

# 软件学院设计性实验报告

学院：软件学院                      专业： 智能应用                      年级/班级： 大二/2 班  
2024—2025 学年第一学期

课程名称	数据结构	指导教师	张磊
学号姓名	2328624035 李陈歆烨		
实验地点	向真楼 403	实验时间	11 月 20 日
项目名称	二叉链表存储结构的建立及遍历	实验类型	综合性

一、实验目的（1）构建二叉树的二叉链表；（2）先序遍历二叉树中所有结点；（3）中序遍历二叉树中所有结点；（4）后序遍历二叉树中所有结点；（5）层次遍历二叉树中所有结点。

。

二、实验仪器或设备

学院提供公共机房，1 台微型计算机/学生

三、实验说明（设计方案）

1) 编写完成下列功能的函数：（1）构建二叉树的二叉链表；（2）先序遍历二叉树中所有结点；（3）中序遍历二叉树中所有结点；（4）后序遍历二叉树中所有结点；（5）层次遍历二叉树中所有结点；

2) 用主函数调用你所编写的函数，以验证你编程序的正确性。

四．实验步骤（包括主要步骤、代码分析等）

```
#include <stdio.h>

#include <stdlib.h>

// 二叉树节点中数据元素的类型假设为 char，可按需替换

typedef char TElemType;

// 二叉树的二叉链表存储结构定义

typedef struct BiTNode {

    TElemType data;
```

```

        struct BiTNode *lchild, *rchild;
    } BiTNode, *BiTree;

// 按照先序遍历序列创建二叉树（以#表示空节点）
BiTree CreateBiTree() {
    char ch;

    scanf("%c", &ch);

    if (ch == '#') {
        return NULL;
    }

    BiTree T = (BiTree)malloc(sizeof(BiTNode));

    T->data = ch;

    T->lchild = CreateBiTree();

    T->rchild = CreateBiTree();

    return T;
}

// 先序遍历二叉树
void PreOrderTraverse(BiTree T) {
    if (T == NULL) {
        return;
    }

```

```
    printf("%c ", T->data);

    PreOrderTraverse(T->lchild);

    PreOrderTraverse(T->rchild);

}
```

// 中序遍历二叉树

```
void InOrderTraverse(BiTree T) {

    if (T == NULL) {

        return;

    }

    InOrderTraverse(T->lchild);

    printf("%c ", T->data);

    InOrderTraverse(T->rchild);

}
```

// 后序遍历二叉树

```
void PostOrderTraverse(BiTree T) {

    if (T == NULL) {

        return;

    }

    PostOrderTraverse(T->lchild);

    PostOrderTraverse(T->rchild);

}
```

```

        printf("%c ", T->data);
    }

// 定义一个结构体用于辅助层次遍历（队列节点）
typedef struct LinkNode {
    BiTree data;
    struct LinkNode *next;
} LinkNode;

// 定义队列结构
typedef struct {
    LinkNode *front, *rear;
} LinkQueue;

/ 初始化队列
void InitQueue(LinkQueue *Q) {
    Q->front = Q->rear = (LinkNode*)malloc(sizeof(LinkNode));
    Q->front->next = NULL;
}

// 判断队列是否为空
int IsEmpty(LinkQueue *Q) {

```

```
        return Q->front == Q->rear;
    }
}
```

// 入队操作

```
void EnQueue(LinkQueue *Q, BiTree e) {
    LinkNode *s = (LinkNode *)malloc(sizeof(LinkNode));
    s->data = e;
    s->next = NULL;
    Q->rear->next = s;
    Q->rear = s;
}
```

// 出队操作

```
BiTree DeQueue(LinkQueue *Q) {
    if (IsEmpty(Q)) {
        return NULL;
    }
    LinkNode *p = Q->front->next;
    BiTree e = p->data;
    Q->front->next = p->next;
    if (Q->rear == p) {
        Q->rear = Q->front;
    }
}
```

```
    }  
  
    free(p);  
  
    return e;  
}
```

// 层次遍历二叉树

```
void LevelOrderTraverse(BiTree T) {  
  
    LinkQueue Q;  
  
    InitQueue(&Q);  
  
    if (T!= NULL) {  
        EnQueue(&Q, T);  
    }  
  
    while (!IsEmpty(&Q)) {  
        BiTree p = DeQueue(&Q);  
        printf("%c ", p->data);  
        if (p->lchild!= NULL) {  
            EnQueue(&Q, p->lchild);  
        }  
        if (p->rchild!= NULL) {  
            EnQueue(&Q, p->rchild);  
        }  
    }  
}
```

```
}
```

```
int main() {  
    BiTree T = CreateBiTree();  
    printf("先序遍历结果: ");  
    PreOrderTraverse(T);  
    printf("\n");  
    printf("中序遍历结果: ");  
    InOrderTraverse(T);  
    printf("\n");  
    printf("后序遍历结果: ");  
    PostOrderTraverse(T);  
    printf("\n");  
    printf("层次遍历结果: ");  
    LevelOrderTraverse(T);  
    printf("\n");  
    return 0;  
}
```

结果分析与总结

（可参考以下内容：）

```
C:\Users\LENOVO\Desktop\7. x + v
ABD##E##CF##G##
先序遍历结果: A B D E C F G
中序遍历结果: D B E A F C G
后序遍历结果: D E B F G C A
层次遍历结果: A B C D E F G

-----
Process exited after 1.781 seconds with return value 0
请按任意键继续. . .
```

根据用户页面先序，中序，后序，层次遍历正确，结果正确

2024 年 10 月 22 日

**实验报告格式说明：**

1. 页脚插入页码：宋体小五号，居中
2. 课程名称标题：黑体四号加粗
3. 表格内容：宋体五号
4. 标题：黑体小四
5. 正文：宋体五号
6. 页面设置：纸张A4, 页边距上下2cm, 左右2.8cm