Binary Search Trees

Learn Vid Fun...

Time Complexities of BST Operations

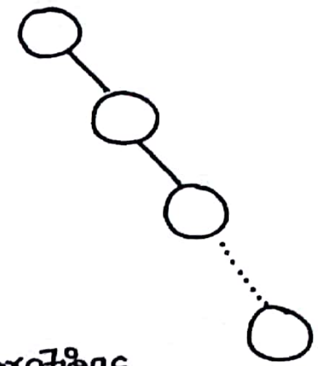
The time complexity for all Binary Search Tree operations be it search operation or Insert operation or Delete operation is $O(h)$ where h is the height of a Binary Search Tree.

Thus, In general-

Time complexity of BST Operations = O(height)

Worst Case-

In worst case, the binary search tree is a skewed binary search tree and we have to travel from root to the deepest leaf node.



In that case, the height of the binary search Tree becomes 'n'.

Thus,

In worst case, Time complexity for BST operations = $O(n)$

In this case, BST is as good as unordered list with no benefits.

Binary Search Tree in Worst Case-

Example-

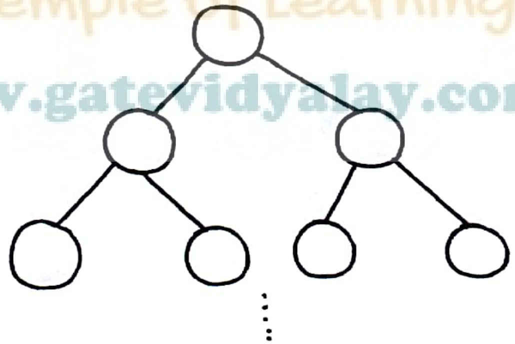
Skewed Binary Search Tree

Best Case-

In best case, the binary search tree is a balanced binary search tree with height 'log n'

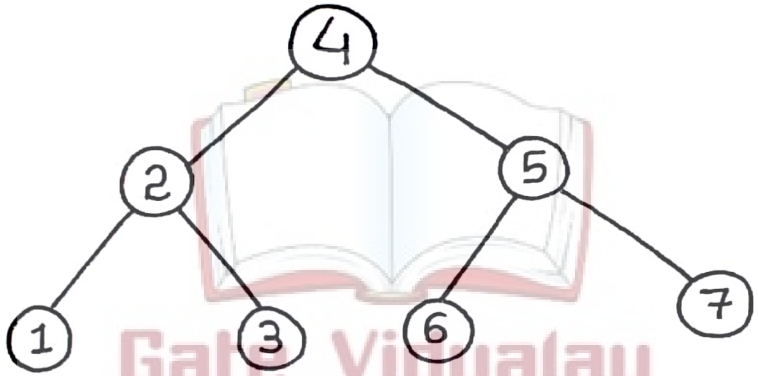
Thus,

In Best Case, Time Complexity of BST Operations $= O(\log n)$



Binary Search Tree in Best Case-

Example-



Balanced Binary Search Tree