Day-9

09-01-2021

level-Easy

Q1 Single Number (every element appears twice except for one. find that single one.

Input: nums = [2,2,1]
output: 1

// Brute force.
// using loop in if condition check num[i] != num[i+1]
return num[i], else i++.

Optimal (by Bit manipulation)

```
int singleNumber(vector<int>& nums) {
    int n = nums.size();
    int a=0;
    for(int i=0; i<n; i++)
    {
        a = a^nums[i];
    }
    return a;
}
```

level medium

Q-2 Remove duplicate from sorted array - I
// Unique element appears at most twice.

input: nums = [1,1,1,2,2,3]
Output: 5, nums = [1,1,2,2,3]

```
// take counter variable.
int cnt = 1;
// store size in variable n
int n = num.size();

int size = 0;
if (n == 0)
    return 0;
for (int i = 1; i < n; i++)
{
    if (num[i] != num[size])
    {
        size++;
        num[size] = num[i];
        cnt = 1;
    }
    else if (cnt < 2) {
        size++;
        num[size] = num[i];
        cnt++;
    }
}
return size + 1;
}
```

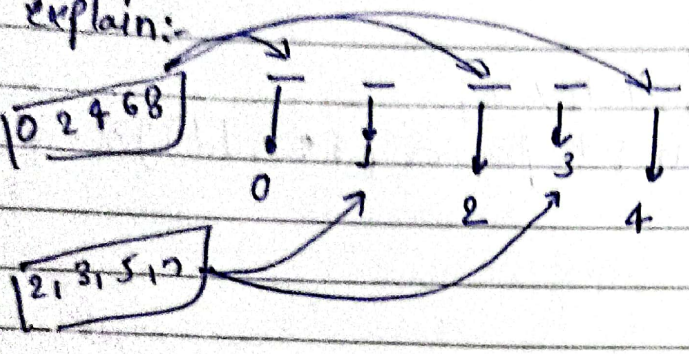
level - medium (Recursion)

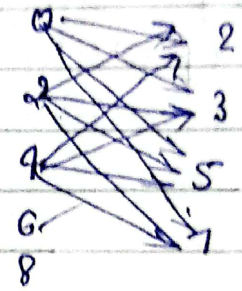
Q.1 Count good numbers (A digit string is good if the digit (0-indexed) at even indices are even and the digits at odd indices are prime (2, 3, 5 or 7)

Input: n = 1
output: 5

| even = 0, 2, 4, 6, 8 {0-9}
| prime = 2, 3, 5, 7

explain:-





even $5c_1$
odd $4c_1$

logic: ans = ans * 5 ; if (i % 2 == 0)
ans = ans * 4 ; if (i % 2 == 1)
ans = ans % mod
↳ range ($10^9 + 7$)

// optimal soln

// fn create

```
long long power(long long x, long long y) // return x^y
{
    if (y == 0)
        return 1;
    long long ans = power(x, y/2);
    ans = ans * ans;
    ans = ans % mod;
```