19l-1316 lab 11:

q#1

#include<iostream>

#include<Windows.h>

using namespace std;

void SWAP(int a, int b)

{

int temp;

temp=a;

a=b;

b=temp;

}

void quicksort(int list[], int left, int right)

{

int pivot, i, j;

if(left<right)

{ i=left; j=right+1; pivot=list[left];

do

{

do

{

i++;

}

while(list[i]<pivot);

do

{

j--;

}

while(list[j]>pivot);

if(i<j)

{

swap(list[i],list[j]);

}

}

while(i<j);

swap(list[left],list[j]);

quicksort(list, left, j-1);

quicksort(list, j+1, right);

}

}

void BubbleSort(int list[], int left, int right)

{

for (int i=left; i < right; i++)

for (int j=right ; j > i; j--)

{

if (list[j] < list[j-1])

{

int temp=list[j];

list[j]=list[j-1];

list[j-1]=temp;

}

}

}

int main()

{

int size;

cout<<"Enter array size: ";

cin>>size;

int \*arrQ=new int[size];

int \*arrB=new int[size];

DWORD start, end;

for (int i=0; i<size; i++)

arrQ[i]=rand()%100;

for (int i=0; i<size; i++)//arrB is going to be a copy of arrQ

arrB[i]=arrQ[i];

if(size<=100)//display array contents on console before Quicksorting

for(int i=0; i<size;i++)

cout<<arrQ[i]<<" ";

cout<<endl;

start= GetTickCount();

quicksort(arrQ, 0, size-1);

end= GetTickCount();

double cpu\_time\_used = end - start;

cout<<" Time taken by Quicksort = "<<cpu\_time\_used<<"milliseconds"<<endl;

if(size<=100)//display array contents on console after Quicksorting

{

for(int i=0; i<size;i++)

cout<<arrQ[i]<<" ";

}

cout<<endl<<endl;

if(size<=100)//display array contents on console before Bubblesorting

for(int i=0; i<size;i++)

cout<<arrB[i]<<" ";

cout<<endl;

start= GetTickCount();

BubbleSort(arrB, 0, size-1);

end= GetTickCount();

cpu\_time\_used = end - start;

cout<<" Time taken by Bubblesort = "<<cpu\_time\_used<<"milliseconds"<<endl;

if(size<=100)//display array contents on console after Bubblesorting

for(int i=0; i<size;i++)

cout<<arrB[i]<<" ";

cout<<endl;

system("pause");

return 0;

}

Q#2

#include<iostream>

#include<windows.h>

using namespace std;

void Adjust(int \*tree , const int root, const int n)

{

int a = 0;

int e=tree[root];

int k=e;

for(int j=2\*root;j<=n;j\*=2)

{

if(j<n)

if(tree[j]<tree[j+1]) j++;

if(k>=tree[j]) break;

tree[j/2]=tree[j];

a=j;

}

tree[a/2]=e;

}

void HeapSort(int \*list, const int n)

{ for(int i=n/2; i>=1;i--)

Adjust(list,i,n);

for(int i=n-1; i>=1;i--)

{ int t=list[i+1];

list[i+1]=list[1];

list[1]=t;

Adjust(list,1,i);

}

}

void BubbleSort(int list[], int left, int right)

{

for (int i=left; i < right; i++)

for (int j=right ; j > i; j--)

{

if (list[j] < list[j-1])

{

int temp=list[j];

list[j]=list[j-1];

list[j-1]=temp;

}

}

}

int main()

{

int size;

cout<<"Enter array size: ";

cin>>size;

int \*arrQ=new int[size];

int \*arrB=new int[size];

DWORD start, end;

for (int i=0; i<size; i++)

arrQ[i]=rand()%100;

for (int i=0; i<size; i++)//arrB is going to be a copy of arrQ

arrB[i]=arrQ[i];

if(size<=100)//display array contents on console before Quicksorting

for(int i=0; i<size;i++)

cout<<arrQ[i]<<" ";

cout<<endl;

start= GetTickCount();

HeapSort(arrQ, size);

end= GetTickCount();

double cpu\_time\_used = end - start;

cout<<" Time taken by Heap Sort = "<<cpu\_time\_used<<" milliseconds"<<endl;

if(size<=100)//display array contents on console after Quicksorting

{

for(int i=0; i<size;i++)

cout<<arrQ[i]<<" ";

}

cout<<endl<<endl;

if(size<=100)//display array contents on console before Bubblesorting

for(int i=0; i<size;i++)

cout<<arrB[i]<<" ";

cout<<endl;

start= GetTickCount();

BubbleSort(arrB, 0, size-1);

end= GetTickCount();

cpu\_time\_used = end - start;

cout<<" Time taken by Bubblesort = "<<cpu\_time\_used<<" milliseconds"<<endl;

if(size<=100)//display array contents on console after Bubblesorting

for(int i=0; i<size;i++)

cout<<arrB[i]<<" ";

cout<<endl;

system("pause");

return 0;

}