

1、分析并回答下列问题：

- ① 图中顶点的度之和与边数之和的关系？
- ② 有向图中顶点的入度之和与出度之和的关系？
- ③ 具有 n 个顶点的无向图，至少应有多少条边才能确保是一个连通图？若采用邻接矩阵表示，则该矩阵的大小是多少？
- ④ 具有 n 个顶点的有向图，至少应有多少条弧才能确保是强连通图的？为什么？

解：

1. ① 对无向图 $G=(V, E)$, 有 $\sum_{v \in V} d(v) = 2|E|$
 对 有向图 $G=(V, E)$, 有 $\sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v) = |E|$

② 由①可知, $\sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v) = |E|$

③ $n-1$ 条边; 矩阵大小为 $n \times n$

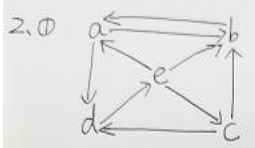
④ n ; 因为在有向图 G 中, 如果对于任何两个不相同的点 a, b , 从 a 到 b 和从 b 到 a 都存在路径, 则称 G 是强连通图, 强连通图必须从任一点出发都可以回到原处, 每个节点至少要 1 条出路

2、设一有向图 $G=(V, E)$, 其中 $V=\{a, b, c, d, e\}$, $E=\{\langle a, b \rangle, \langle a, d \rangle, \langle b, a \rangle, \langle c, b \rangle, \langle c, d \rangle, \langle d, e \rangle, \langle e, a \rangle, \langle e, b \rangle, \langle e, c \rangle\}$

- ① 请画出该有向图，并求各顶点的入度和出度。
- ② 分别画出有向图的正邻接链表和逆邻接链表。

解：

2. ①



设入度为 $d^-(v)$
出度为 $d^+(v)$

$a: d^+(v)=2, d^-(v)=2$
 $b: d^+(v)=1, d^-(v)=3$
 $c: d^+(v)=2, d^-(v)=1$
 $d: d^+(v)=1, d^-(v)=2$
 $e: d^+(v)=3, d^-(v)=1$

② 正邻接链表:

顶点	出度	邻接点
a	2	b, d
b	1	a
c	2	b, d
d	1	e
e	3	a, b, c

逆邻接链表:

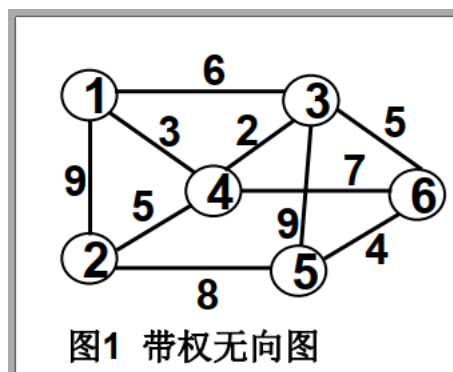
顶点	入度	前驱点
a	2	e, b
b	3	a, c, e
c	1	e
d	2	a, c
e	1	d

3、对图 1 所示的带权无向图。

① 写出相应的邻接矩阵表示。

② 写出相应的边表表示。

③ 求出各顶点的度。



解：

3、① 邻接矩阵：

∞	9	6	3	∞	∞
9	∞	∞	5	8	∞
6	∞	∞	2	9	5
3	5	2	∞	∞	7
∞	8	9	∞	∞	4
∞	∞	5	7	4	∞

② 边表： 边表：

0	1	9
0	2	6
0	3	3
1	3	5
1	4	8
2	3	2
2	4	9
2	5	5
3	5	7
4	5	4

顶点表：

0	1
1	2
2	3
3	4
4	5
5	6

③

1: $d(v) = 3$

2: $d(v) = 3$

3: $d(v) = 4$

4: $d(v) = 4$

5: $d(v) = 5$

6: $d(v) = 3$

4、 已知有向图的逆邻接链表如图 2 所示。

- ① 画出该有向图。
- ② 写出相应的邻接矩阵表示。
- ③ 写出从顶点 a 开始的深度优先和广度优先遍历序列。
- ④ 画出从顶点 a 开始的深度优先和广度优先生成树。

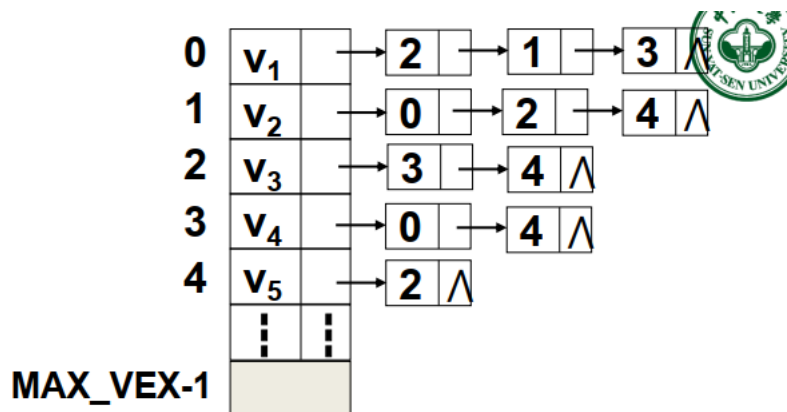
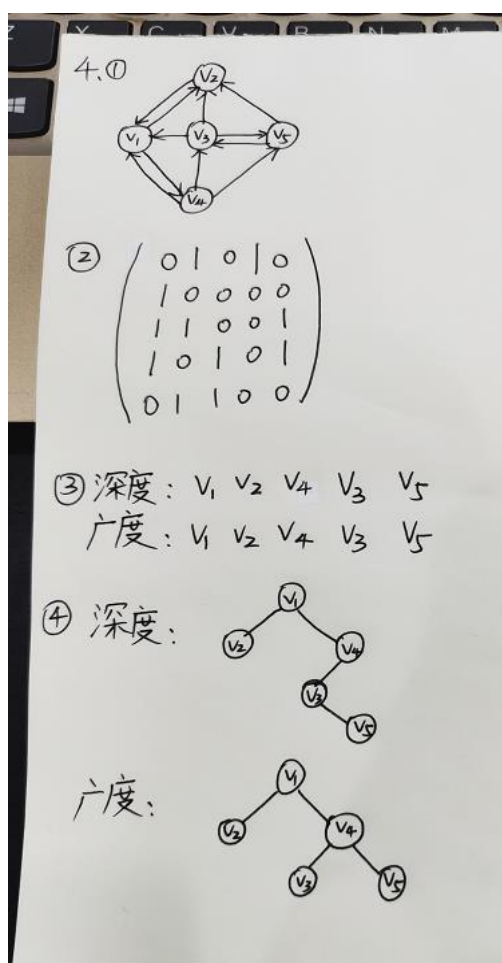


图2 有向图的逆邻接链表

解：



5、一个带权连通图的最小生成树是否唯一?在什么情况下可能不唯一?

解:不一定唯一。当权值都相等的时候,就相当于无权的图,所以此时最小生成树不唯一。

6、对于图 1 所示的带权无向图。

① 按照 Prime 算法给出从顶点 2 开始构造最小生成树的过程。

② 按照 Kruskal 算法给出最小生成树的过程。

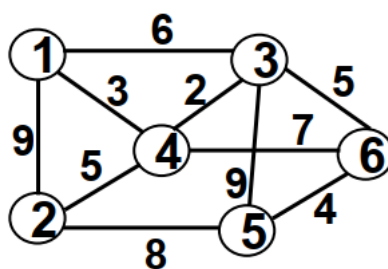
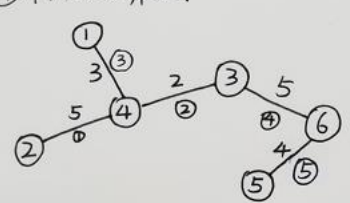


图1 带权无向图

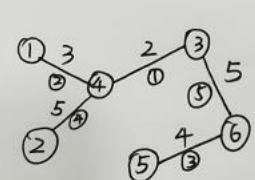
解:

6、① Prime 算法



各边上所标序号即为构造顺序

② Kruskal 算法



各边上所标序号即为构造顺序

7、已知带权有向图如图 3 所示，请利用 Dijkstra 算法从顶点 V_4 出发到其余顶点的最短路径及长度，给出相应的求解步骤。

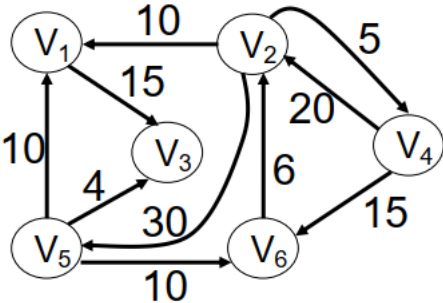


图3 带权有向图

7.	V_4	V_1	V_2	V_3	V_5	V_6	\therefore 到 $V_1: V_4 \rightarrow V_2 \rightarrow V_1, 30$
0	∞/V_4	$20/V_4$	∞/V_4	∞/V_4	$15/V_4$		到 $V_2: V_4 \rightarrow V_2, 20$
1	∞/V_4	$20/V_4$	∞/V_4	∞/V_4	$15/V_4$		到 $V_3: V_4 \rightarrow V_2 \rightarrow V_1 \rightarrow V_3, 45$
2	$30/V_2$	$20/V_4$	∞/V_4	$50/V_2$			到 $V_5: V_4 \rightarrow V_2 \rightarrow V_5, 50$
3	$30/V_2$		$45/V_1$	$50/V_2$			到 $V_6: V_4 \rightarrow V_6, 15$
4			$45/V_1$	$50/V_2$			
5				$150/V_2$			

解：

8、已知带权有向图如图 4 所示，请利用 Floyd 算法求出每对顶点之间的最短路径及路径长度。

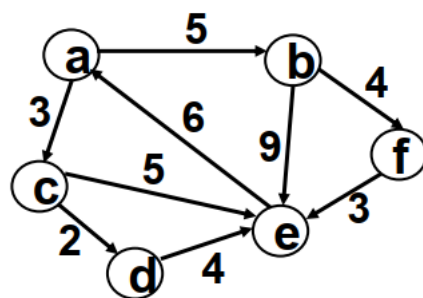


图4 带权有向图

解：

8. 经计算,

$$A_6 = \begin{pmatrix} 0 & 5 & 3 & 5 & 8 & 9 \\ 13 & 0 & 16 & 18 & 7 & 4 \\ 11 & 16 & 0 & 2 & 5 & 20 \\ 10 & 15 & 13 & 0 & 4 & 19 \\ 6 & 11 & 9 & 11 & 0 & 15 \\ 9 & 14 & 12 & 14 & 13 & 0 \end{pmatrix}$$

$$Path = \begin{pmatrix} 1 & 1 & 1 & 3 & 3 & 2 \\ 6 & 2 & 6 & 6 & 6 & 2 \\ 5 & 5 & 3 & 3 & 3 & 5 \\ 5 & 5 & 5 & 4 & 4 & 5 \\ 5 & 1 & 1 & 3 & 5 & 2 \\ 5 & 5 & 5 & 5 & 6 & 6 \end{pmatrix}$$

9、一个 AOV 网用邻接矩阵表示，如图 5。用拓扑排序求该 AOV 网的一个拓扑序列，给出相应的步骤。

$$\begin{matrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

图5 一个AOV网的邻接矩阵

解：

9. 经计算,

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad V = \{V_0, V_1, V_2, V_3, V_4, V_5, V_6\}$$

10、拓扑排序的结果不是唯一的，请给出如图 6 所示的有向图的所有可能的拓扑序列。

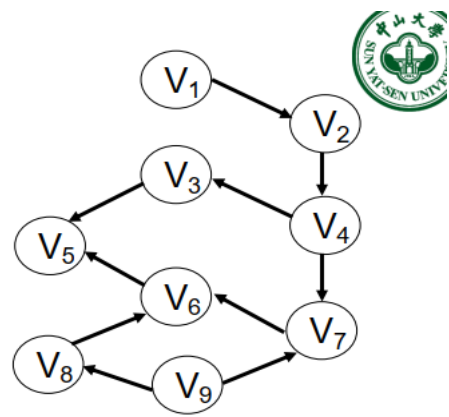


图6 有向图

解：

序列前 4 个点	序列后 5 个点	序列数
(1) 9 8 1 2 (2) 9 1 8 2 (3) 9 1 2 8 (4) 1 2 9 8 (5) 1 9 2 8 (6) 1 9 8 2	(1) 4 7 6 3 5 (2) 4 3 7 6 5 (3) 4 7 3 6 5	18
(1) 9 1 2 4 (2) 1 9 2 4 (3) 1 2 9 4 (4) 1 2 4 9	(1) 3 7 8 6 5 (2) 3 8 7 6 5 (3) 8 7 6 3 5 (4) 8 7 3 6 5 (5) 7 8 6 3 5 (6) 7 8 3 6 5 (7) 7 3 8 6 5	28
(1) 1 2 4 3	(1) 9 7 8 6 5 (2) 9 8 7 6 5	2

11、请在深度优先搜索算法的基础上设计一个对有向无环图进行拓扑排序的算法。

解：通过深度优先搜索访问顶点，并把访问完的顶点添加至链表开头。这里要注意，由于深度优先搜索是逆向确定各顶点的拓扑排序，因此顶点是添加至链表开头的。

伪码为：

```
topologicalSort()
    将所有结点的 color[u] 设置为 WHITE

    for s 从 0 至 |V| - 1
        if color[s] == WHITE
            dfs(s)

dfs(u)
    color[u] = GRAY
    for 与 u 相邻的结点 v
        if color[v] == WHITE
            dfs(v)

    out.push_front(u) //将访问结束的顶点逆向添加至链表
```

具体实现代码为：

```
#include<iostream>
#include<vector>
#include<algorithm>
#include<list>
using namespace std;
static const int MAX = 100000;

vector<int> G[MAX];
list<int> out;
bool V[MAX];
int N;

void dfs(int u){
    V[u] = true;
    for(int i = 0; i < G[u].size(); i++){
        int v = G[u][i];
        if(!V[v]) dfs(v);
    }
    out.push_front(u);
}
```

```

int main(){
    int s, t, M;

    cin>>N>>M;

    for(int i = 0; i < N; i++) V[i] = false;

    for(int i = 0; i < M; i++){
        cin>>s>>t;
        G[s].push_back(t);
    }

    for(int i = 0; i < N; i++){
        if(!V[i]) dfs(i);
    }

    for(list<int>::iterator it = out.begin(); it != out.end(); it++)
        cout<<*it<<endl;

    return 0;
}

```

12、设计一个算法利用图的遍历方法输出一个无向图 G 中从顶点 V_i 到 V_j 的长度为 S 的简单路径，设图采用邻接链表作为存储结构。

解：

```

void FindPath (AGraph *G,int u,int v,int path[],int d){
    int w;//w 是每一次遍历中，当前结点的下一个邻接顶点的代表变量
    ArcNode*p;
    d++;//路径长度增加 1
    path[d]=u;//将当期顶点添加到路径中
    visited[u]=1;//设置已访问结点
    if(u==v)//找到一条路径则输出
        print(path[]); //输出路径上的结点
    p=G->adjlist[u].firstarc;//p 指向 u 的第一个相邻点
    while(p!=NULL){ //遍历 u 的所有相邻点
        w=p->adjvex;//w 为下一个邻接顶点
        if(visited[w]==0)//若顶点 w 未访问，递归访问它
            FindPath(G,w,V,path,d);
        p=p->nextarc;//p 指向 u 的下一个相邻点
    }
}

```

```
visited[u]=0;//恢复环境，使该顶点可重新使用
}
```

13、假设一个工程的进度计划用 AOE 网表示，如图 7 所示。

- ① 求出每个事件的最早发生时间和最晚发生时间。
- ② 该工程完工至少需要多少时间？
- ③ 求出所有关键路径和关键活动。

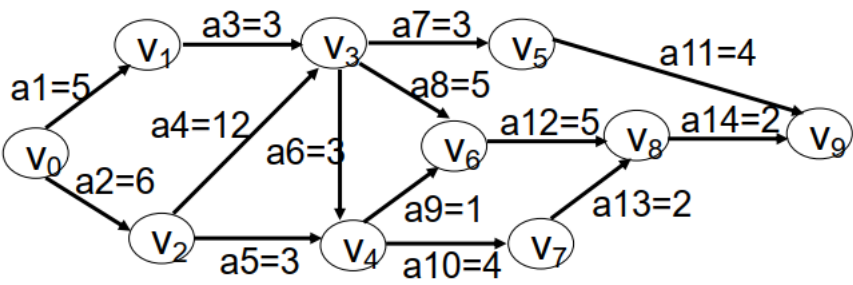


图7 一个AOE网

解：

①

顶点	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9
最早发生时间	0	5	6	18	21	21	23	25	28	30
最晚发生时间	0	15	6	18	21	26	23	26	28	30

活动	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14
最早发 生时间	0	0	5	6	6	18	18	18	21	21	21	23	25	28
最晚发 生时间	10	0	15	6	19	19	23	18	22	22	26	23	26	28

②至少需要 30

③关键路径：V0->V2->V3->V6->V8->V9

关键活动：a2, a4, a8, a12, a14

PS：14，15 题与 7，8 题内容完全一致，不再写出。