

Problem 3102: Minimize Manhattan Distances

Problem Information

Difficulty: Hard

Acceptance Rate: 32.42%

Paid Only: No

Tags: Array, Math, Geometry, Sorting, Ordered Set

Problem Description

You are given an array `points` representing integer coordinates of some points on a 2D plane, where `points[i] = [xi, yi]` .

The distance between two points is defined as their Manhattan distance.

Return _the**minimum** possible value for **maximum** distance between any two points by removing exactly one point_.

Example 1:

Input: points = [[3,10],[5,15],[10,2],[4,4]]

Output: 12

Explanation:

The maximum distance after removing each point is the following:

* After removing the 0th point the maximum distance is between points (5, 15) and (10, 2), which is $|5 - 10| + |15 - 2| = 18$. * After removing the 1st point the maximum distance is between points (3, 10) and (10, 2), which is $|3 - 10| + |10 - 2| = 15$. * After removing the 2nd point the maximum distance is between points (5, 15) and (4, 4), which is $|5 - 4| + |15 - 4| = 12$. * After removing the 3rd point the maximum distance is between points (5, 15) and (10, 2), which is $|5 - 10| + |15 - 2| = 18$.

12 is the minimum possible maximum distance between any two points after removing exactly one point.

Example 2:

Input: points = [[1,1],[1,1],[1,1]]

Output: 0

Explanation:

Removing any of the points results in the maximum distance between any two points of 0.

Constraints:

* `3 <= points.length <= 105` * `points[i].length == 2` * `1 <= points[i][0], points[i][1] <= 108`

Code Snippets

C++:

```
class Solution {
public:
    int minimumDistance(vector<vector<int>>& points) {
        }
};
```

Java:

```
class Solution {
public int minimumDistance(int[][][] points) {
        }
}
```

Python3:

```
class Solution:
    def minimumDistance(self, points: List[List[int]]) -> int:
```

