Experiment Name / No.: 12

Camlin | Page No. 39
Date / /

BINARY SEARCH TREE USING LINKED LIST

AIM

Write a menu driven C program to implement a binary search tree using linked list and perform insertion, deletion, traversal and searching.

ALGORITHM

1 Start
2 Enter choice for choosing between, insertion, deletion, traversal, search
3 case 1 : insert
3.1 If root = NULL
3.1.1 Allocate memory for root
3.1.2 set root -> data = val
3.1.3 set root -> left = root -> right = NULL
3.2 Else
3.2.1 If val < root -> data
3.2.1.1 call insert (tree -> left, val) goto step 3
3.2.2 else
3.2.2.1 call insert (tree -> right, val) goto step 3
4 case 2 : Delete
4.1 If root = NULL
4.1.1 Node not found
4.2 If val < root -> data
4.2.1 set root -> left = delete (root -> left, val) goto step 4

Teacher's Signature:

4.3 Else if val > root -> data
4.3.1 set root -> right = delete(root -> right, val)
4.4 else
4.4.1 If root -> left != NULL and root -> right != NULL
4.4.1.1 set temp = inorder successor(root -> right)
4.4.1.2 set root -> data = temp -> data
4.4.1.3 set root -> right = delete(root -> right, temp -> data)
4.4.2 else
4.4.2.1 If root -> left = NULL and root -> right = NULL
4.4.2.1.1 free(root)
4.4.2.2 else if root -> left != NULL
4.4.2.2.1 set root = root -> left
4.4.2.3 Else set root = root -> right
5. inorder successor
5.1 Repeat until ptr != NULL and ptr -> left != NULL
5.1.1 set ptr = ptr -> left
5.2 return ptr, go to step 4.4.1.1
6. case 3 : search
6.1 If root = NULL or root -> data = val
6.1.1 return root
6.2 Else if val > root -> data
6.2.1 call search(root -> right, val); goto step 6
6.3 Else call search(root -> left, val), goto step 6
7. case 4: preorder traversal.
7.1 If ptr != NULL

Teacher's Signature:

7.1.1 Print ptr -> data
7.1.2 call preorder (ptr -> left)
7.1.3 call preorder (ptr -> right)
case 5: inorder traversal
8.1 if ptr != NULL
8.1.1 call inorder (ptr -> left)
8.1.2 print ptr -> data
8.1.3 call inorder (ptr -> right)
case 6: postorder traversal
9.1 if ptr != NULL
9.1.1 call postorder (ptr -> left)
9.1.2 call postorder (ptr -> right
9.1.3 print ptr -> data

CONCLUSION

The program has been executed correctly and output has been verified