

实验 3 综合应用

【实验目的】

- 加深对哈夫曼树实现算法的理解。
- 加深对二叉树遍历算法的理解。
- 加深对数据结构中链表、顺序表、堆、文件等基本操作及实现算法的理解，以便在解决实际问题中灵活运用它们。
- 加深对各种课程各种基本概念、基本结构和基本操作的理解。

【编程要求】

- 良好的编程风格（可参考附录《C++ Code Standard》/《高质量 C++/C 编程指南》）；
- 必要的注释；（简单要求如下）
 1. readme 文件对上交的实验内容文件或目录作适当的解释；
 2. 每个.cpp 文件中注释信息至少包含以下内容：
 - （1）版权信息。
 - （2）文件名称，标识符，摘要或模块功能说明。
 - （3）当前版本号，作者/修改者，完成日期。
 - （4）版本历史信息。 // （1）--（4）部分写在文件头
 - （5）所有的宏定义，非局部变量都要加注释
 - （6）所有函数前有函数功能说明，输入输出接口信息，以及调用注意事项
 - （7）函数关键地方加注释
 3. 每个.h 文件中对关键接口信息/变量声明信息等加注释

【参考资料】

- 《C++ Coding Standard》
- 《高质量 C++/C 编程指南》
- 《数据结构》教材

【评分标准】

- E<60: 在规定时间内上交实验程序及文档,基本要求中的大部分内容未完成。

- 60<D<70: 在规定时间内上交实验程序及文档,完成了基本要求中的大部分内容,编程风格好,文档基本符合规范,设计思想基本清晰,界面基本符合要求。
- 70<C<80: 在规定时间内上交实验程序及文档,完成了基本要求中的全部内容,文档规范,编程风格好,设计思想基本清晰,界面美观大方。
- 80<B<90: 在规定时间内上交实验程序及文档,完成了基本要求中的全部内容,并完成选做内容中的部分要求,文档规范清晰,编程风格好,设计思想清晰,界面美观大方。
- 90<A<100: 在规定时间内上交实验程序及文档,完成了基本要求和选做内容的全部内容,且功能完善,文档规范清晰,设计思想十分清晰,编程风格好,界面美观大方。

实验题目：文本压缩与解压

【问题描述】

利用哈夫曼编码进行通信可以大大提高信道利用率,缩短信息传输时间,降低传输成本。但是,这就要求在发送端通过一个编码系统对待传数据预先编码,在接收端将传来的数据进行译码(复原)。对于双工信道(即可以双向传输信息的信道),每端都需要一个完整的编/译码系统。试为这样的信息收发站写一个哈夫曼的编/译码系统。

【实验内容】

【必做内容】

1. 一个完整的系统应具有以下功能:

(1) 初始化(Initialization)。从终端读入字符集大小 n 以及 n 个字符和 n 个权值,建立哈夫曼树,并将它存于文件 hfmTree 中。

(2) 编码(Encoding)。利用已建好的哈夫曼树(如不在内存,则从文件 hfmTree 中读入),对文件 ToBeTran 中的正文进行编码,然后将结果存入文件 CodeFile 中。

(3) 译码(Decoding)。利用已建好的哈夫曼树将文件 CodeFile 中的代码进行译码,结果存入文件 TextFile 中。

(4) 打印代码文件(Print)。将文件 CodeFile 以紧凑格式显示在屏幕上,每行 50 个二进制代码。同时将此字符形式的编码文件写入文件 CodePrin 中。

(5) 打印哈夫曼树(Tree Printing)。将已在内存中的哈夫曼树以直观的方式(树或凹入形式)显示在屏幕上,同时将此字符形式的哈夫曼树写入文件 TreePrin 中。

(6) 其他辅助功能:实现各个转换操作的源/目标文件,均由用户在选择此操作时指定。

2. 按《实验报告书 书写规范》进行实验报告的书写。要注意结合测试数据的测试情况进行系统的分析。

【选做内容】

1. 用 MFC 的单文档窗口和菜单设计界面。

【测试数据】

(1) 用下表给出的字符集和频度的实际统计数据建立哈夫曼树，并实现以下报文的编码和译码：“THIS PROGRAM IS MY FAVORITE”。

| | | | | | | | | | | | | | | |
|----|-----|----|----|----|----|-----|----|----|----|----|---|----|----|----|
| 字符 | | A | B | C | D | E | F | G | H | I | J | K | L | M |
| 频度 | 186 | 64 | 13 | 22 | 32 | 103 | 21 | 15 | 47 | 57 | 1 | 5 | 32 | 20 |
| 字符 | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | |
| 频度 | 57 | 63 | 15 | 1 | 48 | 51 | 80 | 23 | 8 | 18 | 1 | 16 | 1 | |

(2) 统计本实验你所编写的所有源程序的各字符的出现频度，并最终实现对你所完成的这个系统的源程序进行编码和译码。

(3) 在互联网上搜索 10 篇英文文章(每篇字数>5000 字)。统计存在于这 10 篇英文文章中字符的出现概率，然后以这些字符和出现概率来建立哈夫曼树，最后将这 10 篇文章进行编码以及译码，并观察其前后的文件内容变化。

【实现提示】

- (1) 注意写入文件 CodeFile 的数据类型。(通过检查 CodeFile 文件容量来进行分析)
- (2) 在程序的一次执行过程中，第一次执行初始化, 编码或译码命令之后，哈夫曼树已经在内存了，不必再读入。每次执行中不一定执行 Initialization 初始化命令，文件文件 hfmTree 可能早已建好。

【实验时间】 4 周(第 11 周-第 15 周)