**《算法与程序综合实际设计》实习指导书**

**（一）线性结构（链表）题目**

**8、【银行业务活动的模拟】**

问题描述:客户业务分为两种。一种是申请从银行得到一笔资金，即取款或借款。一种是向银行投入一笔资金，即存款或还款。银行有两个服务窗口，相应地有两个队列。客户到达银行后先排第一个队列。处理每个客户业务时，如果属于第一种，且申请额超出银行现存资金总额而得不到满足，则立刻排入第二个队等候，直至满足时才离开银行；否则业务处理完后立刻离开银行。每接待完一个第二种业务的客户，则顺序检查和处理（如果可能）第二个队列中的客户，对能满足的申请者予以满足，不能满足者重新排列第二个队列的队尾。注意，在此检查过程中，一旦银行资金总额少于或等于刚才第一个队列中最后一个客户(第二种业务)被接待之前的数额，或者本次已将第二个队列检查或处理了一遍，就停止检查（因为此时已不可能还有能满足者）转而继续接待第一个对列的客户。任何时刻都只开一个窗口，假设检查不需要时间。营业时间结束时所有客户立即离开银行。

写一个上述银行业务的模拟程序，通过模拟方法求出用户在银行内逗留的平均时间。

要求 :利用链式存储结构实现模拟

**（二）栈和队列题目：1、【算术表达式求值】**

**问题描述：**一个算术表达式是由操作数(operand)、运算符(operator)和界限符(delimiter)组成的。假设操作数是正整数，运算符只含加减乘除等四种运算符，界限符有左右括号和表达式起始、结束符“#”，如：#（7+15）\*（23-28/4）#。引入表达式起始、结束符是为了方便。编程利用“算符优先法”求算术表达式的值。

**基本要求：**从键盘读入一个合法的算术表达式，输出正确的结果；显示输入序列和栈的变化过程，操作数类型扩充到实数。

"(3.14159/2+sqrt(1/3^2+4)+1/2^2\*ln(1/1.1\*(2+sqrt(1/3^2+4))))\*23.45@";

**选作内容：**扩充运算符集合；引入变量操作数；

**三、树型结构题目：**

**2、【哈夫曼编/译码器】**

**任务：**建立最优二叉树函数

**要求：**可以建立函数输入二叉树，并输出其赫夫曼树。

在上交资料中请写明：存储结构、基本算法（可以使用程序流程图）、输入输出、源程序、测试数据和结果、算法的时间复杂度、另外可以提出算法的改进方法；

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码的编/译码系统。

**[基本要求]**

一个完整的系统应具有以下功能：

（1）I：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）E：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）D：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）P：印代码文件（Print）。将文件CodeFile以紧凑格式显示在终端上，每行50个代码。同时将此字符形式的编码文件写入文件CodePrin中。

（5）T：印哈夫曼树（Tree printing）。将已在内存中的哈夫曼树以直观的方式（树或凹入表形式）显示在终端上，同时将此字符形式的哈夫曼树写入文件TreePrint中。

**[测试数据]**

（1）利用下面这道题中的数据调试程序。

某系统在通信联络中只可能出现八种字符，其概率分别为0.25，0.29，0.07，0.08，0.14，0.23，0.03，0.11，试设计哈夫曼编码。

（2）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 空格  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

**[实现提示]**

（1）编码结果以文本方式存储在文件CodeFile中。

（2）用户界面可以设计为“菜单”方式：显示上述功能符号，再加上“Q”，表示退出运行Quit。请用户键入一个选择功能符。此功能执行完毕后再显示此菜单，直至某次用户选择了“Q”为止。

（3）在程序的一次执行过程中，第一次执行I，D或C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I命令，因为文件hfmTree可能早已建好。

**4、计算命题演算公式的真值**

所谓命题演算公式是指由逻辑变量（其值为TRUE或FALSE）和逻辑运算符∧（AND）、∨（OR）和┐（NOT）按一定规则所组成的公式（蕴含之类的运算可以用∧、∨和┐来表示）。公式运算的先后顺序为┐、∧、∨，而括号（）可以改变优先次序。已知一个命题演算公式及各变量的值，要求设计一个程序来计算公式的真值。

要求：

（1）利用二叉树来计算公式的真值。首先利用堆栈将中缀形式的公式变为后缀形式；然后根据后缀形式，从叶结点开始构造相应的二叉树；最后按后序遍历该树，求各子树之值，即每到达一个结点，其子树之值已经计算出来，当到达根结点时，求得的值就是公式之真值。

（2）逻辑变元的标识符不限于单字母，而可以是任意长的字母数字串。

（3）根据用户的要求显示表达式的真值表。

提示：

构造二叉树，用堆栈计算结果。要求可以输入任意长的字母数字串，也是通过堆栈来完成。当检测到逻辑运算符或者是结束符是就进栈，如果没有检测到就复制到另一个数组中，完成任意长的字母数字串的输入。真值表输出可以用穷举法实现。

二叉树的节点用结构体定义

typedef struct Node

{

char data[100]; //数据域

struct Node \*leftChild; //左子树指针

struct Node \*rightChild; //右子树指针

struct Node \*father;//根结点指针

}BiTreeNode;//二叉树结点的结构体定义

堆栈节点也是结构体定义

typedef struct

{

BiTreeNode \* dizhi[100];

int top;

}SeqStack1; //定义堆栈结点结构体

由题意可知储存结构用数组和堆栈实现。数组b[100][100]用来存储输入的字母或者是数字，而堆栈用来存储输入的逻辑运算符和后续遍历二叉树。

此题主要的算法用于将二叉树中缀形式变为后缀形式以及后序遍历二叉树得到二叉树根节点的值，也就是逻辑运算左后结果，以及真值表的输出。

中缀形式变为后缀形式，设置一个堆栈，初始时将栈顶元素值设为#，顺序输入中缀表达式，当读到字母或者数字的时候将其输出，并保存在数组b[100][100]中，当读到逻辑运算符时，若前者的优先级高于后者，前者运算符作为后缀表达式的一个单词输出，接着一直向下输入比较，知道读入#号时，算法结束。

**5、二叉排序树和二叉平衡树**

(1)给定一组数

(2)编程实现二叉排序树的创建、插入、删除和查询

(3)编程实现二叉平衡树的创建、插入、删除和查询

(4)对于给定的这组数分别在二叉排序树和二叉平衡树上进行查找，给出两种情况下的查找成功和不成功时的ASL

**（四）图型结构题目：**

**4、这是一个古典问题。**

假设有n个修道士和n个野人准备渡河，但只有一条能容纳c人的小船，为了防止野人侵犯修道士，要求无论在何处，修道士的个数不得少于野人的人数（除非修道士个数为0）。如果两种人都会划船，试设计一个算法，确定他们能否渡过河去，若能，则给出一个小船来回次数最少的最佳方案。

要求：

（1）用一个三元组（x1,x2,x3）表示渡河过程中各个状态。其中，x1表示起始岸上修道士个数，x2表示起始岸上野人个数，x3表示小船位置（0——在目的岸，1——在起始岸）。例如（2,1,1）表示起始岸上有两个修道士，一个野人，小船在起始岸一边。采用邻接表做为存储结构，将各种状态之间的迁移图保存下来。

（2）采用广度搜索法，得到首先搜索到的边数最少的一条通路。

（3）输出数据：若问题有解（能渡过河去），则输出一个最佳方案。用三元组表示渡河过程中的状态，并用箭头指出这些状态之间的迁移：

目的状态←…中间状态←…初始状态。

若问题无解，则给出“渡河失败”的信息。

1. 求出所有的解。

main → solve → check → print

**（3）测试实例**：

请输入野蛮人和修道士人数N: 3

请输入船可乘人数C: 2

运行结果如下：

( 3 3 1 ) → ( 0 2 ) → ( 3 1 0 )

( 3 1 0 ) ← ( 0 1 ) ← ( 3 2 1 )

( 3 2 1 ) → ( 0 2 ) → ( 3 0 0 )

( 3 0 0 ) ← ( 0 1 ) ← ( 3 1 1 )

( 3 1 1 ) → ( 2 0 ) → ( 1 1 0 )

( 1 1 0 ) ← ( 1 1 ) ← ( 2 2 1 )

( 2 2 1 ) → ( 2 0 ) → ( 0 2 0 )

( 0 2 0 ) ← ( 0 1 ) ← ( 0 3 1 )

( 0 3 1 ) → ( 0 2 ) → ( 0 1 0 )

( 0 1 0 ) ← ( 1 0 ) ← ( 1 1 1 )

( 1 1 1 ) → ( 1 1 ) → ( 0 0 0 )

( 0 0 0 )

渡河成功! …… ……

有4种渡河方案。

**五、查找、排序、文件**

**2、【B\_树索引文件的插入、删除和查找】**

功能要求：

1. 从键盘输入一组关键字插入B\_树中；
2. 向B\_树中插入一个关键字；
3. 从B\_树中删除一个关键字；
4. 从B\_树中查找一个关键字对应记录的位置；
5. 遍历输出B\_树中所有关键字；
6. 求出一棵B\_树的深度；
7. 求出一棵B\_树的节点数；
8. 内存B\_树存盘；
9. 由文件中保存的B\_树恢复到内存中；
10. 清除B\_树，即收回B\_树中的所有节点；

等功能。

//定义B\_树的阶数和特定的最大关键字，可自行设定

#define m 3

#define MAXKEY 9999

//定义关键字类型为整型

Typedef int KeyType;

//定义元素类型

Struct ElemType {

KeyType key;//整型关键字域

Char rest[10]； //字符数组域

};

//定义B\_树的节点类型

Struct MBNode {

Int keynum; //关键字个数域

MBNode \*parent ; //指向父节点的指针域

KeyType key[m+1] ；//保存n个关键字域，下标0位置未用

MBNode \* ptr[m+1] ; //保存n+1 个指向子树的指针域

Int recptr[m+1] ; //保存每个关键字对应记录的存储位置域，下标0位置未用

};

**四、上交相关内容要求**

上交成果的内容必须由以下四个部分组成，缺一不可

1．上交源程序：学生按照课程设计的具体要求所开发的所有源程序（应该放到一个文件夹中）；

2．上交程序的说明文件：（保存在.txt中）在说明文档中应该写明上交程序所在的目录，上交程序的主程序文件名，如果需要安装，要有程序的安装使用说明；

3．课程设计报告：（保存在word 文档中，文件名要求按照"姓名-学号-课程设计报告"起名，如文件名为"张三-001-课程设计报告".doc ）按照课程设计的具体要求建立的功能模块，每个模块要求按照如下几个内容认真完成；

其中包括:

1. 需求分析：

（2）在该部分中叙述，每个模块的功能要求

（3）概要设计

在此说明每个部分的算法设计说明（可以是描述算法的流程图），每个程序中使用的存储结构设计说明（如果指定存储结构请写出该存储结构的定义。

（4）详细设计

　　各个算法实现的源程序，对每个题目要有相应的源程序（可以是一组源程序，每个功能模块采用不同的函数实现）

源程序要按照写程序的规则来编写。要结构清晰，重点函数的重点变量，重点功能部分要加上清晰的程序注释。

（5）调试分析

测试数据，测试输出的结果，时间复杂度分析，和每个模块设计和调试时存在问题的思考（问题是哪些？问题如何解决？），算法的改进设想。

4. 课设总结：（保存在word 文档中）总结可以包括 : 课程设计过程的收获、遇到问题、遇到问题解决问题过程的思考、程序调试能力的思考、对数据结构这门课程的思考、在课程设计过程中对《数据结构》课程的认识等内容。

**五、实习成绩考核办法**

要求：

⑴关于课题表及选题：在上面的课题中列出了多个设计课题，每个课题都有相应的要求或说明。参加课程设计的学生首先要了解设计的任务，仔细阅读各题的设计要求，然后根据自己的基础和能力情况从中选择6个题目。一般来说，选择课题应以在规定的时间内能完成，并能得到应有的锻炼为原则。

若学生对课题表以外的相关课题较感兴趣，希望选作课程设计的课题时，应征得指导教师的认可，并写出明确的设计要求和说明。

⑵关于设计的总要求 ：在设计时，要严格按照题意要求独立进行设计，不能随意更改。若确因条件所限，必须要改变课题要求时，应在征得指导教师同意的前提下进行。

⑶验收：在设计完成后，应当场运行和答辩，由指导教师验收，只有在验收合格后才能算设计部分的结束。

⑷**设计报告**：设计结束后要写出课程设计报告，以作为整个课程设计评分的书面依据和存档材料。设计报告以规定格式的电子文档书写、打印并装订，排版及图、表要清楚、工整。内容及要求如下：

①设计任务、要求及所用软件环境或工具

②问题的模型化描述以及求解算法的简要描述

③所选择的存储结构及基本运算、设计的模块构成、各模块的简要说明、流程图、调用关系表

④所设计软件的组成（程序文件和数据文件）及使用说明

⑤程序清单（计算机打印）

⑥设计总结和体会

⑦验收情况

⑸关于课程设计的成绩评定：课程设计的成绩评定以选定课题的难易度、完成情况和设计报告为依据综合评分。从总体来说，所设计的程序应该全部符合要求，问题模型、求解算法以及存储结构清晰；具有友好、清晰的界面；设计要包括所需要的辅助程序，如必要的数据输入、输出、显示和错误检测功能；操作使用要简便；程序的整体结构及局部结构要合理；设计报告要符合规范。

平时表现：40% 上机演示：30% 设计报告：30%。

2012年12月8日