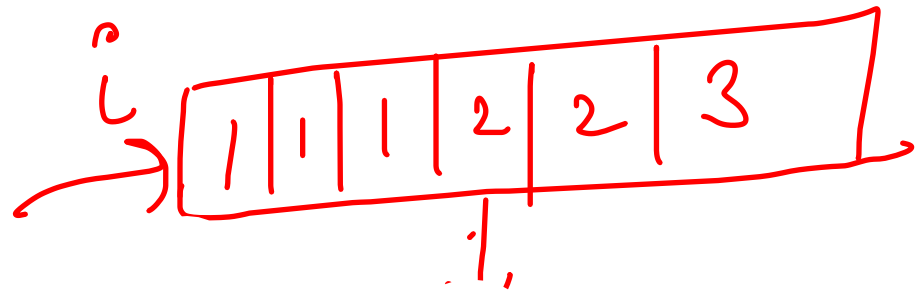
HW_De-Duplication

1 → 3 (1 more)

2 → 2 ✓

3 → 1 ✓

```
6
1 1 1 2 2 3
```



n = 6 - 1 ⇒ 5

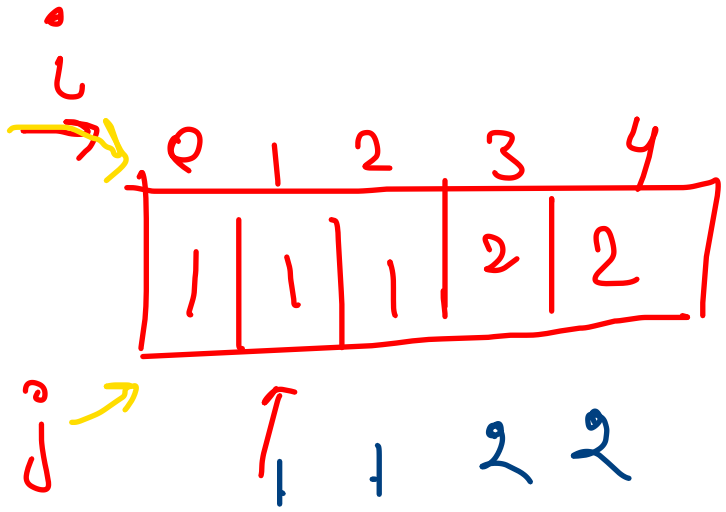
TC → O(n)

SC → O(1)

```java
public static int remove(int arr[], int n){
    if(n <= 2){
        return n;
    }

    int j = 1;
    int count = 1;
    for(int i = 1; i < n; i++){
        if(arr[i] == arr[i-1]){
            count++;
        }
        else{
            count = 1;
        }

        if(count <= 2){
            arr[j++] = arr[i];
        }
    }
    return j;
}
```



n = 5

count = 1 2 3

i = 1

1 == 1 T

3 <= 2 X

i = 2

1 == 1 T

1 <= 2

2 <= 2 T

i = 3

2 == 1 → count = 1 2

i = 4

2 == 2

j = 4

```
8
1 0 2 3 0 4 5 0
0 1 2 3 4 5 6 7
```

count = 3 8+3

index	value	Action	mod. array.	zeros left
		skip (no space)	1 0 2 3 0 4 5 0	2
7	0			
6	5	copy 5 to arr[6+2] (ignore)	1 0 2 3 0 4 5 0	2
5	4	copy 4 to arr[5+2] arr[7]	1 0 2 3 0 4 5 4	2
4	0	copy 0 to arr[4+2] arr[6]	1 0 2 3 0 4 0 4	1
		Duplicate 0 to arr[4+1] = arr[5]	1 0 2 3 0 0 0 4	1

index	value	Action	Array	0 left
3	3	copy 3 to arr[3+1] = arr[4]	1 0 2 3 3 0 0 4	1
2	2	copy 2 to arr[2+1] = arr[3]	1 0 2 2 3 0 0 4	1
1	0	copy 0 to arr[1+1] = arr[2]	1 0 0 2 3 0 0 4	0
		Duplicate 0 to arr[1+0] = arr[1]	1 0 0 2 3 0 0 4	0
0	1	copy 1 to arr[0+0] → arr[0]	1 0 0 2 3 0 0 4	0

o/p → [1, 0, 0, 2, 3, 0, 0, 4]

```java
public static void duplicateZero(int arr[], int n){

    int zeros = 0;

    // count the no. of zeros
    for(int i = 0; i < n; i++){
        if(arr[i] == 0){
            zeros++;
        }
    }
    // traverse the array from last and shift the elem

    for(int i = n-1; i >= 0; i--){
        if(i + zeros < n){
            arr[i + zeros] = arr[i];
        }
        if(arr[i] == 0){
            zeros--;
            if(i + zeros < n){
                arr[i + zeros] = 0;
            }
        }
    }
}
```

n

n

TC → O(n)

SC → O(1)