# 题目

给定一个非负整数 c ，你要判断是否存在两个整数 a 和 b，使得 a2 + b2 = c 。

示例 1：

输入：c = 5

输出：true

解释：1 \* 1 + 2 \* 2 = 5

示例 2：

输入：c = 3

输出：false

提示：

0 <= c <= 231 - 1

# 分析

## 方法一：使用sqrt函数

class Solution {

public:

bool judgeSquareSum(int c) {

for (long a = 0; a \* a <= c; a++) {

// 计算 b = sqrt(c - a^2)

double b = sqrt(c - a \* a);

// 判断 b 是否为整数

if (b == (int)b) {

return true; // 如果是整数，返回 true

}

}

return false; // 循环结束仍未找到符合条件的 a、b，返回 false

}

};

复杂度分析

时间复杂度：O(sqrt(c))。枚举a的时间复杂度为O(sqrt(c))，对于每个a的值，可在O(1)的时间内寻找b。

空间复杂度：O(1)。

## 方法二：双指针

思路：

代码：

class Solution {

public:

bool judgeSquareSum(int c) {

long left = 0; // 左指针初始位置为 0

long right = (int)sqrt(c); // 右指针初始位置为 sqrt(c)

while (left <= right) { // 循环直到左指针大于右指针

long sum = left \* left + right \* right; // 计算当前两数的平方和

if (sum == c) { // 如果和等于 c，返回 true

return true;

} else if (sum > c) { // 如果和大于 c，右指针左移

right--;

} else { // 如果和小于 c，左指针右移

left++;

}

}

return false; // 如果循环结束仍未找到符合条件的数对，返回 false

}

};

复杂度分析

时间复杂度：O(sqrt{c})。最坏情况下a和b一共枚举了0到sqrt{c}里的所有整数。

空间复杂度：O(1)O(1)O(1)。