ECE 2230 MP5

TEST PLAN

lab5.c

Cameron L. Burroughs

Unit Drivers 0 - 3:

These drivers are simple drivers to test different orders of insertion and deletion in a simple binary search tree such as inserting and deleting the root, deleting a node with no children, deleting a node with children, and deleting a node with grandchildren.

Unit Driver 4:

This simple provided driver tests two special cases of insertion and deletion. It first inserts a 10 at the root, then attempts to insert another 10. As this is a repeated key, the original 10 is replaced with the new one at the root. Next, it deletes a 10 at the root, leaving an empty list.

Unit Driver 5:

This unit driver makes a non-optimal binary search tree and then tests the special condition of deleting a node that does not exist. The expected result is that, after 6 comparisons, the program identifies that 10 is not present in the tree and prints such.

Unit Driver 6:

This driver creates a non-optimal binary search tree then attempts to remove the root, 10. Because 10 has two children and grandchildren, the root must be replaced with a successor. In this case, the successor is 12 (to the right child and all the way to the left). Expected output is for 12 to be moved to the root. See results below.

A close up of a screen

Description automatically generated A close up of a screen

Description automatically generated

Unit Driver 7:

This is a simple test driver that tests insertions and deletions. In this case, all deletions are deletions of leaves. This driver does not test removals of nodes with children.

Unit Driver 8:

This driver tests removal of a node with one left child, and grandchildren. In this case, the left child should move to the position of the node being removed. In this case, the node removed is the root and the left child is moved into the position of the root. The grandchildren remain unchanged.

Unit Driver 9:

This driver is similar to driver 8, however, it tests the removal of a node with one right child. Again, the right child should be moved into the position of the node removed and the grandchildren should remain unchanged.

Unit Driver 10:

This driver was created to test the proper values for recent comparison count for the insert, remove, and search functions. With insertion, unless the node inserted is a repeat, the RCC count should always be even. To test this, a 28 is inserted then another 28 is inserted afterwards. The first time a 28 is inserted, it does not already exist in the list and therefore the RCC is an even number, 4, the next time a 28 is inserted it already exists in the list and the RCC count is 5. See results below. The functions tree\_search and tree\_remove should return the same RCC value because search is called right before remove. Again, if the number is not found, an even number should be the RCC value.

Text

Description automatically generated Text

Description automatically generated

Unit Drivers 11-12:

These drivers test removing a node with grandchildren. First, the driver tests removing the roots left node, a node with grandchildren. In this case, the successor should be the new left node of the root. Next, it conducts a similar test where the roots right node, another node with grandchildren, is removed. Again, the successor should be the new right node of the root. Finally, it tests removing the root, a node with two children and grandchildren on both sides. In this case, the successor becomes the new root. Driver 12 conducts a similar test to driver 11, however node 10 now has no right-left child instead of no left-right child.

Unit Driver 13:

This driver also tests the various removals of nodes with grandchildren, however in this case, the tree has 4 levels and the successor is on the farthest level from the root. This test verifies the correct operation of the removal of a node with two children and grandchildren.