有序数组中的单一元素

给你一个仅由整数组成的有序数组，其中每个元素都会出现两次，唯有一个数只会出现一次。

请你找出并返回只出现一次的那个数。

你设计的解决方案必须满足 `O(log n)` 时间复杂度和 `O(1)` 空间复杂度。

示例 1:

输入: `nums = [1,1,2,3,3,4,4,8,8]`

输出: `2`

示例 2:

输入: `nums = [3,3,7,7,10,11,11]`

输出: `10`

```python
class Solution:
    def singleNonDuplicate(self, nums: List[int]) -> int:
        #两两异或 相同为 0 最后剩余则为落单数
        res=0
        for i in range(len(nums)):
            res^=nums[i]
        return res
```