题目地址(69. x 的平方根)

https://leetcode-cn.com/problems/sqrtx

入选理由

1. 最基本的二分类型，讲义来写有哦。那么这是哪一种呢？ 题解里告诉我吧

标签

- 二分

难度

- 简单

题目描述

```
实现 int sqrt(int x) 函数。

计算并返回 x 的平方根，其中 x 是非负整数。

由于返回类型是整数，结果只保留整数的部分，小数部分将被舍去。

示例 1:

输入: 4
输出: 2
示例 2:

输入: 8
输出: 2
说明: 8 的平方根是 2.82842...,
  由于返回类型是整数，小数部分将被舍去
```

前置知识

- 二分法

思路

此题就是我讲义提到的寻找最右边的满足条件的值 的变种。

变种的点在于本题不是给定一个有序数组和 target。不过我们只需要一点点抽象即可轻松转化为我们已知的问题，进而使用模板解决。

简单抽象一下，nums 数组就是 [0,1,2,3,4,...,x] target 就是 x 的平方根。以题目的 8 为例，我们 **先不考虑结果只保留整数的部分**最后再将小数部分去掉即可。·这么来看，2,...,2.82841,2.82842,.... 都是符合的。由于需要返回不带小数的，那不就是返回 **最左边的满足条件的值** 么？但是这种算法比较复杂，原因在于计算误差，比如题目限定了 10 ** -5 以内的误差都可以，那么是可以的。

仍然以 8 为例， 我们想要在 1,2,3,4,5... （注意我这里不考虑小数了）找满足条件的 ans，使得 ans ^ 2 刚好小于等于 8，也就是找所有满足 ans ^ 2 <= 8 的最大值，也就是 **找最右边的满足条件的值** ，这里的条件就是 <= 8。

我们可以找所有满足 ans ^ 2 >= 8 的最小值，也就是 **找最左边的满足条件的值**，这里的条件就是 >= 8 么？很明显不可以。这样算的话，答案就是 3 了。（如果题目让我们向上取整就可以用啦 ^_^ ）

代码

Python：

```python
class Solution:
    def mySqrt(self, x: int) -> int:
        ans, l, r = 0, 0, x
        while l <= r:
            mid = (l + r) // 2
            if mid ** 2 > x:
                r = mid - 1
            if mid ** 2 <= x:
                ans = mid
                l = mid + 1
        return int(ans)
```

Java：

```java
class Solution {
    public int mySqrt(int x) {
        if(x==1)
            return 1;
        int left=0;
        int right=46340;
        while(left<=right){
            int mid=left+(right-left)/2;
            if(mid*mid>x){
                right=mid-1;
```

```
            }else if(mid*mid<x){
                left=mid+1;
            }else{
                return mid;
            }
        }
        return right;
    }
}
```

C++

```
class Solution {
public:
    int mySqrt(int x) {
        int l = 1, r = x, mid;
        while(l <= r){
            mid = l + (r-l)/2;
            if(x/mid > mid) l = mid + 1;
            else if (x/mid < mid) r = mid - 1;
            else return mid;
        }
        return r;
    }
};
```

复杂度分析

- 时间复杂度：$O(\log_x)$
- 空间复杂度：$O(1)$

上一页 下一页