

Logic Used to implement Functions

1. Logic Used:

- **Insert function** : Search for a suitable location and then insert. Maintain **Threaded Tree Structure** using boolean variables in Node structure. If the location already has an element, return and output “duplicate node”, otherwise fix its position.
- **Delete function** : Search for an element to be deleted. There may be 3 cases 1)deletion of leaf node, 2) deletion of node with 1 child 3) deletion of node with 2 children. Code follows caseA, caseB, caseC for the same. Maintain **Threaded Tree Structure**. If the element is not found, output “Deletion can't happen”.
- **Search function** : Search for the element recursively.
- **Reverse Inorder** : Go to the rightmost node of the tree. Until root is reached, traverse in left mode fashion and add it to the Singly Linked list. Add root. Find the min element in the left subtree and continue left traversal until NULL.
- **Successor**: Each node in our threaded binary tree contains link to inorder successor. Return the right node of the given node. If NULL is output “No Successor”.
- **Split**: Search for a node which has the key value of the argument. Split based on this node and write in-order traversal of the threaded tree.
- **allElementsBetween(k1, k2)**: Traverse in in-order fashion. Find a smaller key among k1 and k2 and return a list having elements between them.
- **kthElement(k)**: Uses reverse in-order call to the function to find out kth largest element.
- **PrintTree**: Gives output a DOT File which can be used to visualize the tree.