Moves zeros

283. Move Zeroes

Easy 14114 360 Add to List Share

Given an integer array `nums`, move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

Note that you must do this in-place without making a copy of the array.

Example 1:

```
Input: nums = [0,1,0,3,12]
Output: [1,3,12,0,0]
```

Example 2:

```
Input: nums = [0]
Output: [0]
```

Constraints:

- `1 <= nums.length <= 10^4`
- `-2^31 <= nums[i] <= 2^31 - 1`

Brute force

In Brute force approach we will use extra

1st loop
→ Create an vector and copy all Non Zero number in temp vector

2nd loop
→ Now store all zero in temp vector

Hence we will get required output.

```
class Solution {
public:
    void moveZeroes(vector<int>& v) {
     vector<int> temp;
        for(int i=0;i<v.size();i++){
            if(v[i]!=0) temp.push_back(v[i]);
        }
        for(int i=0;i<v.size();i++){
            if(v[i]==0) temp.push_back(v[i]);
        }
        int tempLen=temp.size();
        for(int i=0;i<tempLen;i++){
            v[i]=temp[i];
        }
    }
};
```

T.C - O(3n)
S.C → O(n)

Optimal Approach: 2 pointer Swap

[0, 1, 0, 3, 12]
 ↑ ↑
 i j

loop (i = 0 → n)

j will be responsible for keeping track of 0

1st Iteration

[0, ...] → Find Non-Zero element using i

i, j

if (v[i] != 0)

swap(v[i], v[j])

j++;

2nd [0, 1] swap → [1, 0, 0, 3, 1, 2]

j i → j i

3rd i++ [1, 0, 0, 3, 1, 2]

j i

4th i++ [1, 0, 0, 3, 1, 2]

j i

swap(0,3)

[1, 3, 0, 0, 1, 2]

j i

i++

5th [1, 3, 0, 0, 1, 2]

j i

swap(0,1)

i++ [1, 3, 1, 0, 0, 2]

j i

[1, 3, 1, 0, 0, 2] swap

j i

[1, 3, 1, 2, 0, 0]

```cpp
class Solution {
public:
    void moveZeroes(vector<int>& v) {
        int j=0;
        for(int i=0;i<v.size();i++){
            if(v[i]!=0){
                swap(v[i],v[j]);
                j++;
            }
        }
    }
};
```

T.C - O(n)

S.C - O(1)