

Day- 8

08-09-2021

level - medium.

P1 Sort colors.

Given an array nums with m objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white or blue.
Use `int i, j, k` without using sorting library.

Input: `nums = [2, 0, 2, 1, 1, 0]`

Output: `[0, 0, 1, 1, 2, 2]`

// Brute force.

// using loop `num[i] > num[i+1]`

swap (`num[i], num[i+1]`)

`num[i] < num[i+1]`

swap (`num[i], num[i+1]`)

`num[i] == num[i+1]`

`i++;`

} }
} Point to remember -

Use DNF for optimal (DNF - Dutch National flag)

Alg - • Take three-pointers, namely - low, mid, high
• we use low and mid pointers at the start, and the high pointer will point at the end of the given array.

cases - if $\text{arr}[\text{mid}] = 0$, swap ($\text{arr}[\text{mid}]$, $\text{arr}[\text{high}]$)
if $\text{arr}[\text{mid}] = 1$, noswap, $\text{mid}++$
if $\text{arr}[\text{mid}] = 2$, swap ($\text{arr}[\text{mid}]$, $\text{arr}[\text{high}]$);

```
int low = 0;  
int mid = 0;  
int high = nums.size() - 1;  
while (mid <= high)  
{  
    if (nums[mid] == 0)  
        swap (nums[low], nums[mid]);  
    low++;  
    mid++;  
}  
else if (nums[mid] == 1)  
{  
    mid++;  
}  
else  
{  
    swap (nums[high], nums[mid]);  
    high--;  
}
```

level - medium

Find longest consecutive subsequence O(n) time.

Input: $\text{num} = [100, 9, 200, 1, 3, 2]$

Output: $9 \rightarrow \{1, 2, 3, 4\} \rightarrow l = 4$
~~log 200 + 15/8 = 22~~

Brute force (N log N + N)

```
// sort the array  
// count = 1, maxx = 1  
// if ( $\text{num}[i] == \text{num}[i+1]$ )  
    {  
        count++;  
        maxx = max(maxx, count);  
    }  
else if ( $\text{num}[i] == \text{num}[i+1]$ )  
    continue;  
else  
    count = 1;
```

optional (O(n))

```
int longestConsecutive(vector<int> &num) {
```

```
// by hashing  
// create a hash set  
set<int> hashSet;  
// iterate over num's array  
for (int num : num)  
{  
    // insert num in hashSet
```

```
hashSet.insert(num);  
}
```

```
int ls = 0; // longer streak  
for (int num : nums)
```

```
{ if (!hashSet.count(num - 1)) // check the num like  
too, so gg is exist or  
not.  
}
```

```
int currNum = num;
```

```
int currStreak = 1;
```

```
while (hashSet.count(currNum + 1)) // check for too,  
so too is exist or  
not.
```

```
currNum += 1;
```

```
currStreak += 1;
```

```
ls = max(ls, currStreak);
```

```
}
```

```
return ls;
```

```
}
```