

《数据结构与算法》第九次作业练习题

开始时间 11/09/2023 8:30:00 PM 结束时间 11/14/2023 11:59:00 PM 答题时长 7409分钟

答卷类型 标准答案

总分 56

单选题

得分: 暂无 总分: 16

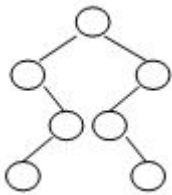
2-1 二叉树后序遍历为：CBEDA，中序遍历为：BCAED。则元素D的左孩子是 (2分)

- ☐ A. A
- ☐ B. B
- ☐ C. C
- ☒ D. E

2-2 如果二叉树的前序遍历结果是12345，后序遍历结果是24531，那么该二叉树的中序遍历结果是什么？ (3分)

- ☐ A. 23145
- ☐ B. 32154
- ☒ C. 21435
- ☐ D. 无法确定

2-3 已知一棵二叉树的树形如下图所示，其后序序列为{ e, a, c, b, d, g, f }。树中与结点a同层的结点是： (3分)



- ☐ A. c
- ☒ B. d
- ☐ C. f
- ☐ D. g

2-4 对M ($M \geq 2$) 个权值均不相同的字符构造哈夫曼树。下列关于该哈夫曼树的叙述中, 错误的是 () (2分)

- ☐ A. 树中一定没有度为1的结点
- ☐ B. 树中两个权值最小的结点一定是兄弟结点
- ☐ C. 树中任一非叶结点的权值一定不小于下一层任一结点的权值
- ☒ D. 该树一定是一棵完全二叉树

2-5 已知字符集{ a, b, c, d, e, f }，若各字符出现的次数分别为{ 6, 3, 8, 2, 10, 4 }，则对应字符集中各字符的哈夫曼编码（3分）可能是：

- ☒ A. 00, 1011, 01, 1010, 11, 100
- ☐ B. 00, 100, 110, 000, 0010, 01
- ☐ C. 10, 1011, 11, 0011, 00, 010
- ☐ D. 0011, 10, 11, 0010, 01, 000

2-6 对 n 个互不相同的符号进行哈夫曼编码。若生成的哈夫曼树共有 115 个结点, 则 n 的值是: (3分)

- ☐ A. 56
- ☐ B. 57
- ☒ C. 58
- ☐ D. 60

6-1 求叶子结点个数 (10分)

以二叉链表作为二叉树的存储结构，求二叉树的叶子结点个数。

函数接口定义：

```
void creat(BiTree &Tree)//创建二叉树
int countleaf(BiTree Tree)//叶子结点计数
```

其中 `Tree` 是指向二叉树根节点的指针。

裁判测试程序样例：

```
#include<stdio.h>
#include<malloc.h>
#define len sizeof(struct BiTNode )

typedef struct BiTNode
{
    char data; //数据域
    struct BiTNode *lchild; //左孩子指针
    struct BiTNode *rchild; //右孩子指针
}BiTNode,*BiTree;

void creat(BiTree &Tree);//构建二叉树
int countleaf(BiTree Tree);//求叶子结点个数

int main()
{
    BiTree Tree;
    creat(Tree);//创建二叉树
    printf("%d\n",countleaf(Tree)); //输入叶子节点个数
    return 0;
}

/* 请在这里填写答案 */
```

输入样例：

```
ABC##DE#G##F####
```

输出样例：

```
3
```

6-2 求二叉树高度 (10分)

二叉树采用二叉链表存储，求该二叉树的高度

函数接口定义：

```
void creat(BiTree &Tree)//构建二叉树
int Depth(BiTree Tree)//求高度
```

`Tree` 为指向二叉树的根节点的指针。

裁判测试程序样例：

```

#include<stdio.h>
#include<malloc.h>
#define len sizeof(struct BiTNode )

typedef struct BiTNode
{
    char data; //数据域
    struct BiTNode *lchild; //左孩子指针
    struct BiTNode *rchild; //右孩子指针
}BiTNode,*BiTree;

void creat(BiTree &Tree);//构建二叉树

int Depth(BiTree Tree);//求深度

int main()
{
    BiTree Tree;
    creat(Tree);//创建二叉树
    printf("%d\n",Depth(Tree));
    return 0;
}
/* 请在这里填写答案 */

```

输入样例:

```
ABC##DE#G##F###
```

输出样例:

```
5
```

编程题

得分：暂无 总分：20

7-1 根据后序序列和中序序列确定二叉树 (10分)

二叉树采用二叉链表存储，要求根据给定的后序遍历序列和中序遍历序列建立二叉树，并输出二叉树的深度及其先序遍历序列。

输入格式:

测试数据有多组，处理到文件尾。每组测试数据的第一行输入结点数 n ($1 \leq n \leq 10$)，第二、三行各输入 n 个整数，分别表示二叉树的后序遍历序列和中序遍历序列。

输出格式:

对于每组测试，在一行上分别输出该二叉树的深度及其先序遍历序列。每两个数据之间留一个空格。

输入样例:

```

9
7 4 2 8 9 5 6 3 1 9
4 7 2 1 8 5 9 3 6

```

输出样例:

```
4 1 2 4 7 3 5 8 9 6
```

来源:

7-2 哈夫曼树 (10分)

哈夫曼树，第一行输入一个数 n ，表示叶结点的个数。

需要用这些叶结点生成哈夫曼树，根据哈夫曼树的概念，这些结点有权值，即weight，题目需要输出哈夫曼树的带权路径长度（WPL）。

输入格式:

第一行输入一个数 n ，第二行输入 n 个叶结点（叶结点权值不超过1000， $2 \leq n \leq 1000$ ）。

输出格式:

在一行中输出WPL值。

输入样例:

```
5
1 2 2 5 9
```

输出样例:

```
37
```