**Binary Search Tree (BST)**

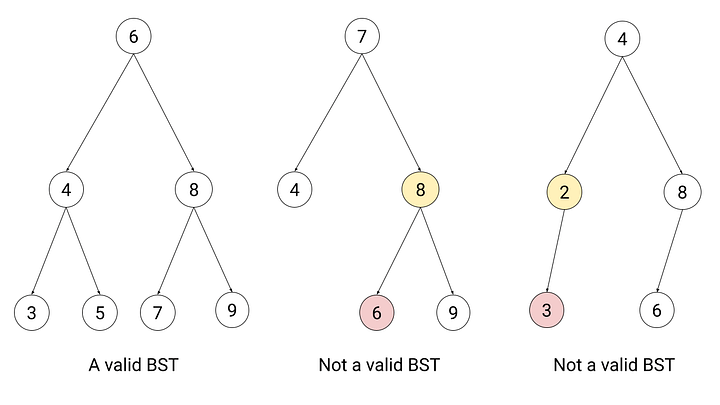
A **Binary Search Tree (BST)**  is a specific type of binary tree that follows a particular ordering and sorting property.

While the ordering rule is commonly associated with numeric values, it's important to note that **binary search tree (BST)** can store any data type as long as there's a well-defined comparison rule for ordering. This means BSTs can be used with strings, custom objects, or any data structure where elements can be ordered (e.g., lexicographically for strings or based on a specific attribute in objects).

For example:

* A BST storing **numbers** follows the rule: left subtree contains smaller numbers, right subtree contains larger numbers.
* A BST storing **strings** follows the same rule alphabetically: left subtree has lexicographically smaller strings, right subtree has larger ones.
* A BST storing **objects** (e.g., employees) could be sorted by attributes like salary, age, or name.

Let's use numeric values for simplicity and better understanding.



The image shows three binary trees, with one being a valid Binary Search Tree (BST) and the other two violating BST properties. Here's an explanation

**1st Tree (Left) - A Valid BST**

* The root node is **6**.
* The left subtree contains **4**, with its children **3** and **5** (both less than 6).
* The right subtree contains **8**, with its children **7** and **9** (both greater than 6).
* Every node follows the BST rule: **Left subtree < Parent < Right subtree**.

✅ **This is a valid BST.**

**2nd Tree (Middle) - Not a Valid BST**

* The root node is **7**.
* The left child is **4** (valid, since 4 < 7).
* The right child is **8** (valid, since 8 > 7).
* However, **6** is in the right subtree of **8**, which violates the BST rule because **6 < 8**.

❌ **This is NOT a valid BST.**

**3rd Tree (Right) - Not a Valid BST**

* The root node is **4**.
* The right subtree contains **8** and **6**, which are correctly placed.
* However, in the left subtree, the node **2** has a right child **3**, which violates the BST rule because **3 > 2** but is on the left subtree of 4.

❌ **This is NOT a valid BST.**

**Time Complexity for Important BST Operations**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Best Case (Balanced BST)** | **Worst Case (Skewed BST)** |
| Search | O(log n) | O(n) |
| Insertion | O(log n) | O(n) |
| Deletion | O(log n) | O(n) |