**第7章 图**

**实验7.1 实现图的邻接矩阵和邻接表存储**

编写一个程序algo7-1.cpp，实现图的相关运算，并在此基础上设计一个主程序exp7\_1.cpp，完成如下功能：

1. 建立如图7.1所示的有向图Ｇ1的邻接矩阵，并输出。
2. 由有向图Ｇ1的邻接矩阵产生邻接表，并输出。
3. 再由（２）的邻接表产生对应的邻接矩阵，并输出。
4. 输出有向图G1从顶点0开始的深度优先遍历序列（递归算法）。
5. 输出有向图G1从顶点0开始的广度优先遍历序列（递归算法）。

３

５

４

８

７

６

１

９

５

５

图７.１　一个带权有向图Ｇ1

提示：图的邻接矩阵表示类型和邻接表表示类型定义到头文件graph.h中。

typedef int InfoType;

#define MAXV 100 /\*最大顶点个数\*/

/\*以下定义邻接矩阵类型\*/

typedef struct

{ int no; /\*顶点编号\*/

InfoType info; /\*顶点其他信息\*/

} VertexType; /\*顶点类型\*/

typedef struct /\*图的定义\*/

{ int edges[MAXV][MAXV]; /\*邻接矩阵\*/

int vexnum,arcnum; /\*顶点数，弧数\*/

VertexType vexs[MAXV]; /\*存放顶点信息\*/

} MGraph; /\*图的邻接矩阵类型\*/

/\*以下定义邻接表类型\*/

typedef struct ANode /\*弧的结点结构类型\*/

{ int adjvex; /\*该弧的终点位置\*/

struct ANode \*nextarc; /\*指向下一条弧的指针\*/

InfoType info; /\*该弧的相关信息,这里用于存放权值\*/

} ArcNode;

typedef int Vertex;

typedef struct Vnode /\*邻接表头结点的类型\*/

{ Vertex data; /\*顶点信息\*/

ArcNode \*firstarc; /\*指向第一条弧\*/

} VNode;

typedef VNode AdjList[MAXV]; /\*AdjList是邻接表类型\*/

typedef struct

{ AdjList adjlist; /\*邻接表\*/

int n,e; /\*图中顶点数n和边数e\*/

} ALGraph; /\*图的邻接表类型\*/

**实验7.2 采用普利姆算法求最小生成树**

编写程序，对于如图7.2所示的无向带权图G2采用普利姆算法输出从顶点0出发的最小生成树。

３

５

４

８

７

６

１

９

５

５

图７.2　一个带权有向图Ｇ2