374 Guess Number Higher or Lower

2018年4月10日 13:42

Question:

We are playing the Guess Game. The game is as follows:
I pick a number from **1** to ***n***. You have to guess which number I picked.
Every time you guess wrong, I'll tell you whether the number is higher or lower.
You call a pre-defined API guess(int num) which returns 3 possible results (-1, 1, or 0):
-1 : My number is lower
1 : My number is higher
0 : Congrats! You got it!
Example:
n = 10, I pick 6.
Return 6.

来自 <https://leetcode.com/problems/guess-number-higher-or-lower/description/>

我们正在玩一个猜数字游戏。 游戏规则如下：
我从 **1** 到 ***n*** 选择一个数字。 你需要猜我选择了哪个数字。
每次你猜错了，我会告诉你这个数字是大了还是小了。
你调用一个预先定义好的接口 guess(int num)，它会返回 3 个可能的结果（-1，1 或 0）：
-1 : 我的数字比较小
1 : 我的数字比较大
0 : 恭喜！你猜对了！
示例:
n = 10, 我选择 6.
返回 6.

Solution for Python3:

```python
# The guess API is already defined for you.
# @param num, your guess
# @return -1 if my number is lower, 1 if my number is higher, otherwise return 0
# def guess(num):

 # Binary Search
# Time complexity : O(log2(n)) Binary Search is used.
# Space complexity : O(1). No extra space is used.
class Solution1(object):
    def guessNumber(self, n):
        """
        :type n: int
        :rtype: int
        """
        low, high= 1, n
        while low <= high:
           mid = low + (high - low) // 2
           val = guess(mid)
           if not val:
               return mid
           elif val < 0:
               high = mid - 1
           else:
               low = mid + 1
        return -1
```

```python

# Ternary Search
# Time complexity : O(log3(n)). Ternary Search is used.
# Space complexity : O(1). No extra space is used.
class Solution2(object):
    def guessNumber(self, n):
        """
        :type n: int
        :rtype: int
        """
        low, high = 1, n
        while low <= high:
           mid1 = low + (high - low) // 3
           mid2 = high - (high - low) // 3
           val1 = guess(mid1)
           val2 = guess(mid2)
           if not val1:
               return mid1
           if not val2:
               return mid2
           elif val1 < 0:
               high = mid1 - 1
           elif val2 > 0:
               low = mid2 + 1
           else:
               low = mid1 + 1
               high = mid2 - 1
        return -1

class Solution3(object):
    def guessNumber(self, n):
        """
        :type n: int
        :rtype: int
        """
        low, high = 1, n
        while low < high:
           mid = low + (high - low) // 2
           if guess(mid) == 1:
               low = mid + 1
           else:
               high = mid
        return low

class Solution4(object):
    def guessNumber(self, n):
        """
        :type n: int
        :rtype: int
        """
        low, high = 1, n
        while low < high:
           mid = low + (high - low) // 2
           low, high = ((mid, mid), (mid + 1, high), (low, mid - 1))[guess(mid)]
        return low
```

Solution for C++:

```cpp
// Forward declaration of guess API.
// @param num, your guess
// @return -1 if my number is lower, 1 if my number is higher, otherwise return 0
int guess(int num);

class Solution {
public:
    int guessNumber(int n) {
        int low = 1, high = n, cur, val;
        while (low <= high) {
            cur = (high - low) / 2 + low;
            val = guess(cur);
            if (!val)
                return cur;
            else if (val == -1)
                high = cur - 1;
            else
                low = cur + 1
        }
        return -1;
    }
};
```