**《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 | 田一曼 |
| 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：94 | |
| 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;  }  void insertNode(Node \*\*root, int data) {  if (\*root == NULL) {  \*root = createNode(data);  return;  }  if (data < (\*root)->data)  insertNode(&((\*root)->left), data);  else  insertNode(&((\*root)->right), data);  }  void inorderTraversal(Node \*root) {  if (root == NULL)  return;  inorderTraversal(root->left);  printf("%d ", root->data);  inorderTraversal(root->right);  }  int main() {  Node \*root = NULL;  int data;  printf("Enter node data (enter -1 to stop): ");  scanf("%d", &data);  while (data != -1) {  insertNode(&root, data);  printf("Enter node data (enter -1 to stop): ");  scanf("%d", &data);  }  printf("Inorder traversal of binary search tree: ");  inorderTraversal(root);  printf("\n");  return 0;  }    #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;  }  void insertNode(Node \*\*root, int data) {  if (\*root == NULL) {  \*root = createNode(data);  return;  }  if (data < (\*root)->data)  insertNode(&((\*root)->left), data);  else  insertNode(&((\*root)->right), data);  }  Node \*searchNode(Node \*root, int data) {  if (root == NULL)  return NULL;  if (root->data == data)  return root;  else if (data < root->data)  return searchNode(root->left, data);  else  return searchNode(root->right, data);  }  int main() {  Node \*root = NULL;  int data, search;  printf("Enter node data (enter -1 to stop): ");  scanf("%d", &data);  while (data != -1) {  insertNode(&root, data);  printf("Enter node data (enter -1 to stop): ");  scanf("%d", &data);  }  printf("Enter data to search: ");  scanf("%d", &search);  Node \*found = searchNode(root, search);  if (found == NULL)  printf("%d not found in binary search tree\n", search);  else  printf("%d found in binary search tree\n", search);  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。