

Greedy Pirates

Ruffus the notorious, Bitland pirate has just acquired some good from a caravan that stopped when a road broke from under them. SCORE! Now he must do the grueling task of sharing the loot with his comrades. Ruffus always has the loot hung up on long wall behind a curtain. When Ruffus' friends come in, they sit in their favorite chairs. Then once Ruffus gives the signal the curtains fully open and the pirates grab the piece loot they desire the most.

Pirates are notoriously tunnel visioned. When they notice something they want, they will want it more than anything else offered to them. This isn't so bad, except it sometimes causes scuffles. Ruffus devised a plan; Ruffus leaves the curtains slightly open so that way each pirate can see a small section of the wall prior to the unveiling. This causes the pirates (in their favorite seats) to latch on to a single piece of wealth in a fixed range.

The problem with Ruffus' plan is that pirates that sit close to each other can potentially see overlapping sections of wall. If pirates see the same item, they may get in a fight, and Ruffus might need to hire more pirates. Unfortunately, the curtains broke when Ruffus was devising his plan, and now the curtains don't close all the way. Now the curtains will always have a gap of a fixed size and position.

You will be given the position of each pirate's chair and the positions of the curtain gap. Your job is to determine the minimum number of times Ruffus needs to unveil the loot to the pirates such that there is no possible conflict. You can assume that the loot wall will be the x-axis (the line $y = 0$). You can also assume that a pirate will always be able to see some part of the loot wall through the curtain.

Assume items have a positive, but arbitrarily small width.

Input Specification

The first line of input will contain 4 integers x_1 , y_1 , x_2 , and y_2 . The curtain's gap will be from (x_1, y_1) to (x_2, y_2) . The curtain (and subsequently its gap) will be parallel to the x-axis.

The second line of input will be single positive integers n ($n < 100,000$), representing the number of pirates. The following n lines will describe a pirate. Each pirate description consists of 1 pair of integers representing the point of the pirate's chair on the 2-dimensional Cartesian grid.

All y-coordinates will be positive, and no pirate will be between the curtain and the loot wall, or on the curtain. No coordinate value will be greater than 100,000 in magnitude.

Output Specification

The first and only line of output should contain a number represent the minimum number of meetings Luffus needs to prevent pirate disagreements.

Input Output Example

Input	Output
-1 1 1 1 4 10 2 -2 2 2 2 1 10	2
1 2 0 2 10 8 6 7 5 3 10 4 6 6 9 2 5 7 7 2 15 6 6 4 9	5

Explanation

Case 1:

The first and fourth pirate can go to the first reveal.

The third and the second pirate can go to a second reveal.

There is no way to fit all the pirates into one reveal.

Case 2:

Pirates 2, 6, and 7 can be in the first reveal.

Pirates 1, 5, and 8 can be in the second reveal.

Pirates 3 and 9 can be in the third reveal.

Pirate 4 will have their own reveal.

Pirate 10 will have their own reveal as well.

There is no way to fit all the pirates into four or less reveals.

Grading Information

Reading from and writing to standard input – 10 points

Turn each pirate into some “interval(s)” – 10 points

Sort the pirates’ interval(s) – 10 points

Greedily make decisions – 10 points

Comments, white space usage, and reasonable variable names – 10 points

Your program will be tested on 10 test cases – 5 points each

No points will be awarded to programs that do not compile.

Only cases that finish within the maximum of {5 times my solution, 10 seconds} will be graded.

This is a much easier problem than the first version.