First Bad Version

Try to solve the First Bad Version problem.

We'll cover the following

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

The latest version of a software product fails the quality check. Since each version is developed upon the previous one, all the versions created after a bad version are also considered bad.

Suppose you have `n` versions with the IDs $[1, 2, \ldots, n]$, and you have access to an API function that returns TRUE if the argument is the ID of a bad version.

Find the first bad version that is causing all the later ones to be bad. Additionally, the solution should also return the number of API calls made during the process and should minimize the number of API calls too.

Constraints:

- $1 \leq$ first bad version $\leq$ `n` $\leq 2^{31} - 1$

Examples

1 of 3

Understand the problem

Let’s take a moment to make sure you’ve correctly understood the problem. The quiz below helps you check if you’re solving the correct problem:

First Bad Version

1 What is the output if `n` = 10 and the first bad version is 6?

A) 6, 5

B) 6, 4

C) 6, 3

D) 6, 2

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Initialize `first` to 1 and `last` to `n`.

Calculate `mid` as the mean of 1 and `n` and call the API function with `mid`. Increment the counter for the number of API calls.

If the API function returns TRUE, set `last` to `mid`.

Else, if the API function returns FALSE, set `first` to `mid+1`.

While `first < last`, repeat the steps to adjust `first` and `last`, to recalculate `mid`, and to call the API function.

Return the tuple containing the first bad version and the number of API calls.

Reset Show Solution Submit

Try it yourself

> **Note**: For each test case, the “Expected Output” column in the “Show Results” tab below will list the first bad version as well as the maximum number of calls to the API function that the correct solution will make.

Implement your solution in the following coding playground:

```
public class FBVersion{
   static Api versionApi = new Api();

   public static boolean isBadVersion(int v){
      return versionApi.isBad(v);
   }

   public static int[] firstBadVersion(int n) {
      // -- DO NOT CHANGE THIS SECTION
      versionApi.n = n;
     // --

      // Write your code here
      return new int[]{0, 0};
   }
}
```

Powered by AI Submit

Test Cases Results

Case 1 Case 2 Case 3

Input #1

100

First Bad Version

← Back Next →

☑ Mark as Completed