

278. First Bad Version

题目描述: <https://leetcode.com/problems/first-bad-version/>

给定一个版本序列[1,2,3,...,n], 找到第一个不合格的版本, 由于后期版本都是在前期版本基础上改编, 则如果i版本不合格则所有大于i编号的版本都是不合格的。

题目给出一个可调用的API, bool isBadVersion(version)用于判断版本号=version的是否为不合格的

要求在log(n)时间复杂度内求出第一个不合格的版本。

解题思路:

利用二分查找

注意: 一定要考虑到 $(\text{max} + \text{min}) > \text{INT_MAX}$ 的情况

1. 使用long long
2. $\text{mid} = \text{min} + (\text{max} - \text{min}) / 2$ 计算

代码:

```
// Forward declaration of isBadVersion API.
bool isBadVersion(int version);

class Solution {
public:
    int firstBadVersion(int n) {
        long long min = 1, max = n, mid;
        while(min <= max){
            mid = (min+max)/2;
            if(isBadVersion(mid))
                max = mid-1;
            else
                min = mid+1;
        }
        return min;
    }
};
```