1779. Find Nearest Point That Has the Same X or Y Coordinate

Easy 272 44 Add to List Share

You are given two integers, `x` and `y`, which represent your current location on a Cartesian grid: `(x, y)`. You are also given an array `points` where each `points[i] = [a_i, b_i]` represents that a point exists at `(a_i, b_i)`. A point is **valid** if it shares the same x-coordinate or the same y-coordinate as your location.

Return *the index* ***(0-indexed)*** *of the* ***valid*** *point with the smallest* ***Manhattan distance*** *from your current location*. If there are multiple, return *the valid point with the* ***smallest*** *index*. If there are no valid points, return `-1`.

The **Manhattan distance** between two points `(x_1, y_1)` and `(x_2, y_2)` is `abs(x_1 - x_2) + abs(y_1 - y_2)`.

Example 1:

```
Input: x = 3, y = 4, points = [[1,2],[3,1],[2,4],[2,3],[4,4]]
Output: 2
Explanation: Of all the points, only [3,1], [2,4] and [4,4] are valid. Of the valid points, [2,4] and [4,4] have the smallest Manhattan distance from your current location, with a distance of 1. [2,4] has the smallest index, so return 2.
```

Example 2:

```
Input: x = 3, y = 4, points = [[3,4]]
Output: 0
Explanation: The answer is allowed to be on the same location as your current location.
```

Example 3:

```
Input: x = 3, y = 4, points = [[2,3]]
Output: -1
Explanation: There are no valid points.
```

Constraints:

- `1 <= points.length <= 10^4`
- `points[i].length == 2`
- `1 <= x, y, a_i, b_i <= 10^4`

Accepted 31,446 Submissions 45,828

Java Autocomplete

```java
class Solution {
    public int nearestValidPoint( int x, int y, int[][] points ) {

        int length = points.length;
        int min = Integer.MAX_VALUE;
        int index = Integer.MAX_VALUE;

        for (int i = 0; i < length; i++) {

            int r1 = x - points[i][0];
            int r2 = y - points[i][1];

            if (r1 == 0) {
                if (min > Math.abs(r2)) {
                    min = Math.abs(r2);
                    index = i;
                }
            } else if (r2 == 0) {
                if (min > Math.abs(r1)) {
                    min = Math.abs(r1);
                    index = i;
                }
            }
        }

        return index == Integer.MAX_VALUE ? -1 : index;

    }
}
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input
```
3
4
```

Output
```
2
0
```

Diff

Expected
```
2
0
```