

实验一 线性结构及其应用

实验编号：1.1.1

实验分值：满分 5 分

实验题目：线性表的顺序存储结构与应用

实验内容：

实现线性表的顺序存储结构，并实现一个典型应用（如年级学生的成绩管理）。

实验要求：

1. 实现线性表的顺序存储结构（结构和基本操作算法）；
2. 利用线性表的顺序存储结构实现的成绩管理。
 - （1） 学生信息包括学号，姓名，各科成绩等，并采用文件形式存储；
 - （2） 能够对学生信息进行添加、查找、删除和修改等，并反映操作前后的结果；
 - （3） 能够对单科成绩和加权平均成绩进行排序；
 - （4） 能够对单科成绩和加权平均成绩进行范围查找。
 - （5） 以“菜单”形式组织成绩管理的各项功能。

实验编号：1.1.2

实验分值：满分 6 分

实验题目：线性表的链式存储结构与应用

实验内容：

实现线性表的链式存储结构，并实现一元多项式的代数运算。

实验要求：

以链表存储一元多项式，在此基础上完成对多项式的代数操作。

1. 能够输入多项式（可以按各项的任意输入顺序，建立按指数降幂排列的多项式，）和输出多项式（按指数降幂排列）
2. 能够计算多项式在某一点 $x=x_0$ 的值，其中 x_0 是一个浮点型常量，返回结果为浮点数。

3. 能够给出计算两个多项式加法、减法、乘法和除法运算的结果多项式，除法运算的结果包括商多项式和余数多项式。
4. 利用循环链表结构和可用空间表的思想，把循环链表表示的多项式返还给可用空间表，以减少乘法和除法运算中间结果的空间占用和结点频繁的分配与回收操作。

实验编号：1.1.3

实验分值：满分 6 分

实验题目：动态存储分配模拟与应用

实验内容：

线性表的游标存储结构来模拟内存的动态存储分配管理，并用线性表的静态存储结构完成一个典型的应用（如一元多项式的代数运算）

实验要求：

1. 实现静态链表的初始化、空间分配和回收操作，以及其他操作；
2. 利用上述存储结构实现一元多项式的代数运算（要求参见实验 1.1.2）

实验编号：1.1.4

实验分值：满分 5 分

实验题目：等价分类问题

等价关系是最基本的数学概念，它是集合上的一种特殊的二元关系，它同时具有自反性、对称性和传递性。常用等价关系来划分集合，等价分类问题是：给定一个集合 X 和在 X 上的一个等价关系 R ，则 X 中的一个元素 a 的等价类是在 X 中等价于 a 的所有元素构成的集合： $[a]_R = \{ y \mid y \in X \wedge y R a \}$ 。等价分类在实际和计算机科学中有着广泛应用。

实验要求：

1. 利用线性表的链式和顺序存储结构相结合，输入并存储等价对。
2. 设计并实现等价分类算法，并实现测试程序。
3. 分析等价分类算法的时间复杂性和空间占用情况。

实验编号：1.2.1

实验分值：满分 5 分

实验题目：算术表达式求值

表达式求值是实现程序设计语言的基本问题之一，也是栈的应用的一个典型例子。设计一个程序，演示用算符优先法对算术表达式求值的过程。

实验要求：

1. 从键盘输入任意一个语法正确的（中缀）表达式，显示并保存该表达式。
2. 利用栈结构，把上述（中缀）表达式转换成后缀表达式，并显示栈的状态变化过程和所得到的后缀表达式。
3. 利用栈结构，对上述后缀表达式进行求值，并显示栈的状态变化过程和最终结果。

实验编号：1.2.2

实验分值：满分 6 分

实验题目：迷宫问题求解

一个迷宫可以看成是由 $m \times n$ 个房间组成的矩形，迷宫内部的每个房间有 8 个方向，每个方向或者有障碍（如墙）而不能通过，或者无障碍（如有门）而能通过。入口为左上角房间，出口为右下角房间，问是否有路径从入口到出口，若有则输出一条这样的路径；否则，提示“迷宫无入口到出口的路径”。

实验要求：

1. 设计一个迷宫及其障碍的表示，并能随机或手动生成迷宫。
2. 设计并实现一个非递归的算法，找出（输出）从入口到出口的一条路径（如存在）。
3. 设计并实现一个递归的算法，找出（输出）从入口到出口的一条路径（如存在）。

以下可以不做，供思考

4. 如果有多条路径，能否找到所有路径。
5. 如果有多条路径，能否找到步数最少的路径（捷径）。

实验编号：1.2.3

实验分值：满分 5 分

实验题目：骑士周游（马踏棋盘，哈密顿路径）问题求解

将马随机放在国际象棋 8×8 的棋盘 `Board[8][8]` 的某个方格中，马按走棋规则进行移动。要求每个方格只进入一次，走遍棋盘全部的 64 个方格。

实验要求：

设计并实现一个非递归的算法，求出马的行走路线，并按求出的行走路线，将数字 1, 2, 3,, 64 一次填入一个 8×8 的方阵输出之。

实验编号：1.3.1

实验分值：满分 6 分

实验题目：银行排队模拟系统

假设某银行有 5 个服务窗口，一个统一的叫号系统。在服务时段内，每个时间单位有随机数目的顾客到来，每个新来的顾客得到一个顺序号，排入等待服务队列，等待叫号。每个服务窗口的职员处理完一个顾客的业务后，顾客即刻离开，职员呼叫下一个顾客，窗口显示下一个顾客的服务号。被叫号的顾客随即得到服务。假定每个顾客需要的服务时间也是随机的。试设计程序模拟上述场景。

实验要求：

1. 在服务时段内显示如下实时信息和统计信息：
 - (1) 每个窗口正在办理业务的顾客的顺序号
 - (2) 目前等待顾客人数
 - (3) 每个顾客得到服务之前已等待的时间单位数
2. 在所有顾客离开后，显示如下统计信息：
 - (1) 顾客等待的平均时间单位数
 - (2) 每个窗口服务的顾客数
3. （可以不做，供思考，做了可以加 1 分）

考虑去银行办业务：一般来说，服务窗口越多，队走的越快，银行经理希望顾客满意，但又不希望雇佣过多的员工。我们模拟的服务窗口有如下假设：(1)

只排一队，并且先到的人先得到服务（这是一个队列）；（2）平均每隔 15 秒就会来一位顾客；（3）如果有空闲的窗口，在顾客抵达之时就会马上处理；（4）从顾客来到窗口到处理完顾客请求，平均需要 120 秒。请模拟高峰期银行开多少个窗口最为合适。

实验编号：1.3.2

实验分值：满分 6 分

实验题目：凯撒加密法

凯撒密码 (caesar) 是罗马扩张时期朱利斯·凯撒 (Julius Caesar) 创造的，用于加密通过信使传递的作战命令，是一种简单的消息编码方式：它根据字母表将消息中的每个字母移动常量位 k (密钥)。举个例子如果 k 等于 3，则在编码后的消息中，每个字母都会向前移动 3 位：a 会被替换为 d；b 会被替换成 e；依此类推。字母表末尾将回卷到字母表开头。于是，w 会被替换为 z，x 会被替换为 a。

实验要求：

1. 设计并实现一个算法，将输入的文本字符串（只包含 a~z 的字符）进行加密后输出。
2. 设计并实现另一个算法，将已加密后的字符串解密后输出。
3. 在不知道每个字母移动位数 k (密钥) 的情况下，设计并实现加密字符串的暴力破解算法，即输出所有可能的加密前的字符串。并说明为什么容易破解？
4. (改进算法) 使用不重复密钥：即不是将每个字母移动常数位，而是利用一个密钥值列表，将各个字母移动不同的位数。在此基础上，设计并实现字符串的加密和解密算法。思考：如果还考虑暴力破解，比改进前的会难吗？为什么？

实验编号：1.4.2

实验分值：满分 6 分

实验题目：稀疏矩阵的压缩存储及其转置算法

假设稀疏矩阵采用三元组表表示，编写程序实现该矩阵的快速转置要求。输入一个稀疏矩阵，由程序将其转换成三元组表存储；转置后的三元组表示，由程序将其转换成矩阵形式后输出。

实验要求：

1. 用户可以根据自己的需求输入任意一个稀疏矩阵，通过程序将其转换成三元组存储方式。
2. 能够完成矩阵的转置功能，要求使用直观的转置和快速转置两种方法。
3. 要能够显示原矩阵和转置后的矩阵的三元组表示和矩阵形式，让用户能进行比较。
4. 程序执行的命令包括：
 - (1) 构造稀疏矩阵 M ；
 - (2) 求转置矩阵 T ；
 - (3) 显示（打印）矩阵
5. 分析直观转置算法和快速转置算法的时间和空间复杂度。
6. 快速转置方法提示

引入两个数组作为辅助数据结构：

$num[nu]$ ：表示矩阵 A 中某列的非零元素的个数；

$cpot[nu]$ ：初始值表示矩阵 A 中某列的第一个非零元素在 B 中的位置。

num 与 $cpot$ 递推关系：

$$\begin{cases} cpot[0]=0; \\ cpot[col]=cpot[col-1]+num[col-1]; & 1 \leq col < nu \end{cases}$$

三元组表实现稀疏矩阵的转置（顺序取，直接存）算法伪代码如下：

1. 设置转置后矩阵 T 的行数、列数和非零元素的个数；
2. 计算 M 中每一列的非零元素个数；
3. 计算 M 中每一列的第一个非零元素在 T 中的下标；
4. 依次取 M 中的每一个非零元素对应的三元组；
 - 4.1 确定该元素在 T 中的下标 pb ；
 - 4.2 将该元素的行号列号交换后存入 T 中 pb 的位置；
 - 4.3 预置该元素所在列的下一个元素的存放位置；