

5680. Find Nearest Point That Has the Same X or Y Coordinate

My Submissions (/contest/biweekly-contest-47/problems/find-nearest-point-that-has-the-same-x-or-y-coordinate/submissions/)

Back to Contest (/contest/biweekly-contest-47/)

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] = [ai, bi]` 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 $\text{abs}(x_1 - x_2) + \text{abs}(y_1 - y_2)$.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Easy

Example 1:

Input: $x = 3, y = 4, \text{points} = [[1,2],[3,1],[2,4],[2,3],[4,4]]$
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 M

Example 2:

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

Example 3:

Input: $x = 3, y = 4, \text{points} = [[2,3]]$
Output: -1
Explanation: There are no valid points.

Constraints:

- $1 \leq \text{points.length} \leq 10^4$
- $\text{points}[i].\text{length} == 2$
- $1 \leq x, y, a_i, b_i \leq 10^4$

JavaScript

kb ↺ ⚙

```
1 /**
2  * @param {number} x
3  * @param {number} y
4  * @param {number[][]} points
5  * @return {number}
6  */
7  const mi = Math.min;
8  const abs = Math.abs;
9  const nearestValidPoint = (x, y, points) => {
10     let min = Number.MAX_SAFE_INTEGER;
11     let n = points.length;
12     for (const p of points) {
13         if (p[0] == x || p[1] == y) {
14             let tmp = abs(p[0] - x) + abs(p[1] - y);
15             min = mi(min, tmp);
16         }
17     }
18     for (let i = 0; i < n; i++) {
19         if (points[i][0] == x || points[i][1] == y) {
20             let tmp = abs(points[i][0] - x) + abs(points[i][1] - y);
21             if (tmp == min) {
22                 return i;
23             }
24         }
25     }
26     return -1;
27 }
```

```
25     }  
26     return -1;  
27 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result: Accepted** (</submissions/detail/464251943/>) [More Details >](/submissions/detail/464251943/)

Share your acceptance!

Copyright © 2021 LeetCode [Help Center \(/support\)](/support) | [Jobs \(/jobs\)](/jobs) | [Bug Bounty \(/bugbounty\)](/bugbounty) | [Students \(/student\)](/student) | [Terms \(/terms\)](/terms) | [Privacy Policy \(/privacy\)](/privacy)

 [United States \(/region\)](/region)