



宇宙人 (Aliens)

とある人工衛星が、遠い惑星にある宇宙人の文明を発見した。解像度は低いですが、その星の一部を写した写真がすでに得られている。写真には、宇宙人が高度な文明を持っていることを示すいくつかの痕跡が写っていた。専門家は、写真のなかから n 箇所の特徴的な場所を見つけた。これらはそれぞれ 0 から $n-1$ までの番号が振られている。それぞれの特徴的な場所について、より解像度の高い写真を撮影したい。

低解像度の写真として撮影された領域は、正方形のかたちをしていて、人工衛星の内部システムにおいて、 $m \times m$ の正方形のセルからなるグリッド領域として表現されている。すべての行は上から下へ順番に 0 から $m-1$ までの番号が振られ、すべての列も左から右へ順番に 0 から $m-1$ までの番号が振られている。 s 行 t 列にあるセルを (s, t) と表記することにする。 i 番目の特徴的な場所はセル (r_i, c_i) に含まれている。同じセルの中に複数の特徴的な場所が含まれていることもある。

人工衛星は、グリッド領域のメインの対角線の上を通るような安定軌道にそって周回している。メインの対角線とは、グリッド領域の左上から右下を結ぶ対角線のことである。人工衛星は、以下の条件を満たす任意の領域について高解像度の写真を撮影をすることができる。

- 撮影される領域は正方形である。
- 撮影される領域における 2 つの向かいあう頂点が、メインの対角線の上に乗っている。
- グリッド領域におけるそれぞれのセルは、撮影される領域の中に完全に入っているか、または撮影される領域の外に完全に外出ているかのどちらかである。

人工衛星は、最大で k 枚の高解像度の写真を撮影することができる。

人工衛星は、写真撮影を終えたあと、写真に写っていたすべてのセルのデータを地上基地へと送信する。そのセルに特徴的な場所が写っていない場合でもデータを送信する。撮影されたセルのデータは、複数の写真に写っていた場合でも、ちょうど 1 回だけ送信される。

したがって、以下の条件を満たすように最大 k 枚の写真を撮影する必要がある。

- 特徴的な場所を含むようなすべてのセルは、少なくとも 1 回以上撮影されている。
- 1 回以上撮影されたセルの数が最小化されている。

あなたの課題は、 1 回以上撮影されるセルの数を最小化したときに、撮影されるセルの数を求めることである。

実装の詳細 (Implementation details)

あなたは以下の関数を実装しなければならない。

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
 - n : 特徴的な場所の数。
 - m : グリッド領域における行 (および列) の数。
 - k : 人工衛星が撮影できる写真の枚数の上限。

- r および c : 特徴的な場所を含むセルの位置を表現する、長さ n の配列. i 番目 ($0 \leq i \leq n-1$) のセルの場所は $(r[i], c[i])$ として表される.
- この関数は、すべての特徴的な場所を含むように撮影するときの、撮影されるセルの数の最小値を返さなければならない.

実装の詳細については、各言語のテンプレートファイルを参照せよ.

例 (Examples)

例 1 (Example 1)

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

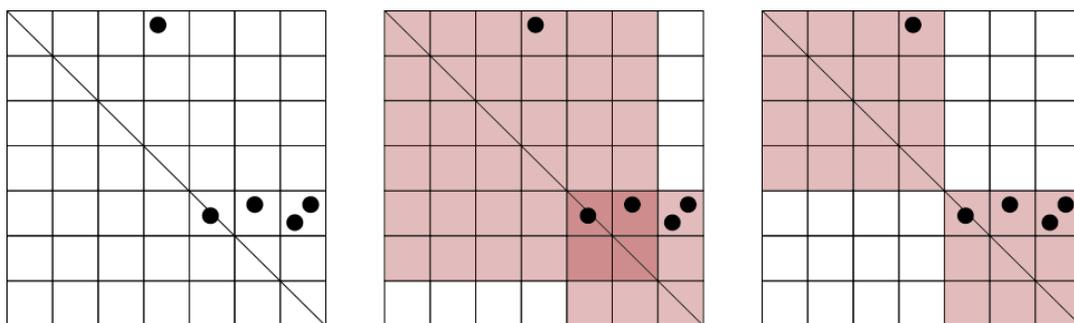
この例では、 7×7 のグリッド領域の上に 5 箇所の特徴的な場所がある. 特徴的な場所は $(0,3)$, $(4,4)$, $(4,5)$, $(4,6)$ の 4 つの相異なるセルに含まれている. 最大で 2 枚までの高解像度の写真を撮影することができる.

5 箇所の特徴的な場所すべてを撮影するには、例えば、 $(0,0)$ および $(5,5)$ を含む 6×6 の写真を撮影し、 $(4,4)$ および $(6,6)$ を含む 3×3 の写真を撮影すれば良い. このとき、人工衛星は合計で 41 個のセルのデータを地上へ送信するが、これは最適ではない.

$(0,0)$ および $(3,3)$ を含む 4×4 の写真を撮影し、 $(4,4)$ および $(6,6)$ を含む 3×3 の写真を撮影すると、最適となる. このとき、人工衛星は合計で 25 個のセルのデータを地上へ送信する. よって `take_photos` は 25 を返さなければならない.

$(4,6)$ は 2 箇所の特徴的な場所を含んでいるが、このセルの撮影は 1 回でよいことに注意せよ.

この例を以下に図示する. 一番左の図はこの例におけるグリッド領域を示している. 中央の図は 41 個のセルを撮影するような最適でない撮影方法を示している. 一番右の図は最適解における撮影方法を示している.



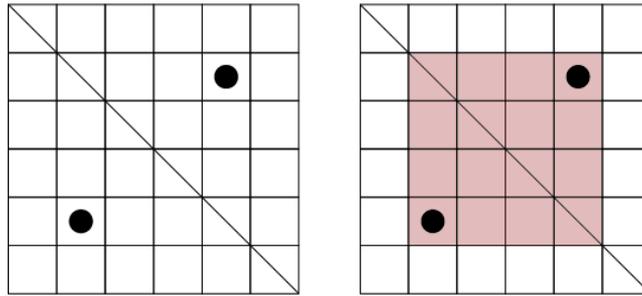
例 2 (Example 2)

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

この例では、 $(1,4)$ および $(4,1)$ という、メインの対角線に対して線対称な位置に 2 箇所の特徴的な場所が存在する. このうちの片方のセルを含むように写真を撮影すると、かならずもう片方のセルも写真に含まれる. よって、写真を 1 枚撮影するだけで十分である.

この例におけるグリッド領域および、最適解における撮影方法を以下に図示する. 最適解にお

いては、16 個のセルを含む 1 枚の写真を撮影するだけで良い。



小課題 (Subtasks)

すべての小課題で、 $1 \leq k \leq n$ を満たす。

1. (4 点) $1 \leq n \leq 50$ かつ $1 \leq m \leq 100$ かつ $k = n$ を満たす。
2. (12 点) すべての i ($0 \leq i \leq n-1$) に対して $r_i = c_i$ を満たし、 $1 \leq n \leq 500$ かつ $1 \leq m \leq 1000$ を満たす。
3. (9 点) $1 \leq n \leq 500$ かつ $1 \leq m \leq 1000$ を満たす。
4. (16 点) $1 \leq n \leq 4000$ かつ $1 \leq m \leq 1000000$ を満たす。
5. (19 点) $1 \leq n \leq 50000$ かつ $1 \leq k \leq 100$ かつ $1 \leq m \leq 1000000$ を満たす。
6. (40 点) $1 \leq n \leq 100000$ かつ $1 \leq m \leq 1000000$ を満たす。

採点プログラムのサンプル (Sample grader)

採点プログラムのサンプルは、以下のフォーマットで入力を読み込む。

- 1 行目: 整数 n , m , k .
- $2+i$ 行目 ($0 \leq i \leq n-1$): 整数 r_i , c_i .