

7.1

- 入/出度

1: 3/0

2: 2/2

3: 1/2

4: 1/3

5: 2/1

6: 2/3

- 邻接矩阵

0 0 0 0 0 0

1 0 0 1 0 0

0 1 0 0 0 1

0 0 1 0 1 1

1 0 0 0 0 0

1 1 0 0 1 0

- 邻接表

1#

2—>1—>4#

3—>2—>6#

4—>3—>5—>6#

5—>1#

6—>1—>2—>5#

- 逆邻接表

1—>2—>5—>6#

2—>3—>6#

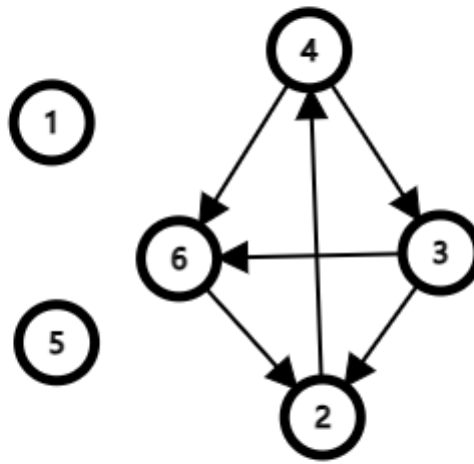
3—>4#

4—>2#

5—>4—>6#

6—>3—>4#

- 强连通分量



7.14

```

typedef struct ArcNode
{
    char adjvex;
    struct ArcNode *nextarc;
    InfoType *info;
} ArcNode;

typedef struct VNode
{
    char vex;
    ArcNode *firstarc;
    InfoType *info;
} VNode, AdjList[MAX_VERTEX_NUM];

typedef struct ALGraph
{
    AdjList vexs;
    int vexnum, arcnum;
} ALGraph;

int create(ALGraph *g)
{
    cin >> g->vexnum >> g->arcnum;
    for (int i = 0; i < g->vexnum; i++)
    {
        cin >> g->vexs[i].vex;
        cin >> g->vexs[i].info;
    }
}
  
```

```

        g->vexs[i].firstarc = NULL;
    }
    for (int i = 0; i < g->arcnum; i++)
    {
        char a, b;
        cin >> a >> b;
        int i = LocateVex(g, a);
        int j = LocateVex(g, b);
        ArcNode *p, *q;
        p = (ArcNode *)malloc(sizeof(ArcNode));
        q = (ArcNode *)malloc(sizeof(ArcNode));
        p->adjvex = a;
        p->nextarc = g->vexs[j].firstarc;
        g->vexs[j].firstarc = p;
        q->adjvex = b;
        q->nextarc = g->vexs[i].firstarc;
        g->vexs[i].firstarc = q;
        cin >> q->info >> q->info;
    }
}

```