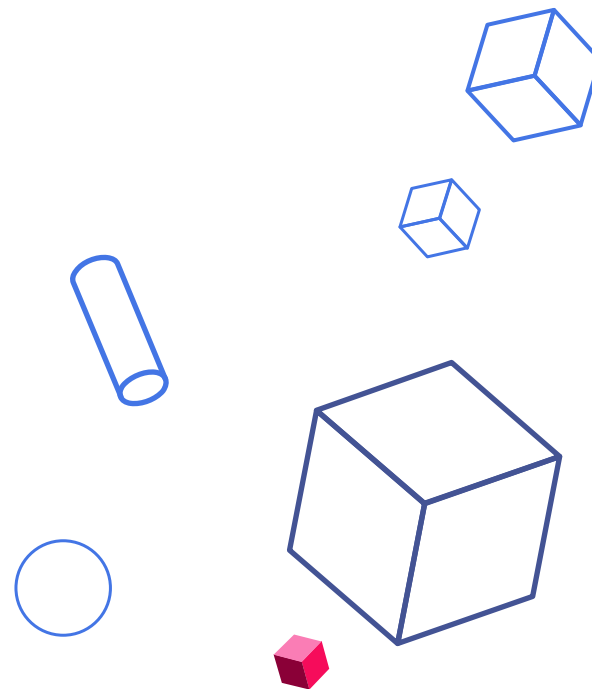


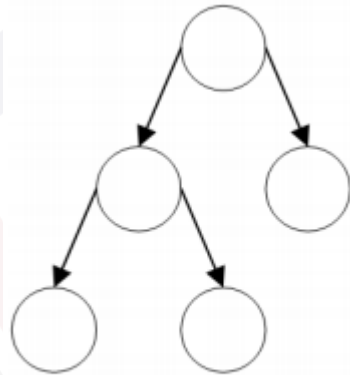
# 图论

## 拓扑排序与关键路径

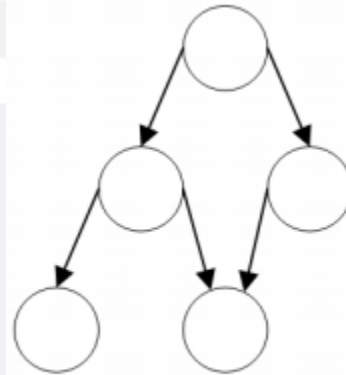
# 拓扑排序



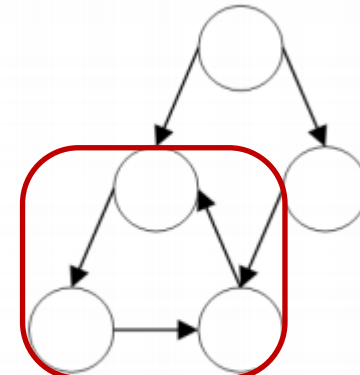
## 有向无环图：无环的有向图，简称DAG图



有向树



DAG图



有向图

有向无环图常用来描述一个工程或者系统的进行过程。

一个工程可以分为若干个子工程，只要完成了这些子工程（活动），就完成了整个工程。



## AOV网：拓扑排序

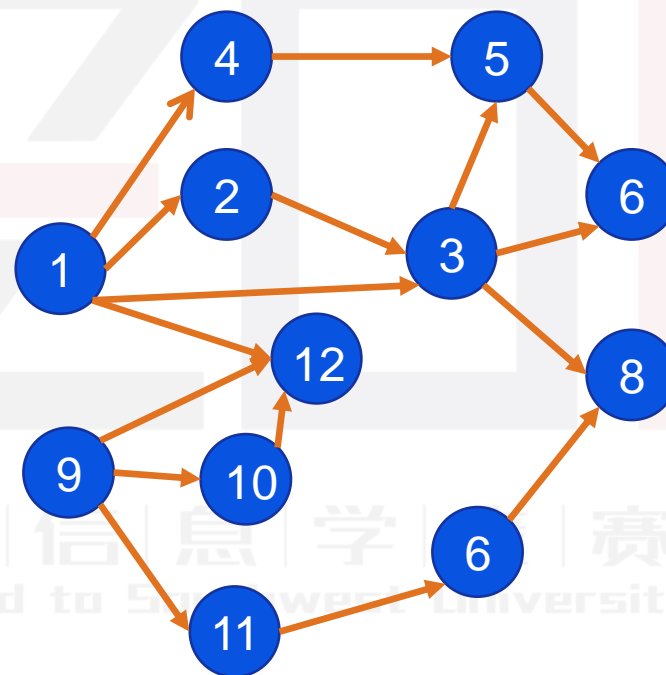
又称为“**顶点表示活动网络**”，用顶点表示活动，边表示活动（顶点）发生的先后关系。

## AOE网：关键路径

又称为“**边表示活动网络**”，用顶点表示事件，用有向边表示活动，顶点表示活动的开始或者结束事件，每个事件表示在它之前的活动已经完成，在它之后的活动可以开始。边上的权值表示活动的持续时间。

代号	知识点	前导知识点
1	程序设计基础	无
2	离散数学	1
3	数据结构	1 2
4	汇编语言	1
5	语言设计与分析	3 4
6	计算机原理	11
7	编译原理	3 5
8	操作系统	3 6
9	高等代数	无
10	线性代数	9
11	普通物理	9
12	数值分析	1 9 10

大学选课需要学习先修课程，可以表示为如下的图：





# AOV网的特点:



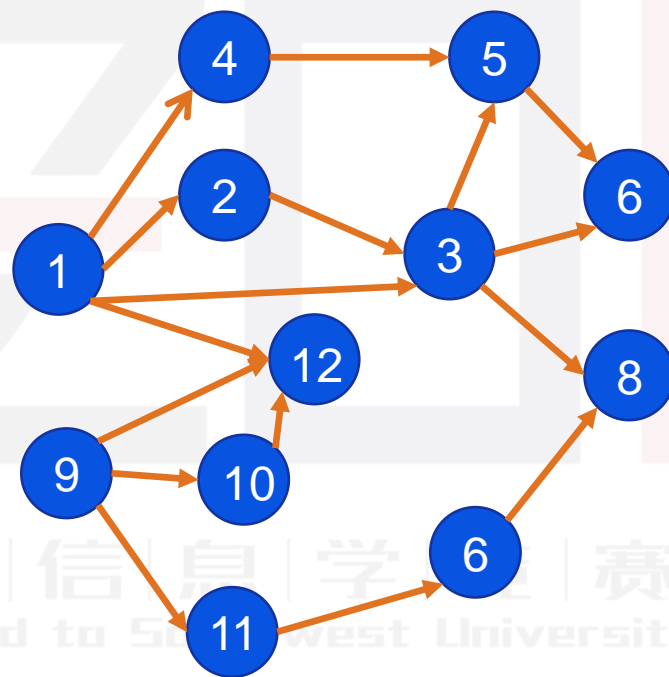
西南大学附属中学  
High School Affiliated to Southwest University

若从 $i$ 到 $j$ 有一条有向路径,  $i$ 就是 $j$ 的前驱,  $j$ 就是 $i$ 的后继

若 $\langle i, j \rangle$ 是网中的有向边,  $i$ 就是 $j$ 的直接前驱,  $j$ 就是 $i$ 的直接后继

AOV网中不允许出现回路

怎么判读网中是否有回路?





对于一个有向无环图(Directed Acyclic Graph简称DAG)  $G$ , 将 $G$ 中所有顶点排成一个线性序列, 同时这个序列满足下面两个条件:

- (1) 每个顶点出现且只出现一次。
- (2) 若存在一条从顶点  $A$  到顶点  $B$  的路径, 那么在序列中顶点  $A$  出现在顶点  $B$  的前面。

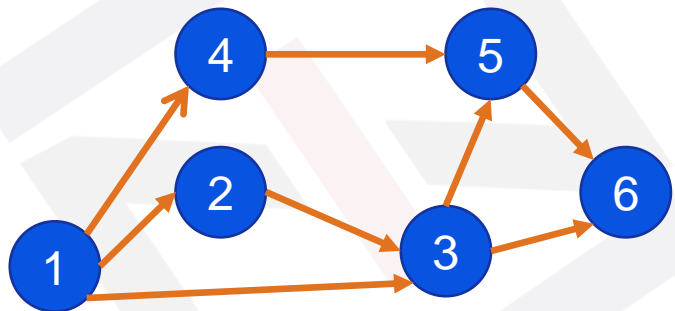
## 拓扑排序



# 求拓扑排序的方法:



西南大学附属中学  
High School Affiliated to Southwest University



在有向图中，选一个没有前驱的顶点输出

从图中删除该顶点和以它为起点的所有边。

重复上两步，直到全部顶点都输出，  
或者图中不存在无前驱的顶点 **(说明图中存在环)**

## 拓扑序列

1 2 3 4 5 6

1 4 2 3 5 6

一个AOV网络的拓扑序列不是唯一的





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。

给出每个人的孩子的信息。

输出一个序列，使得每个人的后辈都比那个人后列出。

## 拓扑排序

### 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 $i$ 个人的儿子。两个数之间用一个空格隔开

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。



## 参考代码



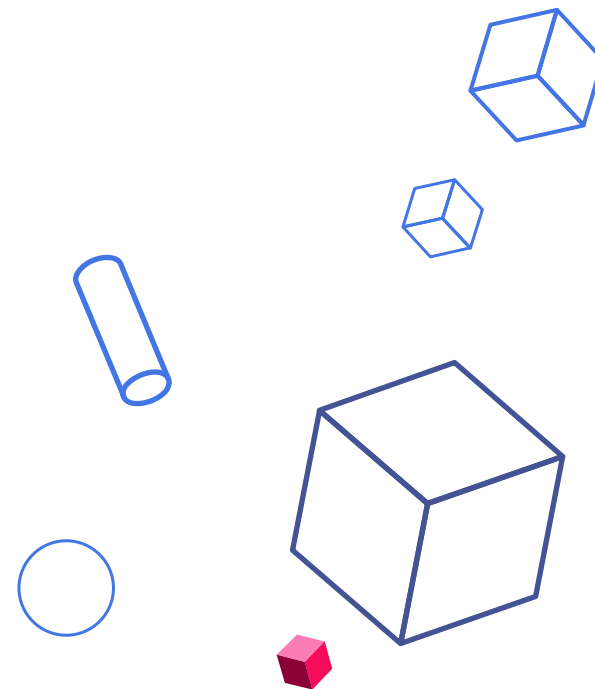
西南大学附属中学  
High School Affiliated to Southwest University

```
int G[N][N]; //储存有序对信息
int indegree[N]; //储存入度数量
int topo[N]; //储存拓扑序列
int main(){
    int n;
    cin>>n;
    for(int i=1;i<=n;i++){
        while(1)
        {
            int j;
            cin>>j;
            if(j==0) break;
            G[i][j]=1; //添加一条i指向j的一条边
            indegree[j]++; //j的入度+1
        }
    }
}
```

```
int top=0;
for(int i=1;i<=n;i++){
    if(indegree[i]==0)
        topo[++top]=i; //入度为0就将对应点入栈
}
while(top>0) //当栈不为空时重复执行
{
    int t=topo[top]; //输出入度为0的元素，并删除
    cout<<t<<" ";
    top--;
    for(int i=1;i<=n;i++){
        if(G[t][i]) //遍历所有与它相邻的点
        {
            G[t][i]=0; //删除边
            indegree[i]--; //入度减1，如果变为了0，就入栈
            if(indegree[i]==0)
                topo[++top]=i;
        }
    }
}
return 0;
}
```

# 关键路径

---



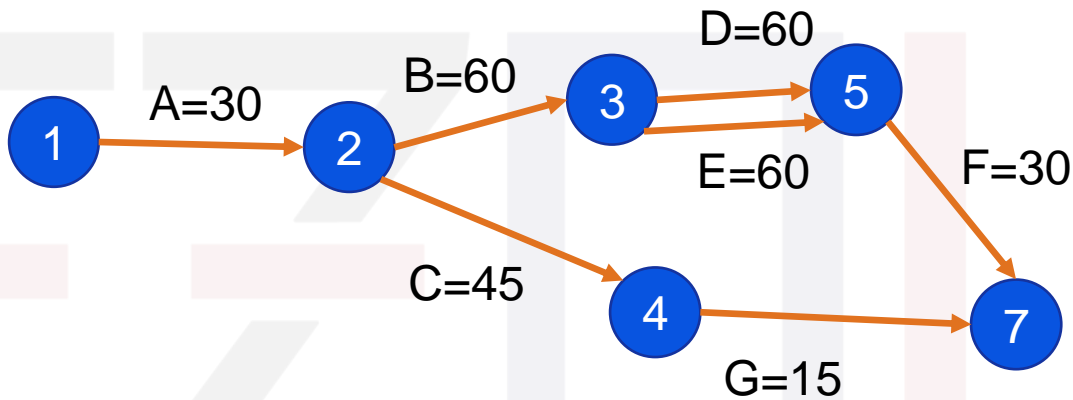


# 关键路径



聚餐晚上6点开始，最迟几点要开始准备？压缩那项活动时间可以让总时间减少？

代号	活动描述	时间	前置任务
A	菜单定制	30	
B	原料采购	60	A
C	餐具准备	45	A
D	甜点准备	60	B
E	原料清洗	60	B
F	烹饪	30	DE
G	桌椅布置	15	C
H	宴会开始	0	FG



事件1 表示整个工程的开始 (源点: 入度为0)

事件7 表示整个工程的结束 (汇点: 出度为0)

**关键路径:** 路径长度最长的路径

**路径长度:** 路径上各活动持续时间之和 (权值之和)

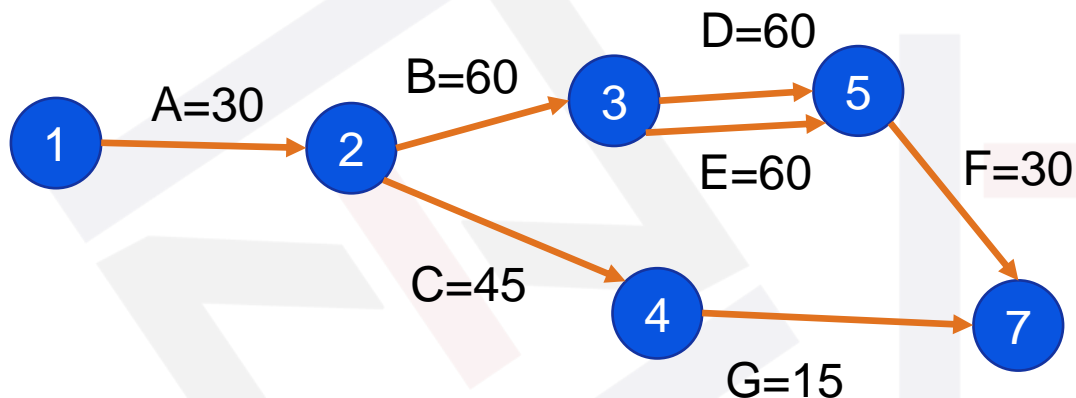


# 关键路径



西南大学附属中学  
High School Affiliated to Southwest University

解决关键路径问题，需要定义4个描述量：



$l(i) - e(i)$  完成活动 $a_i$ 的时间余量  
 $l(C) - e(C) = 90$

$Ve(v_j)$  事件 $v_j$ 最早发生的时间  
 $Ve(1) = 0$      $Ve(2) = 30$

$v_l(v_j)$  事件 $v_j$ 最晚发生的时间  
总时间180时,  $v_l(4) = 165$

$e(i)$  活动 $a_i$ 最早发生的时间  
 $e(C) = 30$

$l(i)$  活动 $a_i$ 最晚发生的时间  
总时间180时,  $l(C) = 120$

**关键活动** 关键路径上的活动,  $l(i) == e(i)$  时间余量为0



# 关键路径



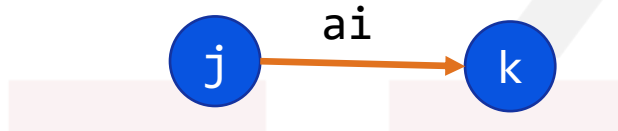
西南大学附属中学  
High School Affiliated to Southwest University

如何找 $l(i)=e(i)$ 的关键活动?

设活动 $a_i$ 用边 $\langle j, k \rangle$ 表示, 其持续时间记为:  $w_{j,k}$

则: (1)  $e(i)=ve(j)$

(2)  $l(i)=vl(k)-w_{j,k}$



如何求 $ve(j)$ 和 $vl(j)$ ?

(1) 从 $ve(1)=0$ 开始向前递推

$$ve[j]=\max(ve[j], ve[i]+w[i][j])$$

(2) 从 $vl(n)=ve(n)$ 开始向后递推

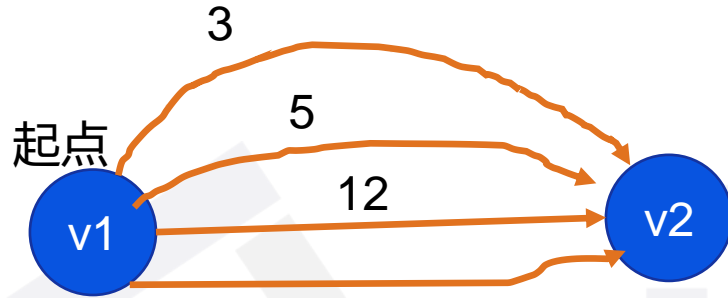
$$vl[i]=\min(ve[i], ve[j]-w[i][j])$$



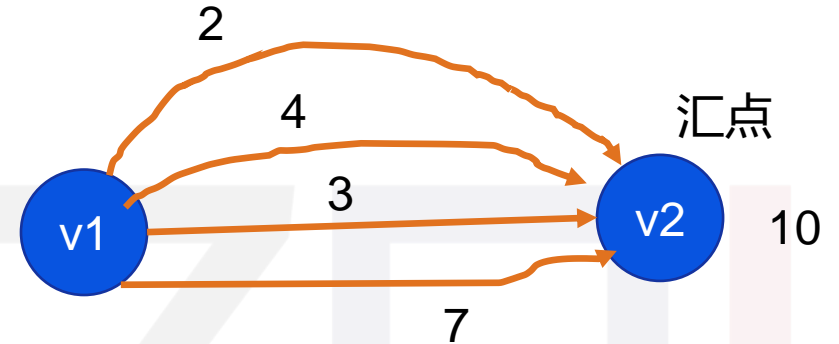
# 计算方法



西南大学附属中学  
High School Affiliated to Southwest University

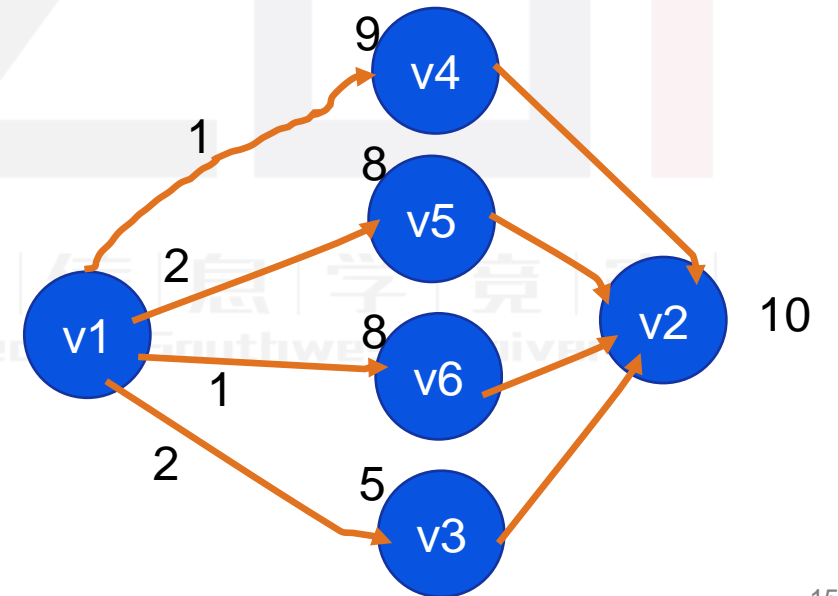
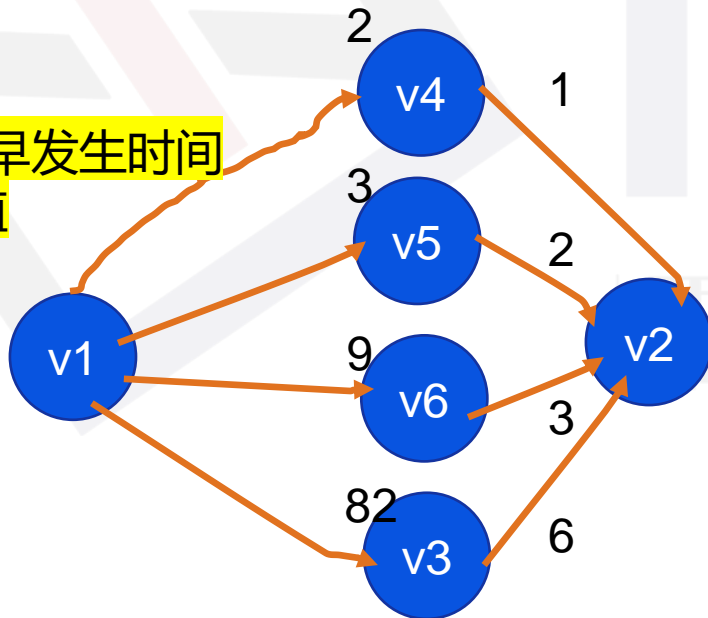


ve[v2]=3、5、12、88中 最大值88



vl[v2]=10-2、10-4、10-3、10-7中 最小值3

Ve[v2]==  
v2起始结点的最早发生时间  
+各自的边的权值  
和的最大值



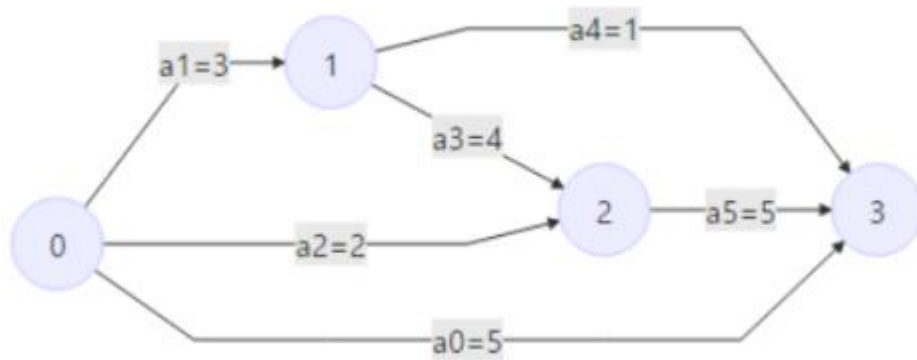


# 求关键路径的步骤



西南大学附属中学  
High School Affiliated to Southwest University

- (1) 求 $ve(i)$ 、 $vl(i)$
- (2) 求 $e(i)$ 、 $l(i)$
- (3) 计算 $l(i) - e(i)$



顶点	$Ve$	$VI$
0	0	0
1	3	3
2	7	7
3	12	12

顶点	$e$	$l$
a0	0	7
a1	0	0
a2	0	5
a3	3	3
a4	3	11
a5	7	7

✓

✓

✓



THANKS

