Hash application: finding the max coordinate sum of a point with a diagonal neighbor

The goal of this assignment is to solve a problem fast using hashing.

Input The input has the following format. The first line has one integer, which indicates the number of points. Each of the following lines has two non-negative integers, which indicates the x and y-coordinates of a point, respectively.

Output Let $p_1, p_2, ..., p_n$ be the points in the input. We say that a point p_i has a monotonically diagonal neighbor p_j if

- $p_i.x - p_j.x = 1$ and $p_i.y - p_j.y = 1$; or

- $p_i.x - p_j.x = -1$ and $p_i.y - p_j.y = -1$

Let S be the set of points that have a monotonically diagonal neighbor. Our goal is to output:

- -1 if $S = \emptyset$

- $\max_{p_i \in S} p_i.x + p_i.y$ otherwise.

Examples of input and output

```
Input 1
2
0 0
1 0
```

```
Output 1
-1
```

Note: In Input 1 no points have a monotonically diagonal neighbor, so, the desired output is 1.

```
Input 2
4
0 0
1 1
2 2
3 2
Output 2
4
```

Note: In Input 2, (0, 0), (1, 1), (2, 2) have a monotonically diagonal neighbor; (0, 0) has one, (1, 1) has two, and (2, 2) has one. So, here $S = \{(0,0), (1,1), (2,2)\}$ and $(2,2)$ has the highest sum of the coordinates, so we output 4.

Note that the output is a single number and has no white character.

Algorithm Create a hash table. Insert all points into the table. For each point p_i, we can check if p_i has a monotonically (smaller) diagonal neighbor by trying to find $(p_i.x - 1, p_i.y - 1)$ in the table in $O(1)$ time.[1] If p_i does, and it has a higher coordinate sum than we have observed so far, we update the max coordinate sum.

As you may not be familiar with how to use hashing, an example file will be provided. Although this problem can be solved using set data structure, we used map in the file as it is more versatile. You may use and modify the file as needed.

Warning We will test your code against large inputs. If you don't use hashing, your implementation will be slow and we will/can catch it. If you don't use hashing, you will get 0 points.

See the lab guidelines for submission/grading, etc., which can be found in Files/Labs.

---

[1]We don't have to check $(p_i.x + 1, p_i.y + 1)$. Do you see why?