Aliens

Unser Satellit hat eben eine fremde Zivilisation auf einem fernen Planeten entdeckt. Wir haben schon ein Foto in niedriger Auflösung von einer quadratischen Fläche des Planeten erhalten. Das Foto zeigt viele Hinweise auf intelligentes Leben. Unsere Experten haben n Sehenswürdigkeiten auf dem Foto entdeckt. Die Sehenswürdigkeiten sind von 0 bis $n-1$ durchgehend nummeriert. Wir wollen hochauflösende Fotos machen, die all diese n Sehenswürdigkeiten enthalten.

Intern hat der Satellit die Fläche des niedrigauflösenden Fotos in ein Raster von $m \times m$ quadratischen Zellen eingeteilt. Sowohl die Zeilen als auch die Spalten sind durchgehend von 0 bis $m-1$ nummeriert (von oben beziehungsweise von links). Die Notation (s,t) bezeichnet die Zelle aus Zeile s und Spalte t. Die Sehenswürdigkeit Nummer i befindet sich in der Zelle (r_i, c_i). In jeder Zelle kann sich eine beliebige Anzahl Sehenswürdigkeiten befinden.

Unser Satellit befindet sich auf einer stabilen Umlaufbahn, die über die *Hauptdiagonale* des Rasters verläuft. Die Hauptdiagonale entspricht der Linie, welche die Eckzellen oben links und unten rechts verbindet. Der Satellit kann ein hochauflösendes Foto einer beliebigen Fläche machen, die folgende Bedingungen erfüllt:

- die Form der Fläche ist ein Quadrat,
- die Eckzellen oben links und unten rechts des Quadrats liegen beide auf der Hauptdiagonalen des Rasters,
- jede Zelle des Rasters befindet sich entweder ganz innerhalb oder ganz ausserhalb der fotografierten Fläche.

Nachdem der Satellit das Fotografieren abgeschlossen hat, schickt er ein einzelnes Foto von jeder hochauflösend fotografierten Zelle zu unserer Basisstation (egal, ob diese Zelle Sehenswürdigkeiten enthält). Die Daten jeder hochauflösend fotografierten Zelle werden nur *einmal* übertragen, sogar, wenn die Zelle mehrmals fotografiert wurde.

Der Satellit kann höchstens k hochauflösende Fotos nehmen.

Somit müssen wir höchstens k quadratische Flächen auswählen und hochauflösend fotografieren. Dabei müssen wir sicherstellen, dass:

- jede Zelle, die mindestens eine Sehenswürdigkeit enthält, mindestens einmal hochauflösend fotografiert wird und
- die Anzahl der Zellen, die mindestens einmal hochauflösend fotografiert werden, minimiert ist.

Deine Aufgabe besteht darin, die kleinstmögliche Gesamtzahl der hochauflösend

fotografierten Zellen zu ermitteln.

Implementierungsdetails

Du sollst folgende Funktion (Methode) implementieren:

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: die Anzahl der Sehenswürdigkeiten,
  - `m`: die Anzahl der Zeilen (und auch Spalten) des Rasters,
  - `k`: die maximale Anzahl hochauflösender Fotos, die der Satellit machen kann,
  - `r` und `c`: zwei Arrays der Länge n, welche die Koordinaten der Zellen angeben, die Sehenswürdigkeiten enthalten. Für $0 \leq i \leq \texttt{n} - 1$ ist die i-te Sehenswürdigkeit in der Zelle (`r[i]`, `c[i]`) enthalten.
  - Die Funktion sollte die kleinstmögliche Gesamtzahl Zellen, die wenigstens einmal hochauflösend fotografiert wurden, ausgeben (ausgehend davon, dass das Foto alle Sehenswürdigkeiten enthält).

Benutze bitte die beigefügten Templatefiles für Implementierungsdetails in deiner Programmiersprache.

Beispiele

Beispiel 1

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

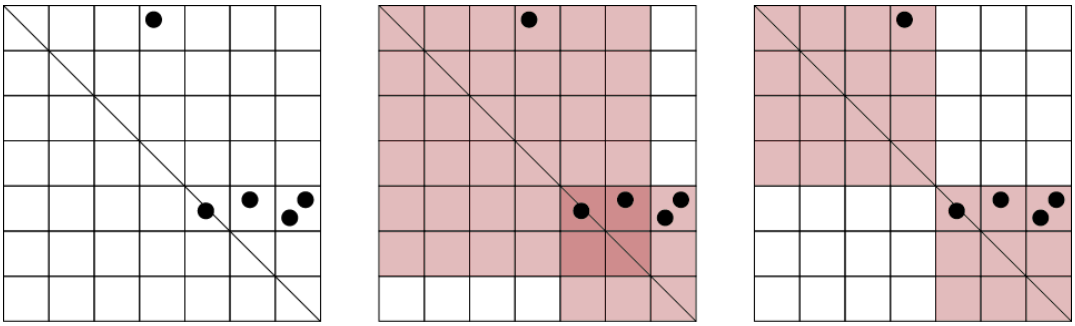
In diesem Beispiel haben wir ein 7×7 Raster mit 5 Sehenswürdigkeiten. Die Sehenswürdigkeiten befinden sich in vier verschiedenen Zellen: $(0, 3)$, $(4, 4)$, $(4, 5)$ und $(4, 6)$. Du kannst maximal 2 hochauflösende Fotos machen.

Eine Möglichkeit, alle fünf Sehenswürdigkeiten zu erfassen, besteht darin, zwei hochauflösende Fotos zu machen: eines im Format 6×6 mit den Zellen $(0, 0)$ oben links und $(5, 5)$ unten rechts, und das andere im Format 3×3 mit den Zellen $(4, 4)$ oben links und $(6, 6)$ unten rechts. Falls wir diese beiden Fotos machen, wird der Satellit die Fotos von 41 Zellen übertragen. Diese Anzahl ist nicht optimal.

Die optimale Lösung benutzt ein erstes Foto mit dem 4×4 Quadrat der Zellen $(0, 0)$ bis $(3, 3)$ und ein zweites Foto mit dem 3×3 Quadrat mit den Zellen $(4, 4)$ bis $(6, 6)$. Daraus resultieren nur 25 zu übertragende fotografierte Zellen, was optimal ist. Entsprechend sollte die Funktion `take_photos` 25 ausgeben.

Es sei bemerkt, dass es genügt, die Zelle $(4, 6)$ einmal zu fotografieren, obwohl sie zwei Sehenswürdigkeiten enthält.

Dieses Beispiel wird unten dargestellt. Die linksstehende Darstellung entspricht diesem Beispiel. Die mittlere Darstellung zeigt die nicht optimale Lösung mit 41 fotografierten Zellen. Die rechtsstehende Darstellung zeigt die optimale Lösung.

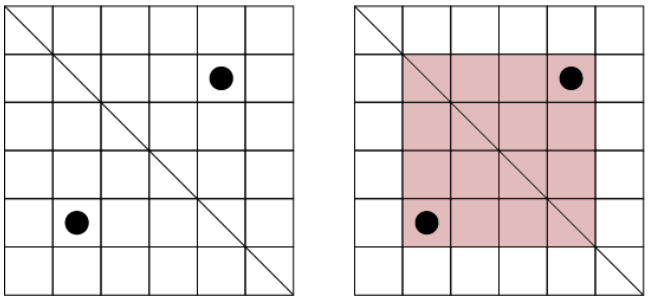


Beispiel 2

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

Hier haben wir 2 Sehenswürdigkeiten, die sich symmetrisch zueinander befinden: in den Zellen $(1, 4)$ und $(4, 1)$. Jedes zugelassene Foto, das die eine Sehenswürdigkeit enthält, enthält auch die andere. Deshalb genügt es, ein Foto zu machen.

Die Darstellung unten zeigt dieses Beispiel und dessen optimale Lösung. In dieser Lösung erfasst der Satellit ein einziges hochauflösendes Foto mit 16 Zellen.



Subtasks

Für alle Subtasks, $1 \leq k \leq n$.

1. (4 Punkte) $1 \leq n \leq 50$, $1 \leq m \leq 100$, $k = n$,
2. (12 Punkte) $1 \leq n \leq 500$, $1 \leq m \leq 1000$, für jedes i mit $0 \leq i \leq n - 1$, $r_i = c_i$,
3. (9 Punkte) $1 \leq n \leq 500$, $1 \leq m \leq 1000$,
4. (16 Punkte) $1 \leq n \leq 4000$, $1 \leq m \leq 1\,000\,000$,
5. (19 Punkte) $1 \leq n \leq 50\,000$, $1 \leq k \leq 100$, $1 \leq m \leq 1\,000\,000$,
6. (40 Punkte) $1 \leq n \leq 100\,000$, $1 \leq m \leq 1\,000\,000$.

Beispielgrader

Der Beispielgrader liest die Eingabe in folgendem Format:

- Zeile 1: Integers n, m und k,
- Zeile 2 + i ($0 \leq i \leq n - 1$): Integers r_i und c_i.