《数据结构课程设计》指导书

东北林业大学 信息学院

计算机科学与技术专业

2022年11月

**《数据结构课程设计》实习指导书**

**一、实习目的**

数据结构课程设计是一项综合性设计活动，要求在教师的指导下，利用本课程内的以及到目前为止所学到的有关知识和技术解决一些不太复杂但却是综合性的问题。从规模来说，课程设计是在平时作业的基础上进一步扩大的大作业。在设计中，要求学生要全面考虑相互联系的各个方面及问题。

通过课程设计，使学生对整个课程的知识体系有较深入的理解，在运用本课程的知识解决实际问题方面得到锻炼，对锻炼学生的实践能力以及运用本课程的知识、方法解决更为复杂的实际问题有较好的启发和指导作用，从而为后续课程的学习、毕业设计环节以及将来的实际工作打好坚实的基础。

通过对给定问题的求解，使学生在运用《数据结构》、程序设计以及迄今为止所学课程中的各种基本技术和理论，在建立问题模型、构造求解算法、设计数据结构、编程及上机调试等方面得到全面的锻炼，从而能更深刻地理解算法、数据结构的精髓，为后续软件课程的学习及软件设计能力的提高奠定良好的基础。

**二、数据结构课程设计要求**

1.学生必须仔细阅读《数据结构课程设计》实习方案，认真主动完成课设的要求。有问题及时主动通过各种方式与教师联系沟通。

2.学生要发挥自主学习的能力，充分利用时间，安排好课程设计的时间计划，并在课程设计过程中不断检测自己的计划完成情况，及时向教师汇报。

3.课程设计按照教学要求需要两周时间完成（2周共十天）。

**三、实验所用仪器及实验环境（硬件改为实验所用仪器及元器件）**

PC机，Codeblocks软件环境。

**四、实习基本内容**

本次课程设计要求在（一）、（二）、（三）、（四）、（五）组中每组选择至少一个题目完成或者多人合作自己设计题目，要求设计题目尽量用到多种数据结构，题目需经过指导教师审核通过。

**（一）线性结构（链表）题目**

**1、员工管理系统**

问题描述：每个员工的信息包括：编号、姓名、性别、出生年月、学历、职务、电话、住址等。系统能够完成员工信息的查询、更新、插入、删除、排序等功能。

基本要求：排序：按不同关键字，对所有员工的信息进行排序；查询：按特定条件查找员工；更新，按编号对某个员工的某项信息进行修改；插入，加入新员工的信息；删除，按编号删除已离职的员工的信息。

选作内容：实现图形用户界面。

通过链表实现

数据结构：

struct workers

{

char name[15];//姓名

char department[18];//单位

char gender;//性别

unsigned int age;//年龄

unsigned long telephone;//电话

unsigned long wage;//工资

unsigned long num;//职工号

struct workers \*next;

};

**2、运动会分数统计**

任务：参加运动会有n个学校，学校编号为1,2,…,n。比赛分成m个男子项目和w个女子项目。男子项目编号为1,2,…,m，女子项目编号为m+1,m+2,…,m+w。不同的项目取前五名或前三名积分；取前五名的积分分别为：7、5、3、2、1，前三名的积分分别为：5、3、2；哪些取前五名或前三名由学生自己设定。（m<=20，n<=20，w<=20）。

功能要求：可以输入各个项目的前三名或前五名的积分；能统计各学校总分；可以按学校编号、学校总分、男女团体总分排序输出；可以按学校编号查询学校某个项目的情况；可以按项目编号查询取得前三或前五名的学校。

规定：输入数据形式和范围为20以内的整数（如果做得更好可以输入学校的名称，运动项目的名称）；输出形式，有中文提示，各学校分数为整型；界面要求，有合理的提示，每个功能可以设立菜单，根据提示，可以完成相关的功能要求。

存储结构：学生自己根据系统功能要求自己设计，请在最后的上交资料中指明你用到的存储结构。

参考的数据结构：

（1）项目信息表结构

typedef struct xm\_table{

int item; // 项目编号

char name[20];//项目名称

int count;  //该项目得分人的数量

}XM\_TABLE;

（2）学生信息表结构

struct STUDENT{

char name[20];        //姓名

int score;                //得分成绩

int range;                //得分名次

int item;                  //得分项目

int sex;                   //性别

};

（3）参赛学校信息结构

//参赛学校

typedef struct SchoolStruct

{

int count;       //计算实际运动员个数

int serial;        //学校编号

char Name[20];//学校名称

int menscore;    //男子团体总分

int womenscore; //女子团体总分

int totalscore;    //团体总分

int jifeng;          //学校积分

struct  STUDENT students[10];  //参赛运动员

struct SchoolStruct \*next;      //下一个参赛学校

}SCHOOLSTRUCT;

操作需求：（1）建立参赛学校链表；（2）添加获奖学生；（3）成绩统计；（4）按项目编号查询取得前三或前五名的学校；（5）按学校编号查询学校某个项目；（6）在幕输出数据。

**3、一元多项式计算**

任务：能够按照指数降序排列建立并输出多项式；能够完成两个多项式的相加、相减和相乘，并将结果输出；

在上交资料中请写明：存储结构、多项式相加的基本过程的算法（可以使用程序流程图）、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；

设有一元多项式Am(x)和Bn(x).

Am(x)=A0+A1x1+A2x2+A3x3+… +Amxm

Bn(x)=B0+B1x1+B2x2+B3x3+… +Bnxn

请实现求M(x)=Am(x)+Bn(x)，M(x)=Am(x)-Bn(x)和M(x)=Am(x)×Bn(x)。

要求：

1) 首先判定多项式是否稀疏

2) 分别采用顺序和动态存储结构实现；

3) 结果M(x)中无重复阶项和无零系数项；

4) 要求输出结果的升幂和降幂两种排列情况

**4、订票系统**

通过此系统可以实现如下功能：

录入：可以录入航班情况（数据可以存储在一个数据文件中，数据结构、具体数据自定）

查询：可以查询某个航线的情况（如输入航班号，查询起降时间，起飞抵达城市，航班票价，票价折扣，确定航班是否满仓；输入起飞抵达城市，查询飞机航班情况。）

订票：可以订票，如果该航班已经无票，可以提供相关可选择航班；（订票情况可以存在一个数据文件中，结构自己设定）

退票：可退票，退票后修改相关数据文件；

客户资料有姓名，证件号，订票数量及航班情况，订单要有编号。

修改航班信息：当航班信息改变可以修改航班数据文件

要求：根据以上功能说明，设计航班信息，订票信息的存储结构，设计程序完成功能；

航班信息数据结构

typedef char keytype;

//航班信息结构

typedef struct

{

char start[6];//起点站

char end[6];//终点站

char sche[10];//航班期

char time1[5];//起飞时间

char time2[5];//到达时间

char model[4];//机型

int price; //票价

}infotype;

//定义航班节点

typedef struct

{

keytype keys[keylen];//航班号

infotype others;//航班信息

int next; //下一航班

}slnode;

//航班表

typedef struct

{

slnode sl[maxspace];

int keynum;

int length;

}sllist;

操作实现：

（1）录入航班信息：

（2）查询航班信息：

可以查询某个航线的情况（如，输入航班号，查询起降时间，起飞抵达城市，航班票价，票价折扣，确定航班是否满仓）；

可以输入起飞抵达城市，查询飞机航班情况；

航班信息查询系统：可以按航班号、起点站、.终点站、起飞时间、到达时间进行查询。

（3）航班订票：可以订票，如果该航班已经无票，可以提供相关可选择航班；（订票情况可以存在一个数据文件中，结构自己设定）

（4）航班退票：可退票，退票后修改相关数据文件；

客户资料有姓名，证件号，订票数量及航班情况，订单要有编号。

（5）修改航班信息：当航班信息改变可以修改航班数据文件

注：因为航班号为两位字母后跟数字，所以在排序时应该使用多关键字的基数排序对航班号进行排序。

**5、joseph环**

任务：编号是1,2,…,n的n个人按照顺时针方向围坐一圈，每个人只有一个密码（正整数）。一开始任选一个正整数作为报数上限值m，从第一个仍开始顺时针方向自1开始顺序报数，报到m时停止报数。报m的人出列，将他的密码作为新的m值，从他在顺时针方向的下一个人开始重新从1报数，如此下去，直到所有人全部出列为止。设计一个程序来求出出列顺序。

要求：利用单向循环链表存储结构模拟此过程，按照出列的顺序输出各个人的编号。

测试数据：m的初值为20，n=7 ，7个人的密码依次为3，1，7，2，4，7，4，首先m=6，则正确的输出是什么？

要求：输入数据：建立输入处理输入数据，输入m的初值，n ，输入每个人的密码，建立单循环链表。

输出形式：建立一个输出函数，将正确的输出序列

数据结构：

typedef struct Node

{

int data;

int password;

struct Node \*next;

}Node, \*LinkList;

基本操作：初始化单链表；给每个人赋密码；确定需要处理的人数；确定开始的上限值；得到正确的顺序；输出结果。

**6、学生成绩管理**

实现功能：输入、输出、插入、删除、查找、追加、读入、显示、保存、拷贝、排序、索引、分类合计、退出。

能实现对学生信息的简单管理。

具体要求：

建立一个4个学生的信息登记表，每个学生的信息包括：学号，姓名，和3门课程的成绩（FOX,C,ENGLISH）。

 程序运行时显示一个简单的菜单，例如：

（1）信息输入（INPUT）

（2）总分统计（COUNT）

（3）总分排序（SORT）

（4）查询(QUERY）

其中：

（1）对4个学生的信息进行输入；

（2）对每个学生的3门课程统计总分；

（3）对4个学生的总分按降序排序并显示出来；

（4）查询输入一个学号后，显示出该学生的有关信息；

数据结构：

struct student

{

int num;//学号

char name[20];//姓名

int foxscore;//fox成绩

int cscore;//C语言

int englishscore;//英语成绩

struct student \*next;

};

操作：成绩信息输入；统计总分；排序；查询。

**7、模拟旅馆管理系统中的床位的分配与回收**

问题描述：某旅馆有n个等级的房间，第i等级有a个房间，每个等级有b个床位(1<=i<=n)。模拟旅馆个管理系统中床位的分配和回收功能，设计能为单个旅客分配床位，在其离店便回收床位（供下次分配）的算法。

数据结构：

struct Room

{

int roomgrade; //房间等级

int roomprice;

int roomnumber;

int bed[N]; //每个等级i的每个房间有对应等级i铺床

int sex;

int peoplein;

int arrtime;

}room[100];

数据结构的具体说明：

程序中假设有N（10）个不同等级的房间，每个等级 i 有对应 i 个房间，每个房有第 i 铺床，如等级为3的房间有3个房间，每个房间有3铺床．并且程序用静态room[100]数组来实现。

程序设计思路及其测试提示：

（1）首先先创建旅馆房间的信息，即对房间信息进行初始化，用函数creat()来实现．并且创建的信息保存在文件里，每执行一次服务都对文件重新写入，保持着最近更新的数据．

（2）先令t=1，满足条件，紧接着输入s的值，第一次输入s的值时输入非0的数，满足进行订房服务的条件，接着进行旅客订房登记服务，需要注意的是本程序只适合对单个旅客进行服务。

（3）根据程序的提示，一直对输入s的值进行测试，如果输入s的值为0则进行退房服务，接着输入t的值，如果输入t的值非0，则可以继续进行服务，接下去是输入s的值，如果输入的s为0，则可以进行退房服务，否则进行订房登记服务。

（4）程序中实现连续多次订房服务是利用输入s的非0来实现的，如果想连续多次进行退房服务，则在输入t（非0）之后，输入s的值为0，循环这种操作，即可实现连续多次的退房服务。

**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

**9、报数问题**

问题描述：有n个小朋友围成一圈玩游戏，小朋友从1至n编号，2号小朋友坐在1号小朋友的顺时针方向，3号小朋友坐在2号小朋友的顺时针方向，……，1号小朋友坐在n号小朋友的顺时针方向。  
　　游戏开始，从1号小朋友开始顺时针报数，接下来每个小朋友的报数是上一个小朋友报的数加1。若一个小朋友报的数为k的倍数或其末位数（即数的个位）为k，则该小朋友被淘汰出局，不再参加以后的报数。当游戏中只剩下一个小朋友时，该小朋友获胜。  
　　例如，当n=5, k=2时：  
　　1号小朋友报数1；  
　　2号小朋友报数2淘汰；  
　　3号小朋友报数3；  
　　4号小朋友报数4淘汰；  
　　5号小朋友报数5；  
　　1号小朋友报数6淘汰；  
　　3号小朋友报数7；  
　　5号小朋友报数8淘汰；  
　　3号小朋友获胜。  
　　给定n和k，请问最后获胜的小朋友编号为多少？

输入格式

　　输入一行，包括两个整数n和k，意义如题目所述。

输出格式

　　输出一行，包含一个整数，表示获胜的小朋友编号。

样例输入

5 2

样例输出

3

样例输入

7 3

样例输出

4

数据规模和约定

对于所有评测用例，1 ≤ n ≤ 1000，1 ≤ k ≤ 9。

要求：利用单向循环链表存储结构模拟此过程。

**（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；

**2、迷宫问题求解**

任务：可以输入一个任意大小的迷宫数据，用非递归的方法求出一条走出迷宫的路径，并将路径输出；

要求：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）、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；

**3、停车场管理**https://blog.csdn.net/csf111/article/details/6846131

设停车场（如下图1所示）内只有一个可停放几量汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆到达时的先后顺序，依次由北向南排列（大门在最南端，最先到达的第一辆车停放在车场的最北端），若车场内已经停满几量汽车，则后来的汽车只能在门外的便道上等候，一旦停车场内有车开走，则排在便道上的第一辆汽车即可开入；当停车场内某车辆要离开时，由于停车场是狭长的通道，在它之后开入车场的车辆必须先退出车场为它让路，待该车辆开出大门外后，为它让路的车辆再按原次序进入车场。在这里假设汽车不能从便道上开走。试设计一个停车场管理程序（这里只是一个假想的停车场管理，并不代表实际的停车场管理）。

北

停

车

场

大门

入口 便道 出口

图1 停车场示意图

分析：汽车在停车场内进出是按照栈的运算方式来实现的，先到的先进停车场；停车场的汽车离开停车场时，汽车场内其它汽车为该辆汽车让路，也是按栈的方式进行；汽车在便道上等候是按队列的方式进行的。因此，将停车场设计成一个栈，汽车让路也需要另一个栈来协助完成，汽车进出便道用队列来实现。

本设计，栈采用顺序栈结构，队列用链式存储结构。

存储结构定义如下：

#define stacksize 10

typedef struct sqstack

{

int data[stacksize];

int top;

} SqStackTp;

typedef struct linked\_queue

{

int data;

struct linked\_queue \* next;

}LqueueTp;

typedef struct

{

LqueueTp \*front , \*rear ;

} QueptrTp;

停车场管理的算法描述如下：

1）接受命令和车号，若是汽车要进停车场，先判断停车场栈是否满，若不满，则汽车入栈，否则汽车进入便道队列等候。

2）若是汽车要离开停车场，为给汽车让路，将停车场栈上若干辆汽车入临时栈，等这辆车出停车场后，临时栈中的汽车出栈，在回到停车场栈，然后看便道队列是否为空，若不空则说明有汽车等候，从队头取出汽车号，让该车进入停车场栈。

3）重复1），2）直到为退出命令（车号为0或负数）。

**（三）树型结构题目**

**1、二叉树的构造**

任务：已知二叉树的层序和中序遍历序列，或已知二叉树的先序序列、中序序列，试编写算法建立该二叉树（ 用递归或非递归的方法都可以）。

要求：能够输入树的各个结点，并能够输出用不同方法遍历的遍历序列；分别建立建立二叉树存储结构的的输入函数、输出层序遍历序列的函数、输出先序遍历序列的函数；

**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，并输出其哈夫曼树。

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。

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。

**3、采用树型结构实现如下系统功能：**

1)输入文件以存放最初家谱中各成员的信息。 成员的信息中均应包含以下内容：

姓名、出生日期、婚否、地址、健在否、死亡日期（若其已死亡）也可附加其它信息、但不是必需的。

2)实现数据的存盘和读盘。

3)以图形方式显示家谱。

4)显示第n代所有人的信息。

5)按照姓名查询，输出成员信息（包括其本人、父亲、孩子的信息）。

6)按照出生日期查询成员名单。

7)输入两人姓名，确定其关系。

8)某成员添加孩子。

9)删除某成员（若其还有后代，则一并删除）。

10)修改某成员信息。

11)按出生日期对家谱中所有人排序。

12)打开一家谱时，提示当天生日的健在成员。

测试要求：

1)建立至少10个成员的数据，以较为直观的方式显示结果，并提供文稿形式以便检查。

2)对界面的要求是：有合理的提示，每个功能可以设立菜单，根据提示，可以完成相关的功能要求。

**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

**6、线索二叉树**

要求：程序从文件中读取每个结点的信息，然后建立一个二叉树。通过菜单的形式，选择不同的线索化方式，然后再对线索化的二叉树经行遍历并输出对应的结果。

提示：

（1）文件的读入：程序首先要从一个文件中读入结点的信息。故需建立一个文件，在文件中，给出每个结点的信息。

（2）菜单的实现：由于需要建立两种不同的线索二叉树，故需要一个菜单，来选择需要进行的操作，并输出操作的结果。

（3）树的建立：从文件中，读入每个结点的信息。然后建立起树，然后对已建立的树进行对应的线索化。

（4）树的线索化：根据菜单的选择，对已建立的树经行相对于的线索化。

（5）输出遍历的结果：对每种不同的线索化的树，给出想对应的算法。然后，根据算法把每种遍历的结果输出。

**7、树与二叉树的转换**

树与二叉树的转换的实现。以及树的前序、后序的递归、非递归遍历算法，层次序的非递归遍历算法的实现，应包含建树的实现。

**（四）图型结构题目**

**1、校园导游程序**

问题描述：用无向网表示你所在学校的校园景点平面图，图中顶点表示主要景点，存放景点的编号、名称、简介等信息，图中的边表示景点间的道路，存放路径长度等信息。要求能够回答有关景点介绍、游览路径等问题。

基本要求：查询各景点的相关信息；查询图中任意两个景点间的最短路径；查询图中任意两个景点间的所有路径；增加、删除、更新有关景点和道路的信息。

选作内容：

①求多个景点的最佳（最短）游览路径。

②区分机动车道和人行道。

③实现导游图的仿真界面。

数据结构：

typedef struct message

{

int num; //景点代码

char name[100]; //景点名称

char pro[500]; //简介

}Ciceroni;

Ciceroni school[10]={{1,"行政楼\n"},{2,"食堂\n"},{3,"赛博楼,信息分院办公室所在地\n"},{4,"求是楼,实验楼计算机中心\n"},{5,"格致楼,法学管理学院"},

{6,"工程实习中心,金工实习\n"},{7,"仰仪楼,机电计测分院\n"},{8,"体育馆,旁边有篮球场`足球场`还有网球场\n"},{9,"一号教学楼，主要以阶梯教室为主\n"},{10,"二号教学楼，小教室为多\n"}}; /\*景点名称和简介\*/

操作：给景点之间的路径赋最大值；最短路径的C语言函数；输出最短路径和最短距离函数；输入景点代码查景点名称和简介；输入景点代码查到其它景点的最短距离。

**2、拓扑排序**

任务：编写函数实现图的拓扑排序。

**3、图的最小生成树**

若要在n个城市之间建设通信网络，只需要架设n-1条线路即可。如何以最低的经济代价建设这个通信网，是一个网的最小生成树问题。

（1）建立一个图，其存储方式可以采用邻接矩阵形式，需要定义两个数组，一个存储顶点，一个存储边，存储边的数组表明节点间的连通关系和边的权值；

（2）利用普里姆算法和克鲁斯卡尔算法求网的最小生成树；

（3）按顺序输出生成树中各条边以及它们的权值。

**4、渡河问题https://blog.csdn.net/kilvdn/article/details/881438**

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表示在目的岸，1表示在起始岸）。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。

（4）求出所有的解。

提示：

（1）需求分析

* 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）。
  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

* 1. 求出所有的解。
  2. 程序测试：用户输入修道士与野人个数以及一条船可容纳的人数，则程序输出可行的渡河状态图并输出最优解，程序最后给出可行解的个数。

注意：

1. 程序的输出格式严格按照三元组的形式，给出状态变迁图
2. 必须采用广度搜索算法

（2）设计

设计思想

1)存储结构

定义一个结构体，用于存放各个时刻的状态

typedef struct

{

int xds;//修道士

int ymr;//野蛮人

int zt;//状态

}DataType;

用邻接表存储结构实现图的操作，其存储结构为：

typedef struct Node

{

int dest; //邻接表的弧头结点序号

struct Node \*next;

}Edge; //邻接表单链表的结点结构体

typedef struct

{

DataType data; //结点数据元素

int sorce; //邻接表的弧尾结点序号

Edge \*adj; //邻接边的头指针

int pre; //指向此点的点的序号

}AdjLHeight; //数组的数据元素类型结构体

typedef struct

{

AdjLHeight a[10000]; //邻接表数组

int numOfVerts; //结点个数

int numOfEdges; //边个数

}AdjLGraph; //邻接表结构体

基本思想

1)由题意知，数据结构选用图较为合理，题中图的结点数目较大且边的数目远小于相同结点的完全图的边数，因此采用图的邻接表存储结构效率较高

2)由于在每个状态时刻均要判断修道士是否安全，可考虑将安全检查编写为一个函数

设计表示法

1)过程或函数调用关系图

main → work → jiancha → print

2)基于数据结构的操作组

该程序数据结构相对简单，只运用了邻接表结构的图，work()函数建立一个广度表，实现广度搜索算法；jiancha()函数用于检查各个状态下修道士是否安全；print()函数打印安全渡河的过程

3)过程或函数接口规格说明

void work(AdjLGraph \*p) //广搜建立表

int jiancha(DataType x) //检查当前情况下，修道士是否安全

int print(AdjLGraph \*p,int g) //打印安全渡河的过程

（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。

**5、中国道路交通网络信息查询系统**

（1）问题描述

出于不同的目的的旅客对交通工具有不同的要求。例如，因公出差的旅客希望在旅途中的时间尽可能短，出门旅游的游客则期望旅费尽可能省，而老年旅客则要求中转次数最少。编制一个全国城市间的交通咨询程序，为旅客提供两种或三种最优决策的交通咨询。

（2）基本要求

1）提供对城市信息进行编辑（如：添加或删除）的功能。

2）城市之间有两种交通工具：火车和飞机。提供对列车时刻表和飞机航班进行编辑（增设或删除）的功能。

3）提供两种最优决策：最快到达或最省钱到达。全程只考虑一种交通工具；

4）旅途中耗费的总时间应该包括中转站的等候时间；

5）咨询以用户和计算机的对话方式进行。由用户输入起始站、终点站、最优决策原则和交通工具，输出信息：最快需要多长时间才能到达或者最少需要多少旅费才能到达，并详细说明依次于何时乘坐哪一趟列车或哪一次班机到何地。

（3）实现提示

1)对时刻表和飞机航班进行编辑,应提供文件输入和键盘输入两种形式。飞机航班信息包括:起始站的出发时间,终点站的到达时间和票价;列车时刻表则需 根据交通图给出各个路段的详细信息,如:对从北京到上海的火车,给出北京至天 津,天津至徐州及徐州至上海各段的出发时间,到达时间及票价等信息。

2)以邻接表作交通图的存储结构,表示边的结构内除含有邻接点的信息外,还包括交通工具,路途中耗费的时间和花费以及出发和到达时间等多种属性。

（4）选做内容

增加旅途中中转次数最少的最优决策。

**6、教学计划编制问题**

（1）问题描述

大学的每个专业都要制定教学计划。假设任何专业都有固定的学习年限，每学年含两学期，每学期的时间长度和学分上限值均相等。每个专业开设的课程都是确定的，而且课程在开设时间的安排必须满足先修关系。每门课程有哪些先修课程是确定的，可以有任意多门，也可以没有。每门课程恰好占一个学期。试在这样的前提下设计一个教学计划编制程序。

（2）基本要求

1）输入参数包括：学期总数，一学期的学分上限，每门课的课程号（固定占3位的字母数字串）、学分和直接先修课的程程号。

2）允许用户指定下列两种编排策略之一：是使学生在各学期中的学习负担尽量均匀；二是使课程尽可能地集中在前几个学期中。

3）若根据给定的条件问题无解，则报告适当的信息；否则将教学计划输出到用户指定的文件中。计划的表格格式自行设计。

（3）实现提示

可设学期总数不超过12，课程总数不超过100.如果输入的先修课程号不在该专业开设的课程序列中，则作为错误处理。应建立内部课程序号与课程号之间的对应关系。

（4）选择内容

产生多种（例如5种）不同的方案，并使方案之间的差异尽可能地大。

**7、行车路线https://blog.csdn.net/Windlight\_/article/details/79515343**

小明和小芳出去乡村玩，小明负责开车，小芳来导航。  
　　小芳将可能的道路分为大道和小道。大道比较好走，每走1公里小明会增加1的疲劳度。小道不好走，如果连续走小道，小明的疲劳值会快速增加，连续走s公里小明会增加s2的疲劳度。  
　　例如：有5个路口，1号路口到2号路口为小道，2号路口到3号路口为小道，3号路口到4号路口为大道，4号路口到5号路口为小道，相邻路口之间的距离都是2公里。如果小明从1号路口到5号路口，则总疲劳值为(2+2)2+2+22=16+2+4=22。  
　　现在小芳拿到了地图，请帮助她规划一个开车的路线，使得按这个路线开车小明的疲劳度最小。

输入格式

输入的第一行包含两个整数n, m，分别表示路口的数量和道路的数量。路口由1至n编号，小明需要开车从1号路口到n号路口。  
　　接下来m行描述道路，每行包含四个整数t, a, b, c，表示一条类型为t，连接a与b两个路口，长度为c公里的双向道路。其中t为0表示大道，t为1表示小道。保证1号路口和n号路口是连通的。

输出格式

　　输出一个整数，表示最优路线下小明的疲劳度。

样例输入

6 7

1 1 2 3

1 2 3 2

0 1 3 30

0 3 4 20

0 4 5 30

1 3 5 6

1 5 6 1

样例输出

76

样例说明

从1走小道到2，再走小道到3，疲劳度为52=25；然后从3走大道经过4到达5，疲劳度为20+30=50；最后从5走小道到6，疲劳度为1。总共为76。

数据规模和约定

对于30%的评测用例，1≤n≤8，1≤m≤10；  
　　对于另外20%的评测用例，不存在小道；  
　　对于另外20%的评测用例，所有的小道不相交；  
　　对于所有评测用例，1≤n≤500，1≤m≤105，1≤a, b≤n，t是0或1，c≤105。保证答案不超过106。

**（五）查找、排序、文件**

**1、二叉排序树与文件操作**

功能要求：

（1）从键盘输入一组学生记录建立二叉排序树；

（2）中序遍历二叉排序树；

（3）求二叉排序树深度；

（4）求二叉排序树的所有节点数和叶子节点数；

（5）向二叉排序树插入一条学生记录；

（6）从二叉排序树中删除一条学生记录；

（7）从二叉排序树中查询一条学生记录；

（8）以广义表的形式输出二叉排序树

//定义学生记录类型

Struct student

{

Char num[6]; //学号

Int grade; //成绩

};

//定义二叉排序树节点值的类型为学生记录类型

typedef student ElemType;

//定义二叉排序树的节点类型

typedef Struct BSTNode

{

ElemType data;

Struct BSTNode \*left;

Struct BSTNode \*rchild;

} BSTNode;

**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. 清除B\_树，即收回B\_树中的所有节点；

//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

};

**3、散列表的插入、删除和查找**

功能要求：

（1）初始化散列表；

（2）向散列表中插入一个元素；

（3）从散列表中删除一个元素；

（4）从散列表中查找一个元素。

散列表通常采用链接法处理冲突，散列文件中每个节点的类型定义为：

Struct FLNode

{ //散列主文件中的节点类型

ElemType data ; //值域

Int next; //指向下一个节点的指针域

};

其中data域用来存储待散列的元素，next域用来存储下一个同义词元素在散列表中的存储位置，即所在节点的位置号。

**五、上交相关内容要求**

上交成果的内容由以下两个部分组成。

（1）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）课程设计报告：按照附件1要求书写课程设计报告。

**六、实习成绩考核办法**

最终成绩=验收成绩\*70%+报告成绩\*30%。

具体成绩评价指标和标准如下所示：

**数据结构课程设计验收（占总成绩70%）**

|  |  |  |
| --- | --- | --- |
| 评价指标 | 评分标准 | 得分 |
| 1、学生的参与意愿（出勤情况）（5分） | 根据学生出勤情况给分，点5次名，每次点名1分。 |  |
| 2、学生的协作意愿（5分） | A：积极帮助他人：4-5分  B：较为积极的帮助他人：3-4分  C：有一定的意愿：2-3分  D：不愿意帮助他人：2分以下 |  |
| 3、算法功能介绍情况（20分） | A：算法功能介绍全面、流畅、重点突出：18-20分  B：算法功能介绍全面、流畅，重点内容较突出：16-18分  C：算法功能介绍全面、较流畅，重点内容较突出：14-16分  D：算法功能介绍较全面、较流畅、重点内容较突出：12-14分  E：算法功能介绍不全面，表述不清楚：12分以下 |  |
| 4、回答问题情况（20分） | A：回答问题准确：18-20分  B：回答问题较准确：16-18分  C：回答问题不全面：14-16分  D：能够回答基本问题：12-14分  E：不能正确回答问题：12分以下 |  |
| 5、课程设计内容完成情况  （20分） | A：很好的完成了所承担的设计任务，算法设计有新意，程序调试顺利，结果正确：18-20分  B：较好的完成了所承担的设计任务，算法设计完整，程序调试顺利，结果正确：16-18分  C：基本完成了所承担的设计任务，算法设计较完整，程序调试较顺利，结果正确：14-16分  D：基本完成了所承担的设计任务，算法设计基本完整，程序基本调试成功，结果基本正确：12-14分  E：部分完成了所承担的设计任务，算法设计不合理，结果不完全正确：12分以下 |  |
| 总分（百分制）： |  | |

**数据结构课程设计报告（占总成绩30%）**

|  |  |  |
| --- | --- | --- |
| 评价指标 | 评分标准 | 得分 |
| 1、需求和规则说明、设计思想、设计表示（每个题目8分，要求完成5个题目，总计40分） | 非常合理：7-8分；合理：6-7分；比较合理：5-6分；一般：4-5分；不合理：4分以下。 |  |
| 2、解决方案（每个题目12分，要求完成5个题目，总计60分） | 非常合理：11-12分；合理：10-11分；比较合理：8-10分；一般：6-8分；不合理：6分以下。 |  |
| 总分（百分制）： |  | |

**附件1 课程设计报告书写模板**

题目名称：二叉树的基本操作演示系统

# 1需求和规格说明

1.1根据先序字串建立一棵二叉树。

1.2用递归的方式遍历一棵二叉树，先序，中序和后序任意选取一种，并打印出访问字串。

1.3用非递归的方式遍历一棵二叉树，先序，中序和后序任意选取一种，并打印出访问字串。

# 2设计

## 2.1设计思想

树的显示方式采用树控件，并同时在内存中建立一棵同样的树。程序先要求用户输入合法的先序字串，当用户输入完成后，程序会自动检验字串是否正确。如果字串正确，程序首先在树控件中递归加入节点，建立一棵可视化的树。当用户点击先序递归遍历，程序会直接打印出先序遍历的字串；当用户点击中序非递归遍历，程序也会打印出中序访问的字串；当用户选择演示功能，在树控件中就会动态的显示出遍历的过程。

## 2.2设计表示（存储结构）

typedef char ElemType;

typedef struct node

{

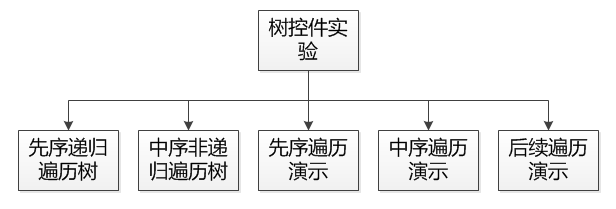
ElemType data;

struct node \*lchild, \*rchild;

}BNode, \*BinTree;

## 3 解决方案

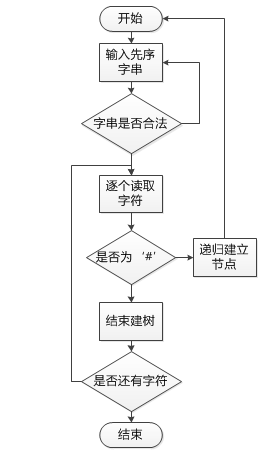
### 3.1功能示意图



图表 总功能示意图

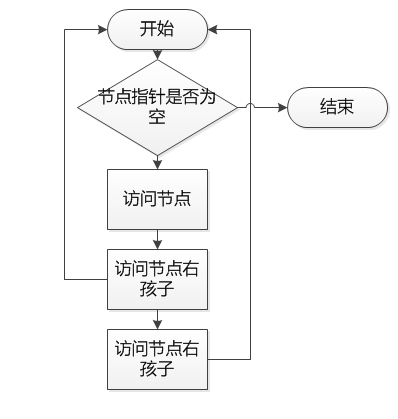
### 3.2流程示意图

**（一）建树流程**



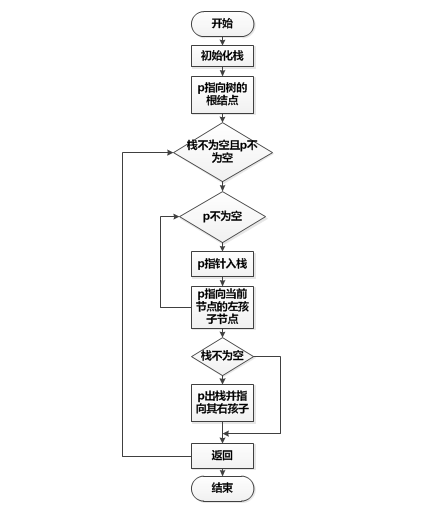
图表 建树流程示意图

**（二）先序递归遍历**



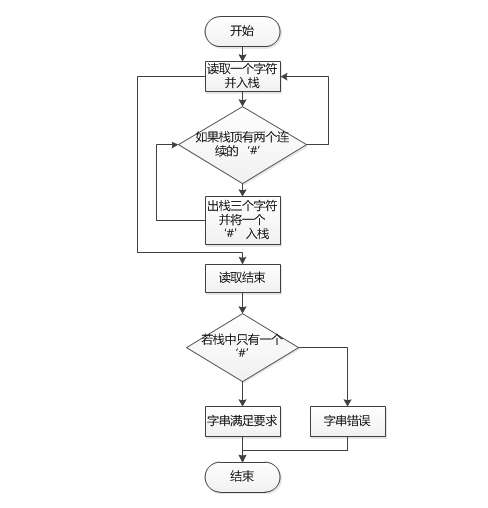
图表 先序遍历流程示意图

**（三）中序非递归遍历**



图表 中序非递归遍历示意图

**（四）先序字串错误检验**



图表 字串检验流程示意图

# 4调试报告

（1）输入“123###5##” 建树成功，先序遍历结果：1234；中序遍历结果：3214；

（2）输入“1248#F##9##5M##AGI###H##36B##C##7D##E##” 建树成功，先序遍历结果: 1248F95MAGIH36BC7DE；中序遍历结果：8F492M5IGAH1B6C3D7E;

（3）输入“1231###2121##211”建树错误；

（4）输入“1323##67hgv####gyhhu###”建树错误；

注：这部分内容主要写实验运行情况，输入和输出之间的关系，可以把程序运行的界面写下来