To construct an Abstract Data Type (ADT) for a binary search tree, it can be implemented by first defining an array and inserting the data one by one. The function 'BuildBst' is responsible for storing the data from array 'a' into a binary search tree by a loop. For every key, it is stored in a 'bstnode' structure that has one key, a left child, and a right child. By traversing the array 'a', the 'InsertBst' function inserts the key into the binary search tree through recursion. 图形用户界面, 文本

描述已自动生成

The 'SearchBst' function is designed to search for a key in the binary search tree. It returns the position of the key if successful, and NULL pointer if the search fails through recursion.文本

描述已自动生成

The 'DeleteBst' function is designed to search for and delete a key. It returns 1 if the deletion is successful, and 0 otherwise. The recursion can either find the key or reach a NULL pointer. If a key is found, two cases are considered: if the node has fewer than two children, the 'DeleteNode' function is executed. If the node has two children, it will find the smallest key in the right subtree using the inorder traversal method. Then, the four cases will be evaluated, and the delete node will be exchanged with the smallest node from the subtree.文本

描述已自动生成

The ‘PrintBst’ function will print the binary search tree using the preorder traversal method.

文本

描述已自动生成