**Assignment No. 06**

Name:- Anirudha Shripad Kale.

Roll No. :- 223029

PRN No :- 22210288

Batch: C1

**Problem Statement:** Read the marks obtained by students of second year in an online examination of particular subject. Find out maximum and minimum marks obtained in that subject. Use heap data structure. Analyse the algorithm.

**Objectives:**

1. To study the heap data structure.
2. To understand the concept of max and min heap.
3. For implementing the heaps understand the heapify algorithm.
4. To analyse the algorithms with time and space complexity.

**Theory:**

Heap Data Structure:

A heap is a specialized tree-based data structure that satisfies the heap property. Heaps are commonly implemented as binary trees, where each node in the tree satisfies the heap property, which depends on the type of heap (max heap or min heap).

Max Heap:

In a max heap, for any given node 𝑖, the value of 𝑖's parent is greater than or equal to the value of 𝑖. In other words, the largest element in the heap is stored at the root. This property ensures that the maximum element can be efficiently retrieved from the heap.

Min Heap:

In a min heap, for any given node 𝑖, the value of 𝑖's parent is less than or equal to the value of 𝑖. In other words, the smallest element in the heap is stored at the root. This property ensures that the minimum element can be efficiently retrieved from the heap.

In a binary tree (including binary heaps), the following formulas can be used to calculate the indices of the left child, right child, and parent of a given node at index 𝑖: (0 based indexing)

Left Child Index (𝑖 left):

The left child of a node at index 𝑖 is located at index 2𝑖 + 1.

Right Child Index (𝑖 right):

The right child of a node at index 𝑖 is located at index 2𝑖 + 2.

Parent Index (𝑖 parent):

The parent of a node at index 𝑖 (except for the root node) is located at index (𝑖 - 1) / 2. Note that integer division should be used to ensure that the result is an integer.

**Algorithm:**

1. BuildingHeaps Class:

a. buildMaxHeap(int[] arr): This method builds a max heap from the input array of marks using the maxHeapify() function.

b. maxHeapify(int[] arr, int size, int index): This method applies the max-heap property to the element at the given index index in the array arr. It recursively maintains the max heap property downwards.

c. buildMinHeap(int[] arr): This method builds a min heap from the input array of marks using the minHeapify() function.

d. minHeapify(int[] arr, int size, int index): This method applies the min-heap property to the element at the given index index in the array arr. It recursively maintains the min heap property downwards.

e. getMax(int[] arr): This method returns the maximum mark by building a max heap from the input array of marks and returning the value at index 0.

f. getMin(int[] arr): This method returns the minimum mark by building a min heap from the input array of marks and returning the value at index 0.

1. MinMaxMarks Class (Main Class):

a. It defines the main() method where the program execution starts.

b. It initializes an array of subject names and shortcuts.

c. It prompts the user to enter the marks for each subject and stores them in an array.

d. It calculates the maximum and minimum marks using the getMax() and getMin() methods from the BuildingHeaps class.

e. It prints out congratulatory messages for the maximum score and improvement suggestions for the minimum score.

1. Input/Output:

The program prompts the user to input marks for each subject.

It then calculates and displays the maximum and minimum marks along with corresponding subject names.

**Input:**

import java.util.\*;

class BuildingHeaps {

    public void buildMaxHeap(int[] arr) {

        int n = arr.length;

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

            maxHeapify(arr, n, i);

        }

    }

    public void maxHeapify(int[] arr, int size, int index) {

        int largest = index;

        int left = 2 \* index + 1;

        int right = 2 \* index + 2;

        if (left < size && arr[largest] < arr[left])

            largest = left;

        if (right < size && arr[largest] < arr[right])

            largest = right;

        if (largest != index) {

            int temp = arr[largest];

            arr[largest] = arr[index];

            arr[index] = temp;

            maxHeapify(arr, size, largest);

        }

    }

    public void buildMinHeap(int[] arr) {

        int n = arr.length;

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

            minHeapify(arr, n, i);

        }

    }

    public void minHeapify(int[] arr, int size, int index) {

        int smallest = index;

        int left = 2 \* index + 1;

        int right = 2 \* index + 2;

        if (left < size && arr[smallest] > arr[left])

            smallest = left;

        if (right < size && arr[smallest] > arr[right])

            smallest = right;

        if (smallest != index) {

            int temp = arr[smallest];

            arr[smallest] = arr[index];

            arr[index] = temp;

            maxHeapify(arr, size, smallest);

        }

    }

    public int getMax(int[] arr) {

        buildMaxHeap(arr);

        return arr[0];

    }

    public int getMin(int[] arr) {

        buildMinHeap(arr);

        return arr[0];

    }

}

public class Assignment6 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        BuildingHeaps heap1 = new BuildingHeaps();

        System.out.println("\t\*\*\* Subjects in second year \*\*\*\t");

        String[] subjects = { "Fundamentals of Data structures (FDS)", "Discrete Mathematics (DM)",

                "Computer Architecture & Organization (CAO)", "Analog and Digital Electronics (ADE)",

                "Universal Human Values (UHV)", "Advanced Data Structure (ADS)", "Theory of Computation (TOC)",

                "Database Managment Systenm (DBMS)", "Probability & Statistics (PS)", "Operating System (OS)" };

        String[] shortCuts = { "FDS", "DM", "CAO", "ADE", "UHV", "ADS", "TOC", "DBMS", "PS", "OS" };

        System.out.println("----------------------------------------");

        for (int i = 0; i < 10; i++) {

            System.out.println((i + 1) + ". " + subjects[i]);

        }

        System.out.println("----------------------------------------");

        System.out.println("Enter the marks of respective subject (out of 100): ");

        int[] marks = new int[10];

        int[] temp = new int[10];

        for (int i = 0; i < 10; i++) {

            System.out.print(shortCuts[i] + " : ");

            marks[i] = sc.nextInt();

            temp[i] = marks[i];

        }

        System.out.println("----------------------------------------");

        int maxMark = heap1.getMax(temp);

        int minMark = heap1.getMin(temp);

        for (int i = 0; i < 10; i++) {

            if (marks[i] == maxMark) {

                System.out.println("Congrats !!\nYour maximum score is " + maxMark + " / 100 in " + shortCuts[i]);

            }

        }

        System.out.println("&");

        for (int i = 0; i < 10; i++) {

            if (marks[i] == minMark) {

                System.out

                        .println("Your minimum score is " + minMark + " / 100 \nNeeded improvement in " + shortCuts[i]);

            }

        }

        System.out.println("----------------------------------------");

    }

}

**Output:**

\*\*\* Subjects in second year \*\*\*

----------------------------------------

1. Fundamentals of Data structures (FDS)

2. Discrete Mathematics (DM)

3. Computer Architecture & Organization (CAO)

4. Analog and Digital Electronics (ADE)

5. Universal Human Values (UHV)

6. Advanced Data Structure (ADS)

7. Theory of Computation (TOC)

8. Database Managment Systenm (DBMS)

9. Probability & Statistics (PS)

10. Operating System (OS)

----------------------------------------

Enter the marks of respective subject (out of 100):

FDS : 98

DM : 98

CAO : 90

ADE : 92

UHV : 87

ADS : 93

TOC : 84

DBMS : 97

PS : 99

OS : 85

----------------------------------------

Congrats !!

Your maximum score is 99 / 100 in PS

&

Your minimum score is 84 / 100

Needed improvement in TOC

----------------------------------------