/\*

Follow up for "Find Minimum in Rotated Sorted Array":

What if duplicates are allowed?

Would this affect the run-time complexity? How and why?

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).

Find the minimum element.

The array may contain duplicates.

way-1:去除重复

way-2:vector<int>::itearator it=min\_element(nums.begin(),nums.end());

way-3:二分查找，有点诀

\*/

class Solution {

public:

int findMin(vector<int>& nums)

{

//way-2

/\*

vector<int>::iterator it=min\_element(nums.begin(),nums.end());

return nums[it-nums.begin()];

\*/

//way-3

int left=0;

int right=nums.size()-1;

while(left<right && nums[left]>=nums[right]) //如果左边比右边小，说明局部已经排序完成

{

int mid=(left+right)/2;

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

if(nums[mid]>nums[right])

left=mid+1;

else if(nums[mid]==nums[right])

left++;

else

right=mid;

}

return nums[left];

}

};