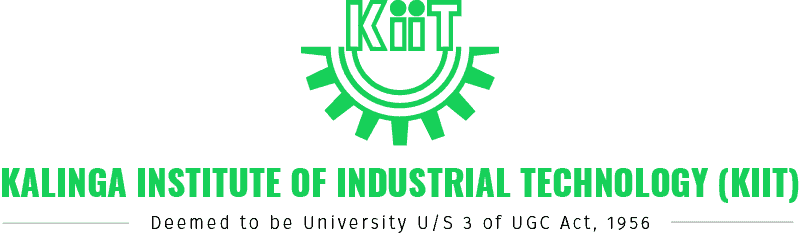
****

**NAME : SOHAM SAMANTA**

**ROLL NUMBER : 20051107**

**SECTION : CSE-14**

# Lab-06:- Evaluation

1. Build Heap

**Solution:**

//SOHAM SAMANTA CODES

#include<bits/stdc++.h>

using namespace std;

#define ll long long int

#define mod 1000000007

#define PI 3.1415926535897932384626433832

#define ss ios\_base::sync\_with\_stdio(false);cin.tie(NULL);

void heapify(int arr[], int N, int i){

int largest = i; // Initialize largest as root

int l = 2 \* i + 1; // left = 2\*i + 1

int r = 2 \* i + 2; // right = 2\*i + 2

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

largest = l;

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

largest = r;

if (largest != i) {

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

heapify(arr, N, largest);

}

}

void buildHeap(int arr[], int N){

int startIdx = (N / 2) - 1;

for (int i = startIdx; i >= 0; i--) {

heapify(arr, N, i);

}

}

void printHeap(int arr[], int N){

cout << "Array representation of Heap is:\n";

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

cout << arr[i] << " ";

cout << "\n";

}

int32\_t main(){

ss;

/\* Binary Tree Representation of input array

1

/ \

3 5

/ \ / \

4 6 13 10

/ \ / \

9 8 15 17

\*/

int arr[] = {1, 3, 5, 4, 6, 13, 10, 9, 8, 15, 17};

int N = sizeof(arr) / sizeof(arr[0]);

buildHeap(arr, N);

printHeap(arr, N);

/\* Final Heap:

17

/ \

15 13

/ \ / \

9 6 5 10

/ \ / \

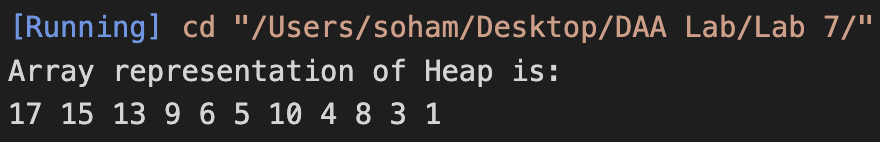
4 8 3 1

\*/

return 0;

}

**Output:**

****

1. Wap to implement Heap Sort .

**Solution :**

//SOHAM SAMANTA CODES

#include<bits/stdc++.h>

using namespace std;

#define ll long long int

#define mod 1000000007

#define PI 3.1415926535897932384626433832

#define ss ios\_base::sync\_with\_stdio(false);cin.tie(NULL);

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

int largest = i; // Initialize largest as root

int l = 2 \* i + 1; // left = 2\*i + 1

int r = 2 \* i + 2; // right = 2\*i + 2

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

largest = l;

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

largest = r;

if (largest != i) {

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

heapify(arr, n, largest);

}

}

void heapSort(int arr[], int n){

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

heapify(arr, n, i);

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

swap(arr[0], arr[i]);

heapify(arr, i, 0);

}

}

void printArray(int arr[], int n){

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

cout << arr[i] << " ";

cout << "\n";

}

int32\_t main(){

ss;

int arr[] = { 12, 11, 13, 5, 6, 7 };

int n = sizeof(arr) / sizeof(arr[0]);

heapSort(arr, n);

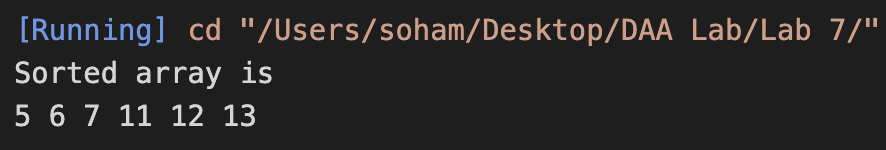
cout << "Sorted array is \n";

printArray(arr, n);

return 0;

}

**Output:**

****

