LeetCode Explore Problems Interview New Contest Discuss Store Premium

Description Solution Discuss (999+) Submissions

Java Autocomplete

973. K Closest Points to Origin

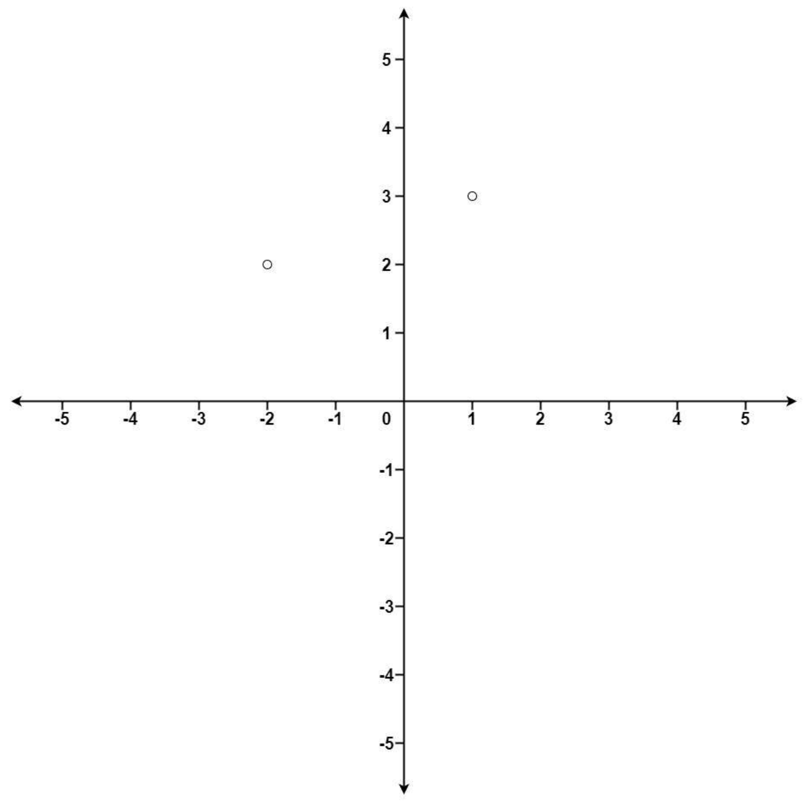
Medium 5146 202 Add to List Share

Given an array of `points` where `points[i] = [xᵢ, yᵢ]` represents a point on the **X-Y** plane and an integer `k`, return the `k` closest points to the origin `(0, 0)`.

The distance between two points on the **X-Y** plane is the Euclidean distance (i.e., $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$).

You may return the answer in **any order**. The answer is **guaranteed** to be **unique** (except for the order that it is in).

Example 1:



```
Input: points = [[1,3],[-2,2]], k = 1
Output: [[-2,2]]
Explanation:
The distance between (1, 3) and the origin is sqrt(10).
The distance between (-2, 2) and the origin is sqrt(8).
Since sqrt(8) < sqrt(10), (-2, 2) is closer to the origin.
We only want the closest k = 1 points from the origin, so the answer is just [[-2,2]].
```

```java
class Solution {

    private int squareOfDistanceFromRoot( int x, int y ) {
        return x * x + y * y;
    }

    public int[][] kClosest( int[][] points, int k ) {

        int[][] kClosestPoints = new int[k][2];

        PriorityQueue<int[]> priorityQueue = new PriorityQueue<>
(Comparator.comparingInt(o ->
                squareOfDistanceFromRoot(o[1], o[2])));

        for (int[] point : points) {
            int distance = squareOfDistanceFromRoot(point[0], point[1]);
            priorityQueue.offer(new int[]{distance, point[0],
point[1]});
        }

        for (int i = 0; i < k; i++) {
            int[] poll = priorityQueue.poll();
            kClosestPoints[i] = new int[]{poll[1], poll[2]};
        }

        return kClosestPoints;
    }

}
```

Problems Pick One < Prev 973/2209 Next > Console Contribute i Run Code Submit