Aliens

Току-що е открита планета с извънземна цивилизация. Разполагаме с фотография на квадратен участък от планетата. Фотографията е с ниска разделителна способност, но показва много признаци на интелигентен живот. Експертите са намерили n точки, които са забележителни, защото представляват голям интерес за нас. Забележителните точки са номерирани от 0 до $n-1$. Искаме да направим фотография с голяма разделителна способност, която да съдържа тези точки.

Сателитът, който заснема, разделя площта на фотографията с ниска разделителна способност на мрежа от m на m единични квадратни клетки. И редовете и стълбовете на мрежата са номерирани с последователни числа от 0 до $m-1$ (съответно от горе и от ляво). Означаваме с (i, j) клетката от ред i и стълб j. Всяка забележителна точка се намира вътре в една от клетките. Всяка клетка може да съдържа и повече от една забележителна точка.

Сателитът се намира на стабилна орбита, която минава над *главния* диагонал на мрежата. Главният диагонал минава през клетките (i, i) за всички $0 \leq i \leq m-1$. Сателитът може да направи фотография с висока разделителна способност на всяка област, която удовлетворява следните ограничения:

- областта има квадратна форма
- единият диагонал на квадрата изцяло се съдържа в главния диагонал в главния диагонал на мрежата
- всяка клетка от мрежата е изцяло вътре или е изцяло извън областта

Сателитът може да направи най-много k фотографии с висока разделителна способност.

След като сателитът направи фотографиите, той ще изпрати на Земята всички фотографирани клетки, независимо дали съдържат или не съдържат забележителна точка. Всяка фотографирана клетка се изпраща точно веднъж, даже и да участва в няколко фотографии.

Така трябва да изберем най-много k квадратни области, които да бъдат фотографирани, така че

- всяка клетка, която съдържа поне една забележителна точка е фотографирана поне веднъж.
- броят на клетките, които са фотографирани поне веднъж е минимален.

Вашата задача е да намерите този брой.

Детайли по реализацията

Трябва да реализирате следната функция

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: брой на забележителните точки,
  - `m`: брой на редовете (и на стълбовете) на мрежата,
  - `k`: максимален брой на фотографиите, които може да направи сателитът,
  - `r` и `c`: два масива с дължина n, задаващи координатите на клетките от мрежата, които съдържат забележителни точки. За $0 \le i \le \texttt{n} - 1$, i-тата забележителна точка се намира в клетка (`r[i]`, `c[i]`),
  - функцията трябва да върне най-малкия възможен брой на клетките, които са фотографирани поне веднъж (като фотографирането трябва да покрие всички забележителни точки).

Примери

Пример 1

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

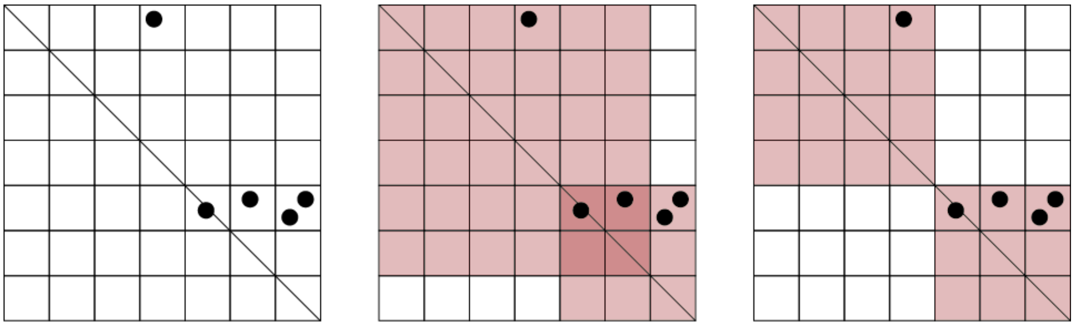
В този пример има мрежа 7×7 с 5 забележителни точки. Забележителните точки се намират в 4 различни клетки: $(0,3)$, $(4,4)$, $(4,5)$ и $(4,6)$. Трябва да се направят най-много 2 фотографии с висока разделителна способност.

Един начин да се направи това е чрез две фотографии: едната с клетки $(0,0)$ и $(5,5)$ в срещуположните си върхове, а другата с клетки $(4,4)$ и $(6,6)$ в срещуположните си върхове. Ако направим тези фотографии, сателитът ще изпрати фотографии на 41 клетки. Тази стойност не е оптимална.

Оптималното решение има една фотография за квадрат 4×4, съдържащ клетките $(0,0)$ и $(3,3)$ и още една фотография за квдрат 3×3, съдържащ клетките $(4,4)$ и $(6,6)$. Така общо са фотографирани 25 клетки, което е оптимално. Функцията `take_photos` трябва да върне 25.

Забележете, че е достатъчно да се фотографира клетката $(4,6)$ веднъж, въпреки че тя съдържа 2 забележителни точки.

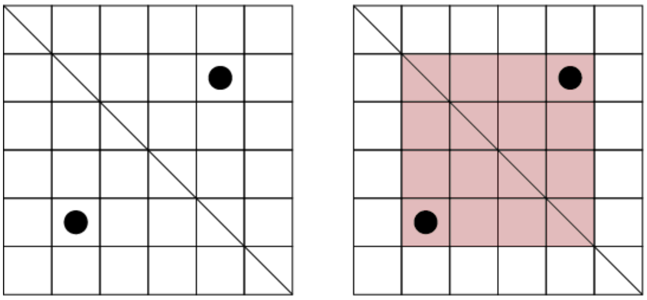
По-долу са изобразени примерът и двата начина за правени на фотографии. Фигурата отдясно показва оптималното решение.



Пример 2

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

Има 2 забележителни точки, разположени симетрично: в клетките $(1, 4)$ и $(4, 1)$. Всяка валидна фотография, която съдържа една от тези клетки, съдържа и другата. Следователно, достатъчно е да се направи само една фотография. Оптималното решение (изобразено по-долу) има фотографии за 16 клетки.



Подзадачи

За всички подзадачи, $1 \leq k \leq n$.

1. (4 points) $1 \leq n \leq 50$, $1 \leq m \leq 100$, $k = n$.
2. (12 points) $1 \leq n \leq 500$, $1 \leq m \leq 1000$, за всички i такива, че $0 \leq i \leq n-1$, $r_i = c_i$.
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) $1 \leq n \leq 100\,000$, $1 \leq m \leq 1\,000\,000$.

Примерен грейдер

Примерният грейдер чете вход в следния формат:

- ред 1: цели числа n, m and k,
- ред 2 + i ($0 \leq i \leq n-1$): цели числа r_i and c_i.