



# C++ 算法基础B



图论——拓扑排序

# 目录

 02 拓扑排序

 03 课堂例题

 01 AOV网

 04 课后习题





编程兔



# 01 AOV网

# 知识点讲解——AOV网

## 1. DAG有向无环图与AOV网

对于一个有向图，若图中不存在回路（环），则称该图是一个DAG图。

若以DAG图中的顶点表示活动，以边表示活动的先后次序，则称其是一个AOV网。

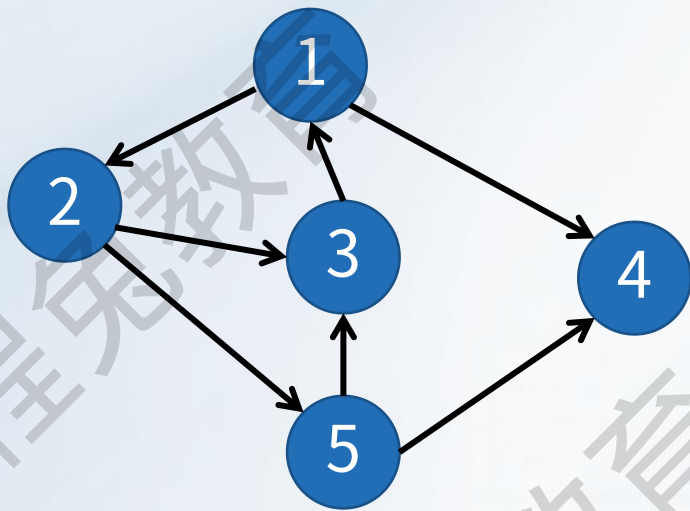


图1

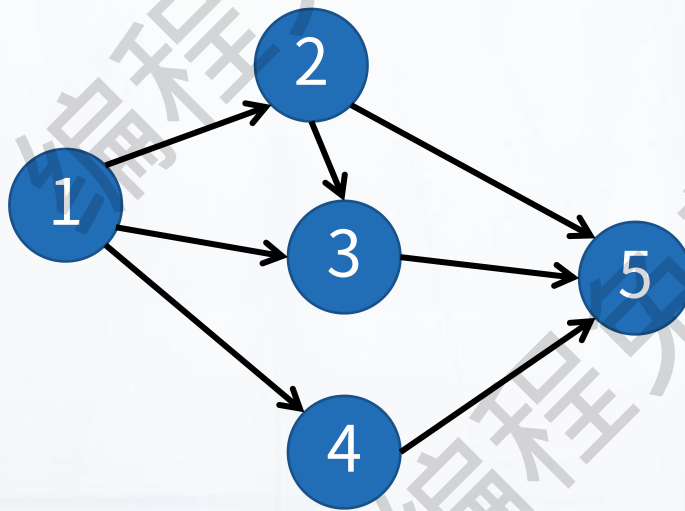
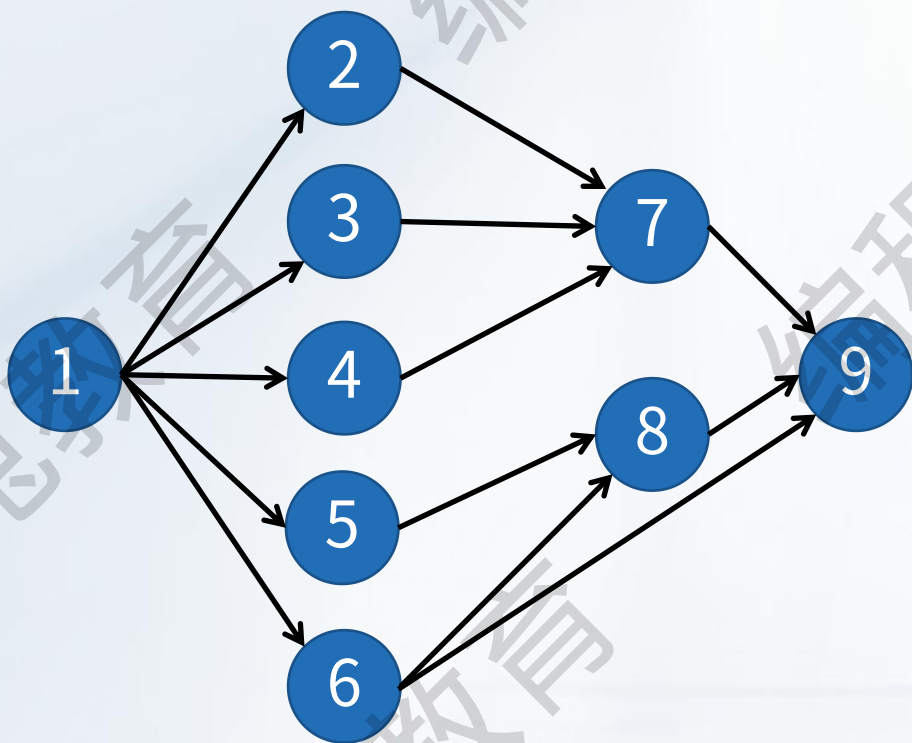


图2

图1中存在环  
不是DAG。  
图2是DAG。

# 知识点讲解——图的基本概念

在AOV网中，有向边代表子工程（活动）的**先后关系**，  
只有当一个活动全部的前驱活动全部都完成之后，这个活动才能进行。  
例如在下图中，只有当工程1完成之后，工程2、3、4、5、6才能开始进行。  
只有当2、3、4全部完成之后，7才能开始进行。以此类推



AOV网

一个AOV网必定是一个有向无环图，  
即不应该带有回路。  
否则，会出现先后关系的自相矛盾。





编程兔



## 02 拓扑排序



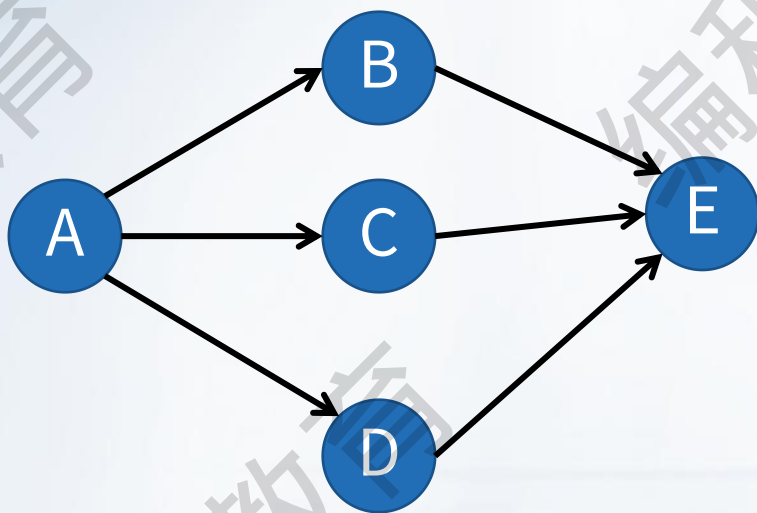
# 知识点讲解——拓扑排序



拓扑排序算法，只适用于AOV网。

把AOV网中的所有活动排成一个序列，使得每个活动的所有前驱活动都排在该活动的前面，这个过程称为“拓扑排序”，所得到的活动序列称为“拓扑序列”。

一个AOV网的拓扑序列是不唯一的，例如下面的这张图，它的拓扑序列可以是：ABCDE，也可以是ACBDE，或是ADBCE。在下图所示的AOV网中，工程B和工程C显然可以同时进行，先后无所谓；但工程E却要等工程B、C、D都完成以后才能进行。



构造拓扑序列可以帮助我们合理安排一个工程的进度。  
由AOV网构造拓扑序列具有很高的实际应用价值。

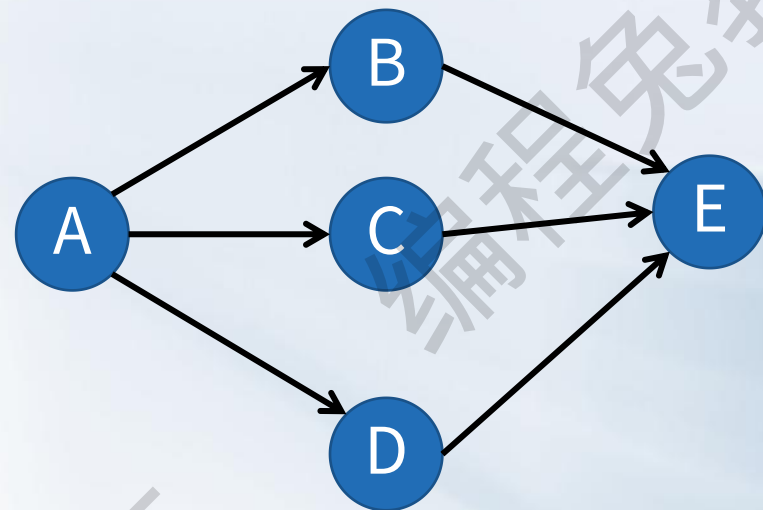
# 知识点讲解——拓扑排序

算法思想：

1. 从图中找选择一个没有前驱，即入度 $ID(v)=0$ 的顶点；
2. 删除该顶点且删除由该顶点的所有起始边（出度的边），并将该顶点输出；
3. 重复(1)和(2)步骤，直到不存在入度为0的顶点为止；
4. 最后得到一个序列，即拓扑排序。

此外：

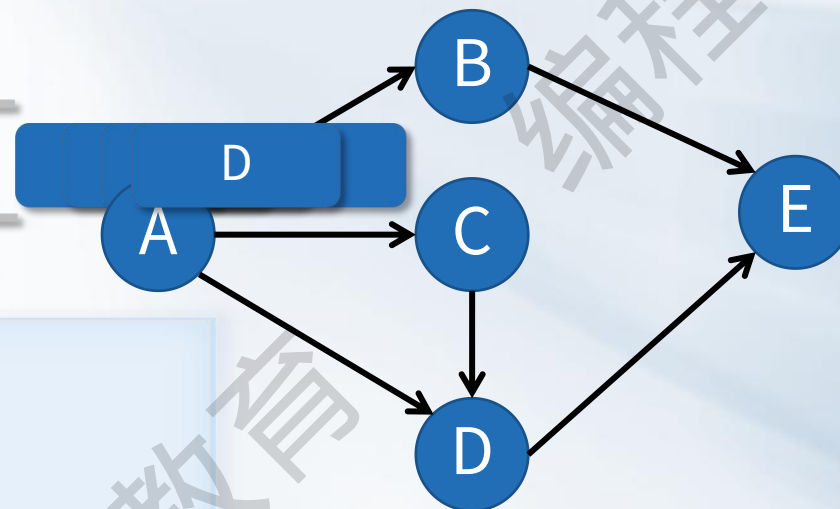
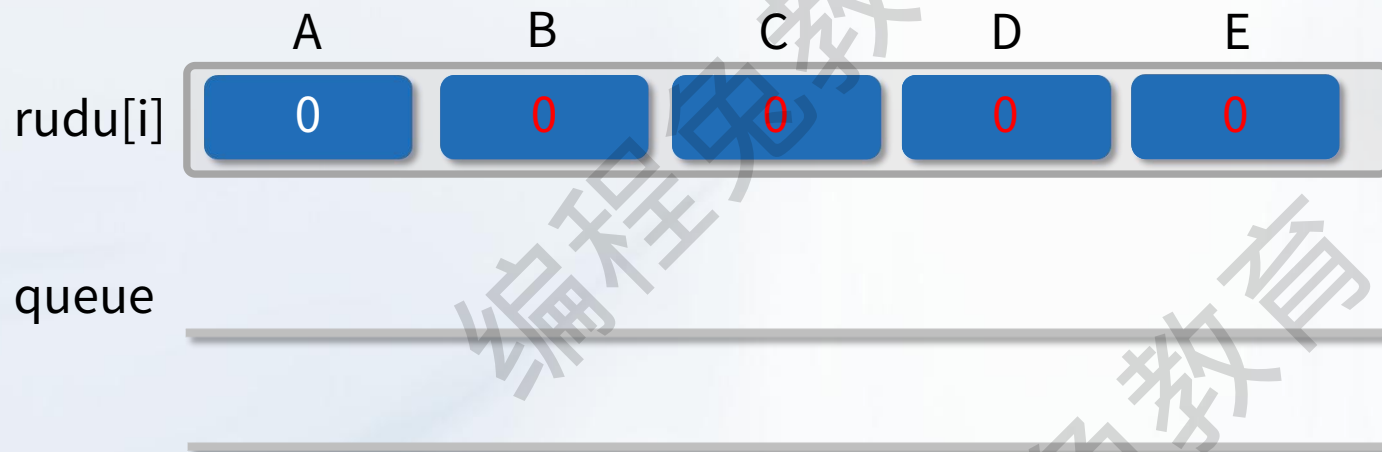
若输出的顶点数小于AOV网中的顶点数，说明有环的存在。



拓扑排序可以用来判断一个有向图是否有环。



# 知识点讲解——拓扑排序流程



算法实现:

1. 先建图，将入度为0的节点入队
2. 循环从队列中“删除”入度为0的节点

删除：使对应节点所有**出边的终点**的入度-1

检查入度减一的这个节点是否入度也变成0

变成0则入队

对应节点出队。



## 03 课堂例题

## 例题1——B3644 【模板】拓扑排序 / 家谱树

有个人的家族很大，辈分关系很混乱，请你帮整理一下这种关系。给出每个人的后代的信息。输出一个序列，使得每个人的后辈都比那个人后列出。



输入

第1行一个整数 $N$  ( $1 \leq N \leq 100$ )，表示家族的人数。接下来 $N$ 行，第 $i$ 行描述第 $i$ 个人的后代编号 $a_{i,j}$ ，表示 $a_{i,j}$ 是 $i$ 的后代。每行最后是0表示描述完毕。



输出

输出一个序列，使得每个人的后辈都比那个人后列出。如果有多重不同的序列，输出任意一种即可。



样例输入：

```
5
0
4 5 1 0
1 0
5 3 0
3 0
```



样例输出：

```
2 4 5 3 1
```



## 例题1——B3644 解析



- 1.根据题目描述建图，并统计每个点的入度。
- 2.将入度为0的节点入队。
- 3.循环取出队头，并输出，并遍历队头的所有出边，将这些边的终点的读书减一，若被减为0，入队。
- 4.直到队列为空，停止循环。



# 例题1——B3644 代码示例

## 拓扑排序模板

```
#include<iostream>
#include<vector>
#include<queue>
using namespace std;
vector <int> G[110];
queue <int> q;
int rudu[110];
int n,t,x;
int main(){
    cin >> n;
    for(int i = 1;i<=n;i++){
        while(cin >> t && t!=0){
            rudu[t]++; //点t入度+1
            G[i].push_back(t);
        }
    }
    //入度为0的节点入队
    for(int i = 1;i<=n;i++){
        if(rudu[i]==0) q.push(i);
    }
```

```
while(!q.empty()){
    x = q.front(); //获取队头
    q.pop();
    cout << x << ' ';
    //遍历队头所有的出边
    for(int i = 0;i<G[x].size();i++){
        //第i条边的终点G[x][i]对应入度-1
        rudu[G[x][i]]--;
        //若该点入度为0则入队
        if(rudu[G[x][i]]==0){
            q.push(G[x][i]);
        }
    }
}
return 0;
```





## 例题2——P4017 最大食物链计数

### 题目描述

给你一个食物网，你要求出这个食物网中最大食物链的数量。

(这里的“最大食物链”，指的是生物学意义上的食物链，即最左端是不会捕食其他生物的生产者，最右端是不会被其他生物捕食的消费者。)

Delia 非常急，所以你只有 1 秒的时间。

由于这个结果可能过大，你只需要输出总数模上 80112002 的结果。

### 输入格式

第一行，两个正整数  $n$ 、 $m$ ，表示生物种类  $n$  和吃与被吃的关系数  $m$ 。

接下来  $m$  行，每行两个正整数，表示被吃的生物A和吃A的生物B。

### 输出格式

一行一个整数，为最大食物链数量模上 80112002 的结果。

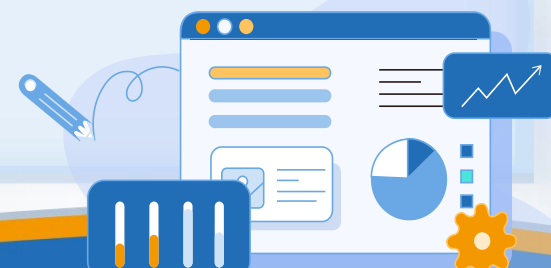


## 例题2——P4017 解析



- 1.根据题目描述可知：食物网中捕食关系一定是单向的（猫吃鱼），并且捕食关系一定无环的。所以可以建立DAG进行拓扑排序。
- 2.本题中一旦a吃了b,也代表a继承了b的所有食物链数量，我们可以设置 $f[i]$ 代表第i个点的食物链数量。
- 3.初始化生产者 $f[i]=1$ ，从生产者(入度为0)开始进行拓扑排序，本题目问最大食物链数量，其实是计算总食物链数量，所以一旦一条食物链到头（到顶端捕食者(出度为0)），我们需要将此时该点的食物链计数加到总数Ans中。

【注】要不断模 80112002



## 例题2——P4017 代码示例

```
#include<iostream>
#include<vector>
#include<stack>
using namespace std;
vector<int> G[5010];
stack<int> st;
int rudu[5010];
int chudu[5010];
int f[5010]; // f[i] 代表第i个点的食物链数量
int main(){
    int n,m,x,y;
    cin >> n >> m;
    // 建图, 统计入度和出度
    for(int i = 1; i<=m; i++){
        cin >> x >> y;
        rudu[y]++;
        G[x].push_back(y);
        chudu[x]++;
    }
    // 统计所有生产者
    for(int i = 1; i<=n; i++){
        if(rudu[i]==0){
            st.push(i);
            f[i] = 1;
        }
    }
    int ans = 0;
```

```
while(!st.empty()){
    //1. 输出栈顶
    int num = st.top();
    st.pop();
    //2. 删边
    for(int i = 0; i<G[num].size(); i++){
        rudu[G[num][i]]--;
        f[G[num][i]] += f[num];
        f[G[num][i]] %= 80112002;
    }
    //3. 看有没有新的入度为0的结点, 入栈
    if(rudu[G[num][i]]==0){
        st.push(G[num][i]);
        // 出度为0代表一条食物链结束, 加入ans
        if(chudu[G[num][i]]==0){
            ans += f[G[num][i]];
            ans %= 80112002;
        }
    }
}
cout << ans;
return 0;
```





# 04 课后习题

## 课后习题

基础习题：

B3644 【模板】拓扑排序 / 家谱树

P1113 杂务

P4017 最大食物链计数

进阶练习：

P1983 [NOIP2013 普及组] 车站分级

P1038 [NOIP2003 提高组] 神经网络

P1347 排序

P3243 [HNOI2015] 菜肴制作

# C++ 编程兔

