

Intuition:

The Intuition is to perform two BFS traversals: one to build the parent-child relationship and another to find all nodes at a distance of k from the target node.

Explanation:

1. Start a BFS from the root node of the binary tree. We will use a queue to perform the BFS traversal. Initialize an empty queue and push the root node into it.
2. While the queue is not empty, perform the following steps:
 - Get the current size of the queue (number of nodes at the current level).
 - Iterate through all the nodes at the current level.
 - For each node, check if it has a left child. If it does, store the parent-child relationship by mapping the left child's value to its parent node in an `unordered_map` called "parent".
 - Similarly, if the current node has a right child, store the parent-child relationship for the right child.
 - After processing the children of the current node, remove it from the queue.
3. After the BFS traversal, we will have the parent-child relationship stored in the "parent" `unordered_map`.
4. Now, we need to perform another BFS traversal to find all the nodes at a distance of k from the target node. Initialize another empty queue and push the target node into it.
5. While the value of k is greater than 0 and the queue is not empty, perform the following steps:
 - Get the current size of the queue (number of nodes at the current level).
 - Iterate through all the nodes at the current level.
 - Mark the current node as visited by adding its value to the "visited" `unordered_map`.
 - If the current node has a left child and it has not been visited yet, push it into the queue.
 - If the current node has a right child and it has not been visited yet, push it into the queue.
 - If the current node has a parent and the parent node has not been visited yet, push the parent node into the queue.
 - After processing the current node and its children/parent, remove it from the queue.
6. After the BFS traversal, we will have all the nodes at a distance of k from the target node stored in the queue.
7. Finally, we can extract the values of the nodes from the queue and store them in a vector called "ans". Return the "ans" vector as the result.