

实验题目与指导

实验分为基础实验、应用实验、扩展实验三类。

- 1) 基础实验：主要验证教材中提到的基础类，深化理解和掌握理论知识；
- 2) 应用实验：主要目标是应用教材中教授的某一个知识点，自己设计方案解决实际的问题，培养学生简单的应用能力；
- 3) 扩展实验：该类实验逻辑结构较为复杂，需要将多个知识点融会贯通，设计较为复杂的方案，以解决实际的问题并具备扩展到数据结构课程设计的功能。该类实验代码实现量较大，一般可两人合作完成。

实验二 扩展线性表

1 实验目的

通过选择下面八个题目之一进行实现，掌握如下内容：

- 进一步掌握指针、模板类、异常处理的使用
- 掌握栈的操作的实现方法
- 掌握队列的操作的实现方法
- 学习使用栈解决实际问题的能力
- 学习使用队列解决实际问题的能力
- 学习使用多维数组解决实际问题的能力

2 实验内容

2.1 题目 1——基础实验

根据栈和队列的抽象数据类型的定义，按要求实现一个栈或一个队列的基本功能（四选一）。要求：

- 1、实现一个共享栈
- 2、实现一个链栈
- 3、实现一个循环队列
- 4、实现一个链队列

编写测试 main() 函数测试栈或队列的正确性。

2.2 题目 2——基础实验

根据三元组的抽象数据类型的定义，使用三元组表实现一个稀疏矩阵。三元组的基本功能：

- 1、三元组的建立
- 2、三元组转置
- 3、三元组相乘
- 4、其他：自定义操作

编写测试 main() 函数测试三元组的正确性

测试数据：

$$\begin{pmatrix} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 13 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2.3 题目 3——应用实验

题目：八皇后问题。

八皇后问题 19 世纪著名的数学家高斯于 1850 年提出的。他的问题是：在 8*8 的棋盘上放置 8 个皇后，使其不能互相攻击，即任意两个皇后都不能处于同一行、同一列、同一斜线上。请设计算法打印所有可能的摆放方法。

提示：

- 1、可以使用递归或非递归两种方法实现
- 2、实现一个关键算法：判断任意两个皇后是否在同一行、同一列和同一斜线上

2.4 题目 4——应用实验

题目：迷宫求解问题。

迷宫求解问题如下：心理学家把一只老鼠从一个无顶盖的大盒子的入口赶进迷宫，迷宫中设置很多隔壁，对前进方向形成了多处障碍，心理学家在迷宫的唯一出口放置了一块奶酪，吸引老鼠在迷宫中寻找通路以到达出口，测试算法的迷宫如下图所示。

提示：

- 1、可以使用递归或非递归两种方法实现
- 2、老鼠能够记住已经走过的路，不会反复走重复的路径
- 3、可以自己任意设置迷宫的大小和障碍
- 4、使用“穷举求解”的方法

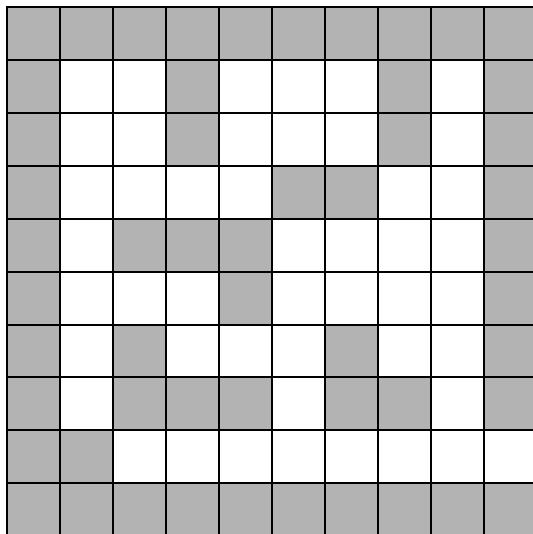


图 1-1 迷宫地图示例

2.5 题目 5——应用实验

表达式求值是程序设计语言编译中最近本的问题，它要求把一个表达式翻译成能够直接求值的序列。例如用户输入字符串“ $14+((13-2)*2-11*5)*2$ ”，程序可以自动计算得到最终的结果。在这里，我们将问题简化，假定算数表达式的值均为非负整数常数，不包含变量、小数和字符常量。

试设计一个算术四则运算表达式求值的简单计算器。

基本要求：

- 1、操作数均为非负整数常数，操作符仅为+、-、*、/、(和)；
- 2、编写 main 函数进行测试。

2.6 题目 6——应用实验

利用队列结构实现车厢重排问题。车厢重排问题如下：

一列货车共有 n 节车厢，每个车厢都有自己的编号，编号范围从 $1 \sim n$ 。给定任意次序的车厢，通过转轨站将车厢编号按顺序重新排成 $1 \sim n$ 。转轨站共有 k 个缓冲轨，缓冲轨位于入轨和出轨之间。开始时，车厢从入轨进入缓冲轨，经过缓冲轨的重排后，按 $1 \sim n$ 的顺序进入出轨。缓冲轨按照先进先出方式，编写一个算法，将任意次序的车厢进行重排，输出每个缓冲轨中的车厢编号。

提示：

- 1、一列火车的每个车厢按顺序从入轨进入不同缓冲轨，缓冲轨重排后的进入出轨，重新编排成一列货车。比如：编号为 3 的车厢进入缓冲轨 1，则下一个编号小于 3 的车厢则必须进入下一个缓冲轨 2，而编号大于 3 的车厢则进入缓冲轨 1，

排在 3 号车厢的后面,这样,出轨的时候才可以按照从小到大的顺序重新编排。

2.7 题目 7——扩展实验

在仅有黑色像素和白色像素的图像中,将相邻的黑色像素构成的点集称为一个连通域。连通域标记算法把连通区域所有像素设定同一个标记,常见的标记算法有四邻域标记算法和八邻域标记算法。四邻域标记算法中,当前黑点与上、下、左、右任意相邻黑点属于同一连通域,图 4 (a) 给出了四邻域连通域示意图。八邻域标记算法中,当前黑点与上、下、左、右及左上、左下、右上、右下任意相邻黑点属于同一连通域,图 4 (b) 给出了八邻域连通域示意图。

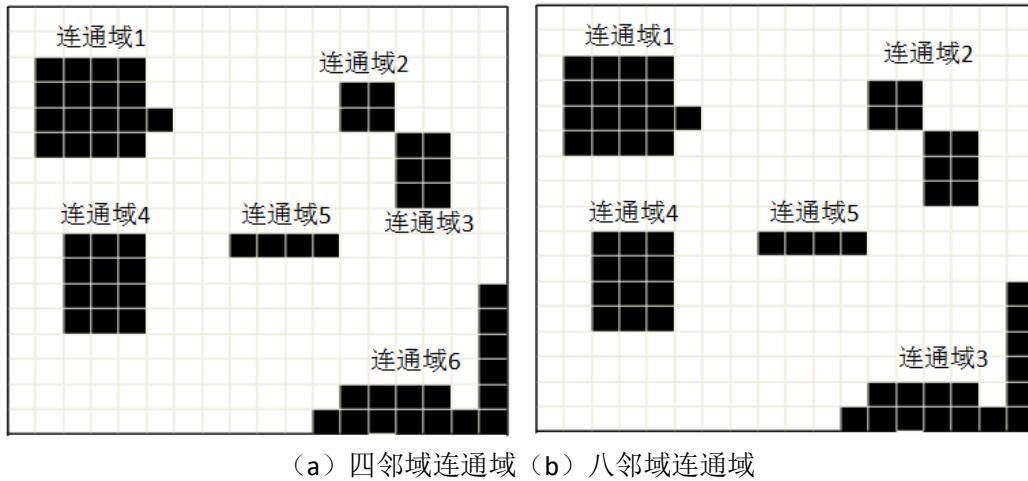


图 1-2 图像连通域示意图

试编写二值图像四邻域连通域标记算法,设图像采用 01 矩阵表示。要求:

- 1、算法尽可能优化
- 2、输出每个像素点所属的连通域标记
- 3、编写测试 main() 函数测试三元组的正确性

测试数据:

```
1010101  
10  1  0  1  11  
1011001  
1001111  
1110001  
0010001  
1111111
```

2.8 题目 8——扩展实验

实现一个识别 BMP 文件的图像类，能够进行以下图像处理。

基本要求：

- 1、能够将 24 位真彩色 Bmp 文件读入内存；
- 2、能够将 24 位真彩色 Bmp 文件重新写入文件；
- 3、能够将 24 位真彩色 Bmp 文件进行 24 位灰度处理；
- 4、能够将 24 位灰度 Bmp 文件进行 8 位灰度处理；
- 5、能够将 8 位灰度 Bmp 文件转化成黑白图像；
- 6、能够将图像进行平滑处理；
- 7、其他：自定义操作，比如翻转、亮度调节、对比度调节、24 位真彩色转 256 色等。

提示：

1、参考教材《数据结构与 STL》第四章 4.4 小节。

2、灰度处理的转换公式

$$\text{Grey} = 0.3 * \text{Red} + 0.59 * \text{Blue} + 0.11 * \text{Green}$$

3、平滑处理采用邻域平均法进行，分成 4 邻域和 8 邻域平滑，基本原理就是将每一个像素点的值设置为其周围各点像素值得平均值。

4、亮度调节公式， a 为亮度调节参数， $0 < a < 1$ ，越接近 0，变化越大

$$R = \text{pow}(R, a) * \text{pow}(255, 1 - a)$$

$$G = \text{pow}(G, a) * \text{pow}(255, 1 - a)$$

$$B = \text{pow}(B, a) * \text{pow}(255, 1 - a)$$

5、对比度调节公式， a 为对比度调节参数， $-1 < a < 1$ ，（中间值一般为 128）

$$R = \text{中间值} + (R - \text{中间值}) * (1 + a)$$

$$G = \text{中间值} + (G - \text{中间值}) * (1 + a)$$

$$B = \text{中间值} + (B - \text{中间值}) * (1 + a)$$

注意：调整对比度的时候容易发生越界，需要进行边界处理

6、24 位真彩色转 256 色，需要手动添加颜色表在 BMP 头结构中，可以使用位截断法、流行色算法、中位切分算法、八叉树算法等方法实现。

3 代码要求

1、注意内存的动态申请和释放，是否存在内存泄漏；

- 2、优化程序的时间性能；
- 3、递归程序注意调用的过程，防止栈溢出
- 4、保持良好的编程的风格：

- 代码要简洁
- 代码段与段之间要有空行和缩近
- 标识符名称应该与其代表的意义一致
- 函数名之前应该添加注释说明该函数的功能
- 关键代码应说明其功能