

Complexity of different operations in Binary tree, Binary Search Tree and AVL tree

In this article, we will discuss complexity of different operations in binary trees including BST and AVL trees. Before understanding this article, you should have basic idea about: [Binary Tree](#), [Binary Search Tree](#) and [AVL Tree](#).

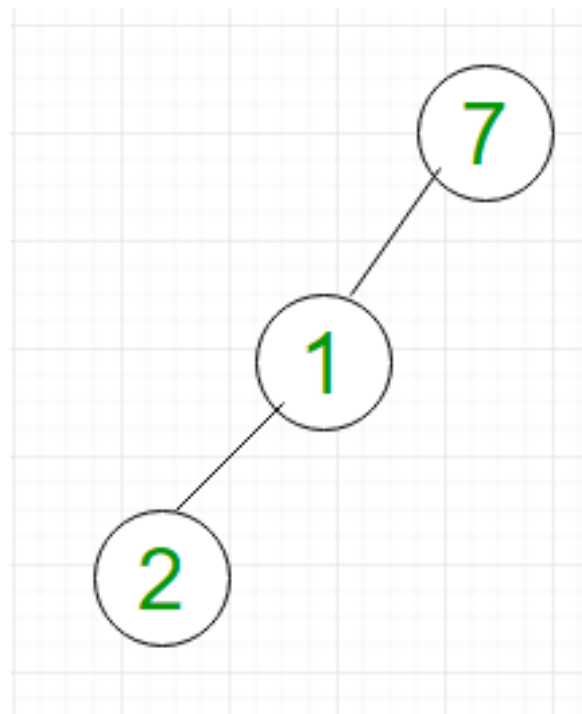
[Hire with us!](#)

The main operations in binary tree are: search, insert and delete. We will see the worst case time complexity of these operations in binary trees.

Binary Tree –

In a binary tree, a node can have maximum two children. Consider the left skewed binary tree shown in Figure 1.



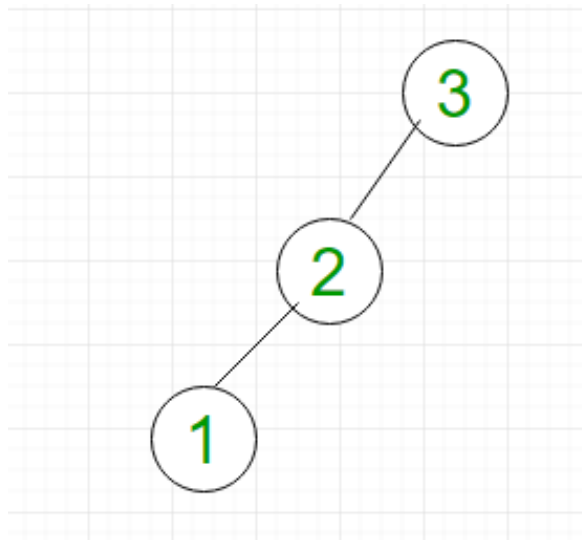


- **Searching:** For searching element 2, we have to traverse all elements (assuming we do breadth first traversal). Therefore, searching in binary tree has worst case complexity of $O(n)$.
- **Insertion:** For inserting element as left child of 2, we have to traverse all elements. Therefore, insertion in binary tree has worst case complexity of $O(n)$.
- **Deletion:** For deletion of element 2, we have to traverse all elements to find 2 (assuming we do breadth first traversal). Therefore, deletion in binary tree has worst case complexity of $O(n)$.

→ Binary Search Tree (BST) –

BST is a special type of binary tree in which left child of a node has value less than the parent and right child has value greater than parent. Consider the left skewed BST shown in Figure 2.





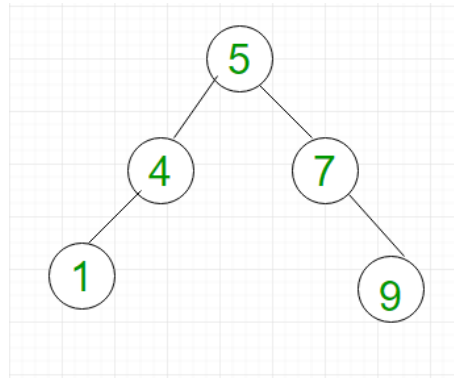
- **Searching:** For searching element 1, we have to traverse all elements (in order 3, 2, 1). Therefore, searching in binary search tree has worst case complexity of $O(n)$. In general, time complexity is $O(h)$ where h is height of BST.
- **Insertion:** For inserting element 0, it must be inserted as left child of 1. Therefore, we need to traverse all elements (in order 3, 2, 1) to insert 0 which has worst case complexity of $O(n)$. In general, time complexity is $O(h)$.
- **Deletion:** For deletion of element 1, we have to traverse all elements to find 1 (in order 3, 2, 1). Therefore, deletion in binary tree has worst case complexity of $O(n)$. In general, time complexity is $O(h)$.

AVL/ Height Balanced Tree –

AVL tree is binary search tree with additional property that difference between height of left sub-tree and right sub-tree of any node can't be more than 1. For example, BST shown in Figure 2 is not AVL as difference between left sub-tree and right sub-tree of node 3 is 2. However, BST shown in Figure 3 is AVL tree.

--->





- **Searching:** For searching element 1, we have to traverse elements (in order 5, 4, 1) = 3 = $\log_2 n$. Therefore, searching in AVL tree has worst case complexity of $O(\log_2 n)$.
- **Insertion:** For inserting element 12, it must be inserted as right child of 9. Therefore, we need to traverse elements (in order 5, 7, 9) to insert 12 which has worst case complexity of $O(\log_2 n)$.
- **Deletion:** For deletion of element 9, we have to traverse elements to find 9 (in order 5, 7, 9). Therefore, deletion in binary tree has worst case complexity of $O(\log_2 n)$.

We will discuss questions based on complexities of binary tree operations.

Que-1. What is the worst case time complexity for search, insert and delete operations in a general Binary Search Tree?

- (A) $O(n)$ for all
- (B) $O(\log n)$ for all
- (C) $O(\log n)$ for search and insert, and $O(n)$ for delete
- (D) $O(\log n)$ for search, and $O(n)$ for insert and delete

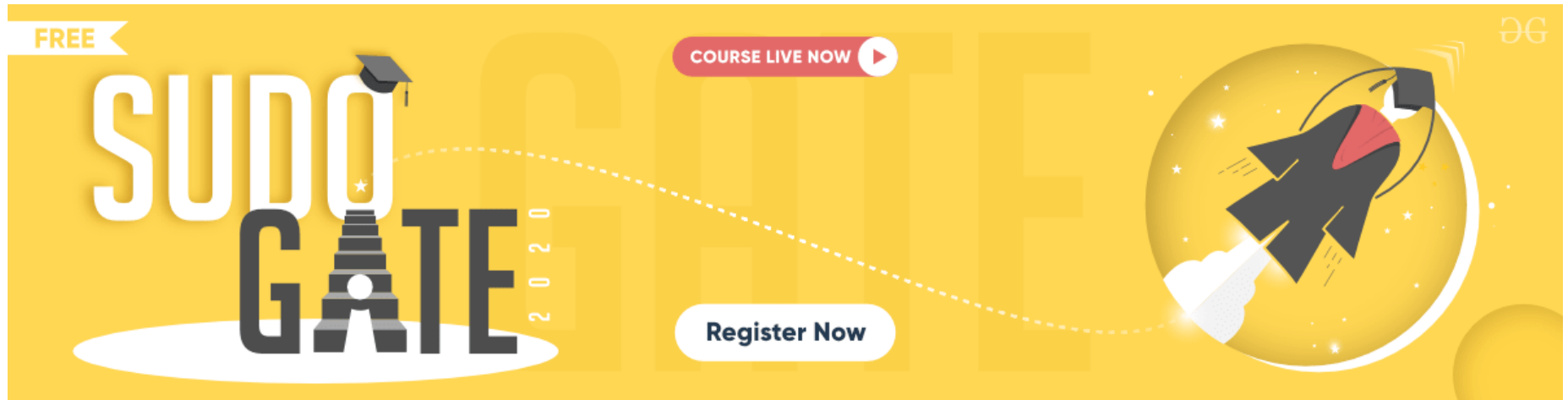
→ **Solution:** As discussed, all operations in BST have worst case time complexity of $O(n)$. So, the correct option is (A).

Que-2. What are the worst case time complexities of searching in binary tree, BST and AVL tree respectively?

- (A) $O(n)$ for all
- (B) $O(\log n)$ for all
- (C) $O(n)$ for binary tree, and $O(\log n)$ for others
- (D) $O(n)$ for binary tree and BST, and $O(\log n)$ for AVL



Solution: As discussed, search operation in binary tree and BST have worst case time complexity of $O(n)$. However, AVL tree has worst case time complexity of $O(\log n)$. So, the correct option is (D).



Recommended Posts:

- Minimum swap required to convert binary tree to binary search tree
- Difference between Binary Tree and Binary Search Tree
- Binary Tree to Binary Search Tree Conversion using STL set
- Binary Tree to Binary Search Tree Conversion
- Count the Number of Binary Search Trees present in a Binary Tree
- Check whether a binary tree is a full binary tree or not | Iterative Approach
- Convert a Binary Tree to Threaded binary tree | Set 1 (Using Queue)
- Convert a Binary Tree to Threaded binary tree | Set 2 (Efficient)
- Iterative Search for a key 'x' in Binary Tree
- Search a node in Binary Tree
- Check if a binary tree is subtree of another binary tree | Set 2



Check if a binary tree is subtree of another binary tree | Set 1

Check whether a binary tree is a full binary tree or not

Check whether a given binary tree is skewed binary tree or not?

Convert an arbitrary Binary Tree to a tree that holds Children Sum Property

This article is contributed by **Sonal Tuteja**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Improved By : [nitinshivanandmesta](#)

Article Tags : [Analysis](#) [GATE CS](#) [Tree](#)

Practice Tags : [Tree](#)



17

1.3

☐ To-do ☐ Done

Based on **8** vote(s)

[Feedback/ Suggest Improvement](#)

[Add Notes](#)

[Improve Article](#)



Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

PRACTICE

Courses
Company-wise
Topic-wise
How to begin?

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved

