**🔷 THEORY**

**📌 Problem Overview**

You are given the marks of second-year students in a subject. The goal is to:

* Find the **maximum** and **minimum** marks
* Use **heap data structures** (Min Heap & Max Heap)

**📌 Heap Data Structure**

A **Heap** is a special **complete binary tree** used mainly to implement **priority queues**. It satisfies the **heap property**:

**🔹 Min Heap:**

* **Parent node** is always **less than or equal** to its children.
* **Root node** gives the **minimum element**.

**🔹 Max Heap:**

* **Parent node** is always **greater than or equal** to its children.
* **Root node** gives the **maximum element**.

**📌 How the Program Works**

* The array of marks is used to build:
  + A **Min Heap** to find the **minimum mark** at index 0.
  + A **Max Heap** to find the **maximum mark** at index 0.
* The heap is built using a **"bubble up"** method, ensuring the heap property is restored after each insertion.

**🔷 ALGORITHM**

**🔧 1. MinHeap(arr, i)**

**Purpose:** Maintain Min Heap property by bubbling up the inserted element.

text

CopyEdit

Input: Array arr[], index i

Output: Modified array with Min Heap property

1. current = i

2. parent = (current - 1) / 2

3. While parent >= 0 and arr[parent] > arr[current]:

a. Swap arr[parent] and arr[current]

b. current = parent

c. parent = (current - 1) / 2

**🔧 2. MaxHeap(arr, i)**

**Purpose:** Maintain Max Heap property by bubbling up the inserted element.

text

CopyEdit

Input: Array arr[], index i

Output: Modified array with Max Heap property

1. current = i

2. parent = (current - 1) / 2

3. While parent >= 0 and arr[parent] < arr[current]:

a. Swap arr[parent] and arr[current]

b. current = parent

c. parent = (current - 1) / 2

**🔧 3. Main Algorithm**

text

CopyEdit

1. Read n (number of students)

2. Input marks into array arr[]

3. For i = 1 to n-1:

- Call MinHeap(arr, i)

4. Output arr[0] as Minimum Score

5. For i = 1 to n-1:

- Call MaxHeap(arr, i)

6. Output arr[0] as Maximum Score

**📈 Time Complexity**

* Building the heap (bubble-up method) takes **O(n log n)** time.
* Accessing min or max is **O(1)** after heap construction.