

Assignment No. 2

Name – Manish Namdev Barage

PRN – 22520007

Batch – T7

Problem Statement – To Implement (Quick Sort / Merge Sort) Sorting algorithms using array as a data structure

Code – Merge Sort

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

void merge(vector<int> &arr, int low, int mid, int high) {
    vector<int> temp;
    int left = low;
    int right = mid + 1;

    while (left <= mid && right <= high) {
        if (arr[left] <= arr[right]) {
            temp.push_back(arr[left]);
            left++;
        }
        else {
            temp.push_back(arr[right]);
            right++;
        }
    }

    while (left <= mid) {
        temp.push_back(arr[left]);
        left++;
    }

    while (right <= high) {
        temp.push_back(arr[right]);
        right++;
    }

    for (int i = low; i <= high; i++) {
        arr[i] = temp[i - low];
    }
}
```

```

void mergeSort(vector<int> &arr, int low, int high) {
    if (low >= high) return;

    int mid = (low + high) / 2 ;
    mergeSort(arr, low, mid);

    mergeSort(arr, mid + 1, high);

    cout<<"===== Partition before sorting ====="<<endl;
    for(int i=low;i<=high;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;

    merge(arr, low, mid, high);

    cout<<"===== Partition after sorting ====="<<endl;
    for(int i=low;i<=high;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}

int main() {

    srand(time(0));

    int n ;
    cout<<"Enter the size :";
    cin>>n;

    vector<int> arr(n);

    for(int i=0;i<n;i++)
    {
        int random = 100 + (rand() % 101);

        if(count(arr.begin(), arr.end(), random)>0)
        {
            i--;
        }
        else
        {
            arr[i] = random;
        }
    }
}

```

```

    }

    cout << "Before Sorting Array: " << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " " ;
    }
    cout << endl;

    mergeSort(arr, 0, n - 1);

    cout << "After Sorting Array: " << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " " ;
    }
    cout << endl;

    return 0 ;
}

```

Output -

```

Enter the size :7
Before Sorting Array:
174 137 192 103 101 188 196
===== Partition before sorting =====
174 137
===== Partition after sorting =====
137 174
===== Partition before sorting =====
192 103
===== Partition after sorting =====
103 192
===== Partition before sorting =====
137 174 103 192
===== Partition after sorting =====
103 137 174 192
===== Partition before sorting =====
101 188
===== Partition after sorting =====
101 188
===== Partition before sorting =====
101 188 196
===== Partition after sorting =====
101 188 196
===== Partition before sorting =====
103 137 174 192 101 188 196
===== Partition after sorting =====
101 103 137 174 188 192 196
After Sorting Array:
101 103 137 174 188 192 196

```

Code – Quick Sort

```
#include<bits/stdc++.h>

using namespace std;

int partitionArray(vector<int>&arr, int low, int high)
{
    cout<<endl<<"Partition: ";

    for(int i=low;i<=high;i++)
    {
        cout<<arr[i]<<" ";
    }

    int pivot = arr[low];
    cout<<endl<<"Pivot Element : "<<pivot<<endl;

    int i=low;
    int j=high;

    while(i < j)
    {
        while(arr[i] <= pivot && i<high)
        {
            i++;
        }

        while(arr[j] > pivot && j>low)
        {
            j--;
        }

        if(i<j)
        {
            swap(arr[i], arr[j]);
        }
    }

    swap(arr[low], arr[j]);
    return j;
}

void quickSort(vector<int>&arr, int low, int high)
{
    if(low < high)
    {
        int pivotIndex = partitionArray(arr, low, high);
```

```

        quickSort(arr, low, pivotIndex-1);

        quickSort(arr, pivotIndex+1, high);
    }

}

int main() {

    srand(time(0));

    int n ;
    cout<<"Enter the size :";
    cin>>n;

    vector<int> arr(n);

    for(int i=0;i<n;i++)
    {
        int random = (rand() % 1000)+1;

        if(count(arr.begin(), arr.end(), random)>=1)
        {
            i--;
        }
        else
        {
            arr[i] = random;
        }
    }

    cout << "Before Sorting Array: " << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " " ;
    }
    cout << endl;

    quickSort(arr, 0, n - 1);

    cout <<endl<<endl<< "After Sorting Array: " << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " " ;
    }
    cout << endl;

    return 0 ;
}

```

Output-

```
Enter the size :10
Before Sorting Array:
797 14 377 436 74 183 37 531 862 181

Partition: 797 14 377 436 74 183 37 531 862 181
Pivot Element : 797

Partition: 181 14 377 436 74 183 37 531
Pivot Element : 181

Partition: 74 14 37
Pivot Element : 74

Partition: 37 14
Pivot Element : 37

Partition: 436 183 377 531
Pivot Element : 436

Partition: 377 183
Pivot Element : 377

After Sorting Array:
14 37 74 181 183 377 436 531 797 862
```