/\*

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

思路：二分查找，找中间点和两边的大小关系

\*/

class Solution {

public:

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

{

//二分查找

int left=0;

int right=nums.size()-1;

int mid;

while(left<=right)

{

mid=(left+right)/2;

cout<<left<<" "<<mid<<" "<<right<<endl;

if(nums[mid]==target)

return mid;

else if(nums[left]<=nums[mid])

{

if(nums[left]<=target && target<=nums[mid])

right=mid-1;

else

left=mid+1;

}

else

{

if(nums[left]<=target || target<=nums[mid])

right=mid-1;

else

left=mid+1;

}

}

return -1;

}

};