Points on a Rectangle ☆

Points: 477.99 Rank: 2874

Problem Submissions Leaderboard Discussions Editorial

You are given q queries where each query consists of a set of n points on a two-dimensional plane (i.e., (x, y)). For each set of points, print `YES` on a new line if all the points fall on the edges (i.e., sides and/or corners) of a non-degenerate rectangle which is axis parallel; otherwise, print `NO` instead.

Input Format

The first line contains a single positive integer, q, denoting the number of queries. The subsequent lines describe each query in the following format:

1. The first line contains a single positive integer, n, denoting the number of points in the query.
2. Each line i of the n subsequent lines contains two space-separated integers describing the respective values of x_i and y_i for the point at coordinate (x_i, y_i).

Constraints

- $1 \leq q \leq 10$
- $1 \leq n \leq 10$
- $-10^4 \leq x, y \leq 10^4$

Output Format

For each query, print `YES` on a new line if all n points lie on the edges of some non-degenerate rectangle which is axis parallel; otherwise, print `NO` instead.

Sample Input

```
2
3
0 0
0 1
1 0
4
0 0
0 2
2 0
1 1
```

Sample Output

```
YES
NO
```

Explanation

We perform the following $q = 2$ queries:

1. In the first query, all points lie on the edges of a non-degenerate rectangle with corners at $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$. Thus, we print `YES` on a new line.

Author	gorbunovdv
Difficulty	Easy
Max Score	30
Submitted By	196

NEED HELP?

View discussions

View editorial

View top submissions

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

Download problem statement

Download sample test cases

Suggest Edits

2. In the second query, points $(0,0)$, $(0,2)$, and $(2,0)$ *could* all lie along the edge of some non-degenerate rectangle, but point $(1,1)$ would have to fall *inside* that rectangle. Thus, we print `NO` on a new line.

Current Buffer (saved locally, editable)

Java 7

```java
import java.io.*;
import java.math.*;
import java.text.*;
import java.util.*;
import java.util.regex.*;

public class Solution {

    // Complete the solve function below.
    static String solve(int[][] coordinates) {


    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        int q = scanner.nextInt();
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        for (int qItr = 0; qItr < q; qItr++) {
            int n = scanner.nextInt();
            scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

            int[][] coordinates = new int[n][2];

            for (int i = 0; i < n; i++) {
                String[] coordinatesRowItems = scanner.nextLine().split(" ");
                scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

                for (int j = 0; j < 2; j++) {
                    int coordinatesItem = Integer.parseInt(coordinatesRowItems[j]);
                    coordinates[i][j] = coordinatesItem;
                }
            }

            String result = solve(coordinates);

            bufferedWriter.write(result);
            bufferedWriter.newLine();
        }

        bufferedWriter.close();

        scanner.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input **Run Code** **Submit Code**