

LeetCode

Explore

Problems

Mock

Contest

Articles

Discuss

Store

Premium

Back to Chapter

Week 1: May 1st–May 7th

Problems appear at midnight, Pacific Time

First Bad Version

Jewels and Stones

Ransom Note

Number Complement

First Unique Character in a String

Majority Element

Cousins in Binary Tree

Week 2: May 8th–May 14th

Problems appear at midnight, Pacific Time

Week 3: May 15th–May 21st

The first problem for this section is

Week 4: May 22nd–May 28th

First Bad Version

Solution

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.

Example:

Given `n = 5`, and `version = 4` is the first bad version.

call `isBadVersion(3)` -> false
call `isBadVersion(5)` -> true
call `isBadVersion(4)` -> true

Then 4 is the first bad version.

Java

```
1  /* The isBadVersion API is defined in the parent class VersionControl.
2     boolean isBadVersion(int version); */
3
4  public class Solution extends VersionControl {
5      public int firstBadVersion(int n) {
6          int left=1;
7          int right=n;
8          while(left<right){
9              int mid=left+(right-left)/2;
10             if(isBadVersion(mid)){
11                 right=mid;
12             }
13             else{
14                 left=mid+1;
15             }
16         }
17         return right;
18     }
19 }
```

Custom Testcase (Contribute)

Run Code

Submit