

## Guess Number Higher or Lower

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`.

## Solution 1

-1 : My number is lower

1 : My number is higher

0 : Congrats! You got it!

Here "My" means the number which is given for you to guess not the number you put into **guess(int num)**.

written by [Nakanu](#) original link [here](#)

## Solution 2

Using binary search to find the smallest number that's not too small.

```
def guessNumber(self, n):  
    class C: __getitem__ = lambda _, i: -guess(i)  
    return bisect.bisect(C(), -1, 1, n)
```

Alternatively, without using the library:

```
def guessNumber(self, n):  
    lo, hi = 1, n  
    while lo < hi:  
        mid = (lo + hi) / 2  
        if guess(mid) == 1:  
            lo = mid + 1  
        else:  
            hi = mid  
    return lo
```

Funny variation:

```
def guessNumber(self, n):  
    lo, hi = 1, n  
    while lo < hi:  
        mid = (lo + hi) / 2  
        lo, hi = ((mid, mid), (mid+1, hi), (lo, mid-1))[guess(mid)]  
    return lo
```

written by [StefanPochmann](#) original link [here](#)

## Solution 3

This problem is a binary search problem has an  $O(\log n)$  complexity.

```
public int guessNumber(int n) {  
    int i = 1, j = n;  
    while(i < j) {  
        int mid = i + (j - i) / 2;  
        if(guess(mid) == 0) {  
            return mid;  
        } else if(guess(mid) == 1) {  
            i = mid + 1;  
        } else {  
            j = mid;  
        }  
    }  
    return i;  
}
```

written by [yfcheng](#) original link [here](#)

From [LeetCoder](#).