Advanced Data Structures Assignment

Name : Shrirang R Mhalgi

Roll number : 222006

Class : SE B

Batch : B1

--------------------------------------------------------------------------------------------------------------------------------------

Problem Statement:

Implement the Heap/Shell sort algorithm implemented in Java demonstrating heap/shell data structure with modularity of programming language.

Code:

package heapsort;

import java.util.Scanner;

public class Heapsort {

int i;

public void heapsortf(int arr[], int n){

//build heap

for(i=(n/2)-1;i>=0;i--)

heapify(arr,n,i);

//extract each element from root

for(i=n-1;i>=0;i--)

{

int temp = arr[0];

arr[0]=arr[1];

arr[1]=temp;

heapify(arr,i,0);

}

}

public void heapify(int arr[] ,int n, int i ){

int largest = i;

int l = 2\*i+1;

int r = 2\*i+2;

if(l<n && arr[l]>arr[largest])

largest=l;

if(r<n && arr[r]>arr[largest])

largest=r;

if(largest!=i)

{

int temp = arr[i];

arr[i]=arr[largest];

arr[largest] = temp;

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

heapify(arr,n,largest);

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Heapsort o = new Heapsort();

int arr[] = new int[10];

int n = sc.nextInt();

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

arr[i] = sc.nextInt();

o.heapsortf(arr, n);

for(int j=0;j<n;j++)

System.out.println(arr[j]);

}

}

Output:

N : 6

21

34

12

67

03

56

After sorting :

67

34

56

21

3

12