

SORT COLORS;

-> <https://leetcode.com/problems/sort-colors/submissions/>

APPROACH 1->COUNTING APPROACH;

1=> MAKE ONE ITERATION ->TAKE THREE VAR FOR 1COUNT ,2COUNT AND 3COUNT;

2=> COUNT ALL THE OCCURENCE OF 1,2,3;

3=> MAINTAIN AN IDX FOR INSERTING POS;

4=> WHILE(0 COUNT IS NOT ZERO)->INSERT 0 IN ARRray and idx++ and zero--;

5=> WHILE(1 COUNT IS NOT ZERO)->INSERT 1 IN ARRray and idx++ and ONE--;

6=> WHILE(2 COUNT IS NOT ZERO)->INSERT 2 IN ARRray and idx++ and ONE--;

APPROACH ->2-> TWO POINTERS;

ASLI POSITION KYA HAI APPROCH TO 0 KA 1 PHEL POSITION KAHA HOGA OVISULY JO PHELA ZERO MILNA CHIYE WO ZERO TH PLACE PE RAKH JAYENGA AND SECONDE SERON SECOND IDX PE SAME HI JO PAHLE 2 HOGA WO KAHA RHEGA LAST IDX PE THEN SECOND 2 SECOND LASTT PE

1=> TAKE TWO POINTERS THAT WILL SHOW ON WHICH IDX WE HAVE TO INSERT ZERO AND ONE;

2=> RUN A LOOP TILL SIZE-1

3=> IF ARR[I] == 0 SWAP WITH S AND S++ AND I++;

**\*\* edge case=> while swpping we also have to check I<=e =>other wise we may swap the elemt that is already at its correct position;**

**Why equal to => becoz we are taking the last index as the first correct place of 2 and if there alrady exist 2 and no swap occurs then e will not decremnt and hence it will be error;**

4=> IF ARR[I] == 2 SWAP WITH E AND E--

**\*\*EDGE CASE=> WHILE SWAPING WITH E WE DO NOT KNOW WHICH ELEMENT IS COMING SO WE HAVE TO CHECK AGAIN THE SAME IDEX SO DO NOT I++;**

5=> done;

```
i C++ | Auto
class Solution {
public:
    void sortColors(vector<int>& nums) {
        int s=0;
        int e =nums.size()-1;

        for(int i=0;i<nums.size() ;){
            if(nums[i]==0 ){
                cout<<nums[i]<<nums[s]<<endl;
                swap(nums[i],nums[s]);

                s++;
                i++;

            }else if(nums[i]==2 && i<=e){
                cout<<nums[i]<<nums[s]<<endl;
                swap(nums[i],nums[e]);
                e--;

            }else{
                i++;
            }
        }
    }
};
```

Saved to local

Ln 27, Col 3