

数据结构与算法

实验三 树

实验目的

- ◆ 通过选择下面两个题目之一进行实现，掌握如下内容：
 - 掌握二叉树基本操作的实现方法
 - 了解哈夫曼树的相关概念
 - 学习使用二叉树解决实际问题的能力

题目一：二叉树

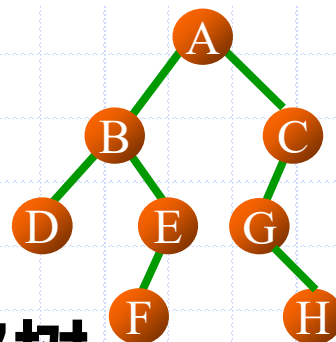
◆ 根据二叉树的抽象数据类型的定义，使用二叉链表实现一个二叉树。

◆ 二叉树的基本功能：

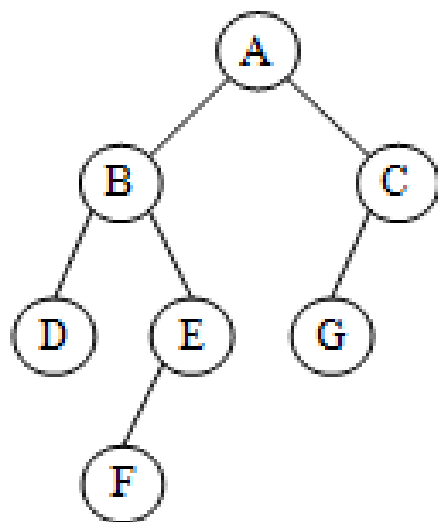
要求：

- 1、二叉树的建立
- 2、前序遍历二叉树
- 3、中序遍历二叉树
- 4、后序遍历二叉树
- 5、按层序遍历二叉树
- 6、求二叉树的深度
- 7、求指定结点到根的路径
- 8、二叉树的销毁
- 9、其他：自定义操作

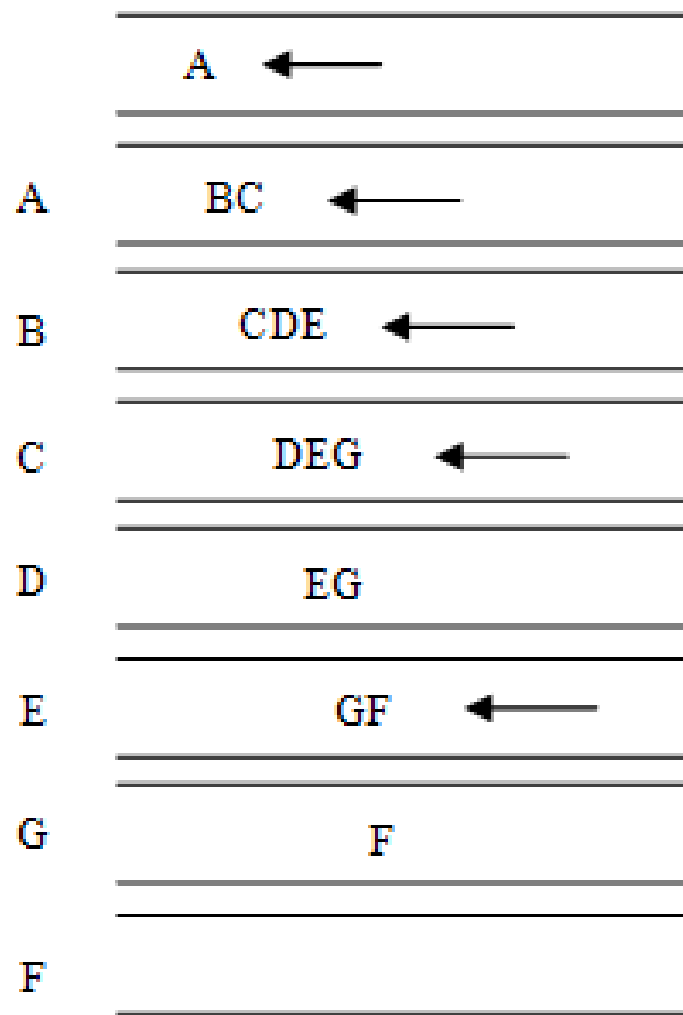
编写测试main()函数测试二叉树的正确性



题目一：二叉树



层序遍历: ABCDEGF



题目一：二叉树

◆ 思考问题（选作）：

1、若数据量非常大，如何使得构造二叉树时栈不溢出？使用非递归方式编写新的二叉树的构造函数，建立二叉树。

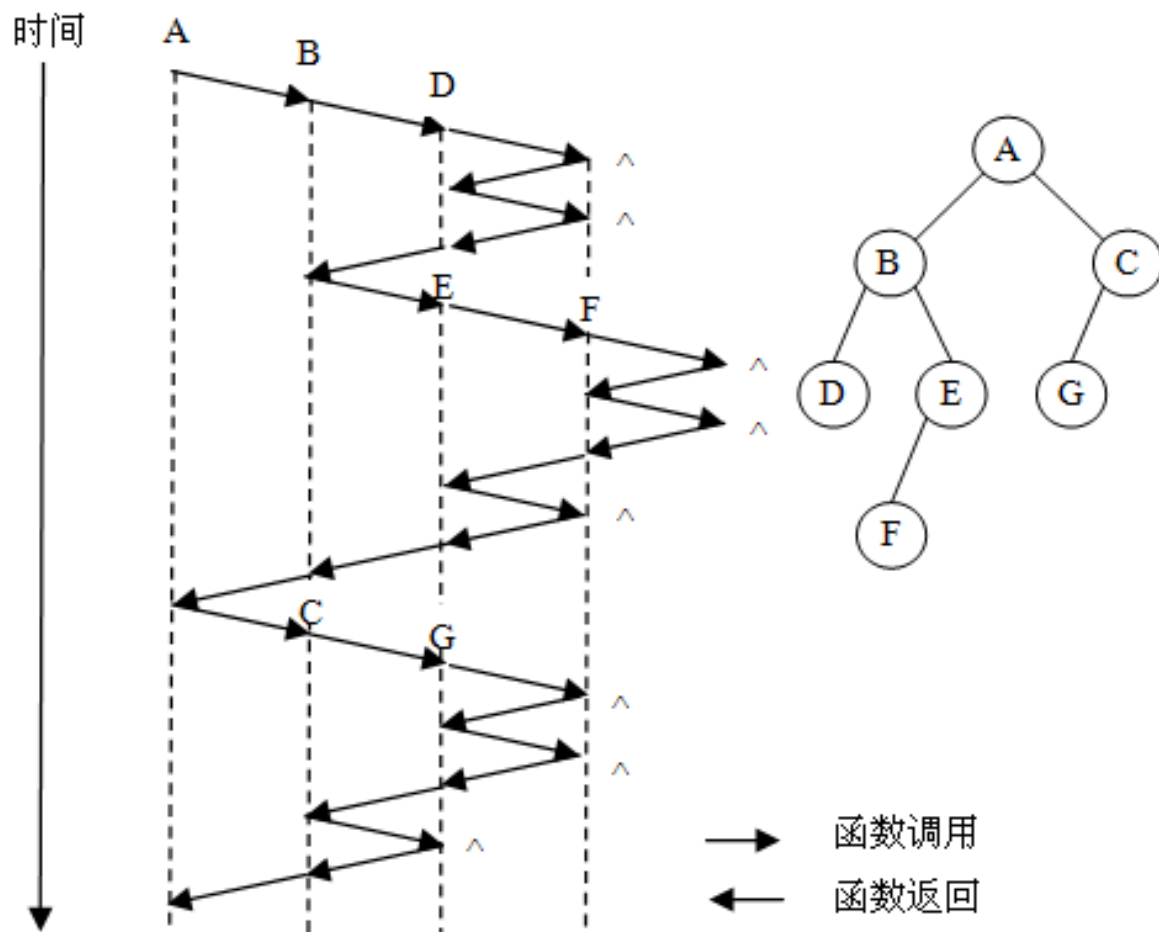
提示：可以使用STL中的stack来辅助实现。

2、若二叉树的每一个结点具有数值，如何搜索二叉树，找到指定值的叶子结点？

3、若已知叶子结点的指针，如何输出从根到该叶子的路径？

题目一：二叉树

位置	1	2	3	4	5	6	7	8	9	10	11	12	13	...
data	A	B	C	D	E	G	^	^	^	F	^	^	^	...



题目二：哈夫曼编/解码器

◆ 利用二叉树结构实现哈夫曼编/解码器。

基本要求：

- 1、初始化(Init)：**能够对输入的任意长度的字符串s进行统计，统计每个字符的频度，并建立哈夫曼树
- 2、建立编码表(CreateTable)：**利用已经建好的哈夫曼树进行编码，并将每个字符的编码输出。
- 3、编码(Encoding)：**根据编码表对输入的字符串进行编码，并将编码后的字符串输出。

题目二：哈夫曼编/解码器

- 4、译码(Decoding)：利用已经建好的哈夫曼树对编码后的字符串进行译码，并输出译码结果。
- 5、打印(Print)：以直观的方式打印哈夫曼树（选作）
- 6、计算输入的字符串编码前和编码后的长度，并进行分析，讨论赫夫曼编码的压缩效果。
- 7、可采用二进制编码方式（选作）

题目二：哈夫曼编/解码器

测试数据：

I love data Structure, I love Computer. I will try my best to study data Structure.

提示：

- 1、用户界面可以设计为“菜单”方式：能够进行交互。**
- 2、根据输入的字符串中每个字符出现的次数统计频度，对没有出现的字符一律不用编码。**

题目二：哈夫曼编/解码器

提示：代码模块

- 1、统计输入的字符串中字符频率**
- 2、创建哈夫曼树**
- 3、打印哈夫曼树**
- 4、创建哈夫曼编码表**
- 5、对输入字符串进行编码并输出编码结果**
- 6、解码并输出解码后的字符串**
- 7、编写测试函数**

题目二：哈夫曼编/解码器

提示：

1、哈夫曼树结点结构：

结点权值、双亲指针、左孩子指针、右孩子指针

2、哈夫曼类结构：

哈夫曼树、哈夫曼编码表

输入字符串、出现的字符

叶子结点数、叶子结点字符

3、注意内存泄露问题，哈夫曼树和哈夫曼编码表都是动态内存，需在哈夫曼树析构函数中进行内存清理

程序要求

- 1.注意内存的动态申请和释放，是否存在内存泄漏；**
- 2.优化程序的时间性能；**
- 3. 保持良好的编程风格：**
 - 代码要简洁；**
 - 代码段与段之间要有空行和缩进；**
 - 标识符名称应该与其代表的意义一致；**
 - 函数名之前应该添加注释说明该函数的功能；**
 - 关键代码应说明其功能。**