HEAP SORT

1. Definition

Heap Sort is a comparison-based sorting algorithm that uses a **binary heap** (Max-Heap or Min-Heap) to sort elements.
It works by **heapifying** the array and then repeatedly **removing the root** and rebuilding the heap.

2. Types of Heaps Used

- **Max-Heap** → gives **ascending** order
- **Min-Heap** → gives **descending** order

(You can use either, depending on the required sort order.)

Algo HeapSort(A, n)

(General structure: Build Heap → Remove Root Repeatedly)

```
// Algo HeapSort(A, n)
// Input: Array A of size n
// Output: A sorted array

1. // Build Heap (Heapify entire array)
   For i = n/2 down to 1:
         Heapify(A, i, n)          // can be Max-Heap or Min-Heap

2. // Sorting by deleting root repeatedly
   For end = n down to 2:
         Swap(A[1], A[end])        // root element moved to end

         // Delete root: Replace removed root with last element
```

```
        // Now reduce heap size by 1 and rebuild heap
        Heapify(A, 1, end - 1)

3. Return sorted array A
```

Heap Sort – Practice Questions (5 Questions)

🔶 Q1.

Given the numbers:
20, 5, 15, 22, 9

Tasks:

1. Construct the **Max-Heap**
2. Construct the **Min-Heap**
3. Perform **Heap Sort (ascending)** using Max-Heap
4. Perform **Heap Sort (descending)** using Min-Heap

🔶 Q2.

Given the numbers:
12, 3, 25, 7, 18, 10

Tasks:

1. Build the **Max-Heap**
2. Build the **Min-Heap**
3. Sort the numbers using **Heap Sort**
4. Show each deletion of the root

🔶 Q3.

Given the numbers:
40, 12, 9, 30, 25, 16

Tasks:

1. Create the **Max-Heap** using bottom-up heapify
2. Create the **Min-Heap**
3. Perform Heap Sort (ascending)
4. Show the heap after each step

🔶 Q4.

Given the numbers:
14, 7, 28, 18, 3

Tasks:

1. Form the **Max-Heap**
2. Form the **Min-Heap**
3. Apply Heap Sort
4. Show final sorted order

🔶 Q5.

Given the numbers:
50, 22, 35, 10, 5, 40

Tasks:

1. Build **Max-Heap** (show tree + array)
2. Build **Min-Heap** (tree + array)
3. Perform Heap Sort (delete + replace + reheapify)
4. Display the final sorted array