Aliens

Our satellite has just discovered an alien civilization on a remote planet. We have already obtained a low-resolution photo of a square area of the planet. The photo shows many signs of intelligent life. Our experts have identified n points of interest in the photo. The points are numbered from 0 to $n-1$. We now want to take high-resolution photos that contain all of those n points.

Internally, the satellite has divided the area of the low-resolution photo into an m by m grid of unit square cells. Both rows and columns of the grid are consecutively numbered from 0 to $m-1$ (from the top and left, respectively). We use (s,t) to denote the cell in row s and column t. The point number i is located in the cell (r_i, c_i). Each cell may contain an arbitrary number of these points.

Our satellite is on a stable orbit that passes directly over the *main* diagonal of the grid. The main diagonal is the line segment that connects the top left and the bottom right corner of the grid. The satellite can take a high-resolution photo of any area that satisfies the following constraints:

- the shape of the area is a square,
- two opposite corners of the square both lie on the main diagonal of the grid,
- each cell of the grid is either completely inside or completely outside the photographed area.

The satellite is able to take at most k high-resolution photos.

Once the satellite is done taking photos, it will transmit the high-resolution photo of each photographed cell to our home base (regardless of whether that cell contains some points of interest). The data for each photographed cell will only be transmitted *once*, even if the cell was photographed several times.

Thus, we have to choose at most k square areas that will be photographed, assuring that:

- each cell containing at least one point of interest is photographed at least once, and
- the number of cells that are photographed at least once is minimized.

Your task is to find the smallest possible total number of photographed cells.

Implementation details

You should implement the following function (method):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - `n`: the number of points of interest,
  - `m`: the number of rows (and also columns) in the grid,