* **Heap.h**

/\*

\* Heap.h

\* Created on: Nov 28, 2020

\* Author: Megha Sonavane

\*/

**#ifndef** HEAP\_H\_

**#define** HEAP\_H\_

**class** Heap {

**public**:

**Heap**();

**void** **buildHeap**(**int**[],**int**);

**void** **heapSort**(**int**[],**int**);

**void** **heapify**(**int**[],**int**,**int**);

**void** **display**(**int**[],**int**);

**virtual** **~Heap**();

};

**#endif** /\* HEAP\_H\_ \*/

* **Heap.cpp**

/\*

\* Heap.cpp

\* Created on: Nov 28, 2020

\* Author: Megha Sonavane

\*/

**#include**<iostream>

**#include** "Heap.h"

**using** **namespace** std;

**Heap::Heap**() {

// **TODO** Auto-generated constructor stub

}

//============build heap=============================

**void** **Heap::buildHeap**(**int** arr[],**int** n){

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

heapify(arr,n,i);

}

//============Heapify method===========================

**void** **Heap::heapify**(**int** arr[],**int** n,**int** i)

{

**int** largest=i;

//for left

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

//for right

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

**if**(left<n && arr[left]>arr[largest])

largest=left;

**if**(right<n && arr[right]>arr[largest])

largest=right;

**if**(largest!=i)

{

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

heapify(arr,n,largest);

}

}

//=================heap sort====================================

**void** **Heap::heapSort**(**int** arr[],**int** n){

**int** pass=1;

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

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

heapify(arr,i,0);

cout<<**endl**<<"Pass "<<pass<<**endl**;

display(arr,n);

pass++;

}

}

//=============display heap=====================================

**void** **Heap::display**(**int** arr[],**int** n){

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

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

}

}

**Heap::~Heap**() {

// **TODO** Auto-generated destructor stub

}

* **Assignment9.cpp**

//============================================================================

// Name : Assignment9.cpp

// Author : Megha Sonavane

// Description : Heap Sort

//============================================================================

**#include** <iostream>

**#include** "Heap.h"

**#define** MAX 20

**using** **namespace** std;

**int** **main**() {

Heap h;

**int** n,ch;

**int** arr[MAX];

cout<<"Heap Data Struture"<<**endl**;

**do**{

cout<<"Enter number of elememts(Max 20):";

cin>>n;

}**while**(n<1||n>20);

cout<<"Enter "<<n<<" elements:";

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

cin>>arr[i];

**do**{

cout<<**endl**<<"==================================================================="<<**endl**;

cout<<"\t1:Heapify"<<**endl**<<"\t2:Heap Sort"<<**endl**<<"\t3:Insert new element to array"<<**endl**<<"\t0:Exit"<<**endl**;

cout<<"\tEnter choice:";

cin>>ch;

cout<<"==================================================================="<<**endl**;

**switch**(ch){

**case** 1:

//========Build heap===================

h.buildHeap(arr,n);

h.display(arr,n);

**break**;

**case** 2:

//======Heap sort=======================

h.buildHeap(arr,n); //build heap

h.heapSort(arr,n); //sort heap

**break**;

**case** 3:

//=======insert new element to heap=====

**if**(n==20)

cout<<"Sorry..no space available..."<<**endl**;

**else**{

**int** data;

cout<<"\tEnter element:";

cin>>data;

arr[n]=data;

n=n+1;

cout<<"\tElement inserted..."<<**endl**;

}

**break**;

**case** 0:

cout<<"\tThank you...";

**break**;

**default**:

cout<<"\tInvalid choice.."<<**endl**;

}

}**while**(ch!=0);

**return** 0;

}

* **Output:**

**Partially sorted array:**

Heap Data Struture

Enter number of elememts(Max 20):5

Enter 5 elements:5

12

9

3

10

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:1

===================================================================

12 10 9 3 5

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:2

===================================================================

Pass 1

10 5 9 3 12

Pass 2

9 5 3 10 12

Pass 3

5 3 9 10 12

Pass 4

3 5 9 10 12

Pass 5

3 5 9 10 12

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:3

===================================================================

Enter element:4

Element inserted...

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:2

===================================================================

Pass 1

10 5 9 3 4 12

Pass 2

9 5 4 3 10 12

Pass 3

5 3 4 9 10 12

Pass 4

4 3 5 9 10 12

Pass 5

3 4 5 9 10 12

Pass 6

3 4 5 9 10 12

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:0

===================================================================

Thank you...

**Completely Sorted Array:**

Heap Data Struture

Enter number of elememts(Max 20):5

Enter 5 elements:3

5

10

9

12

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:1

===================================================================

12 9 10 3 5

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:2

===================================================================

Pass 1

10 9 5 3 12

Pass 2

9 3 5 10 12

Pass 3

5 3 9 10 12

Pass 4

3 5 9 10 12

Pass 5

3 5 9 10 12

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:0

===================================================================

Thank you...

**Completely unsorted array:**

Heap Data Struture

Enter number of elememts(Max 20):5

Enter 5 elements:12

10

9

5

3

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:1

===================================================================

12 10 9 5 3

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:2

===================================================================

Pass 1

10 5 9 3 12

Pass 2

9 5 3 10 12

Pass 3

5 3 9 10 12

Pass 4

3 5 9 10 12

Pass 5

3 5 9 10 12

===================================================================

1:Heapify

2:Heap Sort

3:Insert new element to array

0:Exit

Enter choice:0

===================================================================

Thank you...