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

```
1 class Solution {
       void\ binary Search Insert (vector < int > \&\ nums,\ int\ target,\ int\ st,\ int\ ed,\ int \&\ res) \{
           if (st==ed){
 6
           if (nums[st]<target) res = st+1;</pre>
 7
               if (st==ed && nums[st]>target) res = st;
               if (st==ed && nums[st]==target) res = st;
 8
 9
               return;
10
11
           else{
12
           int mid = (st+ed)/2;
               if (target<=nums[mid])</pre>
13
14
                   binarySearchInsert(nums, target, st, mid, res);
```

```
15
               else
 16
                   binarySearchInsert(nums, target, mid+1, ed, res);
 17
 18
 19
 20 public:
 21
       int searchInsert(vector<int>& nums, int target) {
 22
           int res = 0;
 23
           binarySearchInsert(nums, target, 0, nums.size()-1, res);
24
25
           return res;
26 };
```