****

**数据结构**

**实 验 报 告**

实验名称：实验四 二叉树的建立

实验学时：2

实验环境：dev511

学生学院：信息科学与技术学院/网络空间安全学院

专业班级：计算机2101

学 号：20211003153

学生姓名：赖永超

指导教师：周咏梅

学 年： 2021-2022第二学期

2022 年 5月 20日

目录

[一、实验目的 3](#_Toc103975178)

[二、实验内容 3](#_Toc103975179)

[1. 先序遍历顺序建立二叉链表 3](#_Toc103975180)

[2. （拓展选作）哈夫曼编码问题 14](#_Toc103975181)

[三、实验总结 17](#_Toc103975182)

**实验四二叉树的建立**

姓名：赖永超 班级：计算机2101 学号：20211003153

## 一、实验目的

熟练掌握二叉树遍历算法及二叉树的建立。

## 二、实验内容

### 1. 先序遍历顺序建立二叉链表

1）定义并写出二叉树的结点结构

利用顺序表储存二叉树，定义结构体tree。其中，lson为左儿子序号，rson为右儿子序号，me为自己的值。定义代码如图1-1。

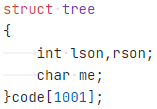


图1-1 树结构的定义

具体代码：sy4\_1\_1\_2101\_20211003153\_赖永超.cpp

Github：

2）写出二叉树的中序遍历算法

利用函数，进行中序输出。当左儿子不为空时，搜索左儿子。当左儿子为空时，输出自己的值，然后搜索右儿子。当右儿子为空时，函数结束，返回。算法代码如图1-2。

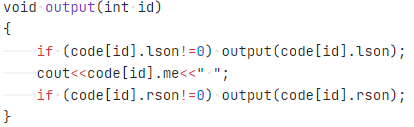


图1-2 中序遍历算法

具体代码：sy4\_1\_1\_2101\_20211003153\_赖永超.cpp

Github：[sy4\_1\_1\_2101\_20211003153\_赖永超.cpp](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/sy4_1_1_2101_20211003153_%E8%B5%96%E6%B0%B8%E8%B6%85.cpp)

3）写出二叉树的先序遍历建立二叉树算法

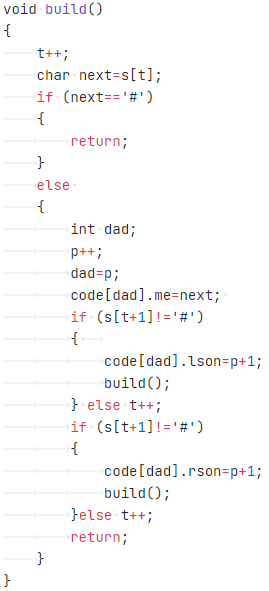
对输入的字符串逐个遍历。t为字符串的指针，p为节点总数。当s[t]不为“#”号时，建立节点，并保存此点编号为父节点。然后建立该父节点的左儿子节点，然后建立其右儿子节点。代码如图1-3。

图1-3 按前序遍历建立二叉树

具体代码：sy4\_1\_1\_2101\_20211003153\_赖永超.cpp

Github：[sy4\_1\_1\_2101\_20211003153\_赖永超.cpp](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/sy4_1_1_2101_20211003153_%E8%B5%96%E6%B0%B8%E8%B6%85.cpp)

4）编写主程序，测试完成图5.10(b)、图5.13、图5.17的二叉树的建立,调用中序遍历函数，检查上述各二叉树中序输出序列是否正确,记录并分析实验过程和结果.

①图5.10（b）的建立和中序遍历

题图如图1-4。

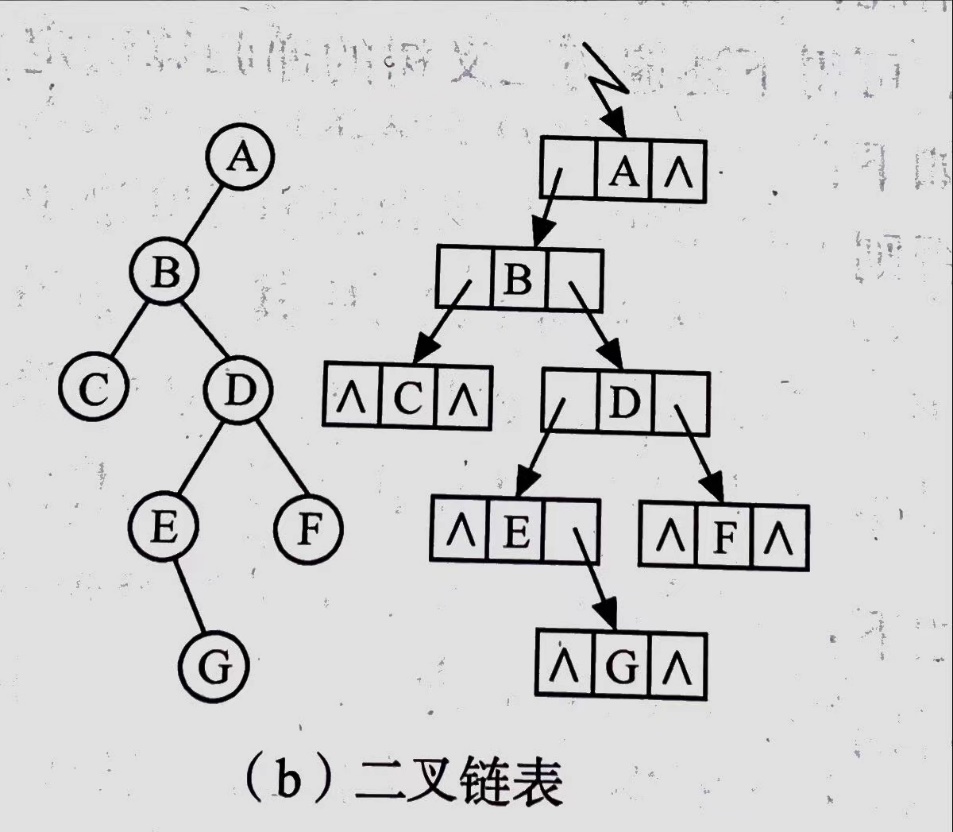


图1-4 题图

输入数据：ABC##DE#G##F###

预计中序遍历结果：C B E G D F A

运行结果如图1-5。



图1-5 图5.10（b）的运行结果

运行结果判断：与预计的中序遍历结果相同，程序运行正确。

②图5.13的建立和中序遍历

题图如图1-6。

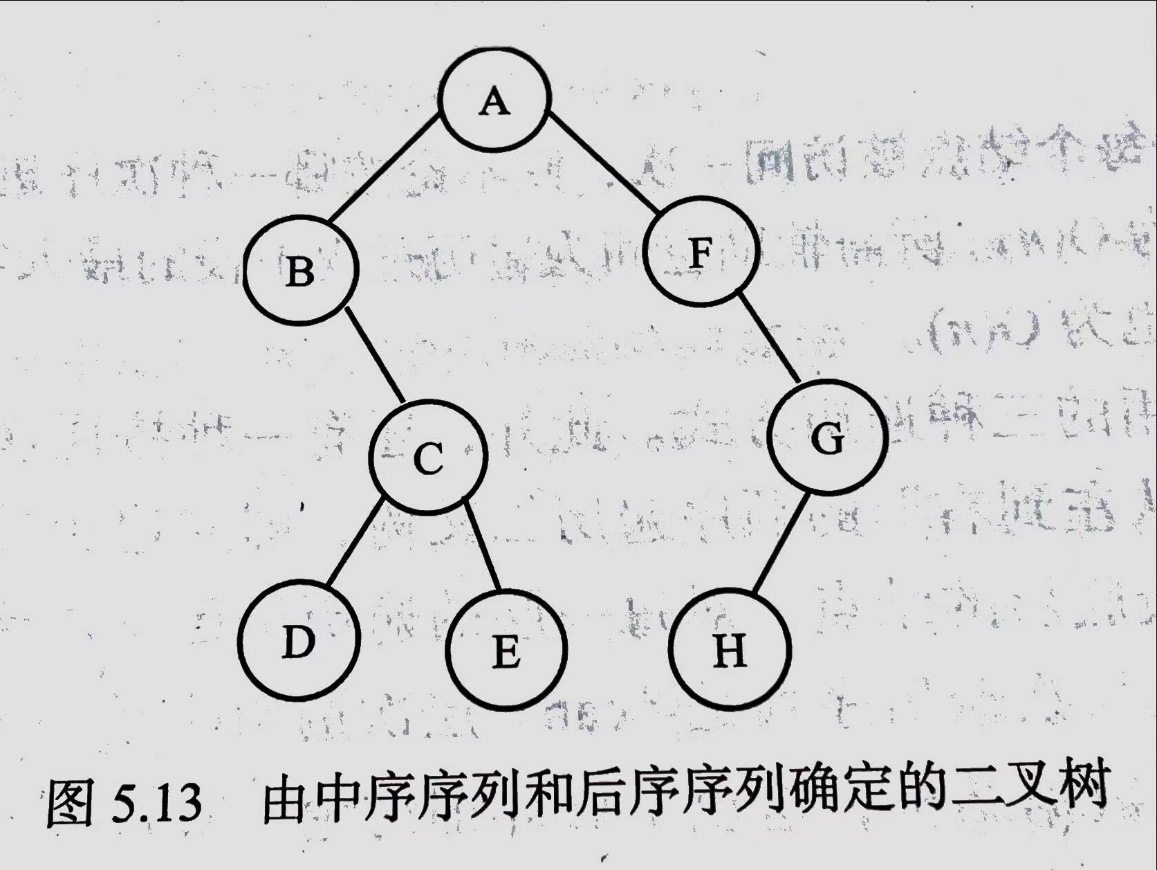


图1-6 题图

输入数据：AB#CD##E##F#GH###

预计中序遍历结果：B D C E A F H G

运行结果如图1-7。



图1-7 图5.13的运行结果

运行结果判断：与预计的中序遍历结果相同，程序运行正确。

③图5.17的建立和中序遍历

题图如图1-8。

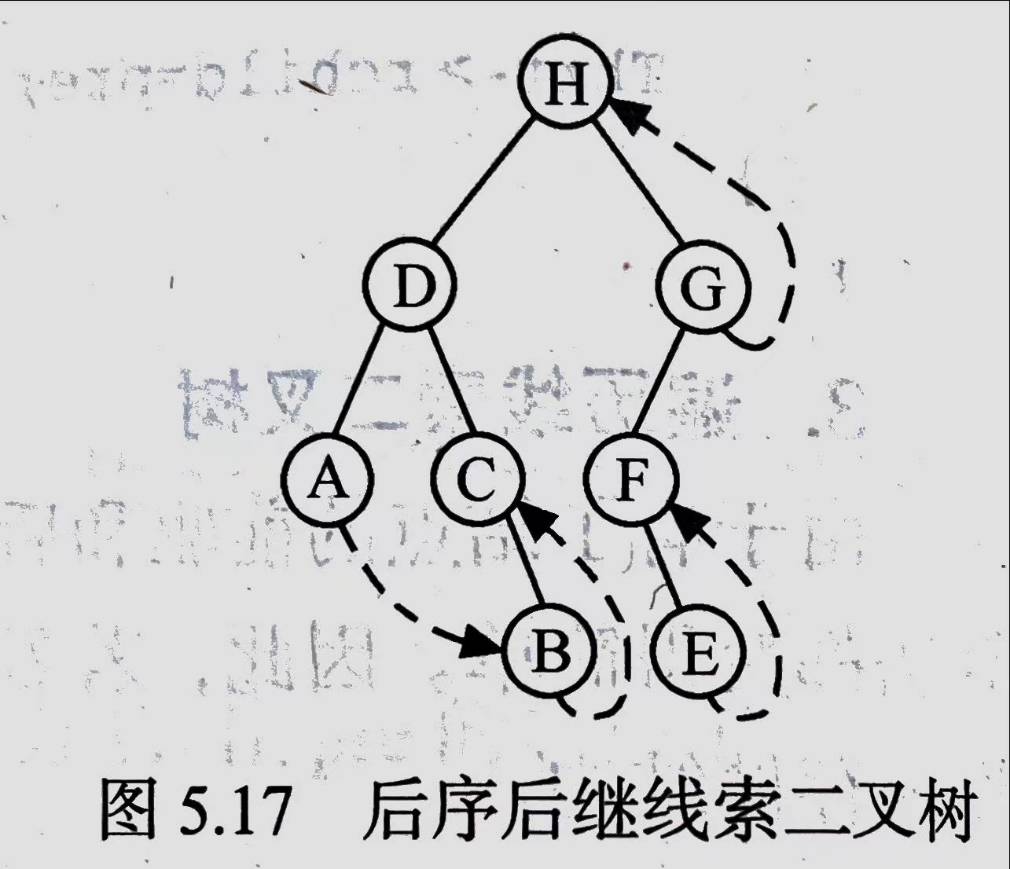


图1-8 题图

输入数据：HDA##C#B##GF#E###

预计中序遍历结果：A D C B H F E G

运行结果如图1-9。



图1-9 图5.17的运行结果

运行结果判断：与预计的中序遍历结果相同，程序运行正确。

具体代码：sy4\_1\_1\_2101\_20211003153\_赖永超.cpp

Github：[sy4\_1\_1\_2101\_20211003153\_赖永超.cpp](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/sy4_1_1_2101_20211003153_%E8%B5%96%E6%B0%B8%E8%B6%85.cpp)

5) 输入数据：ABCDEGF后，再输入若干个#后(记录需输入#的个数)，写出输出结果，画出建立的二叉树。

输入#的个数为8。建立如下图二叉树，二叉树见图1-10。

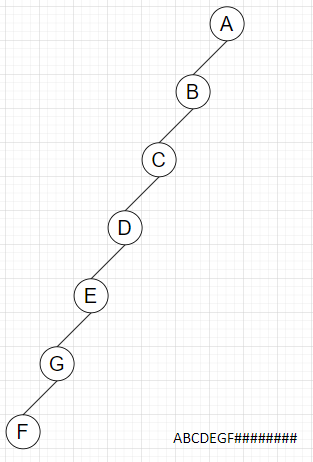


图1-10 输入8个#后建立的二叉树

输出结果如图1-11。



图1-10 输入8个#后输出的结果

具体代码：sy4\_1\_1\_2101\_20211003153\_赖永超.cpp

Github：[sy4\_1\_1\_2101\_20211003153\_赖永超.cpp](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/sy4_1_1_2101_20211003153_%E8%B5%96%E6%B0%B8%E8%B6%85.cpp)

6）增加后序和前序遍历二叉树算法，并在main函数测试建立的上述二叉树的后序、前序及中序序列、层序。

前序遍历代码如图1-11。

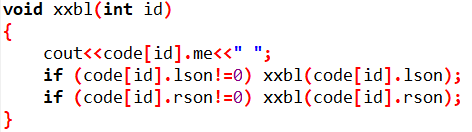


图1-11 前序遍历代码

后续遍历代码如图1-12。

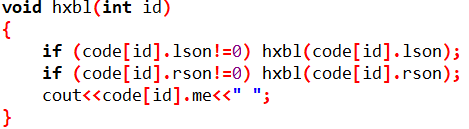


图1-12 前序遍历代码

层序遍历代码如图1-13。

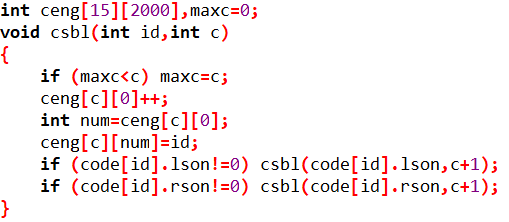


图1-13 层序遍历代码

①图5.10（b）前序遍历、后续遍历和层序遍历

输出结果如图1-14。

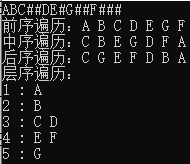


图1-14 图5.10（b）前序遍历、后续遍历和层序遍历

②图5.13前序遍历、后续遍历和层序遍历

输出结果如图1-15。

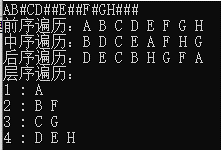


图1-15 图5.13前序遍历、后续遍历和层序遍历

③图5.17前序遍历、后续遍历和层序遍历

输出结果如图1-16。

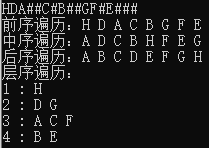


图1-16 图5.17前序遍历、后续遍历和层序遍历

具体代码：sy4\_1\_2\_2101\_20211003153\_赖永超.cpp

Github：[sy4\_1\_2\_2101\_20211003153\_赖永超.cpp](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/sy4_1_2_2101_20211003153_%E8%B5%96%E6%B0%B8%E8%B6%85.cpp)

7）完成pta-ch6作业题-7-1-先序遍历建立二叉树

①实验题目

pta-ch6-7-1 前序序列创建二叉树

②问题描述

编一个程序，读入用户输入的一串先序遍历字符串，根据此字符串建立一个二叉树（以二叉链表存储）。 例如如下的先序遍历字符串： ABC##DE#G##F### 其中“#”表示的是空格，代表一棵空树。然后再对二叉树进行中序遍历，输出遍历结果。

输入格式:

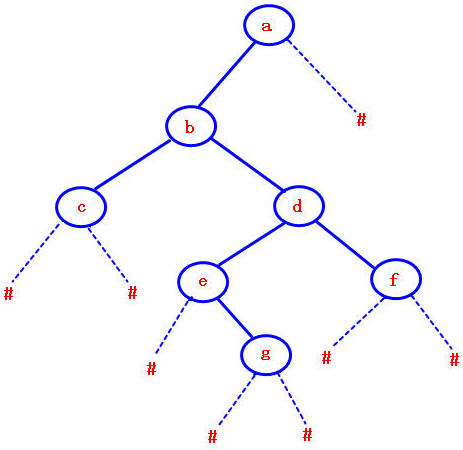
多组测试数据，每组测试数据一行，该行只有一个字符串，长度不超过100。

输出格式:

对于每组数据，

输出二叉树的中序遍历的序列，每个字符后面都有一个空格。

每组输出一行，对应输入的一行字符串。

输入样例:（及其对应的二叉树）

abc##de#g##f###

输出样例:

c b e g d f a

③算法分析

A.数据结构

构建树结构体，代码如上方图1-1。

B.功能分析

输入树的前序遍历，然后输出数的中序遍历

C.算法分析

对输入的字符串逐个遍历。t为字符串的指针，p为节点总数。当s[t]不为“#”号时，建立节点，并保存此点编号为父节点。然后建立该父节点的左儿子节点，然后建立其右儿子节点。代码如上图1-3。

D.具体代码

具体代码：sy4\_1\_1\_2101\_20211003153\_赖永超.cpp

Github：[sy4\_1\_1\_2101\_20211003153\_赖永超.cpp](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/sy4_1_1_2101_20211003153_%E8%B5%96%E6%B0%B8%E8%B6%85.cpp)

E.运行结果

运行结果如图1-17，1-18，1-19。



图1-17 先序遍历建立二叉树 运行截图①



图1-18 先序遍历建立二叉树 运行截图②



图1-19 先序遍历建立二叉树 运行截图③

### 2. （拓展选作）哈夫曼编码问题

已知26个字母使用频度表

|  |  |  |
| --- | --- | --- |
| 序号 | 字母 | 频度(%) |
| 1 | E | 12.25 |
| 2 | T | 9.41 |
| 3 | A | 8.19 |
| 4 | O | 7.26 |
| 5 | I | 7.10 |
| 6 | N | 7.06 |
| 7 | R | 6.85 |
| 8 | S | 6.36 |
| 9 | H | 4.57 |
| 10 | D | 3.91 |
| 11 | C | 3.83 |
| 12 | L | 3.77 |
| 13 | M | 3.34 |
| 14 | P | 2.89 |
| 15 | U | 2.58 |
| 16 | F | 2.26 |
| 17 | G | 1.71 |
| 18 | W | 1.59 |
| 19 | Y | 1.58 |
| 20 | B | 1.47 |
| 21 | K | 0.41 |
| 22 | J | 0.14 |
| 23 | V | 1.09 |
| 24 | X | 0.21 |
| 25 | Q | 0.09 |
| 26 | Z | 0.08 |

请为26个英文字母设计哈夫曼编码。

1. 写出26个字母的哈夫曼编码

|  |  |  |
| --- | --- | --- |
| 字母 | 编码 | 频度(%) |
| E | 011 | 12.25 |
| T | 001 | 9.41 |
| A | 1111 | 8.19 |
| O | 1101 | 7.26 |
| I | 1011 | 7.10 |
| N | 1010 | 7.06 |
| R | 1001 | 6.85 |
| S | 1000 | 6.36 |
| H | 0001 | 4.57 |
| D | 11101 | 3.91 |
| C | 11100 | 3.83 |
| L | 11001 | 3.77 |
| M | 11000 | 3.34 |
| P | 01001 | 2.89 |
| U | 01000 | 2.58 |
| F | 00001 | 2.26 |
| G | 010111 | 1.71 |
| W | 010110 | 1.59 |
| Y | 010101 | 1.58 |
| B | 010100 | 1.47 |
| K | 0000000 | 0.41 |
| J | 000000110 | 0.14 |
| V | 000001 | 1.09 |
| X | 00000010 | 0.21 |
| Q | 0000001111 | 0.09 |
| Z | 0000001110 | 0.08 |

1. 画出哈夫曼树

创建具体过程见链接：[哈夫曼树创建过程](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/%E5%93%88%E5%A4%AB%E6%9B%BC%E6%A0%91%E5%88%9B%E5%BB%BA%E8%BF%87%E7%A8%8B.png)

创建的哈夫曼树见图2-1。高清哈夫曼树图见链接：[哈夫曼树图](https://github.com/Wilson-LYC/Data-Structure/blob/DS/DS-4%20%E4%BA%8C%E5%8F%89%E6%A0%91/%E5%93%88%E5%A4%AB%E6%9B%BC%E6%A0%91.png)

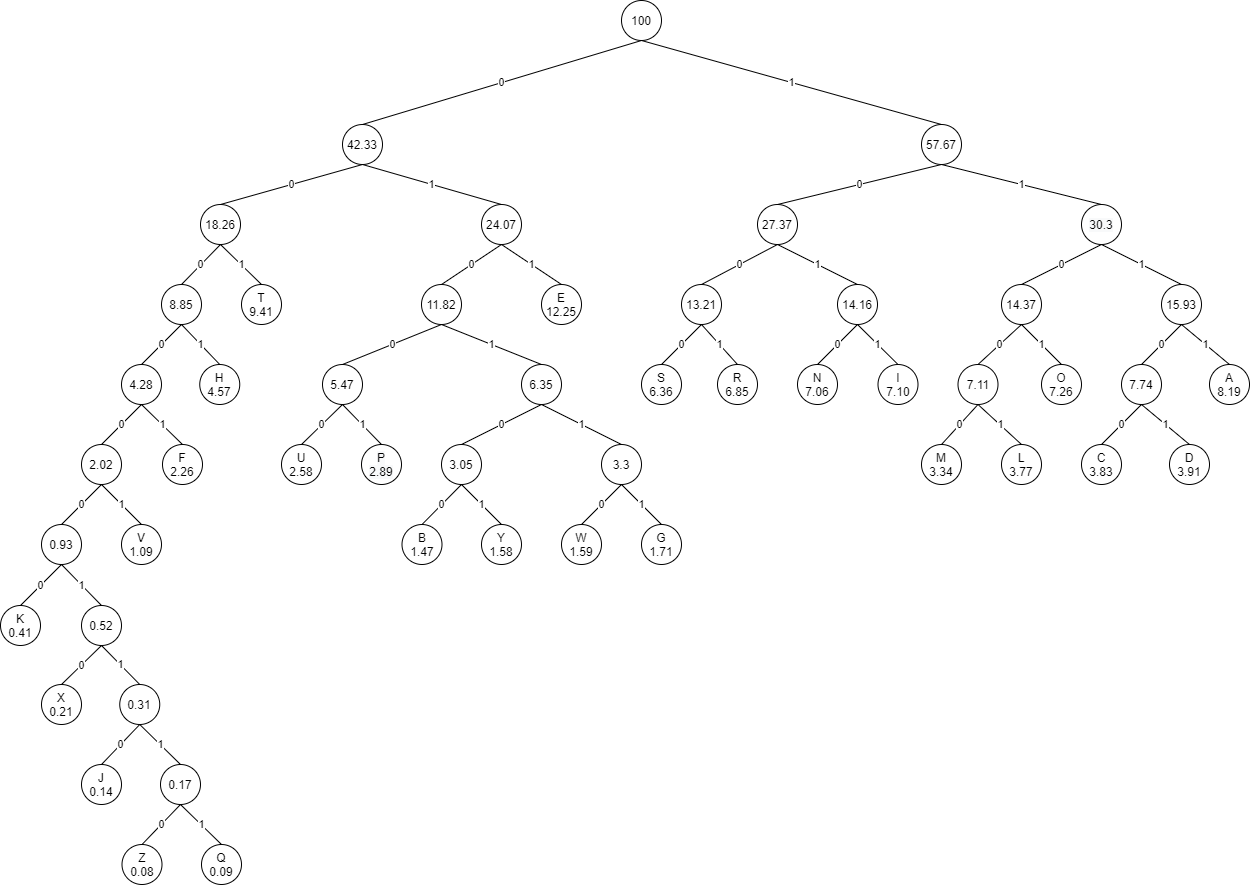


图2-1 哈夫曼树

1. 计算该哈夫曼树的带权路径长度
2. 求26个字母的平均编码长度

## 三、实验总结

在本次实验中，主要使用了研究二叉树的存储、前序遍历、中序遍历、后序遍历和层序遍历。在pta的练习中，学习了根据二叉树的中序遍历和后序遍历的结果，建立出二叉树并输出二叉树的前序遍历。研究哈夫曼树的建立、编码，比较哈夫曼编码和二进制编码的优缺点。哈夫曼编码相较于二进制编码可以缩短编码量，提高信息传递的速率。