

哈尔滨工业大学计算机科学与技术学院

实验报告

课程名称：数据结构与算法设计

课程类型：必修

实验项目名称：树型结构的建立与遍历

实验题目：树型结构的建立与遍历

班级：1503101

学号：1150310116

姓名：李博

设计成绩	报告成绩	指导老师

一、实验目的

树型结构的遍历是树型结构算法的基础，掌握树形结构的遍历和建立算法是数据结构这门课的基础。

二、实验要求及实验环境

实验要求：

1. 至少采用两种方法，编写建立二叉树的二叉链表存储结构的程序，并以适当的形式显示并保存二叉树；
2. 采用二叉树的二叉链表存储结构，编写程序实现二叉树的先序、中序和后序遍历的递归和非递归算法以及层序遍历算法，以适当的形式显示并保存二叉树和相应的遍历序列；
3. 在二叉树的二叉链表存储结构基础上，编写程序实现二叉树的中序线索链表存储结构建立的算法，以适当的形式显示并保存二叉树的相应的线索链表；
4. 在二叉树的线索链表存储结构上，编写程序分别实现求二叉树一个结点的先序、中序和后序遍历的后继结点算法；
5. 以上条要求为基础，编写程序实现对中序线索二叉树进行先序、中序和后序遍历的非递归算法，以适当的形式显示并保存二叉树和相应的遍历序列。

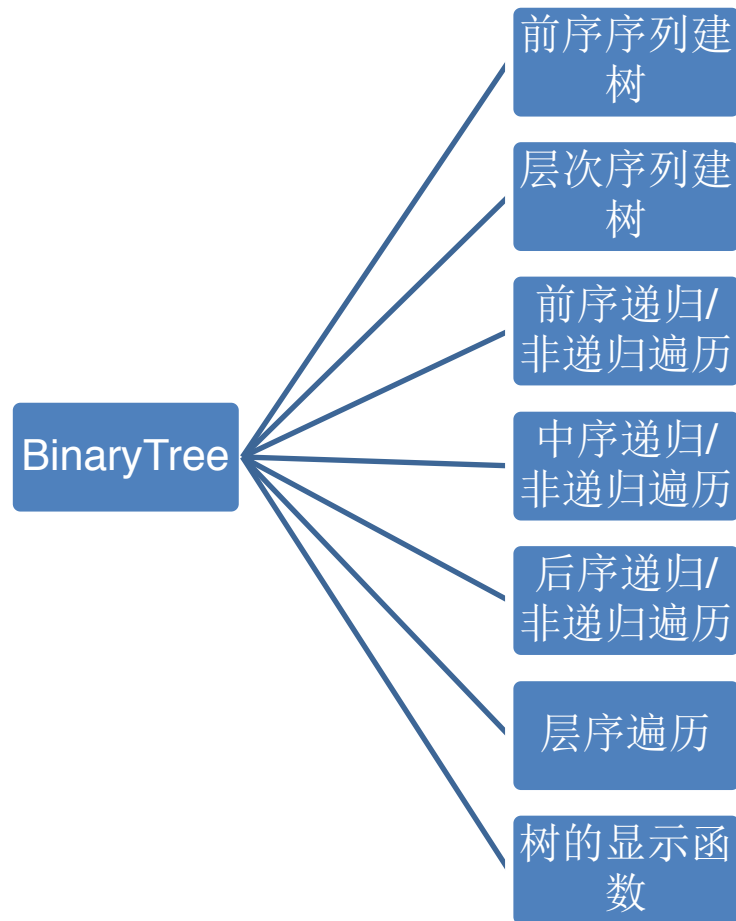
实验环境：

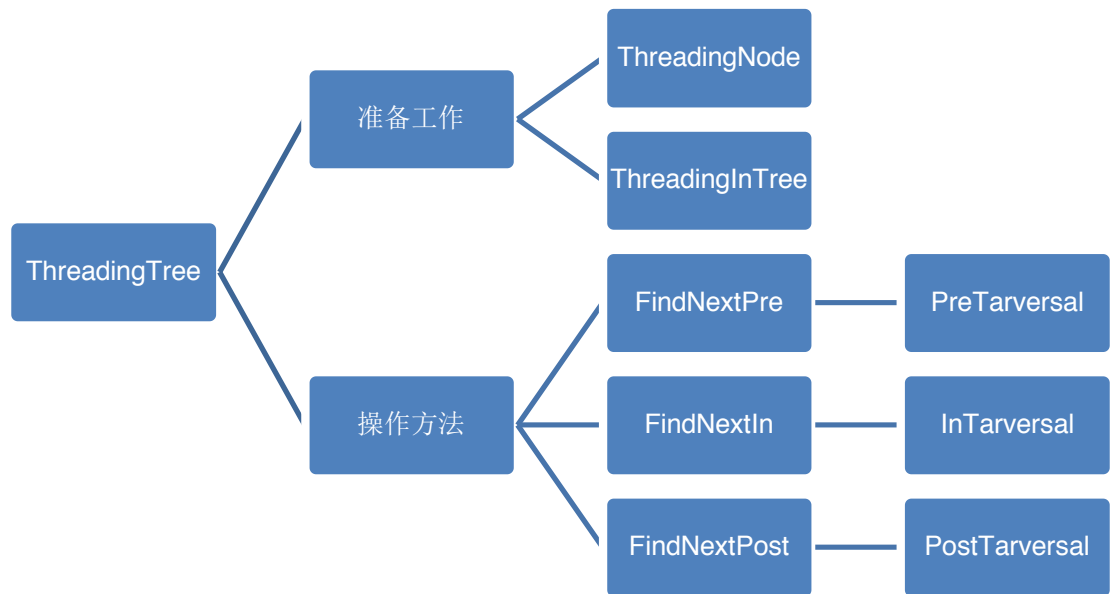
编辑器：xcode 7, sublime 3

编译器：LLVM clang 3.5-C++ 14(否则对于如 nullptr 的命令会报错)

二、设计思想（本程序中的用到的所有数据类型的定义，主程序的流程图及各程序模块之间的调用关系）

逻辑设计





物理设计

此次实验需要解释的部分主要有以下几点。

1. 用中序线索化二叉树求结点的先序后继
如果左子树不为空，返回左儿子即可
如果左子树空，右子树非空，返回右儿子
如果左右子树都空，则找到右线索中第一个有右儿子的的结点，返回这个右儿子
2. 用中序线索化二叉树求结点的中序后继
右子树非空则右子树中的最左结点就是中序后继
右子树为空则右线索就是中序后继
3. 用中序线索二叉树求结点的后序后继
记录 p 的父结点 father
如果 father 没有右子树，此时的 r 为 p 的后序后继
如果 father 有右子树，则 r 右子树进行后序遍历的第一个结点就是后序后继

四、测试结果

1. BinaryTree 部分的输出 out.txt 文件截图如下

Now we uses two ways to construct tree:



Firstly we constructed a tree according full preorder:
A#BC#D###

Traversal with recursion

Preorder Traversal: ABCD
Inorder Traversal: ACDB
Postorder Traversal: DCBA
Levelorder Traversal: ABCD

Traversal without recursion

Preorder Traversal: ABCD
Inorder Traversal: ACDB
Postorder Traversal: DCBA
Then we constructed a tree according full preorder:
ABCDEFGHI##J##

Traversal with recursion

Preorder Traversal: ABDHIECFJG
Inorder Traversal: HDIBEAJFCG
Postorder Traversal: HIDEBJFGCA
Levelorder Traversal: ABCDEFGHIJ

Traversal without recursion

Preorder Traversal: ABDHIECFJG
Inorder Traversal: HDIBEAJFCG
Postorder Traversal: HIDEBJFGCA

2.ThreadingTree 部分的输出 out.txt 文件截图如下

Now we uses two ways to construct tree:



Firstly we constructed a tree according full preorder:
A#BC#D###

Traversal with recursion

Preorder Traversal: ABCD
Inorder Traversal: ACDB
Postorder Traversal: DCBA
Levelorder Traversal: ABCD

Traversal without recursion

Preorder Traversal: ABCD
Inorder Traversal: ACDB
Postorder Traversal: DCBA
Then we constructed a tree according full preorder:
ABCDEFGHI##J##

Traversal with recursion

Preorder Traversal: ABDHIECFJG
Inorder Traversal: HDIBEAJFCG
Postorder Traversal: HIDEBJFGCA
Levelorder Traversal: ABCDEFGHIJ

Traversal without recursion

Preorder Traversal: ABDHIECFJG
Inorder Traversal: HDIBEAJFCG
Postorder Traversal: HIDEBJFGCA

五、系统不足与经验体会

系统不足：在检查时老师指出了，应该将线索化的过程输出，向用户报名线索化这个过程，并且展示这棵树和非线索化树有什么不同。还有就是 display 这个函数，用于图形化显示的时候可能会出现如果树结构较多，左右子树交叉的情况。

经验体会：关于链表这一部分的代码，一定要使用 IDE 辅助 Debug，传统的输出变量的调试方式已经没有办法应付这样的数据结构啦，不过还好恶补了一个晚上之后已经学会了使用 xcode 调试程序的基本方法，极大程度上的加速了此次实验的完成速率。

六、附录：源代码（带注释）