

# Binary Search

## *136. Single Number*

## *540. Single Element in a Sorted Array*

### 解法一：顺序查找

时间复杂度:  $O(n)$ ; 空间复杂度:  $O(1)$

找第一个和后一位不一样的数字，或是在最后一位落单的数字

### 解法二：二分查找

时间复杂度:  $O(\log(n))$ ; 空间复杂度:  $O(1)$

1. 当mid为偶数且下一位和mid位数字相同或 mid为奇数且下一位和mid不同

此时target在左侧

2. 其他情况

此时target在右侧/当前位置

---

## *278. First Bad Version*

### 解法一：二分查找

时间复杂度:  $O(\log(n))$ ; 空间复杂度:  $O(1)$

二分查找，如果当前找到的点是Bad Version，则往前找，否则往后找（包括当前这个）。最后当左侧pointer大于等于右侧pointer时，返回左侧pointer所指的值。