Problem 3275: K-th Nearest Obstacle Queries

Problem Information

Difficulty: Medium
Acceptance Rate: 48.58%
Paid Only: No
Tags: Array, Heap (Priority Queue)

Problem Description

There is an infinite 2D plane.

You are given a positive integer `k`. You are also given a 2D array `queries`, which contains the following queries:

* `queries[i] = [x, y]`: Build an obstacle at coordinate `(x, y)` in the plane. It is guaranteed that there is **no** obstacle at this coordinate when this query is made.

After each query, you need to find the **distance** of the `kth` **nearest** obstacle from the origin.

Return an integer array `results` where `results[i]` denotes the `kth` nearest obstacle after query `i`, or `results[i] == -1` if there are less than `k` obstacles.

Note that initially there are **no** obstacles anywhere.

The **distance** of an obstacle at coordinate `(x, y)` from the origin is given by `|x| + |y|`.

Example 1:

Input: queries = [[1,2],[3,4],[2,3],[-3,0]], k = 2

Output: [-1,7,5,3]

Explanation:

* Initially, there are 0 obstacles. * After `queries[0]`, there are less than 2 obstacles. * After `queries[1]`, there are obstacles at distances 3 and 7. * After `queries[2]`, there are obstacles at distances 3, 5, and 7. * After `queries[3]`, there are obstacles at distances 3, 3, 5, and 7.

Example 2:

Input: queries = [[5,5],[4,4],[3,3]], k = 1

Output: [10,8,6]

Explanation:

* After `queries[0]`, there is an obstacle at distance 10. * After `queries[1]`, there are obstacles at distances 8 and 10. * After `queries[2]`, there are obstacles at distances 6, 8, and 10.

Constraints:

* `1 <= queries.length <= 2 * 105` * All `queries[i]` are unique. * `-109 <= queries[i][0], queries[i][1] <= 109` * `1 <= k <= 105`

Code Snippets

C++:

```
class Solution {
public:
vector<int> resultsArray(vector<vector<int>>& queries, int k) {

}
};
```

Java:

```
class Solution {
public int[] resultsArray(int[][] queries, int k) {

}
}
```

Python3:

```
class Solution:
def resultsArray(self, queries: List[List[int]], k: int) -> List[int]:
```