

## 75. Sort Colors

```
class Solution {
public:
    void sortColors(vector<int>& nums) {
        int low = 0, mid = 0, 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 { // nums[mid] == 2
                swap(nums[mid], nums[high]);
                high--;
            }
        }
    }
};
```

**Accepted** Runtime: 0 ms

• Case 1

• Case 2

Input

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

Output

```
[0,0,1,1,2,2]
```

Expected

```
[0,0,1,1,2,2]
```

### 215. Kth Largest Element in an Array

```
class Solution {  
public:  
    int findKthLargest(vector<int>& nums, int k) {  
        sort(nums.begin(), nums.end()); // O(N log N)  
        return nums[nums.size() - k]; // Kth largest element  
    }  
  
};
```

Accepted Runtime: 0 ms

- Case 1
- Case 2

Input

nums =  
[3,2,1,5,6,4]

k =  
2

Output

5

Expected

5