Systems Programming Assessed Exercise 1

Robert De Venny (2892269d)

Binary Search Tree Operations

Search

The search method traverses the tree using its binary search property (left child values are smaller, right child values are larger).

- If the target value is not found, the function returns `NULL`.
- If the current node matches the target value, it is returned.
- Otherwise, the method determines whether to search the left or right subtree based on comparisons with the current node's value.

Generation

Creation: A new `node_t` is initialized, memory is allocated, and its value is set to the input argument. Child pointers are set to `NULL`, and the node is returned.
Destroy: A post-order recursive algorithm is used to free memory from the leaves upward, ensuring no dangling pointers or orphaned children remain. This approach safely removes all nodes.

Maintenance

Insertion: The insertion method iteratively traverses the tree until it finds a suitable position for a new value.

- If the value is less than the current node, the method explores the left subtree; otherwise, it explores the right subtree.
- When a `NULL` node is reached, the `createTree` method is called to create a new node, which is then assigned as the left or right child of the current node.
- Duplicate values are identified when the new value matches an existing node's value and are not inserted.

Deletion: Removes a specified value while maintaining the tree's structure:

- **Leaf node:** Frees the node and updates the parent's pointer.
- **One child:** Replaces the node with its child, which is then freed.
- **Two children:** Finds the node's in-order successor (smallest value in the right subtree). Replaces the node's value with the successor's, then recursively deletes the successor.

Memory Management

Memory allocated during node creation is properly freed in the `destroy` and `delete` methods, ensuring no memory leaks. The post-order destruction algorithm prevents dangling pointers by freeing child nodes before their parents.