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 	ext{ d} 0) :
-1:我的数字比较小
1:我的数字比较大
0: 恭喜! 你猜对了!
```

Solution for Python3:

示例:

返回 6.

n = 10, 我选择 6.

```
# The guess API is already defined for you.
 1
 2
   # @param num, your guess
 3
    # @return -1 if my number is lower, 1 if my number is higher, otherwise return 0
 4
   # def guess(num):
 5
 6
    # Binary Search
 7
   # Time complexity : O(log2(n)) Binary Search is used.
   # Space complexity : O(1). No extra space is used.
 8
 9
    class Solution1(object):
10
        def guessNumber(self, n):
11
12
            :type n: int
13
            :rtype: int
14
15
            low, high= 1, n
16
            while low <= high:
17
               mid = low + (high - low) // 2
18
               val = guess(mid)
19
               if not val:
20
                   return mid
21
               elif val < 0:
22
                   high = mid - 1
23
               else:
24
                   low = mid + 1
25
            return -1
```

```
26
27
    # Ternary Search
28
    # Time complexity : O(log3(n)). Ternary Search is used.
    # Space complexity : O(1). No extra space is used.
29
30
    class Solution2(object):
31
        def guessNumber(self, n):
32
33
             :type n: int
34
             :rtype: int
35
36
             low, high = 1, n
37
             while low <= high:
                mid1 = low + (high - low) // 3
38
39
                mid2 = high - (high - low) // 3
40
                val1 = guess(mid1)
41
                val2 = guess(mid2)
                if not val1:
42
43
                   return mid1
                if not val2:
44
45
                    return mid2
46
                elif val1 < 0:
47
                    high = mid1 - 1
48
                elif val2 > 0:
49
                    low = mid2 + 1
50
                else:
51
                    low = mid1 + 1
52
                    high = mid2 - 1
53
             return -1
54
55
    class Solution3(object):
56
        def guessNumber(self, n):
57
58
             :type n: int
59
             :rtype: int
             0.00
60
61
             low, high = 1, n
             while low < high:</pre>
62
                mid = low + (high - low) // 2
63
                if guess(mid) == 1:
64
65
                    low = mid + 1
66
                else:
                    high = mid
67
68
             return low
69
70
    class Solution4(object):
71
        def guessNumber(self, n):
72
73
             :type n: int
74
             :rtype: int
75
76
             low, high = 1, n
77
             while low < high:
78
                mid = low + (high - low) // 2
79
                low, high = ((mid, mid), (mid + 1, high), (low, mid - 1))[guess(mid)]
80
             return low
```

Solution for C++:

```
// 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
 3
4
   int guess(int num);
6
   class Solution {
7
    public:
        int guessNumber(int n) {
8
9
            int low = 1, high = n, cur, val;
10
            while (low <= high) {</pre>
                cur = (high - low) / 2 + low;
11
12
                val = guess(cur);
13
                if (!val)
14
                    return cur;
15
                else if (val == -1)
16
                    high = cur - 1;
17
                else
18
                     low = cur + 1
19
            }
20
            return -1;
21
        }
22
   };
```