

数据结构与算法专题实验



张亚明老师、何晖老师

2024. 10



数据结构与算法专题实验

- 实验一 背包问题的求解
- 实验二 农夫过河问题的求解
- 实验三 简易电子表格的设计
- 实验四 八皇后问题
- 实验五 约瑟夫环问题仿真
- 实验六 教学计划编制问题 (*)
- 实验七 二叉排序树与平衡二叉树的实现 (*)
- 实验八 停车场模拟管理程序的设计与实现
- 实验九 学生成绩分析
- 实验十 一元稀疏多项式计算器
- 实验十一、哈夫曼压缩/解压缩算法 (编译码器) (*)
- 实验十二、全国交通咨询模拟系统 (*)
- 实验十三、迷宫问题 (*)



数据结构与算法专题实验

注：从上述题目中任选至少5道题目，打星号至少选2题！



成绩评定

题目完成情况的验收

基本题目

- 验收时间：上机时间完成当场验收
- 验收方式：本人简要介绍完成的情况，并检查程序运行结果
- 成绩评定：按照预先给定的各题分值进行成绩评定

选做题目

- 验收时间：上机时间完成当场验收，并给所指定的分数
- 验收方式：填写成绩评定考核表，本人简要介绍完成的情况，检查程序运行结果，提交专题实验报告
- 成绩评定：实验题目完成情况和报告完成情况，最后核定成绩
 - 结果正确程序没有BUG并有独到之处成绩为所给题目分数的90 - 100%
 - 结果正确程序没有BUG成绩为所给题目分数的75-90%
 - 结果不完全正确成绩为所给题目分数的>60-75%
 - 结果不正确成绩为所给题目分数的<60%

文件名：计算机XX班_学号_姓名

实验报告纸质版和电子版：专题实验课程结束时，各班收齐上交到 何辉老师或张亚明老师

最终成绩

- 总成绩 = 实验题目得分 + 实验报告得分



实验一 背包问题的求解

1. 问题描述

假设有一个能装入总体积为 T 的背包和 n 件体积分别为 w_1, w_2, \dots, w_n 的物品，能否从 n 件物品中挑选若干件恰好装满背包，即使 $w_1 + w_2 + \dots + w_m = T$ ，要求找出所有满足上述条件的解。

例如：当 $T=10$ ，各件物品的体积 $\{1, 8, 4, 3, 5, 2\}$ 时，可找到下列4组解：

(1, 4, 3, 2)

(1, 4, 5)

(8, 2)

(3, 5, 2)。

2. 实现提示

可利用回溯法的设计思想来解决背包问题。首先，将物品排成一列，然后，顺序选取物品装入背包，若已选取第 i 件物品后未满足，则继续选取第 $i+1$ 件，若该件物品“太大”不能装入，则弃之，继续选取下一件，直至背包装满为止。

如果在剩余的物品中找不到合适的物品以填满背包，则说明“刚刚”装入的物品“不合适”，应将它取出“弃之一边”，继续再从“它之后”的物品中选取，如此重复，直到求得满足条件的解，或者无解。

由于回溯求解的规则是“后进先出”，自然要用到“栈”。

进一步考虑：如果每件物品都有体积和价值，背包又有大小限制，求解背包中存放物品总价值最大的问题解——最优解或近似最优解。



实验二 农夫过河问题的求解

1. 问题描述

一个农夫带着一只狼、一只羊和一棵白菜，身处河的南岸。他要把这些东西全部运到北岸。他面前只有一条小船，船只能容下他和一件物品，另外只有农夫才能撑船。如果农夫在场，则狼不能吃羊，羊不能吃白菜，否则狼会吃羊，羊会吃白菜，所以农夫不能留下羊和白菜自己离开，也不能留下狼和羊自己离开，而狼不吃白菜。请求出农夫将所有的东西运过河的方案。

2. 实现提示

求解这个问题的简单方法是一步一步进行试探，每一步搜索所有可能的选择，对前一步合适的选择后再考虑下一步的各种方案。要模拟农夫过河问题，首先需要对问题中的每个角色的位置进行描述。可用4位二进制数顺序分别表示农夫、狼、白菜和羊的位置。用0表在南岸，1表示在北岸。例如，整数5 (0101)表示农夫和白菜在南岸，而狼和羊在北岸。

现在问题变成：从初始的状态二进制0000(全部在河的南岸)出发，寻找一种全部由安全状态构成的状态序列，它以二进制1111(全部到达河的北岸)为最终目标。总状态共16种(0000到1111)，(或者看成16个顶点的有向图)可采用广度优先或深度优先的搜索策略——得到从0000到1111的安全路径。

以广度优先为例：整数队列——逐层存放下一步可能的安全状态；Visited[16]数组标记该状态是否已访问过，若访问过，则记录前驱状态值——安全路径。

最终的过河方案应用汉字显示出每一步的两岸状态。



实验三 简易电子表格的设计

1. 问题描述

设计一个支持基本计算统计功能和其它一些表格管理/处理功能的软件，使用户可在该软件的支持下，用交互方式进行表格建立、数据输入、数据编辑及其它一些表格操作。即类似于简易Excel表格处理软件。

2. 基本要求

- (1) 建立表格：建立空白表格，同时在屏幕上显示，使其处于可输入数据的状态。用户可指定新建表格的行列数。
- (2) 输入数据与编辑数据：通过键盘将数据输入到显示在屏幕上的电子表格上。
- (3) 基本统计计算：可选择按行或列合计、求平均、求最大最小。
- (4) 排序：使任意指定的行或列中的数据按大小(升或降)排列，对字符型数据，还可选择大小写敏感。
- (5) 表格保存：使电子表格以磁盘文件的方式存储在磁盘上，并可随时读入，供继续处理。
- * (6) 公式支持：单元格内可输入公式(表达式)，使对应单元格的最终内容为公式的计算结果。公式最基本的形式是算术计算公式，可按名引用其它单元格。

*该功能可选做。



实验四 八皇后问题

1. 问题描述

设在初始状态下在国际象棋的棋盘上没有任何棋子（这里的棋子指皇后棋子）。然后顺序在第1行，第2行……第8行上布放棋子。在每一行中共有8个可选择的位置，但在任一时刻棋盘的合法布局都必须满足3个限制条件（1）任意两个棋子不得放在同一行（2）任意两个棋子不得放在同一列上（3）任意棋子不得放在同一正斜线和反斜线上。

2. 基本要求

编写求解并输出此问题的一个合法布局的程序。

3、实现提示：

在第 i 行布放棋子时，从第1列到第8列逐列考察。当在第 i 行第 j 列布放棋子时，需要考察布放棋子后在行方向、列方向、正斜线和反斜线方向上的布局状态是否合法，若该棋子布放合法，再递归求解在第 $i+1$ 行布放棋子；若该棋子布放不合法，移去这个棋子，恢复布放该棋子前的状态，然后再试探在第 i 行第 $j+1$ 列布放棋子。



实验五 约瑟夫环问题仿真

1. 问题描述

设编号为 $1, 2, \dots, n$ ($n > 0$) 个人按顺时针方向围坐一圈，每人持有一个正整数密码。开始时任意给出一个报数上限 m ，从第一个人开始顺时针方向自1起顺序报数，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，从他在顺时针方向上的下一个人起重新自1报数；如此下去直到所有人全部出列为止。

2. 基本要求

设计一个程序模拟此过程，给出出列人的编号序列。

3. 实现提示：

可考虑不带头结点的单链表结构。

4. 测试数据：

$N=7$ ，七个人的密码依次为3，1，7，2，4，8，4。

初始报数上限值 $m=20$ 。



实验六 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固定的，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课程号。

(2) 允许用户指定下列两种编排策略之一：一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：

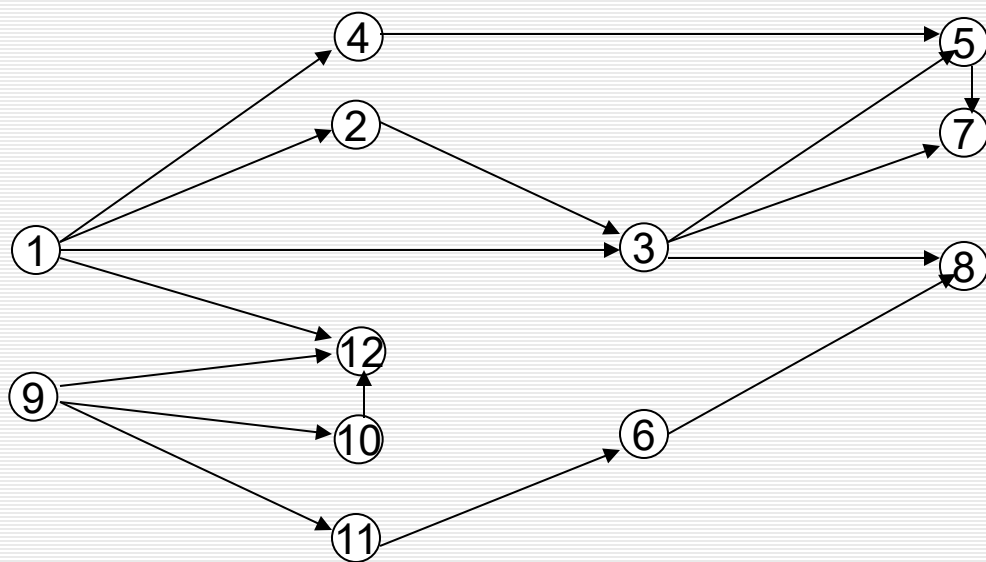
可设学期总数不超过12，课程总数小于100。如果输入的先修课程号不在该专业开设的课程序列中，则作为错误处理。



实验六 教学计划编制问题(续)

3. 测试数据

学期总数：6；学分上限：10；该专业共开设12门课，课程号从C01到C12，学分顺序为2，3，4，3，2，3，4，4，7，5，2，3。先修关系见下图。





实验七 二叉排序树与平衡二叉树的实现

1. 问题描述

分别采用二叉链表和顺序表作存储结构，实现对二叉排序树与平衡二叉树的操作。

2. 基本要求

(1) 用二叉链表作存储结构实现二叉排序树。

1) 以回车符('\n')为输入结束标志，输入数列L，生成一棵二叉排序树T；

2) 对二叉排序树T作中序遍历，输出结果；

3) 计算二叉排序树T查找成功的平均查找长度，输出结果；

4) 输入元素x，查找二叉排序树T，若存在含x的结点，则删除该结点，并作中序遍历(执行操作2)；否则，输出信息“无x”；



实验七 二叉排序树与平衡二叉树的实现(续)

(2) 用顺序表(一维数组)作存储结构——静态链表

- 1) 以回车符('\n')为输入结束标志, 输入数列L, 生成一棵二叉排序树T;
- 2) 对二叉排序树T作中序遍历, 输出结果;
- 3) 计算二叉排序树T查找成功的平均查找长度, 输出结果;
- 4) 输入元素x, 查找二叉排序树T, 若存在含x的结点, 则删除该结点, 并作中序遍历(执行操作2); 否则, 输出信息“无x”;

(3) 用二叉链表作存储结构实现平衡的二叉排序树。

- 1) 用数列L, 生成平衡的二叉排序树BT: 当插入新元素之后, 发现当前的二叉排序树BT不是平衡的二叉排序树, 则立即将它转换成新的平衡的二叉排序树BT;
- 2) 计算平衡的二叉排序树BT的平均查找长度, 输出结果。



实验八 停车场模拟管理程序的设计与实现

1. 问题描述

设停车场只有一个可停放几辆汽车的狭长通道，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的先后顺序依次排列，若车场内已停满几辆汽车，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车即可进入；当停车场内某辆车要离开时，由于停车场是狭长的通道，在它之后开入的车辆必须先退出车场为它让路，待该车辆开出大门，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，试设计这样一个停车场模拟管理程序。

2. 数据结构设计

(1) 为了便于区分每辆汽车并了解每辆汽车当前所处的位置，需要记录汽车的牌照号码和汽车的当前状态，所以为汽车定义一个新的类型CAR，具体定义如下：

```
typedef struct
{char *license_plate; //汽车牌照号码，定义为一个字符指针类型
  char state;          //汽车的当前状态，字符s表示停放在停车位上，
                      //字符p表示停放在便道上，每辆车的初始状态用
                      //字符I表示。
} CAR
```



实验八 停车场模拟管理程序的设计与实现(续)

(2) 由于车位是一个狭长的通道，所以不允许两辆车同时进入停车位，当有车到来要进入停车位的时候车要顺次停放，当某辆车要离开时，比它后到的车要先暂时离开停车位，而且越后到的车就越先离开停车位，显然这和栈的“后进先出”特点相吻合，

所以可以使用一个栈来描述停车位。

由于停车位只能停放有限的几辆车，而且为了便于停车场的管理，要为每个车位分配一个固定的编号，不妨设为1、2、3、4、5(可利用数组的下标)，分别表示停车位的1车位、2车位、3车位、4车位、5车位，针对这种情况使用一个顺序栈比较方便，具体定义如下：

```
#define MAX_STOP 5
typedef struct
{ CAR STOP[MAX_STOP];      //各汽车信息的存储空间
  into top;                //用来指示栈顶位置的静态指针
} STOPPING;
```

(3) 当停车场的停车位上都已经停满了汽车，又有新的汽车到来时要把它调度到便道上，便道上的车辆要按照进入便道的先后顺序顺次存放在便道上，为便道上的每个位置也分配一个固定的编号，当有车从停车位上离开后，便道上的第一辆汽车就立即进入停车位上的某个车位，由于问题描述中限制了便道上的汽车只能从便道上开



实验八 停车场模拟管理程序的设计与实现(续)

走，即便道上的汽车只能在停车位上停放过之后才能离开停车位，这样越早进入便道的汽车就越早进入停车位，而且每次进入停车位的汽车都是处于便道“最前面”的汽车

，显然，这和队列的“先进先出”特点相吻合。所以，这里使用一个顺序队列来描述便道，可以利用数组的下标表示便道的位置，具体定义如下：

```
#define MAX_PAVE 100 //便道不限制停放车辆的数目，设为足够大
typedef struct
{CAR PAVE[MAX_PAVE]; //各汽车信息的存放空间
int front,rear; //用来指示队头和队尾位置的静态指针
}PAVEMENT;
```

(4)当某辆车要离开停车场的时候，比它后进停车位的车要为它让路，而且当它开走之后，让路的车还要按照原来的停放次序再次进入停车位的某个车位上，为了完成这项功能，再定义一个辅助栈，停车位中让路的车依次“压入”辅助栈，待提出请求的

车开走后再从辅助栈的栈顶依次“弹出”到停车位中，对辅助栈也采用顺序栈，具体定义与停车位栈类似，如下：

```
typedef struct
{CAR BUFFER[MAX_STOP]; //各汽车信息的存储空间
int top; //用来指示栈顶位置的静态指针
}BUFFER;
```




实验八 停车场模拟管理程序的设计与实现(续)

3. 功能(函数)设计

(1) 本程序从总体上分为四个大的功能模块，分别为：程序功能介绍和操作提示模块、汽车进入停车位的管理模块、汽车离开停车位的管理模块、查看停车场停车状态的查询模块。具体功能描述如下：

1) 程序功能介绍与操作提示模块：此模块给出程序的欢迎信息，介绍本程序的功能，并给出程序功能所对应的键盘操作的提示。

函数原形为 `void welcome()`；

2) 汽车进入停车位的管理模块：此模块用来登记停车场的汽车的车牌号和对该车的调度过程并修改该车的状态，其中调度过程要以屏幕信息的形式反馈给用户来指导用户对车辆的调度。例如，当前停车位上1, 2, 3车位分别放着牌照为JF001、JF002、JF003的汽车，便道上无汽车，当牌照为JF004的汽车到来后屏幕应给出如下提示信息：

牌照为JF004的汽车进入停车位的4号车位！

按回车键继续程序的运行。

函数原形为 `void come()`；

3) 汽车离开停车位的管理模块：此模块用来为提出离开停车场的车辆作调度处理，并修改相关车辆的状态。其中调度过程要以屏幕信息的形式反馈给用户来指导用户对车辆的调度。当有车离开停车场后应该立刻检查便道上是否有车，如果有的话立



实验八 停车场模拟管理程序的设计与实现(续)

即让便道上的第一辆汽车进入停车位。例如，当前停车位上1、2、3、4、5车位分别停放着牌照为JF001、 JF002、 JF003、 JF004、 JF005的汽车，便道上的1、2位置分别停放着牌照为JF006、 JF007的汽车，当接收到JF003要离开的信息时，屏幕应给出如下提示信息：

牌照为JF005的汽车暂时退出停车位；

牌照为JF004的汽车暂时退出停车位；

牌照为JF003的汽车从停车场开走；

牌照为JF004的汽车停回停车位的3号车位；

牌照为JF005的汽车停回停车位的4号车位；

牌照为JF006的汽车从便道上进入停车位的5号车位；

按回车键继续程序的运行。

函数原形为 `void leave()`；

此函数还要调用其他对于栈和队列的基本操作。

4) 查看停车场停车状态的查询模块：此模块用来在屏幕上显示停车位和便道上各位置的状态。例如，当前停车位上1、2、3、4、5车位分别停放着JF001、JF002、JF003、 JF004、 JF005的汽车，便道上的1、2位置分别停放着牌照为JF006、JF007的汽车，当接收到查看指令后，提示信息如下：



实验八 停车场模拟管理程序的设计与实现(续)

屏幕上应显示：

停车位的情况：

1车位—JF001

2车位—JF002

3车位—JF003

4车位—JF004

5车位—JF005

便道上的情况：

1位置—JF006

2位置—JF007

按回车键继续程序的运行。

函数原形为 `void display();`

此函数还要调用其他对于栈和队列的基本操作。



实验八 停车场模拟管理程序的设计与实现(续)

(2) 以上4个总体功能模块要用到的栈和队列的基本操作所对应的主要函数如下：

函数原形	函数功能
STOPPING * init_stopping()	初始化“停车位栈”
BUFFER * init_buff()	初始化“辅助栈”
PAVEMENT * init_pavement()	初始化“便道队列”
Int car_come(int pos)	将pos指定的汽车信息输入“停车位栈”，并修改该车状态
Int car_leave(int pos)	将pos指定的汽车信息从“停车位栈”删除，并修改该车状态
Int stop_to_buff(int pos)	将pos指定的汽车信息从“停车位栈”移动到“辅助栈”
Int buff_to_stop(int pos)	将pos指定的汽车信息从“辅助栈”移动到“停车位栈”
Int pave_to_stop(int pos)	将pos指定的汽车信息从“便道队列”移动到“停车位栈”
Int car_disp(int pos)	将pos指定的汽车信息显示在屏幕上

3. 界面设计

本程序的界面力求简洁、友好，每一步需要对用户的操作进行提示，并且将操作产生的调度结果以中文的形式显示在屏幕上。文字表述精练、准确。



实验九 学生成绩分析

1. 问题描述

录入、保存一个班级学生多门课程的成绩，并对成绩进行分析。

2. 基本要求

(1) 通过键盘输入各学生的多门课程的成绩，建立相应的文件 `input.dat`。

(2) 对文件 `input.dat` 中的数据进行处，要求具有如下功能：

- 1) 按各门课程成绩排序，并生成相应的文件输出。
- 2) 计算每人的平均成绩，按平均成绩排序，并生成文件。
- 3) 求出各门课程的平均成绩、最高分、最低分、不及格人数、60-69分人数、70-79分人数、80-89分人数、90分以上人数。
- 4) 根据姓名或学号查询某人的各门成绩，重名情况也能处理。



实验九 学生成绩分析(续)

测试数据举例：

学号	姓名	数学	英语	计算机
001	王放	78	77	90
002	张强	89	67	88
003	李浩	56	66	78
004	黄鹏兵	89	86	85
005	李浩	67	88	76
006	陈利风	45	54	67
007	尚晓	78	76	70



实验十 一元稀疏多项式计算器

1. 问题描述

设计一个一元稀疏多项式简单计算器。

2. 基本要求

一元稀疏多项式简单计算器的基本功能是：

(1) 输入并建立多项式；

(2) 输出多项式，输出形式为整数序列：

$n, c_1, e_1, c_2, e_2, \dots, c_n, e_n,$

其中 n 是多项式的项数， c_i, e_i 分别是第 i 项的系数和指数，序列按指数降序排列；

(3) 多项式 a 和 b 相加，建立多项式 $a+b$ ；

(4) 多项式 a 和 b 相减，建立多项式 $a-b$ ；

(5) 计算多项式在 x 处的值；即给定 x 值，计算多项式值。

3. 实现提示

用带表头结点的单链表存储多项式，多项式的项数存放在头结点中。



实验十一、哈夫曼压缩/解压缩算法（编译码器）

1、问题描述：

利用哈夫曼编码进行信息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，要求在发送端通过一个编码系统对传输数据预先编码（压缩）；在接收端将传来的数据进行译码（解压缩复原）。试为这样的通信站编写一个哈夫曼编译码系统——哈夫曼压缩/解压缩算法。

2、基本要求：

- 1) 通信内容可以是任意的多媒体文件；
- 2) 自己设定字符大小，统计该文件中不同字符的种类（字符集、个数）、出现频率（在该文件中）；
- 3) 构建相应的哈夫曼树，并给出个字符的哈夫曼编码；
- 4) 对源文件进行哈夫曼压缩编码形成新的压缩后文件（包括哈夫曼树）；
- 5) 编写解压缩文件对压缩后文件进行解码还原成源文件。

3、实现提示：

不同源文件形成的压缩文件中应该包含相应的哈夫曼树结构，以便解压缩系统直接译码还原之。

参考哈夫曼树一节内容，但要求编写的软件能完整的对任意文件完成压缩/解压缩。



实验十二、全国交通咨询模拟系统

1、问题描述：

处于不同目的的旅客对交通工具具有不同的要求。例如，因公出差的旅客希望在旅途中的时间尽可能地短，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、设计要求

- (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：最快到达和最省钱到达。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。



实验十二、全国交通咨询模拟系统

3、实现提示

- (1) 对全国城市交通图和列车时刻表及飞机航班表进行编辑，应该提供文件形式输入和键盘输入两种方式。飞机航班表的信息应包括：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，例如：对从北京到上海的火车，需给出北京至天津、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，还应包括交通工具、路程中耗费的时间和花费以及出发和到达的时间等多种属性。
- (3) *可考虑增加旅途中转次数最少的最优决策。



实验十三、迷宫问题

1、问题描述：

迷宫实验是取自心理学的一个古典实验。在该实验中，把一只老鼠从一个无顶大盒子的门放入，在盒中设置了许多墙，对行进方向形成了多处阻挡。盒子仅有一个出口，在出口处放置一块奶酪，吸引老鼠在迷宫中寻找道路以到达出口。对同一只老鼠重复进行上述实验，一直到老鼠从入口到出口，而不走错一步。老鼠经多次试验终于得到它学习走迷宫的路线。

2、设计功能要求：

迷宫由m行n列的二维数组设置，0表示无障碍，1表示有障碍。设入口为(1, 1)，出口为(m, n)，每次只能从一个无障碍单元移到周围四个方向上任一无障碍单元。编程实现对任意设定的迷宫，求出一条从入口到出口的通路，或得出没有通路的结论。

算法输入：代表迷宫入口的坐标

算法输出：穿过迷宫的结果。 算法要点：创建迷宫，试探法查找路。