//T**he number of times you can halve the array before you run out of elements is //proportional to log₂(n), where n is the number of elements in the array.**  
  
  
  
findtTarget(nums,target)

{1

//find the pivot of the array  
let pivot=Math.floor(nums.length/2)

//if the nums[pivot]= target return the pivot

if(nums[pivot]==target)

return pivot

//else if nums[pivot]>target that means the target lies in the left of array

//till pivot index--- we need to divide the 0to pivot new (0+pivot)/2

else if(nums[pivot]>target)

{6

console.log(‘nums[pivot]>target’,nums[pivot],target)

return findTarget(getarray(0,pivot),target)

6}

//this will continue till we get the pivot as target or no element matches

//the left array RECURSIVE

//else if nums[pivot]>target that means the target lies in the right of array

//till pivot index++ we need to divide the pivot to //nums.length-1

else if(nums[pivot]>target)

{7

console.log(‘nums[pivot]>target’,nums[pivot],target)

return findTarget(getarray(nums.length-1,pivot),target)

7}

//new (pivot+nums.length-1)/2  
//this will continue till we get the pivot as target or no element matches

//the right array RECURSIVE

else

{8

console.log(‘ Anynums[pivot]!=target’ )

return -1

}8

1}  
  
  
  
getarray(a,pivot,nums)

{2 let j=0

for(let i=a;i<pivot;i++)

{

nums[i]=nums[j]

j++

}

return nums

2}  
console.log(findtTarget([4,5,6,7,0,1,2],0)) ;