35. Search Insert Position

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
```

```
class Solution {
    public int searchInsert(int[] nums, int target) {
        // binary search O(log(n))

        if (nums == null || nums.length == 0)
            return 0;

        int start = 0;
        int end = nums.length-1;

        while(start + 1 < end) {
            int mid = start + (end - start)/2;
            if (nums[mid] == target) {
                return mid;
            } else if (nums[mid] < target) {
                start = mid;
            } else {
                end = mid;
            }
        }

        if (nums[start] >= target) {
            return start;
        } else if (nums[end] >= target) {
            return end;
        } else {
            return end + 1;
        }
    }
}
```