Внеземная цивилизация

Спутник недавно обнаружил внеземную цивилизацию на далекой планете. Со спутника получена фотография низкого качества квадратной области на поверхности этой планеты. Фотография показывает много признаков разумной жизни. Эксперты определили n интересных точек на этой фотографии. Точки пронумерованы от 0 до $n-1$. Теперь требуется получить новые фотографии в более хорошем качестве, содержащие все n интересных точек.

Полученная со спутника фотография низкого качества разделяется на сетку, состоящую из m на m единичных клеток. Строки и столбцы сетки последовательно пронумерованы от 0 до $m-1$ (сверху вниз и слева направо, соответственно). Для клетки в ряду s и в столбце t используется обозначение (s,t). Точка с номером i содержится внутри клетки (r_i, c_i). Каждая клетка может содержать любое количество интересных точек.

Спутник находится на стабильной орбите, проходящей ровно над *главной* диагональю сетки. Главной диагональю называется диагональ, которая соединяет левый верхний и правый нижний углы сетки. Спутник может сделать фотографию в хорошем качестве любой области, удовлетворяющей следующим условиям:

- область должна иметь форму квадрата,
- два противоположных угла квадрата лежат на главной диагонали сетки,
- каждая клетка сетки находится целиком внутри или целиком снаружи фотографируемой области.

Спутник может сделать не более k фотографий в хорошем качестве.

После того как спутник сделал все фотографии, он передает фотографии каждой сфотографированной клетки в хорошем качестве, не зависимо от того, содержит ли клетка интересные точки. Каждая сфотографированная клетка будет передана *ровно один раз*, даже если она была сфотографирована несколько раз.

Таким образом, необходимо выбрать не более k квадратных областей, которые будут сфотографированы, добившись, чтобы:

- каждая интересная точка была сфотографирована хотя бы один раз, и
- количество клеток, которые будут сфотографированы хотя бы один раз, было минимально.

Требуется найти минимальное общее количество клеток которые были сфотографированы хотя бы один раз.

Детали реализации

Требуется реализовать следующую функцию

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: количество интересных точек,
  - `m`: количество строк (а также столбцов) в сетке,
  - `k`: наибольшее количество фотографий, которые спутник может сделать,
  - `r` и `c`: два массива длины n описывающие координаты клетки, содержащей интересные точки. Для $0 \leq i \leq \texttt{n} - 1$, i-ая интересная точка содержится в клетке (`r[i]`, `c[i]`),
  - функция должна возвращать минимальное количество клеток, которые будут сфотографированы хотя бы один раз, при условии, что фотографии должны покрыть все интересные точки.

Используйте приведенный шаблон решения для получения деталей реализации на выбранном вами языке программирования.

Примеры

Пример 1

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

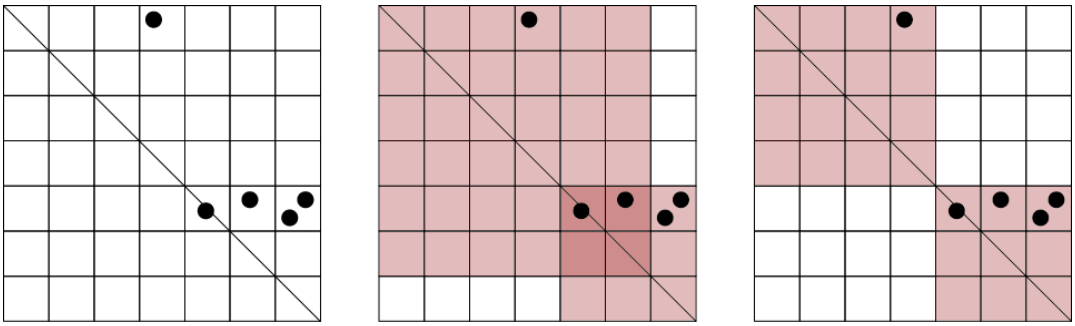
В этом примере есть сетка 7×7 с 5 интересными точками. Интересные точки расположены в 4 различных клетках: $(0, 3)$, $(4, 4)$, $(4, 5)$ и $(4, 6)$. Можно сделать не более 2 фотографий в хорошем качестве.

Например, один вариант сфотографировать все интересные точки двумя фотографиями: сделать одну фотографию в форме квадрата 6×6 с противоположными углами $(0, 0)$ и $(5, 5)$, и вторую с противоположными углами $(4, 4)$ и $(6, 6)$. Если спутник сделает эти две фотографии, спутнику будет необходимо послать фотографии в хорошем качестве для 41 клетки. Это количество не оптимально.

Оптимальное решение -- сделать одну из фотографий в форме квадрата 4×4 с противоположными углами $(0, 0)$ и $(3, 3)$ и другую фотографию в форме квадрата 3×3 с противоположными углами $(4, 4)$ и $(6, 6)$. Для этого потребуется послать только 25 фотографий клеток, что является оптимальным, поэтому `take_photos` должна вернуть 25.

Обратите внимание, что достаточно сфотографировать клетку $(4, 6)$ один раз, несмотря на то, что она содержит две интересных точки.

Этот пример показан на рисунке ниже. Самый левый рисунок показывает сетку, соответствующую примеру. Средняя картинка показывает не оптимальное решение, в котором сфотографирована 41 клетка. На самой правой картинке показано оптимальное решение.

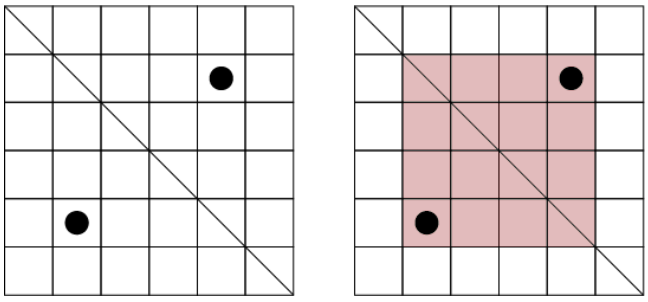


Пример 2

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

В этом примере есть 2 интересных точки расположенных симметрично: в клетках $(1, 4)$ и $(4, 1)$. Любая фотография, содержащая одну из этих точек, содержит и другую. Таким образом, достаточно сделать одну фотографию.

На рисунке ниже изображен пример и его оптимальное решение. В этом решении спутник делает одну фотографию, на которую попадает 16 клеток.



Система оценивания

Во всех подзадачах, $1 \le k \le n$.

1. (4 балла) $1 \le n \le 50$, $1 \le m \le 100$, $k = n$,
2. (12 баллов) $1 \le n \le 500$, $1 \le m \le 1000$, для всех i, таких что $0 \le i \le n - 1$, $r_i = c_i$,
3. (9 баллов) $1 \le n \le 500$, $1 \le m \le 1000$,
4. (16 баллов) $1 \le n \le 4000$, $1 \le m \le 1\,000\,000$,
5. (19 баллов) $1 \le n \le 50\,000$, $1 \le k \le 100$, $1 \le m \le 1\,000\,000$,
6. (40 баллов) $1 \le n \le 100\,000$, $1 \le m \le 1\,000\,000$.

Пример проверяющего модуля

Пример проверяющего модуля читает входные данные в следующем формате:

- Строка 1: целые числа n, m и k.
- Строка 2 + i ($0 \le i \le n - 1$): целые числа r_i и c_i.