

```

public int mySqrt(int x) {
    if (x == 0) {
        return 0;
    }
    if (x < 0) {
        return -1;
    }
    long start = 1;
    long end = x;
    while (start + 1 < end) {
        long mid = start + (end - start) / 2;
        if (mid * mid == x) {
            return (int) mid;
        }
        else if (mid * mid < x){
            start = mid;
        }
        else {
            end = mid;
        }
    }
    if (end * end <= x){
        return (int) end;
    }
    return (int) start;
}

```

```

public int mySqrt(int x) {
    if (x == 0) {
        return 0;
    }
    if (x < 0) {
        return -1;
    }
    int start = 1;
    int end = x;
    while (start + 1 < end) {
        int mid = start + (end - start) / 2;
        if (mid == x / mid) {
            return mid;
        }
        else if (mid < x / mid ){
            start = mid;
        }
        else {
            end = mid;
        }
    }
    if (end <= x / end){
        return end;
    }
    return start;
}

```

搜索旋转排序数组

假设按照升序排序的数组在预先未知的某个关键点上旋转。

(即 0 1 2 4 5 6 7 将变成 4 5 6 7 0 1 2) 。

给你一个目标值来搜索，如果数组中存在这个数则返回它的索引，否则返回 -1。

你可以假设数组中不存在重复。

<https://leetcode-cn.com/problems/search-in-rotated-sorted-array/description/>