## 图论实验一 制作文档

一、实验内容

从文件input.txt中读入一个有向图的权矩阵表示，程序按要求输出这个图的关联矩阵，边列表，正向表，邻接表表示。

本程序以Windows平台下的Visual Studio 2012为开发环境，用C++语言实现上述功能。

二、实验要求

输入数据保存在文件input.txt 中，程序运行后将输出数据保存至**[**学号**].txt**中。这两个文件都与程序在相同的路径下。 输入文件第一行为一个整数n，表示图中结点的个数；接下来有n 行，每一行有n 个用空格隔开的整数，第i 行的第j 个整数表示矩阵中的aij。边的编号按照矩阵中从左到右，从上到下的顺序编号，即第一行第一个不为0 的权值表示的边是e1，第一行第二个不为0 的权值表示的边是e2，……

输出文件应包含这个有向图的关联矩阵、边列表、正向表、邻接表的表示。

具体如下： 第**1~n** 行为关联矩阵表示，每行有m 个数（0 或1），第i 行第j 列的数表示关联矩阵中的aij；接下来的三行为边列表表示，每行有m 个用空格隔开的整数，分别是边列表中的向量A、B 和Z；接下来三行为正向表表示，分别是正向表中的向量A、B 和Z；接下来n 行为图的邻接表表示，第i 行为第i 个结点所链接的边信息，每一条边用两个数表示，分别是权值和边的终结点编号，用空格隔开，边与边之间也用空格隔开。

三、设计思路

鉴于输入图的规模从0至100个节点不等，程序通过new生成动态二维数组，动态数组，动态链表等保存关联矩阵，边列表，正向表，邻接表等结构。

程序首先读入文件中的节点个数v，随后用其生成上述动态结构，并将文件中的权矩阵读入QJ[i][j]中，同时进行边数e的计数。通过e与v生成边列表的动态二维数组BB，并根据权矩阵QJ完成计算。随后生成关联矩阵GJ，并通过边列表计算关联矩阵各项之值。定义正向表，由ZB1，ZB2，ZB3三个动态数组组成，并通过边列表计算各项之值。最后通过结构体Node定义邻接表为动态结构体指针数组LB，通过函数add根据边列表完成所有边的加入。最后通过函数ofileM与ofileA完成对矩阵与数组的文件输出，以及邻接表的输出。

四、关键代码分析

第一个值得注意的代码段是第21行对文件打开是否成功的判断与处理，如果打开失败，则输出“Could not open the file <input.txt>”并结束程序。随后是权矩阵QJ向边列表BB的转化，通过两个for循环遍历QJ的元素，i+1与j+1分别为该边的起点与终点的编号，QJ[i][j]为权值，e+1为该条边的编号，将上述值分别赋给BB[0][e]，BB[1][e]，BB[2][e]，并递增e，即完成了BB矩阵的赋值（代码45-57行）。随后计算关联矩阵GJ，GJ每一列与BB每一列对应，根据BB中保存的起节点与终节点的编号，将GJ中对应点赋值1与-1，其余点赋值为零即可（代码61-72行）。代码76-91行实现对正向表ZB的赋值，用for循环对边列表BB中各点的以该点为始点的边数的计数并赋值给ZB1，随后将BB的第二行与第三行拷贝入ZB2与ZB3即可。邻接表通过结构体Node的结构体指针数组LB实现，从边列表BB依次将各边加入链表即可，考虑到输出顺序，因此各边的加入顺序与BB中的顺序恰好相反，这样在输出以各点为始点的边时能按照编号从大到小输出。

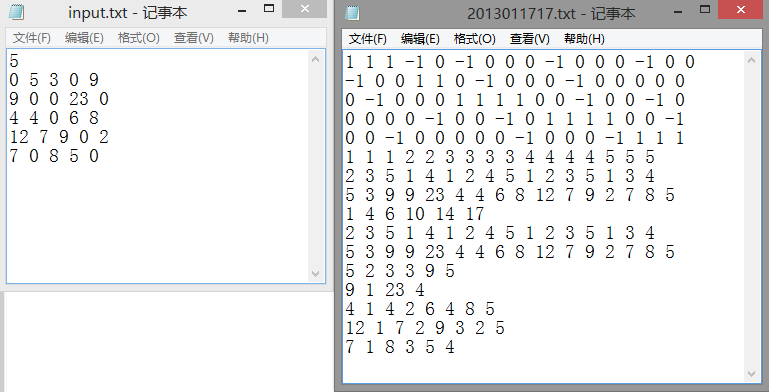
文件输出部分，程序运用C++已有的fstream，编写了函数ofileM以及ofileA，用for完成相应数据的输出并注意了格式的问题，如每一行末尾没有空格。邻接表的输出则包含在main函数中（代码105-126行），并注意了每行末尾没有多余空格以及最后一行后没有多余提行的问题。最后，程序关闭文件input.txt与2013011717.txt。

五、实验结果与分析

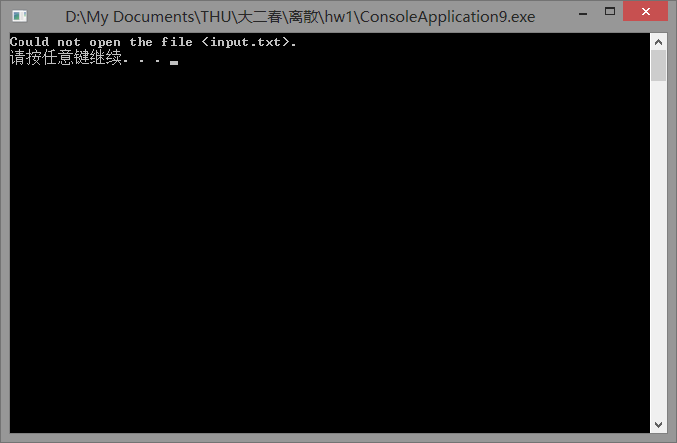
以下为样例输入与输出：



输出与所给输出一致。下面为另一组输入与输出测试：



下面为没有input.txt文件时程序的输出：



综上，程序在正确的输入下能够完成正确的输出，并能在读取文件失败时反馈给用户。

六、实验小结

本次实验要求从文件读入一个有向图的权矩阵表示，输出相应的边列表，正向表，关联矩阵与邻接表。通过这次实验，我练习了对图的不同表示的数据结构的具体实现，加深了对图的上述各种不同表示及其相互转化关系的理解，对具体的程序实现进行了实践与熟悉。

不足的是由于编程经验本身较少，程序在不少地方仍有所缺陷，如未添加判断读入的文件格式是否与要求的一样的部分，在算法方面也还有许多地方有待改进，如均用动态数组可能减慢运行速度。以上就是本次实验的小结。