



What is Heap?

- A Heap is a special type of tree-based data structure.
- It follows the Complete Binary Tree property.
- In a heap, parent nodes are ordered with respect to their children.
- There are two types: Min Heap and Max Heap.



Why We Use Heap?

- To quickly get the minimum or maximum element.
- To efficiently manage priority-based data.
- Used when order matters more than exact structure.
- Provides better performance than simple arrays for priority operations.



Concept of Heap

- Heap is always a Complete Binary Tree.
- Nodes are filled level by level from left to right.
- It follows Heap Property:
- Min Heap \rightarrow Parent \leq Children
- Max Heap \rightarrow Parent \geq Children
- Heap is commonly implemented using arrays, not pointers.



Real-Life Examples

- Hospital emergency system
(most critical patient first)
- CPU scheduling
(highest priority process first)
- Airline ticket booking
(cheapest or highest priority ticket)
- Task scheduling apps
- Priority Queue implementation



Min Heap vs Max Heap

Min Heap:

- Smallest element is always at the root.
- Used when minimum value is required first.

Max Heap:

- Largest element is always at the root.
- Used when maximum value is required first.
- Structure remains the same, only comparison changes.



Heap vs BST

Heap:

- Always a Complete Binary Tree.
- Only root is guaranteed min/max.
- Faster for priority-based access.

BST:

- Left < Root < Right property.
- Can become unbalanced.
- Good for searching sorted data.