991. Broken Calculator

[Java/C++] Recursive & Iterative Solution

1.2K
VIEWS

31

Last Edit: 12 hours ago

(/hi-malik) hi-malik (/hi-malik)
7143

How's going Ladies - n - Gentlemen, today we are going to solve another coolest problem i.e.
Broken Calculator

So, what the question is saying is :-

```
=>Given Two Number's startValue & target
=>find Minimum number of operation's to convert startValue to target
```

Now what operation's question is saying to perform:

```
=>multiply the number on display by 2, or
=>subtract 1 from the number on display.
```

But we'll modify the operation's inorder to get our result! We are going to solve our problem in 2 way's **Recursive** & **Iterative**

```
Recursive Approach
```

if `startValue >= target` , then we have to **subtract by 1**
Otherwise,
if it is `even` , then the only way is to **divide it by 2**
If it is `odd` , then there's no way other than to **add 1 to change to even**

Java

```
class Solution {
    public int brokenCalc(int startValue, int target) {
        if(startValue >= target) return startValue - target;
        if(target % 2 == 0){
            return 1 + brokenCalc(startValue, target / 2);
        }
        return 1 + brokenCalc(startValue, target + 1);
    }
}
```

C++

```
class Solution {
public:
    int brokenCalc(int startValue, int target) {
        if(startValue >= target) return startValue - target;
        if(target % 2 == 0){
            return 1 + brokenCalc(startValue, target / 2);
        }
        return 1 + brokenCalc(startValue, target + 1);
    }
};
```

ANALYSIS :-

- **Time Complexity :-** BigO(logN)
- **Space Complexity :-** BigO(1)

`Iterative Approach`

Similar to recursive,
Run the loop until `target` becomes **>** `startValue`
What you have to perform:-
if it is `even` , then the only way is to **divide it by 2**
If it is `odd` , then there's no way other than to **add 1 to change to even**

Let's understand a way more deeper:

We're only allowed to do "Double" and "Decrement" operation to startValue,
In the case of change startValue to target, considering target = startValue*(2^n) + 1, where n can be any given number.
We can double startValue for n + 1 times, then, do decrement for startValue*(2^n) - 1 times, which make startValue = target.
When n get bigger, which means we perform more double operations, the number of required decrement operation would increase exponentially in this case.

On the other hand, if we change target to startValue,
we can do operations to target which are exactly opposite to what we can do to startValue.
That is:

Division: Divide by 2
Increment: Add by 1
In the same case of target = startValue*(2^n) + 1
more division operations we perform, the number of increment operation would decrease exponentially.
Therefore, do as many division operations as we can which would lead to minumum number of operation needed to change target to startValue.

Java

```
class Solution {
    public int brokenCalc(int startValue, int target) {
        int result = 0;
        while(target > startValue){
            if(target % 2 == 0){
                target /= 2;
            }
            else{
                target++;
            }
            result++;
        }
        return result + (startValue - target);
    }
}
```

C++

```
class Solution {
public:
    int brokenCalc(int startValue, int target) {
        int result = 0;
        while(target > startValue){
            if(target % 2 == 0){
                target /= 2;
            }
            else{
                target++;
            }
            result++;
        }
        return result + (startValue - target);
    }
};
```

ANALYSIS :-

- **Time Complexity :-** BigO(logN)
- **Space Complexity :-** BigO(1)

Comments: 5

Sort By

Login to Comment (/accounts/login/?next=/problems/broken-calculator/discuss/1874813/JavaC++-Recursive-and-Iterative-Solution)

VisD566 (/VisD566) ★ 452 7 hours ago

This type of question was asked in a contest few months back.
I remember myself doing it using **dynamic programming**, lol.
It can be solved using DP easily but the **constaints are not allowing us to apply DP**.
Here is the link : 2139. Minimum Moves to Reach Target Score (https://leetcode.com/contest/weekly-contest-276/problems/minimum-moves-to-reach-target-

Read More

4 Share

SHOW 2 REPLIES

aryannnnn (/aryannnnn) ★ 11 5 hours ago

Best Solution!

1 Share

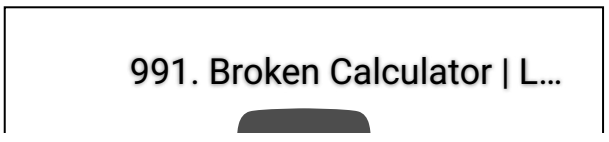
yashshah224 (/yashshah224) ★ 2 7 hours ago

Nice Explanation

1 Share

Vishal_Rajput (/Vishal_Rajput) ★ 260 13 minutes ago

Video solution here: (https://www.youtube.com/watch?v=AcukLKY80bA)



Read More

0 Share

nashvenn (/nashvenn) ★ 13 an hour ago

Clear explanation as usual.
Here's my iterative Python solution: