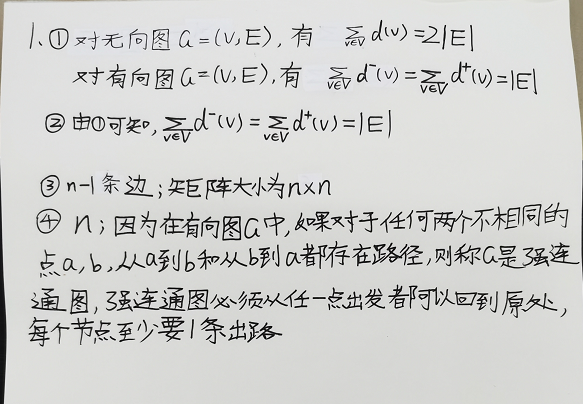
1、分析并回答下列问题：  
① 图中顶点的度之和与边数之和的关系?② 有向图中顶点的入度之和与出度之和的关系?③ 具有n个顶点的无向图，至少应有多少条边才能确保是  
一个连通图? 若采用邻接矩阵表示，则该矩阵的大小是多  
少？④ 具有n个顶点的有向图，至少应有多少条弧才能确保是  
强连通图的? 为什么?

解：

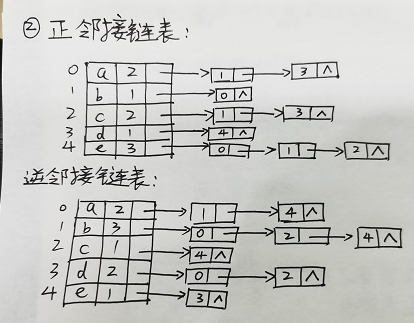
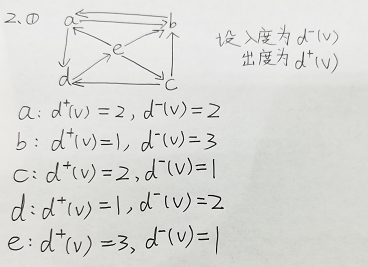


2、 设一有向图G=(V,E)，其中V={a,b,c,d,e}，E={<a,b>,<a,d>,<b,a>,<c,b>, <c,d>,<d,e>,<e,a>,<e,b>,<e,c>}

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：

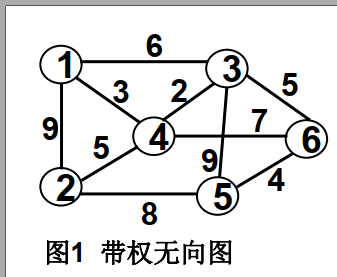


3、对图1所示的带权无向图。

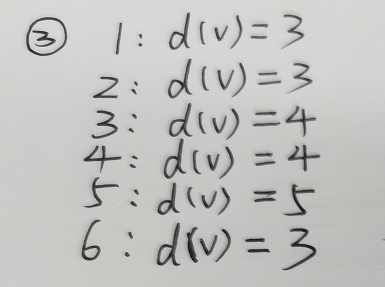
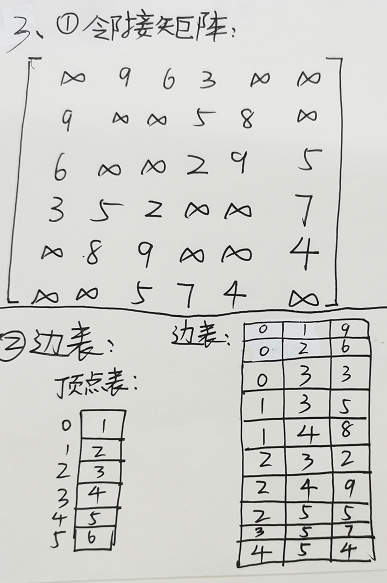
① 写出相应的邻接矩阵表示。

② 写出相应的边表表示。

③ 求出各顶点的度。



解：



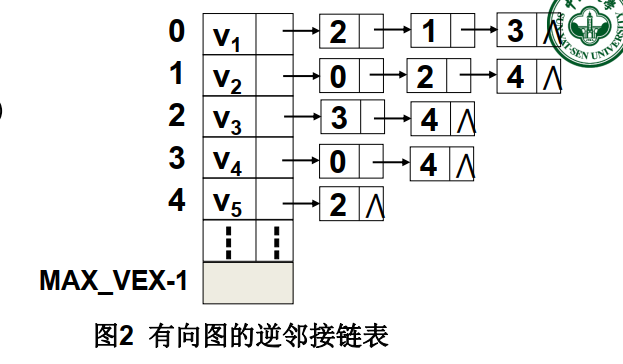
4、 已知有向图的逆邻接链表如图2所示。

① 画出该有向图。

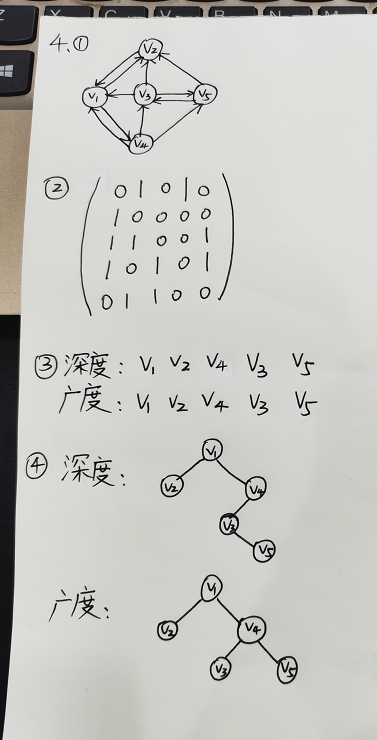
② 写出相应的邻接矩阵表示。

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。



解：



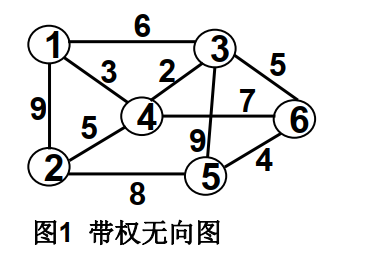
5、 一个带权连通图的最小生成树是否唯一?在什么情况下可能不唯一?

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

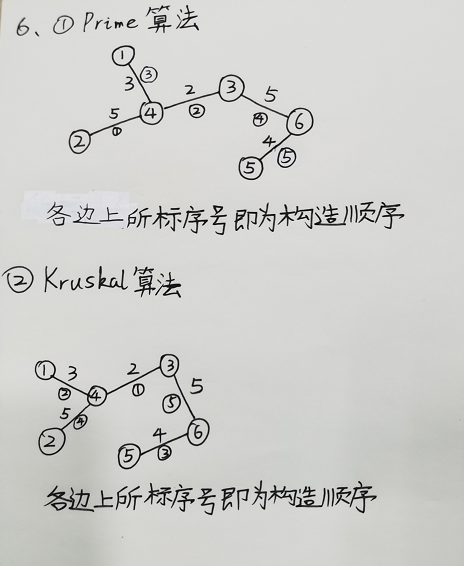
6、对于图1所示的带权无向图。

① 按照Prime算法给出从顶点2开始构造最小生成树的过程。

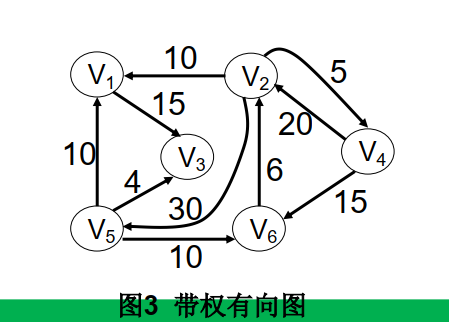
② 按照Kruskal算法给出最小生成树的过程。

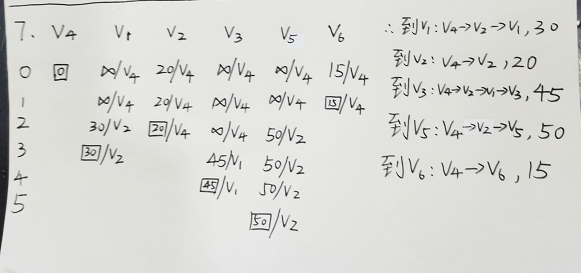


解：

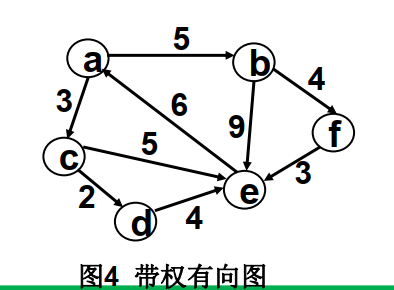


7、已知带权有向图如图3所示，请利用Dijkstra算法从顶点V4出发到其余顶点的最短路径及长度，给出相应的求解步骤。

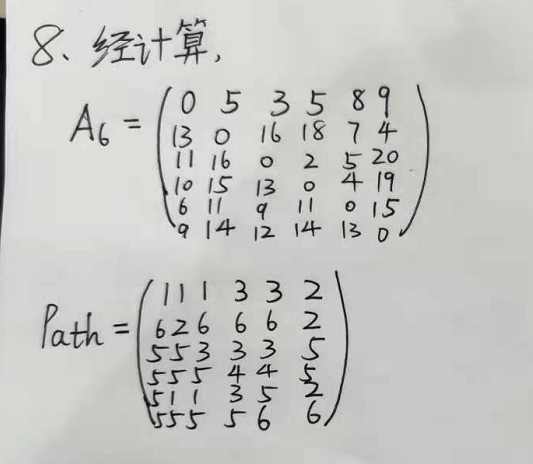


解： 

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。



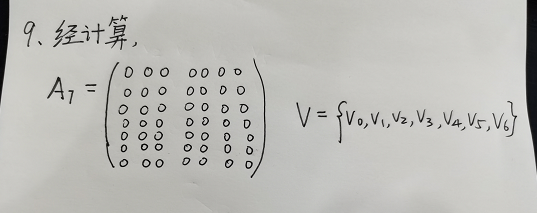
解:



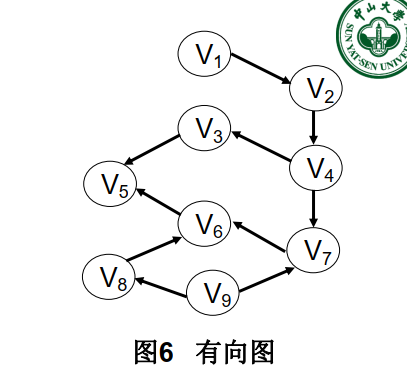
9、一个AOV网用邻接矩阵表示，如图5。用拓扑排序求该AOV网的一个拓扑序列，给出相应的步骤。



解：



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。



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[u] == 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Vi到Vj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

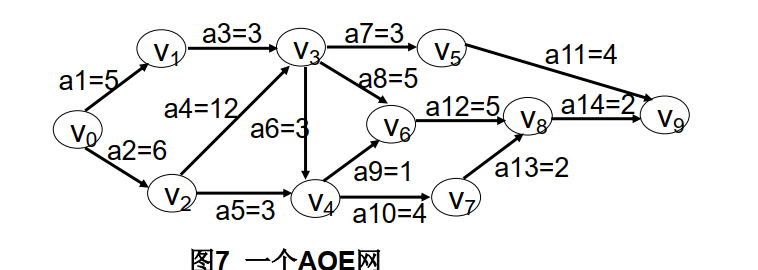
  }

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。



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。