

第一个错误的版本

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：

给定 $n = 5$ ，并且 `version = 4` 是第一个错误的版本。

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。

解题思路：

就是查找一个特定位置，如果直接暴力查找，数据量过大肯定会超时，这就是 leetcode 的小陷阱，其次就要想到二分查找

```
// The API isBadVersion is defined for you.
// bool isBadVersion(int version);

int firstBadVersion(int n) {
    int left=1;
    int right=n;
    while(left<right)
    {
        int mid=left+(right-left)/2;
        if(isBadVersion(mid))
            right=mid;//左部分存在出错地方 缩减范围为【left, mid】
        else
            left=mid+1;//左半部分没有出错地方，右半部分查找【mid+1, right】
    }
    return left;
}
```