# 题目

给定一个非负整数 x ，计算并返回 x 的平方根，即实现 int sqrt(int x) 函数。

正数的平方根有两个，只输出其中的正数平方根。

如果平方根不是整数，输出只保留整数的部分，小数部分将被舍去。

示例 1:

输入: x = 4

输出: 2

示例 2:

输入: x = 8

输出: 2

解释: 8 的平方根是 2.82842...，由于小数部分将被舍去，所以返回 2

提示:

0 <= x <= 231 - 1

注意：本题与主站 69 题相同： https://leetcode-cn.com/problems/sqrtx/

# 分析

## 方法一：二分查找

**思路：**

由于x平方根的整数部分ans是满足 k2≤x的最大k值，因此我们可以对k进行二分查找，从而得到答案。

二分查找的下界为0，上界可以粗略地设定为x。在二分查找的每一步中，我们只需要比较中间元素 mid的平方与x 的大小关系，并通过比较的结果调整上下界的范围。由于我们所有的运算都是整数运算，不会存在误差，因此在得到最终的答案 ans后，也就不需要再去尝试ans+1了。

**代码：**

class Solution {

public:

int mySqrt(int x) {

int l = 0, r = x, ans = -1;

while (l <= r) {

int mid = l + (r - l) / 2;

if ((long long)mid \* mid <= x) {

ans = mid;

l = mid + 1;

} else {

r = mid - 1;

}

}

return ans;

}

};

**复杂度：**

时间复杂度：O(logx)，即为二分查找需要的次数。

空间复杂度：O(1)。