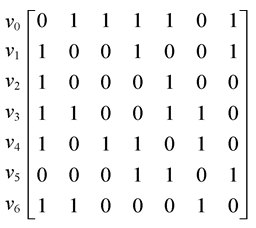
|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 |  | 成  绩 |  |
| 学号 |  |

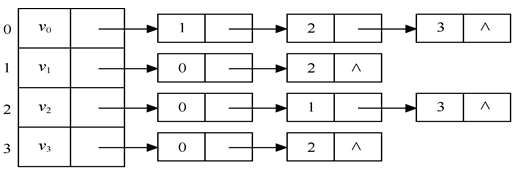
第四章 作业1

【题目】

1、已知图的邻接矩阵，给出从顶点v0出发深度优先和广度优先遍历的结果。



2、已知图的邻接表，给出从顶点v0出发深度优先和广度优先遍历的结果。



3、已知一个图的顶点集V和边集E分别为：

V={ 1,2,3,4,5,6,7 };

E={ (1,2)3, (1,3)5, (1,4)8, (2,5)10, (2,3)6, (3,4)15, (3,5)12, (3,6)9,

(4,6)4, (4,7)20, (5,6)18, (6,7)25 };

用克鲁斯卡尔（Kruskal）算法得到最小生成树，试写出在最小生成树中依次得到的各条边。

1. 设计算法，将已有用邻接矩阵存储的带权图转换成邻接表表示。

给出邻接矩阵和邻接表存储结构如下：

#define INFINITY 2000 //无穷大

#define MAX\_VERTEX\_NUM 20 //最多顶点个数

**//邻接矩阵结构定义**

Typedef struct ArcCell {

int adj;

int weight; //权值

} ArcCell, AdjMatrix [MAX\_VERTEX\_NUM] [MAX\_VERTEX\_NUM];

Typedef struct {

char vex [MAX\_VERTEX\_NUM]; //顶点

AdjMatrix arcs;

int vexnum, arcnum; //顶点和边的个数

} Mgraph;

**//邻接表结构定义**

Typedef struct ArcNode {

int adjvex;

struct ArcNode \*nextarc; //下一个邻接点

int weight; //权值

} ArcNode;

Typedef struct Vnode {

char data; //顶点

ArcNode \*firstarc; //第一个邻接点

} Vnode, AdjList [MAX\_VERTEX\_NUM];

Typedef struct {

AdjList vertices;

int vexnum, arcnum; //顶点和边的个数

} ALGraph;

【答题】（宋体小四号，单倍行距）