/\*

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.

Here are few examples.

[1,3,5,6], 5 → 2

[1,3,5,6], 2 → 1

[1,3,5,6], 7 → 4

[1,3,5,6], 0 → 0

way-1 顺序查找

way-2 二分查找

\*/

class Solution {

public:

int searchInsert(vector<int>& nums, int target)

{

//way-1

/\*

int k=0;

while(true)

{

if(nums[k]<target && k!=nums.size()-1)

k++;

else if(nums[k]<target && k==nums.size()-1)

return nums.size();

else if(nums[k]==target)

return k;

else if(nums[k]>target && k!=0)

return k;

else

return 0;

}

return 0;

\*/

//way-2

int l=0;

int r=nums.size()-1;

int mid;

while(l<=r)

{

if(nums[l]>target)

return l;

else if(nums[r]<target)

return r+1;

mid=l+(r-l)/2;

if(nums[mid]==target)

return mid;

else if(nums[mid]<target)

l=mid+1;

else

r=mid-1;

}

}

};