一、调试成功程序及说明

1. 编程实现书P75 ADT BinaryTree 基本操作20个，用二叉链表结构实现；

算法思想：

有关遍历的算法思想见第二题。

建立二叉树：基于先序遍历。

求树高：使用递归方法，根树的树高等于左右子树树高最大值再加1.

返回父结点指针：利用层序遍历依次遍历结点，若该结点左孩子或右孩子为所指向的孩子，直接返回该结点。

返回左孩子指针：若其左孩子非空，直接返回即可。

返回右孩子指针：若其右孩子非空，直接返回即可。

删除子树：使用递归方法，先删除左子树，再删去右子树，最后释放根结点，并令其置空。

销毁二叉树：基于“删除子树”的函数，将树根作为参数传入“删除子树”的函数即可。

运行结果：

1.求树高

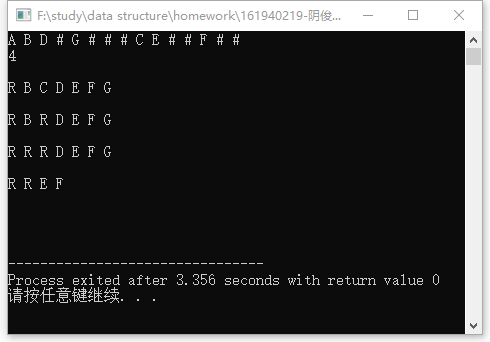
2.将根结点设为R

3.将E的父结点设为R

4.将根结点的左孩子设为R

5.删除根结点的左子树

6.销毁二叉树



结果分析：

求树高：

T(n) = O(n)

S(n) = O(1)

附源程序。

2.实现二叉树的先序、中序、后序遍历，用递归和非递归方法；实现层次遍历。

算法思想：

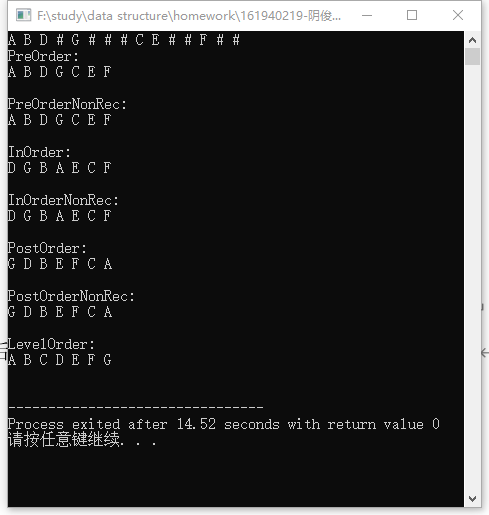
先序非递归：令p指向root，只要p非空，访问它并且入栈，若p空，回溯，弹栈并将p指向弹出元素，再以同样方式遍历右子树。若p空且栈空，表明回溯到根结点，遍历结束。

中序非递归：基本同先序非递归，唯一区别在于访问时机不同，先序非递归是在结点入栈时访问，而中序非递归是在结点出栈时访问。

后序非递归：令p指向root，只要p非空，访问它并且入栈，若p空，弹栈并将p指向弹出元素，若p所指右子树非空且p上一次访问的不是右子树的根结点，以同样的方式遍历右子树，反之访问p自身的数据，并继续回溯。若p空且栈空，表明回溯到根结点，遍历结束。

层序遍历：先将root入队，只要队列非空，出队，令p指向出队结点，访问它，并且令其左右子树依次入队（若非空），若队列空，表明遍历结束，循环终止。

运行结果：



结果分析：

先序遍历（非递归）：

T(n) = O(n)

S(n) = O(n)

中序遍历（非递归）：

T(n) = O(n)

S(n) = O(n)

后序遍历（非递归）：

T(n) = O(n)

S(n) = O(n)

层序遍历：

T(n) = O(n)

S(n) = O(n)

附源程序。

3. 设二叉树采用二叉链表存储，编写函数，对二叉树中每个元素值为x的结点，删除以它为根的子树，并释放相应空间。（习题集6.45）

算法思想：

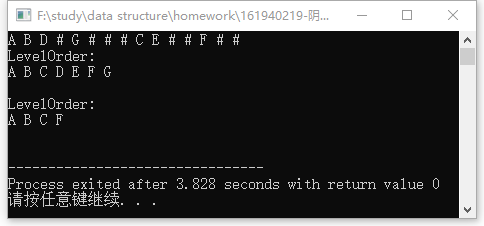
1.层序遍历树中每一结点

2.若该结点左孩子非空且元素值为x，删除左子树，并将其左孩子指针置空。

3.对右孩子操作同左孩子。

4.直到层序遍历结束，循环终止。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(n)

附源程序。

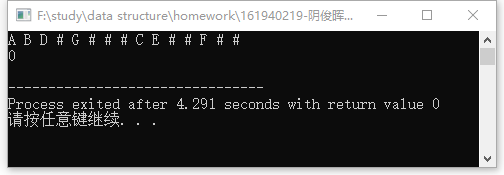
4. 编写函数，判断给定的二叉树是否是完全二叉树。（习题集6.49）

算法思想：

基于层序遍历。

层序遍历所有非空结点的指针，并将它们的左右指针（无论空或非空）都入队，若在队列中存在非空指针在空指针之后的情况，则该树不是完全二叉树，若遍历结束之后，所有空指针都在非空指针之后，则该树是完全二叉树。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(n)

附源程序。

5． CSP题目

题目描述：甲乙丙丁决定玩一个报数的游戏来打发时间。游戏规则为四个人从1开始轮流进行报数，但如果需要报出的数是7的倍数或含有数字7则直接跳过。

此外大家约定，在总共报出了n个数后（不计入被跳过的数）游戏结束。现在需要你来帮忙统计，游戏过程中每个人各自跳过了几次。

输入格式：

从标准输入读入数据。

输入仅一行，包含一个正整数n，表示报出了多少个数后游戏结束。

输出格式：

输出到标准输出。

输出共四行，每行一个整数，依次表示甲乙丙丁四人在游戏过程中跳过的次数。

样例1输入：

20

样例1输出：

2

1

1

0

样例1解释：

报数过程为:

甲：1，乙：2，丙：3，丁：4

甲：5，乙：6，丙：跳过，丁：8

甲：9，乙：10，丙：11，丁：12

甲：13，乙：跳过，丙：15，丁：16

甲：跳过，乙：18，丙：19，丁：20

甲：跳过，乙：22，丙：23，丁：24

在丁报出24后，四个人总计报出了20个数，游戏结束。

样例2输入：

66

样例2输出：

7

5

11

5

评测用例规模与约定：

保证n≤666。

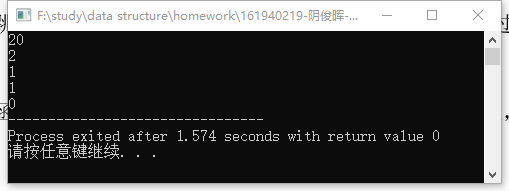
算法思想：

一重循环，初始化i为1，一直报数，直到未跳过的数的个数为n，循环终止。

若i应该跳过，对应报数者计数器加1，若i不该跳过，未跳过的数的个数加1。

Skip（）函数：若i模7为0或i对应字符串中含7，就跳过，反之不跳过。

运行结果：



结果分析：

T(n) = O(n\*strlen(n))

S(n) = O(1)

附源程序。

-------------------------------------------------------------------------

选做题**6.42** 求二叉树中叶子结点的数目

算法思想：

递归思想：根树叶结点数目等于左右子树的叶结点数目之和。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(n)

附源程序。

选做题**6.45**删除以元素值为x的结点为根的子树

***同第三题。***

/\*

算法思想：

运行结果：

结果分析：

T(n) = O()

S(n) = O()

附源程序。

\*/

选做题**6.47 树的层序遍历**

***同第二题。***

/\*

算法思想：

运行结果：

结果分析：

T(n) = O()

S(n) = O()

附源程序。

\*/

二、代码行数及小结

|  |  |
| --- | --- |
| 题目 | 代码行数 |
| binaryTree.h | 32 |
| binaryTree.cpp | 392 |
| 1 | 36 |
| 2 | 32 |
| 3 | 17 |
| 4 | 9 |
| 5 | 60 |
| 选6.42 | 9 |
| 选6.45 | 0 |
| 选6.47 | 0 |
| 总计 | 587 |