

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。

思路，想法很容易，二分查找即可。但是这里有个大坑是，二分的公式需写成  $i/2+h/2$  的形式，而不是  $(i+h)/2$ ，因为第二种会因为大数而溢出，导致时间超限（真的天坑，想了半天没想通，我2分查找都还时间超限？有没有天理了）。

而不是  $(l+r)/2$ ，因为后者在  $l+r$  较大时会溢出，然后就开始了乱七八糟的循环.....

by <https://blog.csdn.net/Accsc/article/details/80315299>

### 代码

```
// Forward declaration of isBadVersion API.
```

```
bool isBadVersion(int version);
```

```
class Solution {
```

```
public:
```

```
    int firstBadVersion(int n) {
```

```
        int i=1;
```

```
        int wz=(i>>1)+(n>>1);
```

```
        while(i<n){
```

```
            if(isBadVersion(wz)){
```

```
                n=wz-1;
```

```
            }else{
```

```
                i=wz+1;
```

```
            }
```

```
            wz=(i>>1)+(n>>1);
```

```
        }
```

```
        if(isBadVersion(i)){
```

```
            return i;
```

```
    }  
    return i+1;  
}  
};
```