**Lab Assignment 9**

**Name = ishwari Sahebrao Jeughale**

**Roll NO = 23527**

**batch = s2**

#include <iostream>

#include <climits>

using namespace std;

class MaxHeap

{

private:

    int \*heapArray;

    int capacity;

    int heapSize;

    int parent(int i)

    {

        return (i - 1) / 2;

    }

    int leftChild(int i)

    {

        return 2 \* i + 1;

    }

    int rightChild(int i)

    {

        return 2 \* i + 2;

    }

    void swap(int &a, int &b)

    {

        int temp = a;

        a = b;

        b = temp;

    }

    void heapify(int i)

    {

        int largest = i;

        int left = leftChild(i);

        int right = rightChild(i);

        if (left < heapSize && heapArray[left] > heapArray[largest])

            largest = left;

        if (right < heapSize && heapArray[right] > heapArray[largest])

            largest = right;

        if (largest != i)

        {

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

            heapify(largest);

        }

    }

public:

    MaxHeap(int cap)

    {

        heapSize = 0;

        capacity = cap;

        heapArray = new int[capacity];

    }

    ~MaxHeap()

    {

        delete[] heapArray;

    }

    void insert(int value)

    {

        if (heapSize == capacity)

        {

            cout << "Heap is full. Cannot insert more elements.\n";

            return;

        }

        int i = heapSize;

        heapArray[i] = value;

        heapSize++;

        while (i != 0 && heapArray[parent(i)] < heapArray[i])

        {

            swap(heapArray[i], heapArray[parent(i)]);

            i = parent(i);

        }

    }

    int extractMax()

    {

        if (heapSize <= 0)

        {

            cout << "Heap is empty\n";

            return INT\_MIN;

        }

        if (heapSize == 1)

        {

            heapSize--;

            return heapArray[0];

        }

        int root = heapArray[0];

        heapArray[0] = heapArray[heapSize - 1];

        heapSize--;

        heapify(0);

        return root;

    }

    void buildHeap(int arr[], int size)

    {

        if (size > capacity)

        {

            cout << "Array size is larger than heap capacity\n";

            return;

        }

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

            heapArray[i] = arr[i];

        heapSize = size;

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

            heapify(i);

    }

    void heapSort(int arr[], int size)

    {

        buildHeap(arr, size);

        for (int i = size - 1; i >= 0; i--)

        {

            arr[i] = extractMax();

        }

    }

    void printHeap()

    {

        if (heapSize == 0)

        {

            cout << "Heap is empty\n";

            return;

        }

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

            cout << heapArray[i] << " ";

        cout << "\n";

    }

    int getSize()

    {

        return heapSize;

    }

};

int main()

{

    int arr[] = {12, 11, 13, 5, 6, 7, 15, 1, 9, 8};

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

    cout << "Original array: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << "\n";

    MaxHeap maxHeap(n);

    maxHeap.heapSort(arr, n);

    cout << "Sorted array: ";

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    cout << "\n";

    cout << "\nDemonstrating heap operations:\n";

    MaxHeap heap(5);

    cout << "Inserting elements: 4, 10, 3, 5, 1\n";

    heap.insert(4);

    heap.insert(10);

    heap.insert(3);

    heap.insert(5);

    heap.insert(1);

    cout << "Heap after insertions: ";

    heap.printHeap();

    cout << "Extracting max elements one by one:\n";

    while (heap.getSize() > 0)

    {

        cout << "Extracted: " << heap.extractMax() << "\n";

    }

    return 0;

}

**GitHub link:-**

[https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment\_ HYPERLINK "https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment\_9.cpp"9 HYPERLINK "https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment\_9.cpp".cpp](https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment_9.cpp)