278 First Bad Version

2018年4月8日 20:51

Question:

You are a product manager and currently leading a team to develop a new product. Unfortunately, the latest version of your product fails the quality check. Since each version is developed based on the previous version, all the versions after a bad version are also bad.
Suppose you have n versions [1, 2, ..., n] and you want to find out the first bad one, which causes all the following ones to be bad.
You are given an API bool isBadVersion(version) which will return whether version is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API.

来自 <https://leetcode.com/problems/first-bad-version/description/>
你是产品经理，目前正在领导一个团队开发一个新产品。不幸的是，您的产品的最新版本没有通过质量检查。由于每个版本都是基于之前的版本开发的，所以错误版本之后的所有版本都是不好的。
假设你有 n 个版本 [1, 2, ..., n]，你想找出第一个错误的版本，导致下面所有的错误。
你可以通过 bool isBadVersion(version) 的接口来判断版本号 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。您应该尽量减少对 API 的调用次数。

Solution for Python3:

```python
# The isBadVersion API is already defined for you.
# @param version, an integer
# @return a bool
# def isBadVersion(version):

class Solution(object):
    def firstBadVersion(self, n):
        """
        :type n: int
        :rtype: int
        """
        s, e, m = 1, n, 0
        while s < e:
            m = s + (e - s) // 2
            if not isBadVersion(m):
                s = m + 1
            else:
                e = m
        return s
```

Solution for C++:

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

class Solution {
public:
    int firstBadVersion(int n) {
        int s = 1, e = n, m;
```

```
        while (s < e) {
            m = s + (e - s) / 2;
            if (!isBadVersion(m))
                s = m + 1;
            else
                e = m;
        }
        return s;
    }
};
```