19l-1316

Q # 1

#include<iostream>

#include"Header.h"

using namespace std;

int main()

{

MaxHeap<int> \*mxHeap; //creating an object of maxheap

mxHeap=new MaxHeap<int>(40);

//insert following data in the MaxHeap

mxHeap->insert(12);

mxHeap ->insert(43);

mxHeap ->insert(9);

mxHeap ->insert(2);

mxHeap ->insert(14);

mxHeap ->insert(16);

mxHeap ->insert(13);

mxHeap ->insert(12);

mxHeap->printContents();

//Carry out 2 deletions from the MaxHeap

int output;

output=mxHeap->del();

cout<<"Output of first deletion is "<<output<<endl;

mxHeap->printContents();

output=mxHeap->del();

cout<<"Output of second deletion is "<<output<<endl;

mxHeap->printContents();

system("pause");

return 0;

}

#include<iostream>

using namespace std;

template<class T>

class MaxHeap

{

public:

//part1: constructor initializes array of size maxsize

MaxHeap(int size)

{

maxsize=size;

arr=new int[maxsize];

n=0;

};

//part2: Inserts temp into its appropriate position

//within the Heap

bool insert(const T largest)

{

if (n==maxsize)

{cout<<"heap is full"<<endl;

return 0;

}

n++;

int i;

for( i=n;1;)

{

if (i==1)

break ;

if (largest<=arr[i/2])

break;

arr[i]=arr[i/2];

i/=2;

}

arr[i]=largest;

return true;

};

//part3: removes the element present in the the root

//of the Heap and readjusts it to form MaxHeap again

T del()

{

int largest=0;

if (n==NULL)

{

cout<<"heap is empty "<<endl;

return 0;

}

largest=arr[1];

int index=arr[n--];

int i,j;

for ( i=1,j=2;j<=n;)

{

if (j<n)

if (arr[j]<arr[j+1])

j++;

if (index>arr[j])

break;

arr[i]=arr[j];

i=j;

j\*=2;

}

arr[i]=index;

return largest;

};

//part4: prints all the temp present in the Heap

//use the appropriate traversal

void printContents()

{

int temp;

for(int i=2;i<=n;++i)

{

for(int j=1;j<=(n-i);++j)

if(arr[j]<arr[j+1])

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

for (int i=1;i<=n;i++)

{

cout<<arr[i]<<endl;

}

};

//part5: destructor, deletes the MaxHeap

~MaxHeap()

{

delete arr;

};

private:

T \*arr;

T maxsize;

int n;

};

Q#2

#include<iostream>

#include"Header.h"

using namespace std;

int main()

{

MinHeap<int> \*mnHeap; //creating an object of MinHeap

mnHeap=new MinHeap<int>(40);

//insert following data in the MinHeap

mnHeap->insert(12);

mnHeap ->insert(43);

mnHeap ->insert(9);

mnHeap ->insert(2);

mnHeap ->insert(14);

mnHeap ->insert(16);

mnHeap ->insert(13);

mnHeap ->insert(12);

mnHeap->printContents();

//Carry out 2 deletions from the MinHeap

int output;

output=mnHeap->del();

cout<<"Output of first deletion is "<<output<<endl;

mnHeap->printContents();

output=mnHeap->del();

cout<<"Output of second deletion is "<<output<<endl;

mnHeap->printContents();

system("pause");

return 0;

}

#include<iostream>

using namespace std;

template<class T>

class MinHeap

{

public:

//part1: constructor initializes array of size maxsize

MinHeap(int size)

{

maxsize=size;

arr=new int[maxsize];

n=0;

}

//part2: Inserts smallest into its appropriate position

//within the Heap

bool insert(const T smallest)

{

if (n==maxsize)

{cout<<"heap is full"<<endl;

return 0;

}

n++;

int i;

for( i=n;1;)

{

if (i==1)

break ;

if (smallest>=arr[i/2])

break;

arr[i]=arr[i/2];

i/=2;

}

arr[i]=smallest;

return true;

}

//part3: removes the element present in the the root

//of the Heap and readjusts it to form MaxHeap again

T del()

{

int smallest=0;

if (n==NULL)

{

return 0;

}

smallest=arr[1]; int r=arr[n--];

int i,j;

for ( i=1,j=2;j<=n;)

{

if (j<n)

if (arr[j]<arr[j+1])

j++;

if (r<arr[j])

break;

arr[i]=arr[j];

i=j;

j=j\*2;

}

arr[i]=r;

return smallest;

}

//part4: prints all the smallest present in the Heap

//use the appropriate traversal

void printContents()

{

int temp;

for(int i=2;i<=n;++i)

{

for(int j=1;j<=(n-i);++j)

if(arr[j]>arr[j+1])

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

for (int i=1;i<=n;i++)

{

cout<<arr[i]<<endl;

}

}

//part5: destructor, deletes the MaxHeap

~MinHeap()

{

delete arr;

}

private:

T \*arr;

T maxsize;

int n;

};