 Binary Search Tree (BST)?

🌳 What is a Binary Search Tree (BST)?

- A Binary Search Tree (BST) is a special type of Binary Tree.
- It follows a specific ordering rule:
- 👉 Left child < Parent < Right child
- Each node can have maximum two children.
- BST helps in fast searching, insertion, and deletion of data.

💡 Why We Need a BST?

- Normal Binary Tree doesn’t store data in any order. (Searching becomes slow)
- BST stores data in a sorted structure, making operations faster.
- It is used when we need quick lookup, sorted traversal, or efficient searching.
- Example: Searching a contact name, dictionary words, or ordered data.

⚙️ Time Complexity (Average Case)

- 🔍 Search: O(log n)
- ➕ Insertion: O(log n)
- ❌ Deletion: O(log n)
- In the worst case (unbalanced tree), time complexity becomes O(n) (like a linked list).
- To fix that, we use Balanced BSTs.