**《数据结构综合设计》实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 人工智能与大数据学院 | 专业 | 虚拟现实技术 | 班级 | 21级2班 | 学生姓名 | 门红岩 |
| 实验  周次 | 10-12 | 实验  日期 | 2023.5.5 | 学时 | 6 | 教师姓名 | 李昊康 |
| 项目名称 | | 查找的应用 | | | | | |
| 实验  类别 | 🗹验证型实验 🞎设计型实验 🞎综合型实验 🞎其它 | | | | | 成绩：90 | |
| 1. 实验目的及具体要求   实验目的：  1. 掌握查找的不同方法，并能用高级语言实现查找算法；  2. 熟练掌握二叉排序树的构造和查找方法。  3. 熟练掌握静态查找表及哈希表查找方法。  具体要求：  从空的二叉树开始，每输入一个结点数据，就建立一个新结点插入到当前已生成的二叉排序树中。  在二叉排序树中查找某一结点。   1. 实验仪器、设备和材料   装有并能运行VS2019的电脑。   1. 实验内容、步骤及实验数据记录   #include <stdio.h>  #include <stdlib.h>  typedef struct node {  int data;  struct node\* left;  struct node\* right;  } Node;  Node\* createNode(int data) {  Node\* newNode = (Node\*)malloc(sizeof(Node));  newNode->data = data;  newNode->left = NULL;  newNode->right = NULL;  return newNode;  }  Node\* insert(Node\* root, int data) {  if (root == NULL) {  return createNode(data);  }  if (data < root->data) {  root->left = insert(root->left, data);  }  else if (data > root->data) {  root->right = insert(root->right, data);  }  return root;  }  void inorderTraversal(Node\* root) {  if (root != NULL) {  inorderTraversal(root->left);  printf("%d ", root->data);  inorderTraversal(root->right);  }  }  int main() {  Node\* root = NULL;  int data;  printf("Enter data for nodes in the BST (enter -1 to stop):\n");  while (1) {  scanf\_s("%d", &data);  if (data == -1) {  break;  }  root = insert(root, data);  }  printf("Inorder traversal of BST:\n");  inorderTraversal(root);  return 0;  }  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  typedef struct Node {  char data[20];  struct Node\* left;  struct Node\* right;  } Node;  Node\* createNode(char\* data) {  Node\* newNode = (Node\*)malloc(sizeof(Node));  strcpy\_s(newNode->data, sizeof(newNode->data), data);  newNode->left = NULL;  newNode->right = NULL;  return newNode;  }  Node\* insert(Node\* root, char\* data) {  if (root == NULL) {  return createNode(data);  }  if (strcmp(data, root->data) < 0) {  root->left = insert(root->left, data);  }  else if (strcmp(data, root->data) > 0) {  root->right = insert(root->right, data);  }  return root;  }  Node\* search(Node\* root, char\* data) {  if (root == NULL || strcmp(root->data, data) == 0) {  return root;  }  if (strcmp(data, root->data) < 0) {  return search(root->left, data);  }  else {  return search(root->right, data);  }  }  void inorderTraversal(Node\* root) {  if (root == NULL) {  return;  }  inorderTraversal(root->left);  printf("%s ", root->data);  inorderTraversal(root->right);  }  int main() {  Node\* root = NULL;  root = insert(root, "李");  insert(root, "云");  insert(root, "豪");  insert(root, "小");  insert(root, "不");  insert(root, "点");  insert(root, "哈");  printf("二叉排序树的中序遍历结果为：\n");  inorderTraversal(root);  printf("\n");  char data[20] = "不";  Node\* result = search(root, data);  if (result == NULL) {  printf("在二叉排序树中未找到%s。\n", data);  }  else {  printf("在二叉排序树中找到%s。\n", data);  }  return 0;  }  分析： | | | | | | | |

说明：1. 实验周次：填写实际上课周，如第5-8周上课填“5-8”或第10周上课填“10”。

1. 实验报告各部分内容需详实填写，按实验指导书上的评分标准给出分数。
2. 实验类型参考实验类型说明文件。