**2. HEAP SORT**

#include <iostream>

#include<cstdlib>

#include <cmath>

#include <fstream>

using namespace std;

void swap(int &a,int &b){

int temp = a;

a = b;

b = temp;

}

int no\_of\_comparisons=0;

void heapify(int \*arr,int size,int i){

int largest =i;

int left = 2\*i;

int right = 2\*i +1;

if(++no\_of\_comparisons && left <=size && (arr[largest]<arr[left]) ){

//count

largest = left;

}

if((++no\_of\_comparisons && right <=size && arr[largest] < arr[right])){

//count

largest = right;

}

if(largest !=i){ // if it is updated.

swap(arr[largest],arr[i]);

heapify(arr,size,largest);

}

}

void printHeap(int \*arr, int len){

cout<<"the sorted heap is :"<<endl;

for(int i=1;i<=len;i++){

cout<<arr[i];

if(i != len)

cout<<" , ";

}

}

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;

}

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;

}

void heapsort(int \*arr,int s){

int t = s;

//build max heap

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

heapify(arr,s,i);

}

//heap sort

for(int i=s;i>=2;i--)

{

swap(arr[1],arr[i]);

s-=1;

heapify(arr,s,1);

}

}

int main(){

ofstream nlogn("nlogn\_quick.txt");

ofstream nsq("nsq\_quick.txt");

ofstream nsize("n\_quick.txt");

ofstream comp("comp\_quick.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(50,siz);

heapsort(arrx,siz);

double n = siz;

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

no\_of\_comparisons=0;

}

return 0;

}

