**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **数据结构与算法分析** |
| **学生姓名：** | **叶子繁** |
| **学生学号：** | **201730684427** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2018-2019第一学期** |

**软件学院**

**2018年10月**

# **二叉树的建立及相关算法的实现**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **地 点：** | | | B7 楼 | | 233 房； | | **实验台号：** | | 233-87 |
| **实验日期与时间：** | | | 2018-10-26 14：00 | | | | **评 分：** | |  |
| **预习检查纪录：** | | |  | | | | **实验教师：** | |  |
| **电子文档存放位置：** | | | /home/Cinkye/下载/exp 2-1/ & /home/Cinkye/下载/exp 2-2/ | | | | | | |
| **电子文档文件名：** | | | BSTNode.h BinNode.h BST.h BSTmain.cpp dictionary.h  heapforHuffman.h hufftree.h huffnode.h main.cpp | | | | | | |
| **批改意见：** | | |  | | | | | | |
| 出勤情况 | 实验报告质量 | | | | | |
| 任务完成情况 | | 源代码注释 | | 报告内容 | |
| □ 出勤  □ 缺勤  □ 早退  □ 请假 | □ 完成  □ 未完成 | | □ 优秀  □ 良好  □ 通过  □ 未通过 | | □ 优秀  □ 良好  □ 通过  □ 未通过 | |

# 报告内容

1、实验目标

2、实验环境

Ubuntu 18.04, Visual Studio Code, Code::Blocks 16.04, Github

3、二叉检索树的实现

3.1 测试数据及预期结果

序列1：10 7 15 3 8 25 11 1 5 9

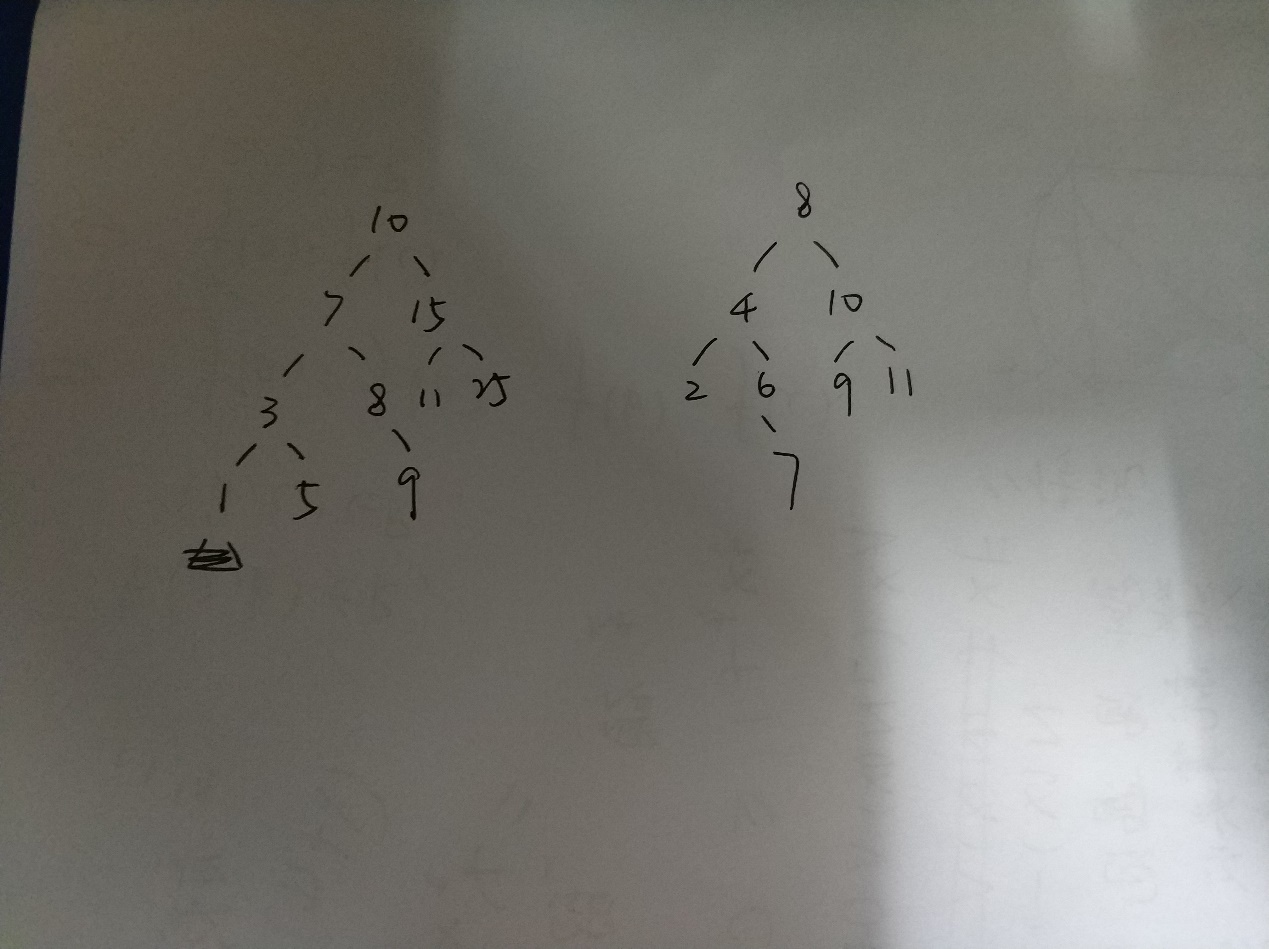
序列2：8 10 4 11 6 9 2 7

**预期结果：**

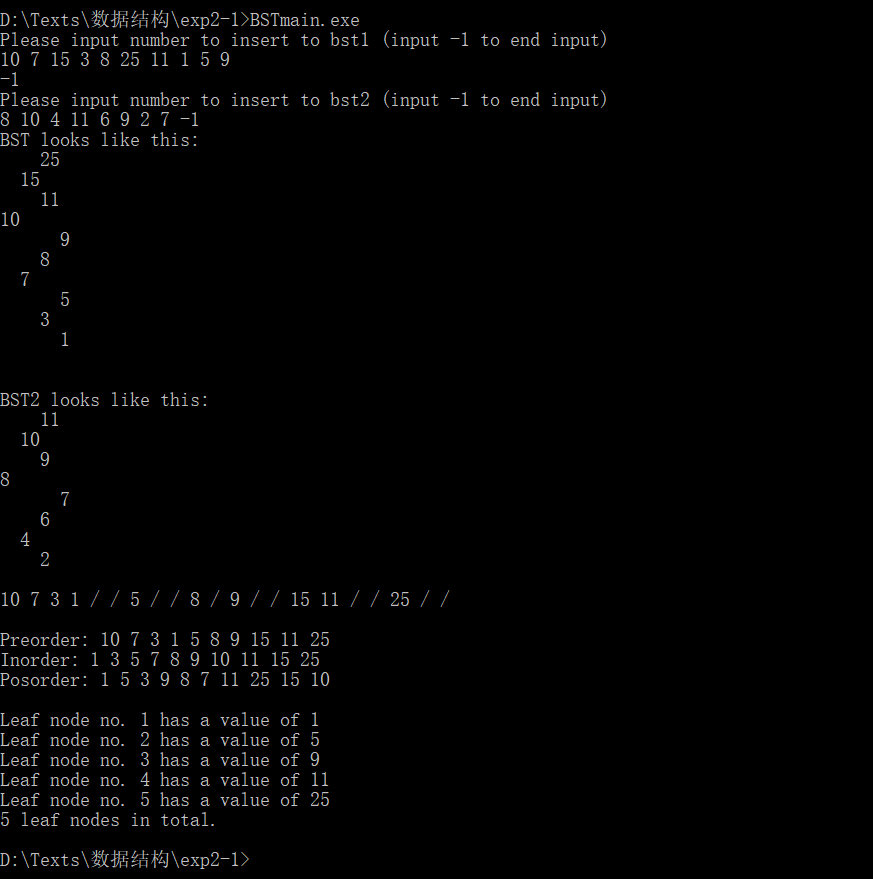
序列1：10 7 3 1 / / 5 / / 8 / 9 / / 15 11 / / 25 / /

序列2：8 4 2 / / 6 / 7 / / 10 9 / / 11 / /

3.2 二叉检索树的形状



3.3 程序运行结果



中序遍历可获得有序数列

3.4 算法分析

（二叉检索树构建算法、各遍历算法分析，要描述分析过程）

二叉检索树构建算法：n个结点，每个节点插入时平均要比较log n层，故二叉树的构建为O(n log n)

遍历算法：各遍历算法只有递归调用与访问节点的顺序区别，且每个节点只访问一次，故为O(n)

4、Huffman编码

4.1 Huffman编码树

1

1

1

0

0

0

1

33

E

1

0

0

6D

7A

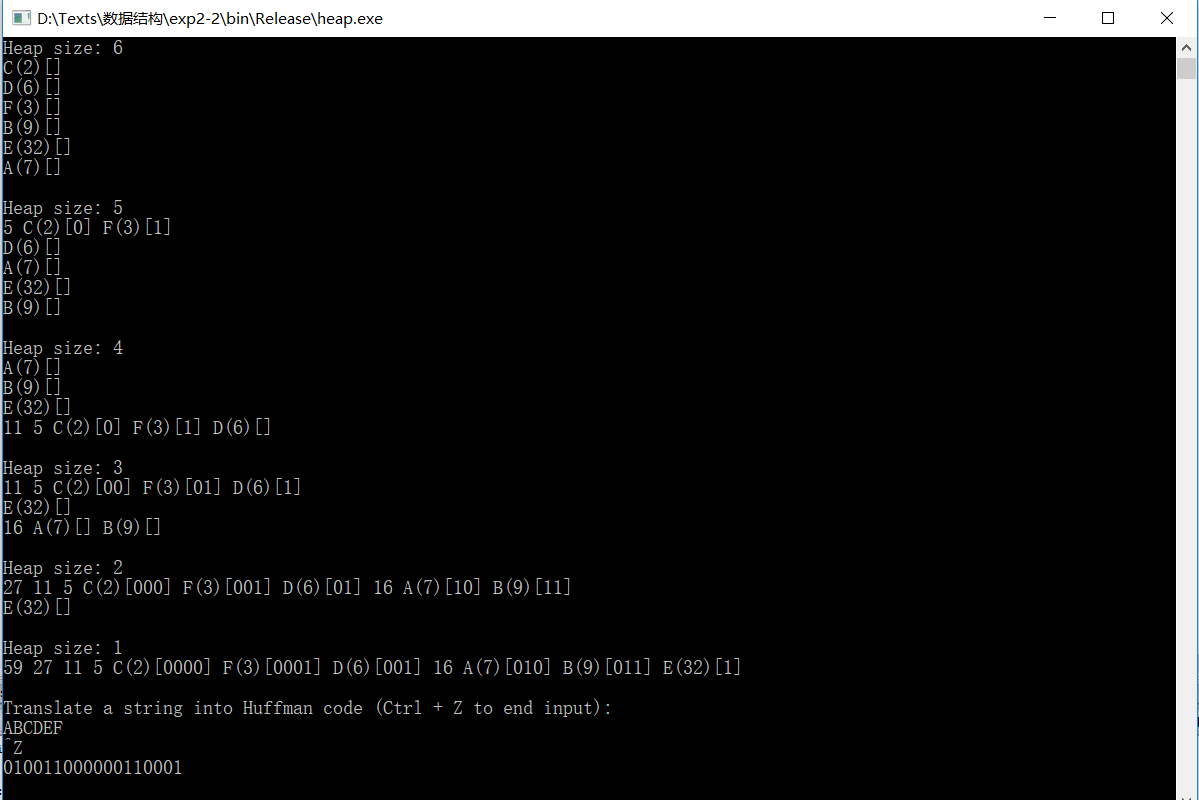
9B

2C

3

F

4.2程序运行结果



4.2 Huffman编码效率分析

A:010 B:011 C:0000 D:001 E:1 F:0001

编码总长度：3+3+4+3+1+4 = 15

单个字母编码长度期望：15/5=3

5、总结

　（包括体会、问题和建议）