069 Sqrt(x)

2018年3月30日 10:21

Question:

Implement int sqrt(int x).

Compute and return the square root of *x*.

x is guaranteed to be a non-negative integer.

Example 1:

Input: 4

Output: 2

Example 2:

Input: 8

Output: 2

Explanation: The square root of 8 is 2.82842..., and since we want to return an integer, the decimal part will be truncated.

来自 <https://leetcode.com/problems/sqrtx/description/>

实现 int sqrt(int x) 函数。

计算并返回 *x* 的平方根。

x 保证是一个非负整数。

案例 1:

输入: 4

输出: 2

案例 2:

输入: 8

输出: 2

说明: 8 的平方根是 2.82842..., 由于我们想返回一个整数，小数部分将被舍去。

Solution for Python3:

```
class Solution1:
    def mySqrt(self, x):
        """
        :type x: int
        :rtype: int
        """
        if x == 0:
            return 0
        l, r = 1, (x + 1) // 2
        while l <= r:
            m = (l + r) // 2
            if m * m == x:
                return m
            elif m * m > x:
                r = m - 1
            else:
                l = m + 1
        return r
```

```python

class Solution2:
    def mySqrt(self, x):
        """
        :type x: int
        :rtype: int
        """
        if x == 0:
            return x
        l, r = 1, (x + 1) // 2
        while 1:
            m = l + (r - l) // 2
            if m > x // m:
                r = m - 1;
            else:
                if (m + 1) > x // (m + 1):
                    return m
                else:
                    l = m + 1

class Solution3:
    def mySqrt(self, x):
        """
        :type x: int
        :rtype: int
        """
        r = (x + 1) // 2
        while r * r > x:
            r = (r + x // r) // 2
        return r
```

Solution for C++:

```cpp
class Solution1 {
public:
    int mySqrt(int x) {
        if (x == 0) {
            return 0;
        }
        int left = 1, right = (x + 1) / 2;
        while (true) {
            int mid = left + (right - left) / 2;
            if (mid > x / mid) {
                right = mid - 1;
            } else {
                if (mid + 1 > x / (mid + 1)) {
                    return mid;
                }
                left = mid + 1;
            }
        }
    }
};

```

```
class Solution2 {
public:
    int mySqrt(int x) {
        long r = (x + 1) / 2;
        while (r * r > x)
           r = (r + x / r) / 2;
        return r;
    }
};
```

Appendix:

牛顿梯度下降法:求如下问题的根x：

$$f\left(x_n\right) = 0$$

求解方法：

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{'}(x_n)}$$

本题应用： $f(\mathrm{r}) = r^2 - \mathrm{x} = 0$

求解过程： $\mathrm{r}_{n+1} = \mathrm{r}_n - \frac{f(\mathrm{r}_n)}{f^{'}(\mathrm{r}_n)} = \mathrm{r}_n - \frac{r_n^2 - x}{2r_n} = \frac{(r_n + (\frac{x}{r_n}))}{2}$

另外： sqrt(x) <= (x+1)/2