Problem 278: First Bad Version

Problem Information

Difficulty: Easy
Acceptance Rate: 46.48%
Paid Only: No
Tags: Binary Search, Interactive

Problem Description

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 returns whether `version` is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API.

Example 1:

Input: n = 5, bad = 4 **Output:** 4 **Explanation:** call isBadVersion(3) -> false call isBadVersion(5) -> true call isBadVersion(4) -> true Then 4 is the first bad version.

Example 2:

Input: n = 1, bad = 1 **Output:** 1

Constraints:

* `1 <= bad <= n <= 231 - 1`

Code Snippets

C++:

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

class Solution {
public:
int firstBadVersion(int n) {

}
};
```

Java:

```
/* The isBadVersion API is defined in the parent class VersionControl.
boolean isBadVersion(int version); */

public class Solution extends VersionControl {
public int firstBadVersion(int n) {

}
}
```

Python3:

```
# The isBadVersion API is already defined for you.
# def isBadVersion(version: int) -> bool:

class Solution:
def firstBadVersion(self, n: int) -> int:
```