第5次作业

## 第5章习题 P157

5.1画出由4个结点所构成的所有形态的树（假设是无序树）。

5.2已知一棵树的度为4，其中度为4的结点的数目为3，度为3的结点的数目为4，度为2的结点的数目为5，度为1的结点的数目为2，请求出该树中的叶子结点的数目。

5.4已知某完全二叉树有100个结点，试用三种不同的方法求出该二叉树的叶子结点数。

5.7设计算法以求解编号为i和j的两个结点的最近的公共祖先结点的编号。

5.8分别求出下图中二叉树的三种遍历序列。

C

H

E

B

G

A

A

J

B

C

E

D

F

I

D

图5-1 题5.8图

a

A

B

C

E

D

A

B

C

E

D

b

c

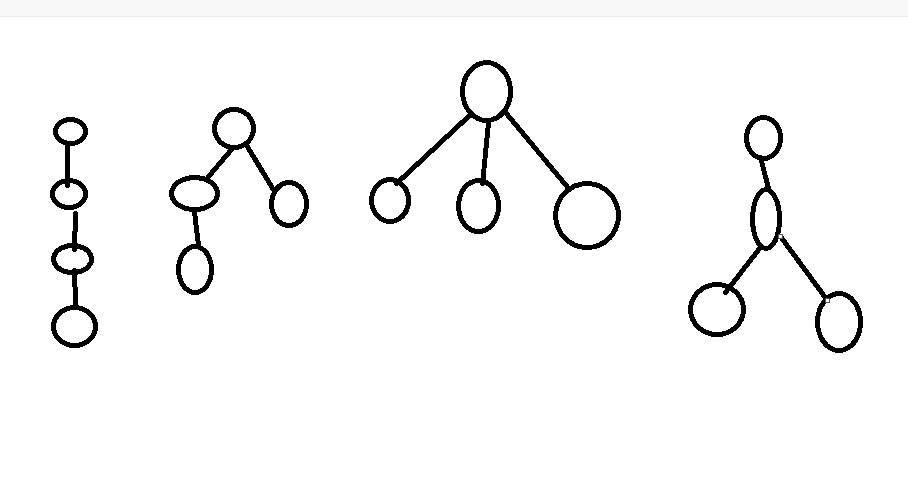
d

5.9分别描述满足下面条件的二叉树特征：

（1）先序序列和中序序列相同。

（2）先序序列和后序序列相反。

5.1



5.2 叶子结点的数目是23

5.4 n1=1 2\*(n2)+1=99 n2=49 则n0=50

5.7

#include<iostream>

using namespace std;

const int maxn = 1e3 + 10;

int a[maxn];

int n = 0;

int fun(int a, int b)//一个树的根结点是其所在位置/2的位置上，所以可以利用类似二分查找的方法来完成共同根结点的查找

{

if (a == b) return a;

if (a > b) return fun(a / 2, b);

if (a < b) return fun(a, b / 2);

}

int main()

{

cin >> n;

for (int i = 1; i <= n; i++) cin >> a[i];

int i, j = 0;

cin >> i >> j;

if (a[i] == 0) //当寻找的结点不存在时

cout << "ERROR: T[" << i << "]" << " is NULL";

else if (a[j] == 0)

cout << "ERROR: T[" << j << "]" << " is NULL";

else

{

int q = fun(i, j);

cout << q << " " << a[q] << endl;

}

return 0;

}

5.8

一

前序遍历ABCDEFGHIJ

中序遍历CDBEFAHGJI

后序遍历DCFEBHJIGA

二

前序遍历ABCDE

中序遍历EDCBA

后序遍历EDCBA

三

前序遍历ABCDE

中序遍历BDECA

后序遍历EDCBA

四

前序遍历ABCDE

中序遍历ABCDE

后序遍历EDCBA