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

```cpp
#include <bits/stdc++.h>
using namespace std;

void bubbleSort(vector<int>&arr){
    int n = arr.size();
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-i-1; j++){
            if(arr[j] > arr[j+1]){
                swap(arr[j], arr[j+1]);
            }
        }
    }
}
```

Approach 2 Using Selection Sort

Code

```cpp
void selectionSort(vector<int>&arr){
    int n = arr.size();
    for(int i=0; i<n-1; i++){
        int minIndex = i;
        for(int j=i+1; j<n; j++){
            if(arr[j] < arr[minIndex]) minIndex = j;
        }
        swap(arr[i], arr[minIndex]);
    }
}
```

Approach 3 Using Insertion Sort

Code

```cpp
void insertionSort(vector<int>&arr){
    int n = arr.size();
    for(int i=1; i<n; i++){
        int min = arr[i];
        int prev = i - 1;
        while(prev >= 0 && arr[prev] > min){
            arr[prev + 1] = arr[prev];
            prev--;
        }
```

```
        arr[prev + 1] = min;
    }
}
```

Approach 4 Using count Variables and in place array replacement

Code

```
// sort array of 0s 1s 2s
// basic approach count no of 0s 1s 2s basically and add in arr
// good approach 0n is DNF 3 pointer

void countAndArrangeInArray(vector<int>& arr) {
    int cntZero = 0, cntOne = 0, cntTwo = 0;
    for (int i : arr) {
        if (i == 0) cntZero++;
        else if (i == 1) cntOne++;
        else if (i == 2) cntTwo++;
    }
    int i = 0;
    while (cntZero--) arr[i++] = 0;
    while (cntOne--) arr[i++] = 1;
    while (cntTwo--) arr[i++] = 2;
}
```

Approach 5 Using Dutch National Flag Algorithm or 3 pointer approach

Code

```
// sort array of 0s 1s 2s
// basic approach count no of 0s 1s 2s basically and add in arr
// good approach 0n is DNF 3 pointer

void countAndArrangeInArray(vector<int>& arr) {
    int cntZero = 0, cntOne = 0, cntTwo = 0;
    for (int i : arr) {
        if (i == 0) cntZero++;
        else if (i == 1) cntOne++;
        else if (i == 2) cntTwo++;
    }
    int i = 0;
    while (cntZero--) arr[i++] = 0;
    while (cntOne--) arr[i++] = 1;
    while (cntTwo--) arr[i++] = 2;
}
```

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

Code

```cpp
vector<int>mergeTwoSortedArray(vector<int>&arr1,vector<int>&arr2){
    int i = 0;
    int j = 0;
    int k = 0;
    int size1 = arr1.size();
    int size2 = arr2.size();
    vector<int>arr3(size1+size2);
    while(i<size1 && j<size2){
        if(arr1[i] < arr2[j]) arr3[k++] = arr1[i++];
        else arr3[k++] = arr2[j++];
    }
    while(i<size1) arr3[k++] = arr1[i++];
    while(j<size2) arr3[k++] = arr2[j++];
    return arr3;
}

vector<int> mergeSort(vector<int>&arr,int start,int end){
    int mid = (start+end)/2;
    if(start<=end) return {arr[start]};
    vector<int>arr1 = mergeSort(arr,start,mid);
    vector<int>arr2 = mergeSort(arr,mid+1,end);
    return mergeTwoSortedArray(arr1,arr2);
}
```

Calling Main Function

Code

```cpp
int main(){
    int n;
    cout<<"Enter the numeber of the objects : ";
    cin>>n;
    vector<int>arr(n);
    cout<<"Enter the objects color 0(Red) , 1(White) and 2(Blue) only :
        \n";
    for(int i=0; i<n; i++){
        cout<<"Enter " <<i+1 <<" th object out of "<<n <<" : ";
        cin>>arr[i];
        if(arr[i] < 0 || arr[i] > 2){
            cout<<"Invalid input! Please enter 0, 1, or 2 only.\n";
            return 1;
        }
    }
    bubbleSort(arr);
    selectionSort(arr);
    insertionSort(arr);
    countAndArrangeInArray(arr);
    dnfAlgo(arr);
    vector<int>mergeSortAns = mergeSort(arr,0,arr.size()-1);
    cout<<"Your sorted array of object is : \n";
    for(auto i : arr) cout<<i <<" ";
}
```

Sample Output (Here i use pass by refrence so it gives same output through all approaches :)

```
Enter the numeber of the objects : 6
Enter the objects color 0(Red) , 1(White) and 2(Blue) only :
Enter 1 th object out of 6 : 0
Enter 2 th object out of 6 : 1
Enter 3 th object out of 6 : 2
Enter 4 th object out of 6 : 2
Enter 5 th object out of 6 : 0
Enter 6 th object out of 6 : 1
Your sorted array of object is :
0 0 1 1 2 2
```