

# Practice Problem B

## Escaping

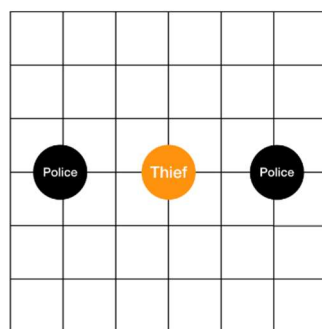
Time Limit: 1 second

We play a pursuit game where a thief is located at a specific grid point on an infinite grid plane. To catch this thief,  $N$  police officers are also deployed at  $N$  different grid points other than the thief's location. One thief and  $N$  police officers take turns moving alternately. The thief moves first. However, at police officers turn, all  $N$  police officers move at the same time. Police officers and thief can move to a point among the four adjacent (neighboring) grid points at their turn.

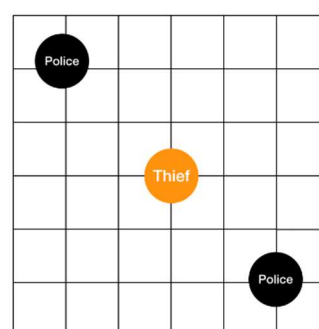
Each police officer can either stay or move, but the thief must move to one of the neighboring points at its turn. If a police officer and the thief meet at the same grid point, the thief is caught by the police officer and the game is over. In this game, we assume that all  $N$  police officers and the thief do their best. So the thief tries to escape eternally and the police officers also do their best to catch the thief as quickly as possible.

If the thief can continue moving without being caught by any police officer no matter what strategy the police officers use, then the initial game configuration is called an "escapable" case for the thief. You have to determine whether or not the given initial configuration is an "escapable" case for a thief.

For example, if the thief keeps moving in the vertical direction for the initial configuration shown in the left figure below, the police officers cannot catch the thief, so the thief can escape forever. However, for the initial configuration the right figure below, no matter how the thief moves in any direction, the thief is eventually caught by the police officers.



Escapable case



Non-escapable case

### Input

Your program is to read from standard input. The input starts with a line containing one integer  $N$ , the number of police officers ( $1 \leq N \leq 500,000$ ). Each of the following  $N$  lines contains two integers,  $x_i$  and  $y_i$ , where  $(x_i, y_i)$  is the initial position (grid coordinate) of each police officer. The next line contains two integers,  $s_x$  and  $s_y$ , where  $(s_x, s_y)$  is the initial position (grid coordinate) of the thief. All the values  $x_i$ ,  $y_i$ ,  $s_x$ , and  $s_y$  are between  $-10^9$  and  $10^9$  inclusively. Note that all the thief and  $N$  police officers are located at different grid positions in the infinite grid plane.

### Output

Your program is to write to standard output. Print exactly one line. The line should contain YES if the thief can escape forever for the initial configuration, NO if the initial configuration is not escapable.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
2 1 3 5 3 3 3	YES

Sample Input 2	Output for the Sample Input 2
2 1 5 5 1 3 3	NO

## Practice Problem B

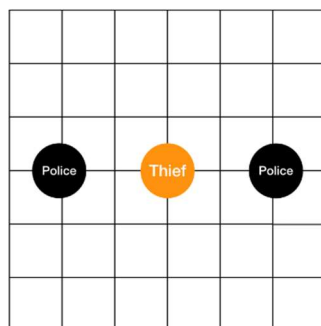
### Escaping

제한 시간: 1 초

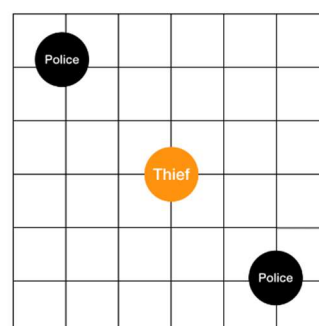
무한 정수 좌표 평면 위 한 격자 점에 있는 도둑 잡기 게임을 해 보자. 이 도둑을 잡기 위해  $N$ 명의 경찰이 도둑이 없는  $N$ 개의 다른 격자 점에 배치되어 있다. 도둑 한 명과  $N$ 명 경찰들은 서로 한번씩 번갈아가면서 움직인다. 도둑이 먼저 시작한다. 경찰 차례일 때는  $N$  명의 경찰 모두가 동시에 움직인다. 경찰과 도둑은 상하좌우 인접한 네 개의 격자 점 중 하나로 이동할 수 있다. 단, 경찰은 그대로 머물러 있을 수 있지만 도둑은 자신의 차례가 오면 반드시 이웃 격자 점으로 이동해야 한다. 이때 경찰과 도둑이 같은 격자 점에서 만나면 도둑은 잡힌 것이 되고 게임은 끝이 난다. 이 게임에서 경찰과 도둑은 최선을 다한다. 즉, 도둑은 최대한 잡히지 않으려는 전략을, 경찰은 최대한 빨리 잡으려는 전략을 쓴다.

입력으로 주어진 초기 위치에서 경찰이 어떤 전략을 사용하더라도 도둑이 경찰에 의해 잡히지 않고 영원히 도망 다닐 수 있다면, 그 초기 조건은 도둑이 “탈출 가능한” 조건이라고 부른다. 여러분은 초기 조건, 즉 한 명의 도둑과  $N$  명의 경찰의 처음 위치를 보고 도둑이 탈출 가능한지 판단해야 한다.

예를 들어, 다음 아래 왼쪽 그림과 같은 초기 조건에서 도둑이 북쪽 또는 남쪽 방향으로만 움직인다면 경찰은 도둑을 영원히 잡을 수 없다. 그러나 아래 오른쪽 그림과 같은 조건이라면 도둑이 어떻게 도망을 가더라도 최선을 다하는 경찰에 의해서 결국은 잡히게 된다.



탈출 가능



탈출 불가능

**Input**

첫 번째 줄에는 경찰의 수  $N$ 이 주어진다. 단,  $1 \leq N \leq 500,000$ 이다. 그 다음  $N$  개의 줄에는 각 경찰의 초기 위치의 좌표  $(x_i, y_i)$ 가 공백을 사이에 두고 주어진다. 다음 줄에는 도둑의 초기 위치의 좌표  $(s_x, s_y)$ 가 공백을 사이에 두고 주어진다. 도둑과 경찰의 좌표는 모두 다른 격자 점으로 주어지며, 정수  $x_i, y_i, s_x, s_y$ 의 범위는  $-10^9 \leq x_i, y_i, s_x, s_y \leq 10^9$ 이다.

**Output**

주어진 초기 조건에서 도둑이 무한히 도망갈 수 있다면, 즉, 초기 조건이 탈출 가능한 조건이라면, YES, 탈출 가능하지 않다면 NO 를 출력한다.

다음은 두 테스트 케이스에 대한 입출력 예이다.

**Sample Input 1**

```
2
1 3
5 3
3 3
```

**Output for the Sample Input 1**

```
YES
```

**Sample Input 2**

```
2
1 5
5 1
3 3
```

**Output for the Sample Input 2**

```
NO
```