/*Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You may assume no duplicates in the array.

Example 1:

Input: [1,3,5,6], 5 Output: 2

Example 2:

Input: [1,3,5,6], 2 Output: 1 Example 3:

Input: [1,3,5,6], 7 Output: 4

Example 1:

Input: [1,3,5,6], 0 Output: 0*/

-
- 思想:
- (1) 二分查找思想，设置start和end，令 mid = (start+end)/2，比较输入值和 nums[mid]的大小，移动start和end，当start > end时循环终止

```
public int searchInsert(int[] nums, int target) {
    int i,l,r,mid;
    int len = nums.length;
    l=0;
    r=len-1;
    while(l<=r){
        mid = l + (r-l)/2;
        if(nums[mid] == target)
            return mid;
        if(nums[mid] < target){
            l = mid + 1;
        }
        else
            r = mid - 1;
    }
    return l;
}
```