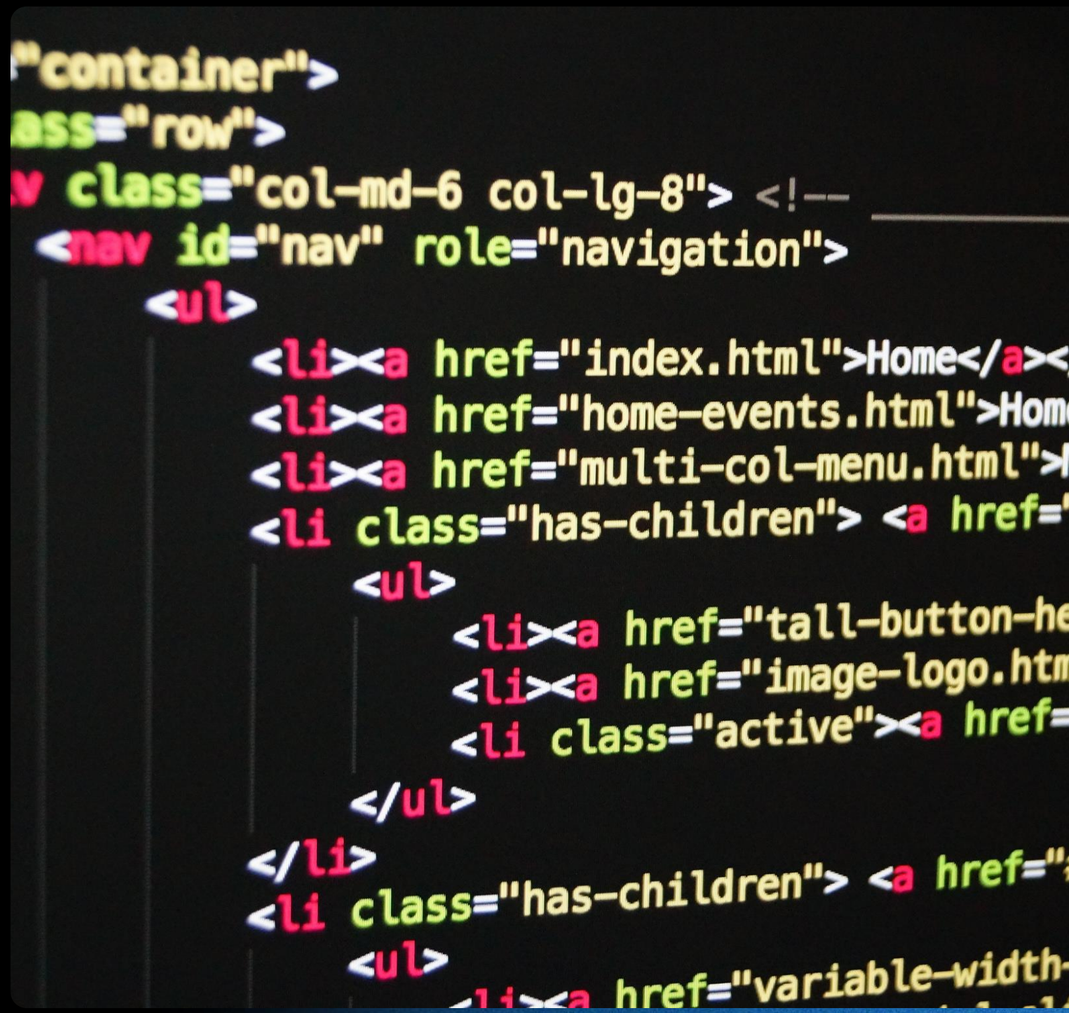
Project

Finding the Kth Ancestor in a Binary Tree

Osaid Ahmed Saeed

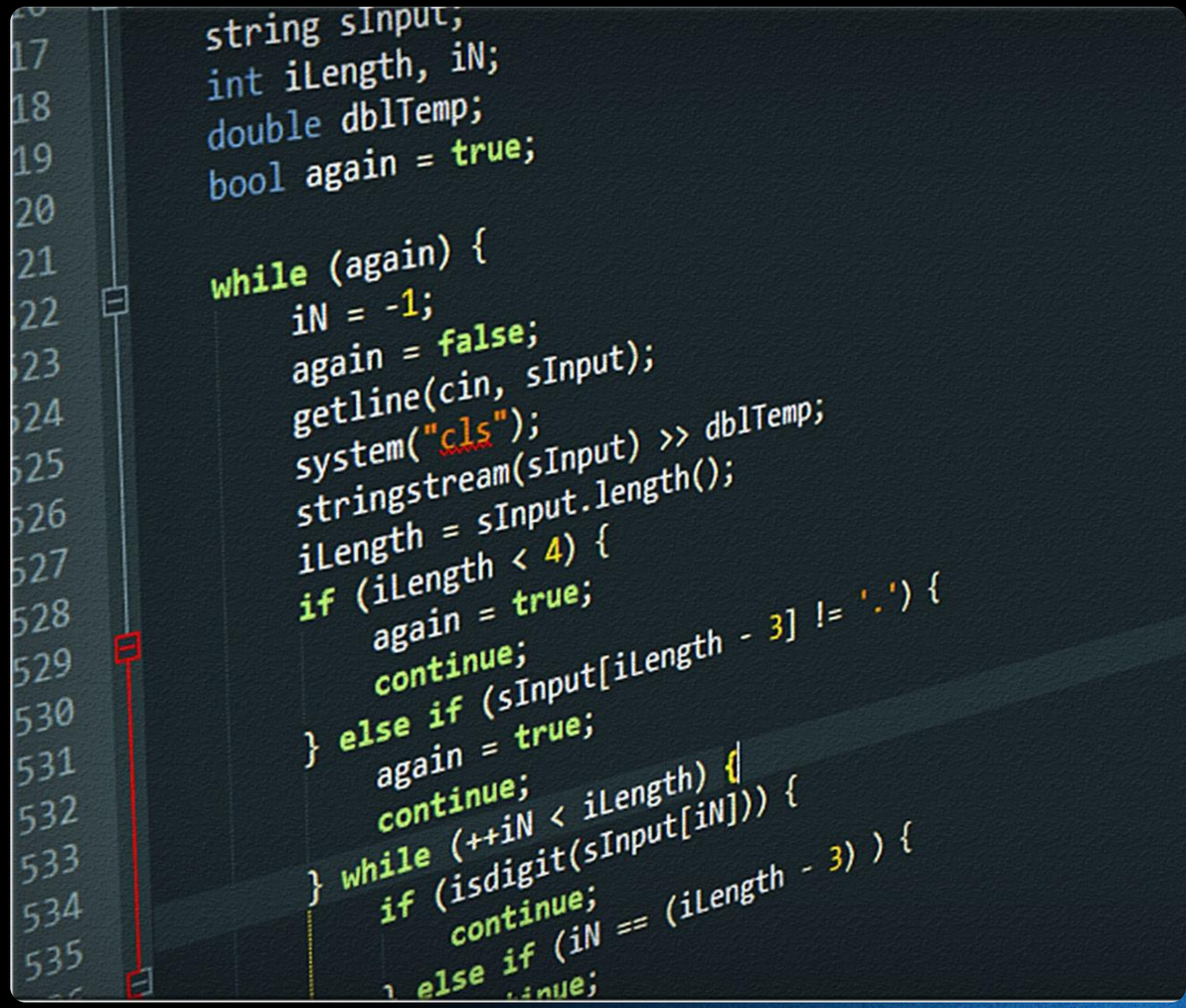
Introduction

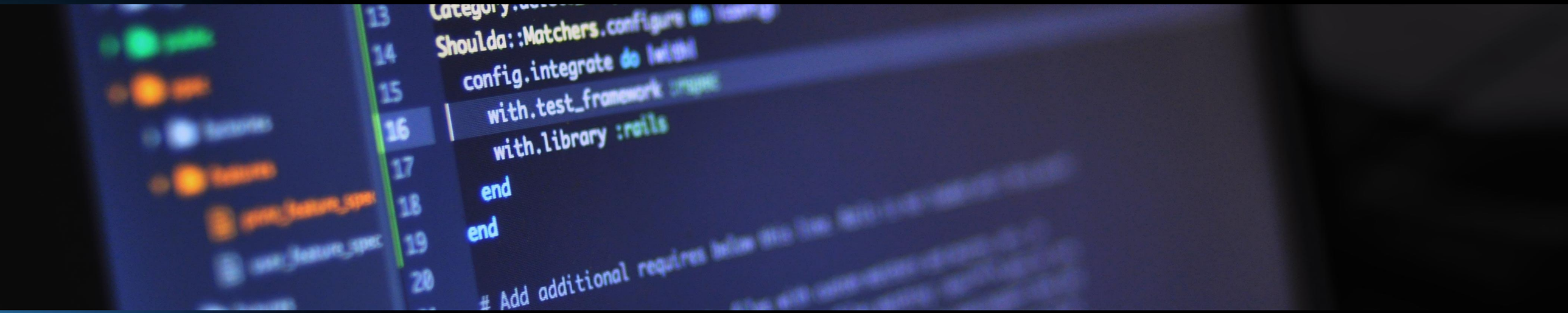
- The concept of ancestors in a binary tree.
- How we can determine the Kth ancestor of a given node.
- Recursion as an approach to solving this problem.



Problem Statement

- Given a binary tree and a specific node, find its Kth ancestor.
- The ancestor at Kth level above the node is required.
- If the Kth ancestor does not exist, return -1.





Binary Tree Structure

- A binary tree is a hierarchical structure where each node has at most two children.
- Parent-child relationships define the hierarchy.
- The importance of left and right child pointers in traversal.

Understanding Tree Traversal

- The process of visiting nodes in a specific order.
- Depth-First Search (DFS) helps in finding ancestors.
- Recursion is used to traverse up the tree.

Recursive Approach for Finding Kth Ancestor

- Start from the root and search for the target node.
- Once the node is found, move upwards to count ancestors.
- Reduce K each time a parent is found.
- Stop when K becomes zero, returning the ancestor.

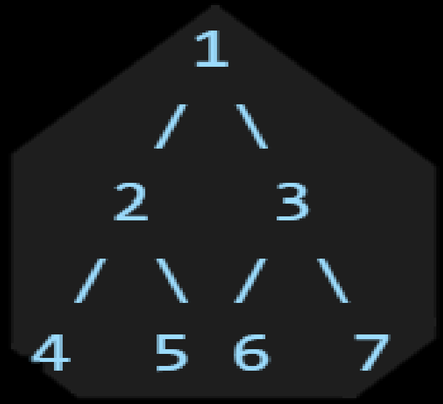
Working of Kth Ancestor Search

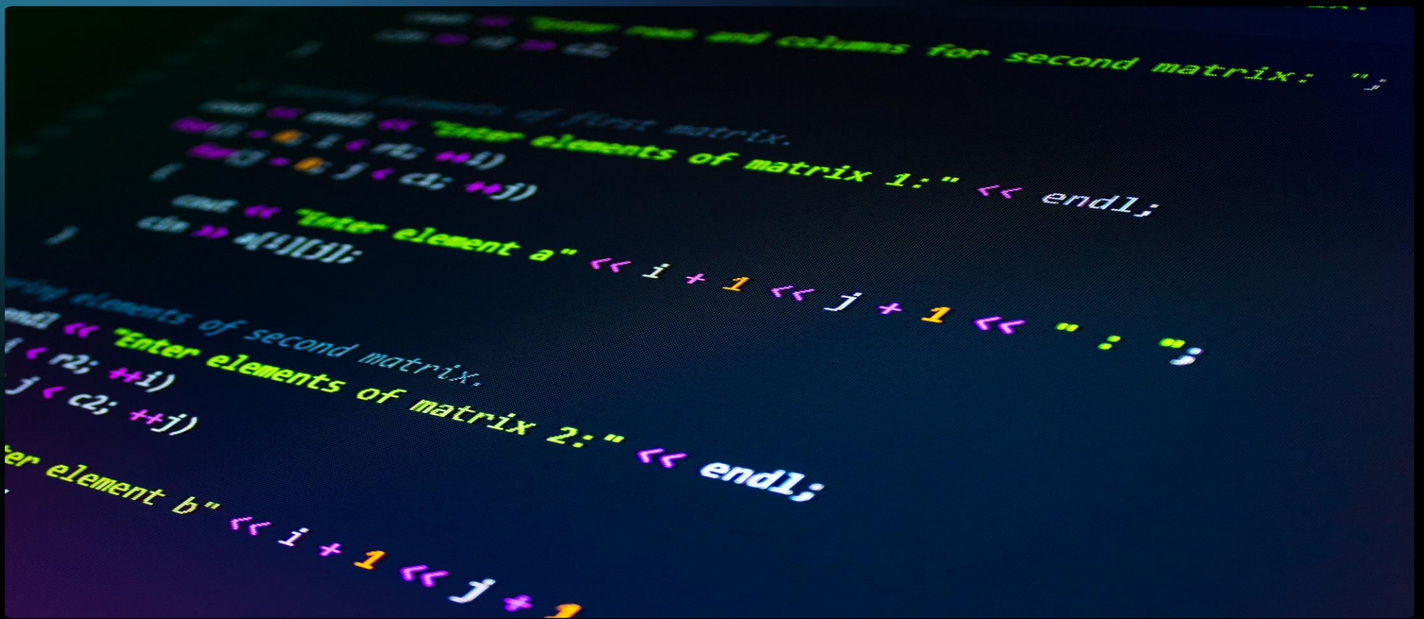
- The function explores both left and right subtrees.
- If the target node is found in a subtree, backtrack upwards.
- Maintain a counter for K to determine when to stop.

Sample Binary Tree Structure

- A simple binary tree example:
- Root node with two children.
- Each child has its own left and right children.

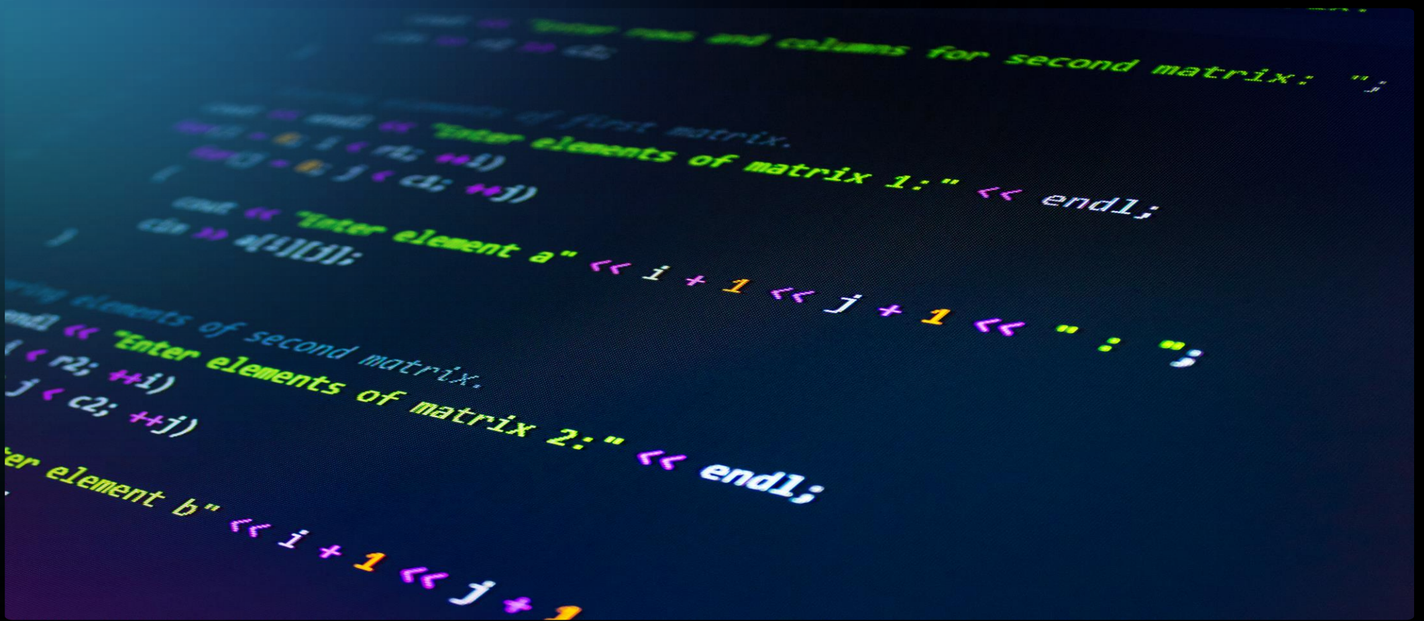
- Input: target = 5, k = 2
- Output: The 2nd ancestor of node 5 is 1





Output Example

- Implemented a recursive approach to find the Kth ancestor.
- Efficiently tracks ancestors during the traversal.
- Can be extended to handle larger trees and different K values.



Conclusion


THANK YOU
THANK YOU