第 5 次大作业

索引

数据结构

根据问题的实际背景，城市交通道路不太可能构成一个稀疏的图，故采用 V×V 的邻接矩阵储存图模型。邻接矩阵的(i,j)非零非对角元素表示 i 节点到 j 节点的权值，依照问题背景权值为大于零的整数，且不会很大，不会接近 int 整形的上限。其余矩阵元素均为初始值 0。

程序中使用数组储存矩阵。为了可读性和使用时的方便性，大部分矩阵都是使用二维数组实现的；算法实现中少量使用了一维数组存储矩阵，是为了方便内存空间的分配和释放。

另外，为了实现 Tarjan 算法，程序中用链表实现了一个简单的栈。

算法描述

1. 强连通分量的判断

本次作业将用于求取强连通分量的 Tarjan 算法稍加改动，来判断给定的有向图是否为强连通。Tarjan 算法的思路是对给定节点进行深度优先搜索，搜索时按照被访问到的次序为节点编号 DFN 并入栈，另外维护一个数组 Low，存储某节点能够回溯到的编号最小的节

点编号。当深搜到达终点时开始回溯弹栈，回溯到 DFN=Low 的时候即到达连通分量的根，期间栈中弹出的节点即构成一个连通分量。由于只要求判断是否强连通，只要弹栈时计数看连通分量中节点个数是否达到总节点个数 V，达到即为强连通，否则说明非强连通。

2. 全源最短路径权值矩阵的计算

本次作业使用 Floyd 算法计算全源最短路径边值矩阵。算法的思想是迭代，k 从 0 到 V-1，每次利用第 k 个节点缩短可能的路径权值，将 0~k 节点之间的全源最短路径权值算出，迭代到 k=V-1 就得到了全部节点的全源最短路径权值矩阵。

3. 单源最短路径的存储和输出

Floyd 算法中可以利用一个 Path 矩阵储存每对节点之间的单源最短路径，Path[i][j]表示 i 到 j 的单源最短路径 j 的前驱，即 i->…->Path[i][j]->j，Path[i][j]=-1 标志 i,j 不可达。迭代开始时对于直接相连的 i,j 有 Path[i][j]=i，迭代过程中若发现 i,j 之间应该经由 k 连接，则 Path[i][j]应改为 Path[k][j]。输出时需要回溯，由终点开始递归输出即可。

算法复杂度分析

Tarjan 算法最多进行一次深搜，时间复杂度为 O(V+E)，空间上需要记录一系列标记，需要递归调用、压栈，空间复杂度为 O(V+E)。

Floyd 算法需要 V 次迭代，每次迭代计算 V^2 个值，时间复杂度为 $O(V^3)$，空间上需要一个 V×V 的临时空间用于迭代，故空间复杂度为 $O(V^2)$。