035. Search Insert Position

035 Search Insert Position

- **Binary Search**+array

Description

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
```

1. Thought line

2. Binary Search+array

```
class Solution {
private:
    void binarySearchInsert(vector<int>& nums, int target, int st, int ed, int& res){
        if (st>ed) return;
        if (st==ed){
            if (nums[st]<target) res  = st+1;
            if (st==ed && nums[st]>target) res = st;
            if (st==ed && nums[st]==target) res = st;
            return;
        }
        else{
            int mid = (st+ed)/2;
            if (target<=nums[mid])
                binarySearchInsert(nums, target, st, mid, res);
```

```
            else
                binarySearchInsert(nums, target, mid+1, ed, res);
        }
    }

public:
    int searchInsert(vector<int>& nums, int target) {
        int res = 0;
        binarySearchInsert(nums, target, 0, nums.size()-1, res);
        return res;
    }
};
```