5718. Queries on Number of Points Inside a Circle

My Submissions (/contest/biweekly-contest-50/problems/queries-on-number-of-points-inside-a-circle/submissions/)

Back to Contest (/contest/biweekly-contest-50/)

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

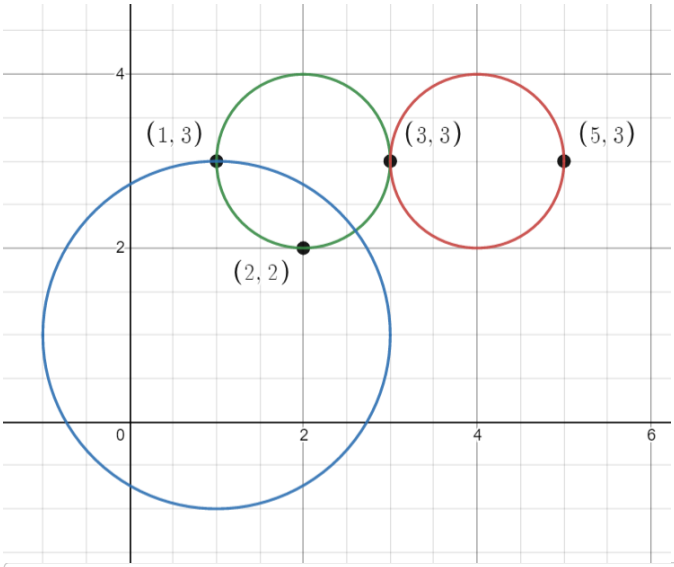
You are given an array `points` where `points[i] = [x_i, y_i]` is the coordinates of the `i^th` point on a 2D plane. Multiple points can have the **same** coordinates.

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**.

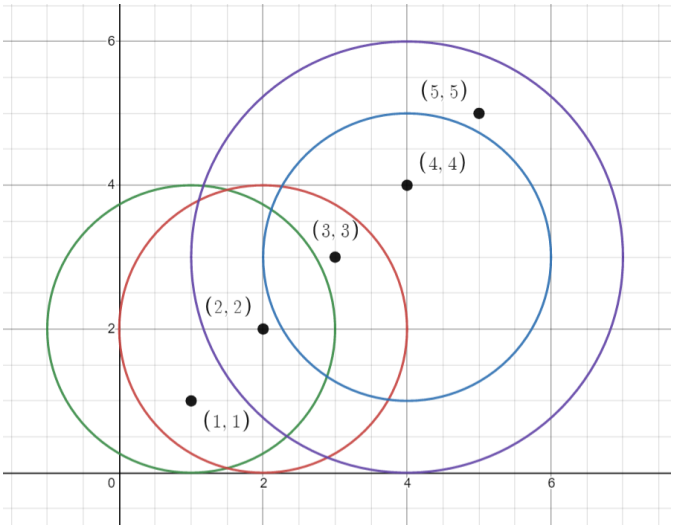
Return *an array* `answer`, *where* `answer[j]` *is the answer to the* `j^th` *query*.

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.
```

Example 2:



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

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.

JavaScript

```javascript
const sq = Math.sqrt;
const countPoints = (points, queries) => {
    let res = [];
    for (const q of queries) {
        let cnt = 0;
        for (const p of points) {
            if (ok(p, q)) cnt++;
        }
        res.push(cnt);
    }
    return res;
};

const ok = (p, q) => {
    let r = q[2];
    let cx = q[0];
    let cy = q[1];
    let px = p[0];
    let py = p[1];
    let dis = sq((px - cx) ** 2 + (py - cy) ** 2);
    return dis <= r;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/481753616/) | More Details > (/submissions/detail/481753616/)

Share your acceptance!