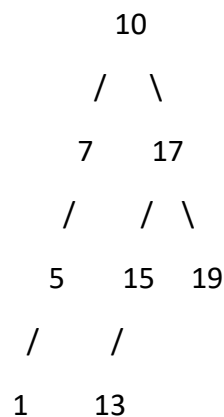


Find the Kth Smallest Element in Binary Search Tree

[LeetCode](#)

Given the root of a binary search tree, and an integer k , return *the k^{th} smallest value (1-indexed) of all the values of the nodes in the tree.*

Example:



$K = 4$

Output: The 4 Smallest Element of BST: 10

Approach 1: Find the kth Smallest Element in a Binary Search Tree using the Inorder Traversal

- **Function Purpose:** This approach finds the Kth smallest element in a BST.
- **Explanation:**
 1. The function performs an inorder traversal of the tree and stores the result in the 'inorderAns' vector.
 2. It returns the Kth smallest element from the vector (1-based index).
- **Time Complexity:** The time complexity of this approach is $O(N)$, where N is the number of nodes in the tree, as it needs to visit all nodes.
- **Space Complexity:** The space complexity is $O(N)$ due to the vector used to store the inorder traversal values.

Approach 2: Find the kth Smallest Element in a Binary Search Tree (Optimized)

- **Function Purpose:** This approach finds the Kth smallest element in a BST in an optimized way.
- **Explanation:**

1. The function uses an optimized approach without storing the entire inorder traversal.
 2. It uses a helper function to keep track of the current index while traversing the tree in ascending order.
 3. When the index matches K, it returns the current node's value as the Kth smallest element.
- **Time Complexity:** The time complexity of this approach is $O(H)$, where H is the height of the tree. In the worst case, it's $O(N)$ for a skewed tree.
 - **Space Complexity:** The space complexity is $O(H)$ due to the recursive call stack, where H is the height of the tree. In the worst case, it's $O(N)$ for a skewed tree.

Conclusion:

- Approach 1 uses an inorder traversal to find the Kth smallest element and is straightforward to implement.
- Approach 2 provides an optimized solution in terms of space complexity, making it more efficient for large trees in terms of memory usage.
- Both approaches accurately find the Kth smallest element in a BST.
- The choice between the two approaches depends on the specific requirements and characteristics of the tree being processed.