



Problem E. Nice Shape

Time limit: 4 seconds

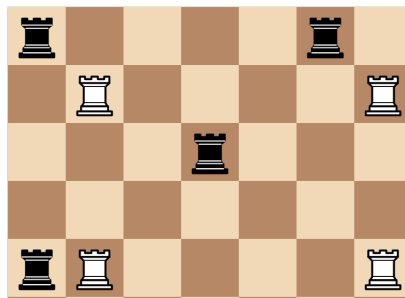
You are given n rooks on the different cells of the infinite chessboard.

The i -th of them is in the cell (r_i, c_i) .

In one move you can move any rook to any cell in the same row/column. In other words, in one move you can choose any i and then either replace r_i to any other integer or replace c_i to any other integer. You can't move a rook to the cell with some other rook.

Four different rooks a, b, c, d form a *nice shape* if you can find a rectangle such that a, b, c, d are its corners. In other words, if the set of cells $\{(r_a, c_a), (r_b, c_b), (r_c, c_c), (r_d, c_d)\}$ is equal to the set of cells $\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$ for some integers x_1, x_2, y_1, y_2 with $x_1 \neq x_2$ and $y_1 \neq y_2$.

For example, the white rooks in the following picture form a nice shape.



Your goal is to find the minimum number of moves that you can perform to get a nice shape.

In other words, you need to find the minimum number of moves that you can perform, such that after them it will be possible to find a rectangle with four rooks in its corners.

Input

The first line of input contains one integer t ($1 \leq t \leq 25\,000$): the number of test cases.

The description of t test cases follows.

The first line contains one integer n ($4 \leq n \leq 100\,000$).

The i -th of the next n lines contains two integers r_i, c_i ($1 \leq r_i, c_i \leq 10^9$)

For each pair i, j with $i \neq j$, $r_i \neq r_j$ or $c_i \neq c_j$.

The total sum of n is at most 100 000.

Output

For each test case, print one integer: the minimum number of moves you need to perform to obtain at least one nice shape among given rooks.

Scoring

Subtask	Score	Constraints
1	10	$n \leq 4$
2	10	$n \leq 50$
3	10	$n \leq 200$
4	30	$n \leq 2000$
5	40	$n \leq 10^5$

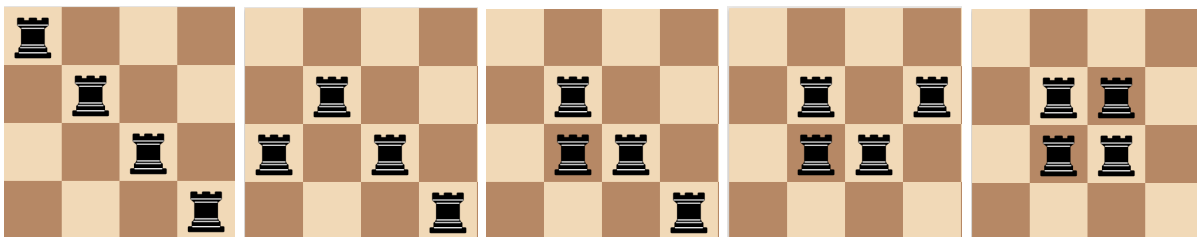


Example

standard input	standard output
5 4 4 4 1 1 2 2 3 3 4 4 4 4 1 1 4 2 2 6 3 2 2 1 1 2 3 3 3 4 3 1 5 1 1 1 2 1 3 1 4 5 5 4 1000000000 1000000000 1000000000 1 2 2 1000000000 999999999	4 2 1 3 3

Note

One of the possible optimal solutions for the first test case of the example:



One of the possible optimal solutions for the second test case of the example:

