1828. Queries on Number of Points Inside a Circle

Medium 581 56 Add to List Share

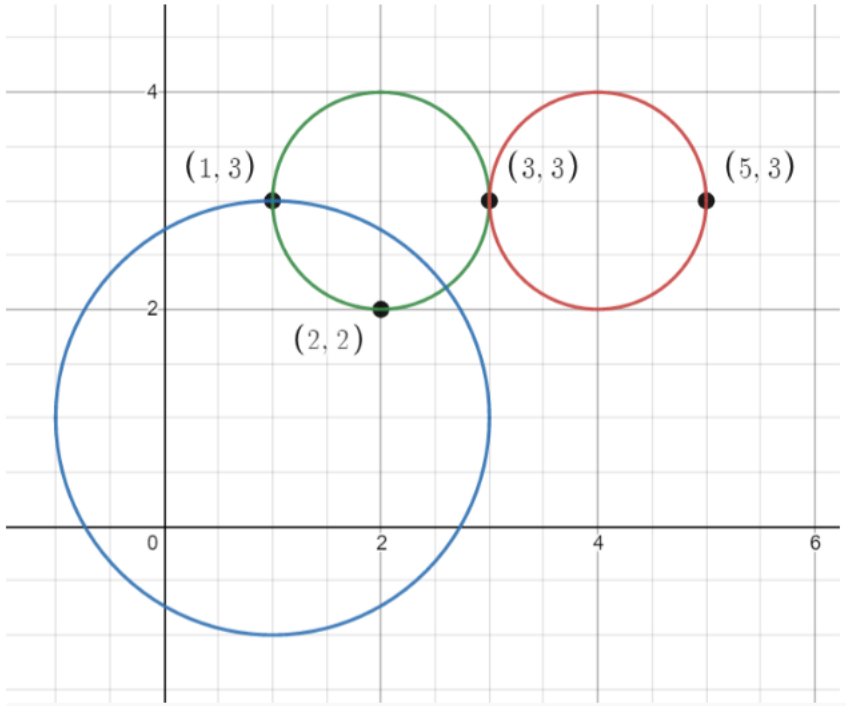
You are given an array `points` where `points[i] = [x_i, y_i]` is the coordinates of the `i^th` point on a 2D plane. Multiple points can have the **same** coordinates.

You are also given an array `queries` where `queries[j] = [x_j, y_j, r_j]` describes a circle centered at `(x_j, y_j)` with a radius of `r_j`.

For each query `queries[j]`, compute the number of points **inside** the `j^th` circle. Points **on the border** of the circle are considered **inside**.

Return *an array* `answer`, *where* `answer[j]` *is the answer to the* `j^th` *query.*

Example 1:



Input: points = [[1,3],[3,3],[5,3],[2,2]], queries = [[2,3,1],[4,3,1],[1,1,2]]
Output: [3,2,2]
Explanation: The points and circles are shown above.
queries[0] is the green circle, queries[1] is the red circle, and queries[2] is the blue circle.

Constraints:

- `1 <= points.length <= 500`
- `points[i].length == 2`
- `0 <= x_i, y_i <= 500`
- `1 <= queries.length <= 500`
- `queries[j].length == 3`
- `0 <= x_j, y_j <= 500`
- `1 <= r_j <= 500`
- All coordinates are integers.

```cpp
vector<int> countPoints(vector<vector<int>>& points, vector<vector<int>>& queries) {
    int n = points.size(), m = queries.size();
    vector<int> result(m, 0);

    for(int i=0; i<m; i++) {
        int x = queries[i][0], y = queries[i][1], r = queries[i][2], count = 0;
        for(int j=0; j<n; j++) {
            int p = points[j][0], q = points[j][1];
            if(((x-p)*(x-p) + (y-q)*(y-q)) <= r*r) {
                count++;
            }
        }
        result[i] = count;
    }
    return result;
}
```

#100daysofDSA





/rvislive

Rakesh Vishwakarma