

Queue & Binary Tree - Exercise

Problem 1: Cool drink Dispenser:

A cool drink dispenser, which holds only Fanta and Coke, operates on a strictly "first in, first out" basis. It will dispense either the "oldest" (based on arrival time) of all the available cool drinks at the dispenser, or, based on the preference of the customer, it will dispense Fanta or Coke (the oldest of that type). Create a data structure to maintain this system and implement the following operations: enqueue, dequeueAny, dequeueFanta, dequeueCoke.

Problem 2: Binary Search Tree:

Implement the following functions in a binary search tree:

- void add(struct TreeNode * root, int val);
- void remove(struct TreeNode * root, int val);
- int search(struct TreeNode * root, int val);
- void get_inorder(struct TreeNode * root, int *arr, int *count);
- void get_postorder(struct TreeNode * root, int *arr, int *count);
- int get_height(struct TreeNode * root);
- int count_leaf_nodes(struct TreeNode * root,);
- int count_non_leaf_nodes(struct TreeNode * root,);

The function 'add' should allocate a node, assign 'val' to the node, and add the node to the tree pointed by 'root'.

The function 'remove' should search for a node with value 'val', remove the node from the tree, and free the memory for the node.

The function 'search' should scan the tree for a node with value 'val'. If such a node is found, it should return 1, 0 otherwise.

The function 'get_inorder' should do an inorder traversal of the tree and store the values in the tree (in inorder form) in 'arr' array. It should also return the number of nodes in the tree.

The function 'get_postorder' should do a postorder traversal of the tree and store the values in the tree (in postorder form) in 'arr' array. It should also return the number of nodes in the tree.

The function 'get_height' should return the height of the binary search tree. The height of a binary search tree is defined as the number of edges in the longest path from root to any leaf node. For example, the height of the tree in the introductory section is 2. That is, there are two edges in the longest path from root to any leaf. Alternately, the height can also be thought of as the number of nodes in the longest path minus 1.

The function 'count_leaf_nodes' should return the number of leaf nodes in the binary search tree.

The function 'count_non_leaf_nodes' should return the number of non-leaf nodes in the binary search tree.