satellite peut prendre,
  - `r` et `c` : deux tableaux de taille `n` décrivant les coordonnées des cellules de la grille qui contiennent des points d'intérêt. Pour  $0 \leq i \leq n - 1$ , le  $i$ -ème point d'intérêt est situé dans la cellule  $(r[i], c[i])$ ,
  - la fonction doit retourner le plus petit nombre total possible de cellules qui sont photographiées au moins une fois (étant donné que les photos doivent couvrir tous les points d'intérêt).

Veillez utiliser le modèle des fichiers fourni pour obtenir des détails de l'implémentation dans votre langage de programmation.

## Exemples

### Exemple 1

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

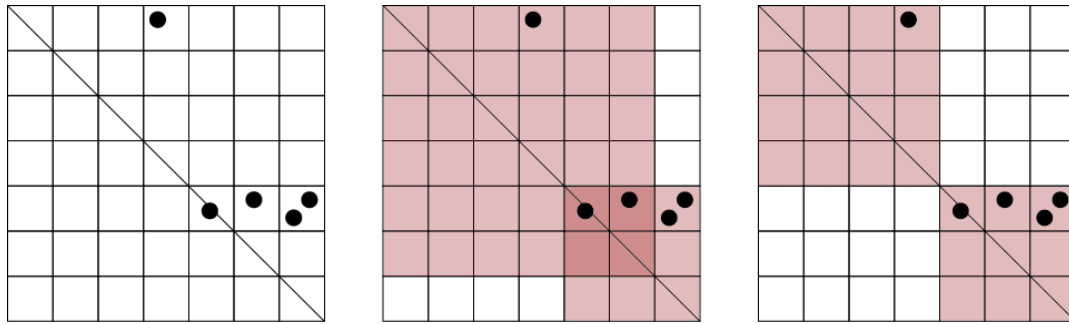
Dans cet exemple nous avons une grille  $7 \times 7$  avec 5 points d'intérêt. Les points d'intérêt sont situés dans quatre cellules différentes :  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$  et  $(4, 6)$ . Vous pouvez prendre au plus 2 photos haute résolution.

Une manière de capturer tous les 5 points d'intérêts est de prendre deux photos : une photo du carré de taille  $6 \times 6$  contenant les cellules  $(0, 0)$  et  $(5, 5)$ , et une photo du carré de taille  $3 \times 3$  contenant les cellules  $(4, 4)$  et  $(6, 6)$ . Si le satellite prend ces deux photos, il transmettra les données de 41 cellules. Cette quantité n'est pas optimale.

La solution optimale utilise une photo pour capturer le carré  $4 \times 4$  contenant les cellules  $(0, 0)$  et  $(3, 3)$  et une autre photo pour capturer le carré  $3 \times 3$  contenant les cellules  $(4, 4)$  et  $(6, 6)$ . Ceci permet d'obtenir seulement 25 cellules photographiées, ce qui est optimal, donc `take_photos` doit retourner 25.

Veillez noter qu'il est suffisant de photographier la cellule  $(4, 6)$  une seule fois, bien qu'elle contienne deux points d'intérêt.

Cet exemple est présenté sur les figures ci-après. La figure de gauche montre la grille qui correspond à cet exemple. La figure du milieu montre une solution sous-optimale dans laquelle 41 cellules ont été photographiées. La figure de droite montre la solution optimale.

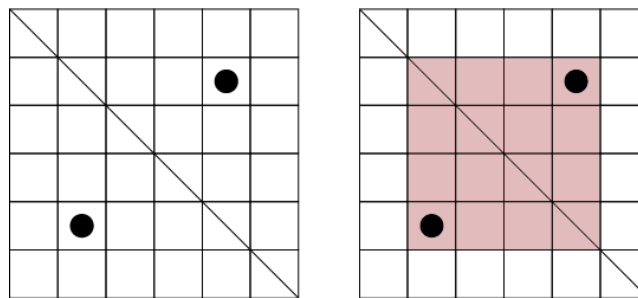


## Exemple 2

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

Dans cet exemple nous avons 2 points d'intérêt situés symétriquement : dans les cellules (1,4) et (4,1). N'importe quelle photo valide qui contient l'un d'entre eux contient l'autre aussi. Par conséquent, il est suffisant d'utiliser une seule photo.

La figure ci-dessous présente cet exemple et sa solution optimale. Dans cette solution, le satellite capture une seule photo de 16 cellules.



## Sous-tâches

Dans toutes les sous-tâches,  $1 \leq k \leq n$ .

1. (4 points)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 points)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , pour tout  $i$  tel que  $0 \leq i \leq n - 1$ ,  
 $r_i = c_i$ ,
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)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

## Evaluateur fourni (grader)

L'évaluateur fourni lit l'entrée selon le format suivant :

- ligne 1 : les entiers  $n$ ,  $m$  et  $k$ ,
- ligne  $2 + i$  ( $0 \leq i \leq n - 1$ ) : les entiers  $r_i$  et  $c_i$ .