

题目：

你是产品经理，目前正在带领一个团队开发新的产品。不幸的是，你的产品的最新版本没有通过质量检测。由于每个版本都是基于之前的版本开发的，所以错误的版本之后的所有版本都是错的。

假设你有 n 个版本 $[1, 2, \dots, n]$ ，你想找出导致之后所有版本出错的第一个错误的版本。

你可以通过调用 `bool isBadVersion(version)` 接口来判断版本号 `version` 是否在单元测试中出错。实现一个函数来查找第一个错误的版本。你应该尽量减少对调用 API 的次数。

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。
```

示例 2：

```
输入：n = 1, bad = 1
输出：1
```

执行结果：**通过** [显示详情](#)

[添加备注](#)

执行用时：**0 ms**，在所有 C++ 提交中击败了 **100.00%** 的用户

内存消耗：**5.9 MB**，在所有 C++ 提交中击败了 **40.63%** 的用户

通过测试用例：**22 / 22**

炫耀一下：



[写题解，分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	0 ms	5.9 MB	C++	2021/11/24 19:33	添加备注

额，比较标准的二分，还是同其他题一样

使用 `mid l r`，这里因为如果最后的答案是 `true`，所以说 `mid` 为 `true` 的时候没有 `-1`，但是当 `mid` 为 `false` 的时候，其答案肯定不是 `false`，所以再这个地方 `mid` 可以 `+1`