**1.ii MERGE SORT**

#include<iostream>

#include<cstdlib>

#include <cmath>

#include <fstream>

using namespace std;

int no\_comp=0;

void merge(int \*arr,int first, int mid, int last){

int size=last-first+1;

int \*arr\_c= new int[size];

int index\_1=first;

int index\_2=mid+1;

int i=0;

while(index\_1<=mid && index\_2<=last ){

no\_comp++;

if(arr[index\_1]>arr[index\_2]){

arr\_c[i]=arr[index\_2++];

}

else{

arr\_c[i]=arr[index\_1++];

}

i++;

}

if(index\_1>mid){

while(index\_2<=last){

arr\_c[i++]=arr[index\_2++];

}

}

if(index\_2>last){

while(index\_1<=mid){

arr\_c[i++]=arr[index\_1++];

}

}

//most important thing in the code

for(int j=0,m=first;j<size && m<=last;j++,m++){

arr[m]=arr\_c[j];

}

delete[] arr\_c;

}

//merge sort made by krishna and Manjit.

void mergesort(int \*arr, int first , int last){

if(first<last){

int mid=(first+last)/2;

mergesort(arr,first,mid);

mergesort(arr,mid+1,last);

merge(arr,first,mid,last);

}

}

int\* getRandomEveryTime(int range,int s){ //gives the sizes of array

int \*arr=new int[s];

int flag=0;

for(int i=0;i<s;){

flag=0;

int x=rand()% range+30;

for(int j=0;j<i;j++){

if(arr[j]==x){

flag=1;

break;

}

}

if(flag==0){

arr[i]=x;

i++;

}

}

return arr;

}

int \*getRandom(int range,int s){ //gives the sizes of array

int \*arr=new int[s];

for(int i=0;i<s ;i++){

arr[i]=rand()% range+30;

}

return arr;

}

void print\_arr(int \*arr,int s){

for(int i=0;i<s ;i++){

cout<<arr[s]<<",";

}

}

int main(){

ofstream nlogn("nlognNew.txt");

ofstream nsq("nsqNew.txt");

ofstream nsize("nNew.txt");

ofstream comp("compNew.txt");

int s =50; // 100 different sizes of inputs

int \*arr=getRandomEveryTime(970,s); // 30 - 1000

for(int i=0 ;i<s;i++){

int siz = arr[i];

int \*arrx = getRandom(100,siz);

mergesort(arrx,0,siz-1);

double n = siz;

cout<<"NO. OF COMP. = "<<no\_comp<<" FOR N = " <<siz<<endl;

no\_comp =0;

}

}

