

Data Structure Lab

Assignment 1

Submitted by:

Navdeep Singh

7th August 2025

Roll No: 24124073

Group: 3

Branch: Information Technology

Year: 2nd Year

- 1 Given an array `nums` with `n` objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers: 0 to represent red 1 to represent white 2 to represent blue

Approach 1 Using Bubble Sort

Code

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 void bubbleSort(vector<int>&arr){
5     int n = arr.size();
6     for(int i=0; i<n-1; i++){
7         for(int j=0; j<n-i-1; j++){
8             if(arr[j] > arr[j+1]){
9                 swap(arr[j], arr[j+1]);
10            }
11        }
12    }
13 }
```

Approach 2 Using Selection Sort

Code

```
1 void selectionSort(vector<int>&arr){
2     int n = arr.size();
3     for(int i=0; i<n-1; i++){
4         int minIndex = i;
5         for(int j=i+1; j<n; j++){
6             if(arr[j] < arr[minIndex]) minIndex = j;
7         }
8         swap(arr[i], arr[minIndex]);
9     }
10 }
```

Approach 3 Using Insertion Sort

Code

```
1 void insertionSort(vector<int>&arr){
2     int n = arr.size();
3     for(int i=1; i<n; i++){
4         int min = arr[i];
5         int prev = i - 1;
6         while(prev >= 0 && arr[prev] > min){
7             arr[prev + 1] = arr[prev];
8             prev--;
9         }
10     }
```

```
10     arr[prev + 1] = min;
11 }
12 }
```

Approach 4 Using count Variables and in place array replacement

Code

```
1 // sort array of 0s 1s 2s
2 // basic approach count no of 0s 1s 2s basically and add in arr
3 // good approach 0n is DNF 3 pointer
4
5 void countAndArrangeInArray(vector<int>& arr) {
6     int cntZero = 0, cntOne = 0, cntTwo = 0;
7     for (int i : arr) {
8         if (i == 0) cntZero++;
9         else if (i == 1) cntOne++;
10        else if (i == 2) cntTwo++;
11    }
12    int i = 0;
13    while (cntZero-->0) arr[i++] = 0;
14    while (cntOne-->0) arr[i++] = 1;
15    while (cntTwo-->0) arr[i++] = 2;
16 }
```

Approach 5 Using Dutch National Flag Algorithm or 3 pointer approach

Code

```
1 // sort array of 0s 1s 2s
2 // basic approach count no of 0s 1s 2s basically and add in arr
3 // good approach 0n is DNF 3 pointer
4
5 void countAndArrangeInArray(vector<int>& arr) {
6     int cntZero = 0, cntOne = 0, cntTwo = 0;
7     for (int i : arr) {
8         if (i == 0) cntZero++;
9         else if (i == 1) cntOne++;
10        else if (i == 2) cntTwo++;
11    }
12    int i = 0;
13    while (cntZero-->0) arr[i++] = 0;
14    while (cntOne-->0) arr[i++] = 1;
15    while (cntTwo-->0) arr[i++] = 2;
16 }
```

Approach 6 Using Merge Sort log n complexity in best or worst case also

Code

```

1 vector<int>mergeTwoSortedArray(vector<int>&arr1,vector<int>&arr2){
2     int i = 0;
3     int j = 0;
4     int k = 0;
5     int size1 = arr1.size();
6     int size2 = arr2.size();
7     vector<int>arr3(size1+size2);
8     while(i<size1 && j<size2){
9         if(arr1[i] < arr2[j]) arr3[k++] = arr1[i++];
10        else arr3[k++] = arr2[j++];
11    }
12    while(i<size1) arr3[k++] = arr1[i++];
13    while(j<size2) arr3[k++] = arr2[j++];
14    return arr3;
15 }
16
17 vector<int> mergeSort(vector<int>&arr,int start,int end){
18     int mid = (start+end)/2;
19     if(start<=end) return {arr[start]};
20     vector<int>arr1 = mergeSort(arr,start,mid);
21     vector<int>arr2 = mergeSort(arr,mid+1,end);
22     return mergeTwoSortedArray(arr1,arr2);
23 }

```

Calling Main Function

Code

```

1 int main(){
2     int n;
3     cout<<"Enter the nuneber of the objects : ";
4     cin>>n;
5     vector<int>arr(n);
6     cout<<"Enter the objects color 0(Red) , 1(White) and 2(Blue) only : \n";
7     for(int i=0; i<n; i++){
8         cout<<"Enter " <<i+1 <<" th object out of "<<n <<" : ";
9         cin>>arr[i];
10        if(arr[i] < 0 || arr[i] > 2){
11            cout<<"Invalid input! Please enter 0, 1, or 2 only.\n";
12            return 1;
13        }
14    }
15    bubbleSort(arr);
16    selectionSort(arr);
17    insertionSort(arr);
18    countAndArrangeInArray(arr);
19    dnfAlgo(arr);
20    vector<int>mergeSortAns = mergeSort(arr,0,arr.size()-1);
21    cout<<"Your sorted array of object is : \n";
22    for(auto i : arr) cout<<i <<" ";
23 }

```