**BST VS BT**

[**Binary Tree**](http://en.wikipedia.org/wiki/Binary_tree) is a specialized form of tree with two child (left child and right Child). It is simply representation of data in Tree structure

[**Binary Search Tree (BST)**](http://en.wikipedia.org/wiki/Binary_search_tree) is a special type of Binary Tree that follows following condition:

1. left child node is smaller than its parent Node
2. right child node is greater than its parent Node

**Full Binary Tree** A Binary Tree is full if every node has 0 or 2 children. Following are examples of full binary tree. We can also say a full binary tree is a binary tree in which all nodes except leaves have two children.

**Complete Binary Tree:** A Binary Tree is complete Binary Tree if all levels are completely filled except possibly the last level and the last level has all keys as left as possible.

**Perfect Binary Tree** A Binary tree is Perfect Binary Tree in which all internal nodes have two children and all leaves are at same level.

**A degenerate (or pathological) tree**A Tree where every internal node has one child. Such trees are performance-wise same as linked list.

**AVL Tree:-**AVL tree is a self-balancing Binary Search Tree (BST) where the difference between heights of left and right subtrees cannot be more than one for all nodes.

**Red-Black tree**: Another variant of binary tree similar to AVL tree it is a self balancing binary search tree. In this tree nodes are either colored red or black.

**Splay tree:**A splay tree is a self-adjusting binary search tree with the additional property that recently accessed elements are quick to access again. All normal operations on a binary search tree are combined with one basic operation, called splaying. Splaying the tree for a certain element rearranges the tree so that the element is placed at the root of the tree.