DSA SERIES

- Learn Coding

Topic to be Covered today

Heap

LETS START TODAY'S LECTURE

Implementation code :-

```
#include <iostream>
using namespace std;


class MaxHeap
{
private:
    int arr[100];
    int size;


public:
    MaxHeap()
    {
        size = 0;
    }
```

```
void insert(int val)
    {
        arr[size] = val;
        int i = size;
        size++;


        while (i > 0)
        {
            int parent = (i - 1) / 2;


            if (arr[parent] < arr[i])
            {
                swap(arr[parent], arr[i]);
                i = parent;
            }
            else
            {
                break;
            }
        }
    }
```

```
void deleteMax()
    {
        if (size == 0)
        {
            cout << "The heap is empty" << endl;
            return;
        }


        arr[0] = arr[size - 1];
        size--;
        int i = 0;


        while (true)
        {
            int leftChild = 2 * i + 1;
            int rightChild = 2 * i + 2;
            int largest = i;


            if (leftChild < size && arr[i] < arr[leftChild])
{
                largest = leftChild;
            }


            if (rightChild < size && arr[largest] < arr[rightChild])
            {
                largest = rightChild;
            }


            if (largest != i)
            {
                swap(arr[largest], arr[i]);
                i = largest;
            }
            else
            {
                break;
            }
        }
    }
```

```
void printHeap()
    {
        for (int i = 0; i < size; i++)
        {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
};
int main()
{


    MaxHeap h;
    h.insert(30);
    h.insert(40);
    h.insert(50);
    h.insert(10);
    h.insert(20);
    h.insert(35);
    h.insert(80);
    cout << "The heap is : ";
    h.printHeap();


    h.deleteMax();
    cout << "The heap after deleting max element : ";
    h.printHeap();


    return 0;
}
```

Heap Sort :

```
#include <iostream>
#include <vector>
using namespace std;


void heapify(vector<int> &arr, int n, int i)
{
    int largest = i;
    int left = 2 * i + 1;
    int right = 2 * i + 2;


    if (left < n && arr[largest] < arr[left])
    {
        largest = left;
    }
```

```
if (right < n && arr[largest] < arr[right])
    {
        largest = right;
    }


    if (largest != i)
    {
        swap(arr[i], arr[largest]);
        heapify(arr, n, largest);
    }
}


void heapSort(vector<int> &arr)
{
    int n = arr.size();
    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);
    }
}


int main()
{


    vector<int> arr = {12, 11, 13, 5, 6, 7};


    cout << "Before sorting : ";


    for (int i = 0; i < arr.size(); i++)
    {
        cout << arr[i] << " ";
    }
    heapSort(arr);


    cout << "\nAfter sorting : ";


    for (int i = 0; i < arr.size(); i++)
    {
        cout << arr[i] << " ";
    }


    return 0;
}
```

Thank You