Rotation in binary search

Ex – 1 2 3 4 5 6 7 8 9

1)Now it Is rotated - 5 6 7 8 9 1 2 3 4

2)Now we have to find 1 (to index number of rotations – find the index of min element)

3)Now observe min element(1 in this case ) is less than 9 and 2 . it is alays less than its left and right elements.

4)SO new check this condition by next=(mid+1)%n ,prev=(mid +n-1)%n

Because the minimum element may present at the edge cases (at last)

5)Also if nums[start]<=nums[end] , then is sorted then we return nums[l]

class Solution {

public int findMin(int[] nums)

{

int l=0;

int h=nums.length-1;

int n=nums.length;

while(l<=h)

{

int mid= l + (h-l)/2;

int next=(mid+1)%n;

int pre=(mid+n-1)%n;

if(nums[mid]<=nums[next] && nums[mid]<=nums[pre])

{

return nums[mid];

}

if(nums[l]<=nums[h])

{

return nums[l];

}

if(nums[l]<=nums[mid])

{

l=mid+1;

}

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

{

h=mid-1;

}

}

return -1;

}

}