Problem 3235: Check if the Rectangle Corner Is Reachable

Problem Information

Difficulty: Hard
Acceptance Rate: 25.40%
Paid Only: No
Tags: Array, Math, Depth-First Search, Breadth-First Search, Union Find, Geometry

Problem Description

You are given two positive integers `xCorner` and `yCorner`, and a 2D array `circles`, where `circles[i] = [xi, yi, ri]` denotes a circle with center at `(xi, yi)` and radius `ri`.

There is a rectangle in the coordinate plane with its bottom left corner at the origin and top right corner at the coordinate `(xCorner, yCorner)`. You need to check whether there is a path from the bottom left corner to the top right corner such that the **entire path** lies inside the rectangle, **does not** touch or lie inside **any** circle, and touches the rectangle **only** at the two corners.

Return `true` if such a path exists, and `false` otherwise.

Example 1:

Input: xCorner = 3, yCorner = 4, circles = [[2,1,1]]

Output: true

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/18/example2circle1.png)

The black curve shows a possible path between `(0, 0)` and `(3, 4)`.

Example 2:

Input: xCorner = 3, yCorner = 3, circles = [[1,1,2]]

Output: false

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/18/example1circle.png)

No path exists from `(0, 0)` to `(3, 3)`.

Example 3:

Input: xCorner = 3, yCorner = 3, circles = [[2,1,1],[1,2,1]]

Output: false

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/18/example0circle.png)

No path exists from `(0, 0)` to `(3, 3)`.

Example 4:

Input: xCorner = 4, yCorner = 4, circles = [[5,5,1]]

Output: true

Explanation:

![](https://assets.leetcode.com/uploads/2024/08/04/rectangles.png)

Constraints:

* `3 <= xCorner, yCorner <= 109` * `1 <= circles.length <= 1000` * `circles[i].length == 3` * `1 <= xi, yi, ri <= 109`

Code Snippets

C++:

```
class Solution {
public:
bool canReachCorner(int xCorner, int yCorner, vector<vector<int>>& circles) {

}
};
```

Java:

```
class Solution {
public boolean canReachCorner(int xCorner, int yCorner, int[][] circles) {

}
}
```

Python3:

```
class Solution:
def canReachCorner(self, xCorner: int, yCorner: int, circles:
List[List[int]]) -> bool:
```