Data structures and Algo in Java - Day 15

Entered Arrays - Medium Level Questions

Solved 3 medium level questions in array which are given below

1. 2 Sum
2. Sort 0’s 1’s 2’s - Used Dutch National Flag Algorithm

* It uses 3 pointers
* It has 3 rules:-

1. Everything from 0 to low-1 is 0
2. Everything from low to mid-1 is 1
3. Everything from high+1 to n-1 is 2

Dry run:-

if arr[mid] == 0 -> swap arr low and arr mid

then increment low and increment mid

if arr[mid] == 1 -> then increment mid

if arr[mid] == 2 -> swap arr mid and arr high then decrement high

3 Find the Majority Element - Uses Moore’s voting Algorithm

It has a counter and an element

if the next element is the same the counter increases , if the next element is not the same the counter decreases and if the counter hits 0 the element gets updated to the next element.

finally a second counter is taken to check how many remaining elements are there in the array , since increasing and decreasing the counter will equalize the array , the remaining numbers are counted in the last

if the second counter is higher than N/2 then the element is the Majority element , if its not then there is no Majority Element.

Here’s the solution for all the 3 questions

1.

class Solution {

public int[] twoSum(int[] arr, int target)

{

int n = arr.length;

HashMap<Integer,Integer> map = new HashMap<>();

for(int i=0;i<n;i++)

{

int sum = target - arr[i];

if(map.containsKey(sum))

{

return new int [] {map.get(sum),i};

}

map.put(arr[i],i);

}

return new int []{};

}

}

2.

class Solution {

public void sortColors(int[] nums)

{

int low = 0;

int mid = 0;

int n = nums.length;

int high = n-1;

while (mid <= high)

{

if(nums[mid]==0)

{

int temp = nums[low];

nums[low] = nums[mid];

nums[mid] = temp;

low++;

mid++;

}

else if (nums[mid]==1)

{

mid++;

}

else

{

int temp = nums[mid];

nums[mid] = nums[high];

nums[high] = temp;

high--;

}

}

}

}

3.

class Solution {

public int majorityElement(int[] nums)

{

int n=nums.length;

int ctr1=0;

int ele=0;

for(int i=0;i<n;i++)

{

if(ctr1==0)

{

ctr1++;

ele=nums[i];

}

else if(nums[i]==ele)

{

ctr1++;

}

else{

ctr1--;

}

}

int ctr2=0;

for(int i=0;i<n;i++)

{

if(nums[i]==ele)

{

ctr2++;

}

}

if(ctr2>n/2)

{

return ele;

}

return -1;

}

}