6042. Count Lattice Points Inside a Circle

My Submissions (/contest/weekly-contest-290/problems/count-lattice-points-inside-a-circle/submissions/) Back to Contest (/contest/weekly-contest-290/)

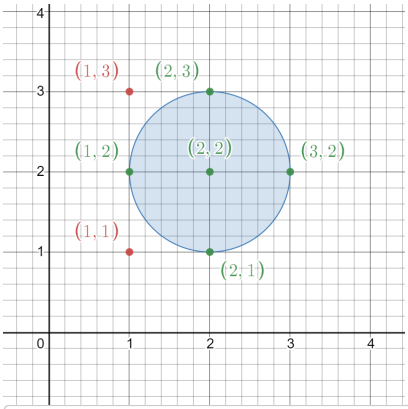
Given a 2D integer array `circles` where `circles[i] = [x_i, y_i, r_i]` represents the center `(x_i, y_i)` and radius `r_i` of the `i^th` circle drawn on a grid, return *the **number of lattice points** that are present inside **at least one** circle*.

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

Note:

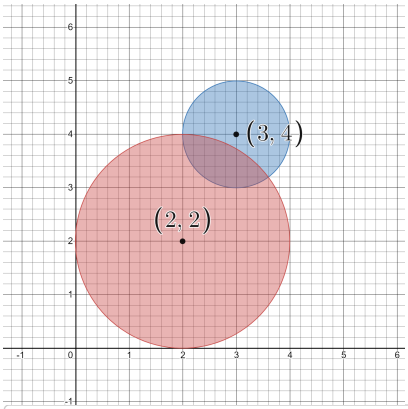
- A **lattice point** is a point with integer coordinates.
- Points that lie **on the circumference of a circle** are also considered to be inside it.

Example 1:




```
Input: circles = [[2,2,1]]
Output: 5
Explanation:
The figure above shows the given circle.
The lattice points present inside the circle are (1, 2), (2, 1), (2, 2), (2, 3), and (3, 2) and are shown in green.
Other points such as (1, 1) and (1, 3), which are shown in red, are not considered inside the circle.
Hence, the number of lattice points present inside at least one circle is 5.
```

Example 2:




```
Input: circles = [[2,2,2],[3,4,1]]
Output: 16
Explanation:
The figure above shows the given circles.
There are exactly 16 lattice points which are present inside at least one circle.
Some of them are (0, 2), (2, 0), (2, 4), (3, 2), and (4, 4).
```

Constraints:

- `1 <= circles.length <= 200`
- `circles[i].length == 3`
- `1 <= x_i, y_i <= 100`

- $1 <= r_i <= min(x_i, y_i)$

JavaScript

```javascript
const countLatticePoints = (circles) => {
    let res = new Set();
    for (const [x, y, r] of circles) {
        // let topRight = [x + r, y + r], bottomRight = [x + r, y - r];
        // let topLeft = [x - r, y + r], bottomLeft = [x - r, y - r];
        // pr(topLeft, bottomLeft, topRight, bottomRight)
        for (let i = x - r; i <= x + r; i++) {
            for (let j = y - r; j <= y + r; j++) {
                if (inCircle(i, j, x, y, r)) res.add(i + ' ' + j);
            }
        }
    }
    return res.size;
};

const inCircle = (x, y, cx, cy, r) => {
    let disPow = Math.abs(x - cx) ** 2 + Math.abs(y - cy) ** 2;
    return disPow <= r * r;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/686335655/) | More Details > (/submissions/detail/686335655/)

Share your acceptance!

2