

0278. 第一个错误的版本

👤 ITCharge 🕒 大约 2 分钟

- 标签：数组、二分查找
- 难度：简单

题目链接

- [0278. 第一个错误的版本 - 力扣](#)

题目大意

描述：给你一个整数 n ，代表已经发布的版本号。还有一个用于检测版本是否出错的接口 `isBadVersion(version):` 。

要求：找出第一次出错的版本号 bad 。

说明：

- 要求尽可能减少对 `isBadVersion(version):` 接口的调用。
- $1 \leq bad \leq n \leq 2^{31} - 1$ 。

示例：

- 示例 1:

输入: $n = 5$, $bad = 4$

输出: 4

解释:

调用 `isBadVersion(3)` -> false

调用 `isBadVersion(5)` -> true

调用 `isBadVersion(4)` -> true

所以, 4 是第一个错误的版本。

py

- 示例 2:

输入: $n = 1$, $bad = 1$
输出: 1

解题思路

思路 1: 二分查找

题目要求尽可能减少对 `isBadVersion(version)`: 接口的调用, 所以不能对每个版本都调用接口, 而是应该将接口调用的次数降到最低。

可以注意到: 如果检测某个版本不是错误版本时, 则该版本之前的所有版本都不是错误版本。而当某个版本是错误版本时, 则该版本之后的所有版本都是错误版本。我们可以利用这样的性质, 在 $[1, n]$ 的区间内使用二分查找方法, 从而在 $O(\log n)$ 时间复杂度内找到第一个出错误的版本。

思路 1: 代码

```
class Solution:
    def firstBadVersion(self, n):
        left = 1
        right = n
        while left < right:
            mid = (left + right) // 2
            if isBadVersion(mid):
                right = mid
            else:
                left = mid + 1
        return left
```

思路 1: 复杂度分析

- **时间复杂度:** $O(\log n)$ 。二分查找算法的时间复杂度为 $O(\log n)$ 。
- **空间复杂度:** $O(1)$ 。只用到了常数空间存放若干变量。

