That's great! A **Heap** is a specialized tree-based data structure that satisfies the **Heap Property**:

* In a **Max-Heap**, for any given node, the value of the node is greater than or equal to the values of its children.
* In a **Min-Heap**, for any given node, the value of the node is less than or equal to the values of its children.

**Key Characteristics:**

1. A heap is a **complete binary tree**, meaning all levels are completely filled except possibly the last, which is filled from left to right.
2. It can be implemented using an array or a list for space efficiency.

**Common Operations:**

1. **Insert**: Adds a new element while maintaining the heap property.
2. **Delete/Extract Root**: Removes the root node (largest in max-heap or smallest in min-heap) and restructures the heap.
3. **Peek**: Retrieves the root element without removing it.

**Use Cases of Heaps**

1. **Priority Queue**: Heaps are used to implement priority queues where the element with the highest or lowest priority is always served first.
2. **Heap Sort**: A sorting algorithm that uses heaps to sort data in O(n log n) time.
3. **Median Finder**: A heap can be used to efficiently calculate the median in a stream of numbers.
4. **Graph Algorithms**: Heaps are essential for algorithms like Dijkstra's shortest path and Prim's minimum spanning tree.