# 题目

给你一个以 (radius, xCenter, yCenter) 表示的圆和一个与坐标轴平行的矩形 (x1, y1, x2, y2) ，其中 (x1, y1) 是矩形左下角的坐标，而 (x2, y2) 是右上角的坐标。

如果圆和矩形有重叠的部分，请你返回 true ，否则返回 false 。

换句话说，请你检测是否存在点 (xi, yi) ，它既在圆上也在矩形上（两者都包括点落在边界上的情况）。

示例 1 ：

输入：radius = 1, xCenter = 0, yCenter = 0, x1 = 1, y1 = -1, x2 = 3, y2 = 1

输出：true

解释：圆和矩形存在公共点 (1,0) 。

示例 2 ：

输入：radius = 1, xCenter = 1, yCenter = 1, x1 = 1, y1 = -3, x2 = 2, y2 = -1

输出：false

示例 3 ：

输入：radius = 1, xCenter = 0, yCenter = 0, x1 = -1, y1 = 0, x2 = 0, y2 = 1

输出：true

提示：

1 <= radius <= 2000

-104 <= xCenter, yCenter <= 104

-104 <= x1 < x2 <= 104

-104 <= y1 < y2 <= 104

# 分析

## 方法一：x和y方向最短距离

思路：

判断圆和矩形是否重叠，可以通过计算圆的圆心到矩形的最短距离，并与圆的半径进行比较。具体步骤如下：

1、确定圆心在矩形的哪个方位：计算圆心在 x 轴和 y 轴方向上相对于矩形的位置，确定圆心是在矩形内部、左侧、右侧、上方还是下方。

2、计算圆心到矩形的最短距离：

如果圆心在矩形内部，最短距离为 0。

如果圆心在矩形左侧，x 方向的最短距离为矩形左边界到圆心的水平距离。

如果圆心在矩形右侧，x 方向的最短距离为圆心到矩形右边界的水平距离。

类似地计算 y 方向的最短距离。

3、判断是否重叠：如果圆心到矩形的最短距离的平方小于或等于圆半径的平方，则圆和矩形重叠。

代码：

class Solution {

public:

bool checkOverlap(int radius, int xCenter, int yCenter, int x1, int y1, int x2, int y2) {

// 计算圆心在x方向到矩形的最短距离

int dx = 0;

if (xCenter < x1) {

dx = x1 - xCenter;

} else if (xCenter > x2) {

dx = xCenter - x2;

}

// 计算圆心在y方向到矩形的最短距离

int dy = 0;

if (yCenter < y1) {

dy = y1 - yCenter;

} else if (yCenter > y2) {

dy = yCenter - y2;

}

// 检查最短距离的平方是否小于等于半径的平方

return dx \* dx + dy \* dy <= radius \* radius;

}

};

解释

计算 x 方向最短距离（dx）：如果圆心的 x 坐标小于矩形左边界 x1，则 dx 为 x1 与圆心 x 坐标的差值；如果圆心的 x 坐标大于矩形右边界 x2，则 dx 为圆心 x 坐标与 x2 的差值；否则圆心在矩形 x 范围内，dx 为 0。

计算 y 方向最短距离（dy）：类似 x 方向的计算，如果圆心的 y 坐标小于矩形下边界 y1，则 dy 为 y1 与圆心 y 坐标的差值；如果圆心的 y 坐标大于矩形上边界 y2，则 dy 为圆心 y 坐标与 y2 的差值；否则圆心在矩形 y 范围内，dy 为 0。

判断重叠：圆心到矩形的最短距离是一个直角三角形的斜边，其两条直角边分别为 dx 和 dy。根据勾股定理，斜边的平方为 dx² + dy²。如果这个值小于或等于圆半径的平方，则说明圆和矩形存在重叠。

这种方法高效地判断了圆和矩形是否重叠，时间复杂度为 O (1)，空间复杂度为 O (1)。

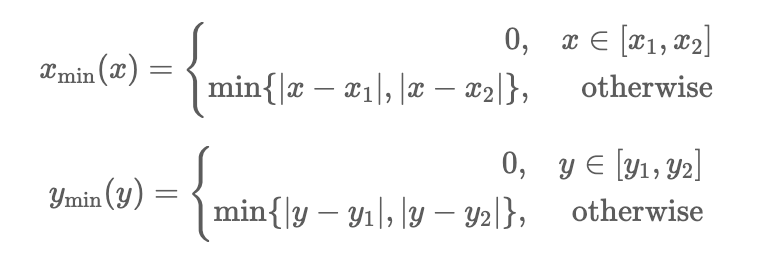
## 方法二：圆心到矩形区域最短距离

思路：

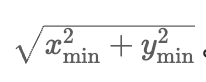
在求圆和直线的位置关系时，我们常常会计算圆心到直线的垂直线段的距离。这条垂直线段的距离小于半径的时候，就说明两者相交。更进一步地考虑，其实是因为这条垂直线段的长度已经是圆心到直线上任意点中最小的了，如果最小的线段长度比圆半径小，说明存在点在圆内。

我们可以类比这个思想，来计算求圆心到矩形区域的最短距离。

我们可以分解成两个问题，即圆心到区域 x1≤x≤x2的最小值 xmin，和圆心到区域 y1≤y≤y2的最小值 ymin，我们可以得到这样的关系：



圆心到矩形区域的最小距离就是：



未了方便理解，我们可以把平面区域根据x=x1，x=x2，y=y1，y=y2四条直线分割成九个区域，分类讨论就可以合并到这个结果。

得到这个距离之后，我们再和半径比较，如果这个距离不大于半径的话，就说明存在公共点。

class Solution {

public:

bool checkOverlap(int radius, int xCenter, int yCenter, int x1, int y1, int x2, int y2) {

long long dist = 0;

if (xCenter < x1 || xCenter > x2) {

dist += min(pow(x1 - xCenter, 2), pow(x2 - xCenter, 2));

}

if (yCenter < y1 || yCenter > y2) {

dist += min(pow(y1 - yCenter, 2), pow(y2 - yCenter, 2));

}

return dist <= radius \* radius;

}

};