21BRS1296 – Anika Kamath

Design and Analysis of Algorithms (Lab)

L37+L38

Experiment No.: 4

Q. Merge Sort Algorithm

1. 10 User inputs

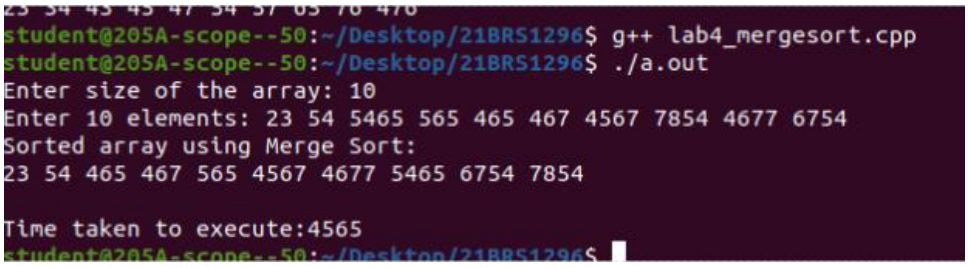
Code:

```
#include <iostream>
#include<ctime>
using namespace std;

void merge(int* A, int p, int q, int r) {
    int n1 = q-p+1;
    int n2 = r-q;
    int i,j,k;
    int *L=new int[n1+1], *R = new int[n2+1];
    for(i=0; i<n1; i++)
        L[i]=A[p+i];
    for(j=0; j<n2;j++)
        R[j]=A[q+j];
    L[n1]=999; //sentinel
    R[n2]=999; //sentinel
    i=0;
    j=0;
    for(k=p; k<r; k++) {
        if(L[i]<=R[j])
            A[k]=L[i++];
```

```
        else
            A[k]=R[j++];
    }
    delete(L);
    delete(R);
}
void mergeSort(int* a, int p, int r) {
    if(p<r) {
        int q=(p+r)/2;
        mergeSort(a,p,q);
        mergeSort(a,q+1,r);
        merge(a,p,q,r);
    }
}
int main() {
    int size;
    cout<<"Enter size of the array: ";
    cin>>size;

    int arr[size];

    cout<<"Enter "<<size<<" elements: ";

    for(int i=0;i<size;i++){
        cin>>arr[i];
    }

    mergeSort(arr,0,size-1);
    for(int i=0; i<size; i++)
        cout << arr[i]<<" ";
```

```
    clock_t tstart=clock();
    double time1=(double)clock()-(tstart)/CLOCKS_PER_SEC;
    cout<<endl<<"Time taken to execute:"<<time1<<endl;
    return 0;
}
```

Output:



2. 10 sets of inputs (random)

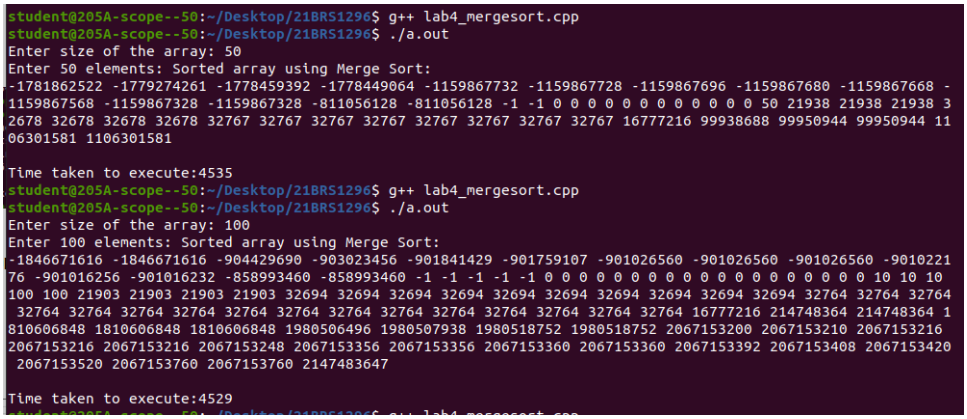
Code:

```
#include <iostream>
#include<ctime>
using namespace std;

void merge(int* A, int p, int q, int r) {
    int n1 = q-p+1;
    int n2 = r-q;
    int i,j,k;
    int *L=new int[n1+1], *R = new int[n2+1];
    for(i=0; i<n1; i++)
        L[i]=A[p+i];
    for(j=0; j<n2;j++)
        R[j]=A[q+j];
```

```
    L[n1]=999; //sentinel
    R[n2]=999; //sentinel
    i=0;
    j=0;
    for(k=p; k<r; k++) {
        if(L[i]<=R[j])
            A[k]=L[i++];

        else
            A[k]=R[j++];
    }
    delete(L);
    delete(R);
}
void mergeSort(int* a, int p, int r) {
    if(p<r) {
        int q=(p+r)/2;
        mergeSort(a,p,q);
        mergeSort(a,q+1,r);
        merge(a,p,q,r);
    }
}
int main() {
    int size;
    cout<<"Enter size of the array: ";
    cin>>size;

    int arr[size];

    cout<<"Enter "<<size<<" elements: ";
```

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

    mergeSort(arr,0,size-1);
    for(int i=0; i<size; i++)
        cout << arr[i]<<" ";

    clock_t tstart=clock();
    double time1=(double)clock()-(tstart)/CLOCKS_PER_SEC;
    cout<<endl<<"Time taken to execute:"<<time1<<endl;
    return 0;
}
```

Output:



Graph:

Number of el	Time
10	4424
50	4535
100	4529
500	4937
1000	5351
5000	7710
7000	8533
10000	10220
12000	10449
15000	10733
18000	10980
20000	11356
25000	11281