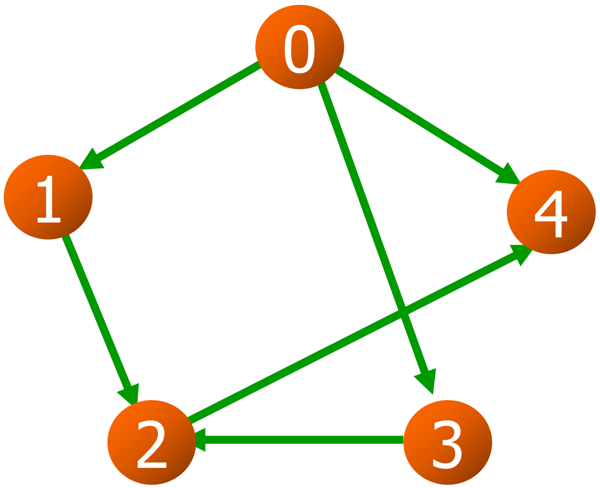
第七章 图

7.1 图的定义和术语
7.2 图的存储结构
7.3 图的遍历
7.4 图的连通
7.5 有向无环图及其应用
7.6 最短路径

7.5 有向无环图及其应用

一. 有向无环图

- 有向无环图(DAG:Directed Acycline Graph)是图中无环的有向图。



7.5 有向无环图及其应用

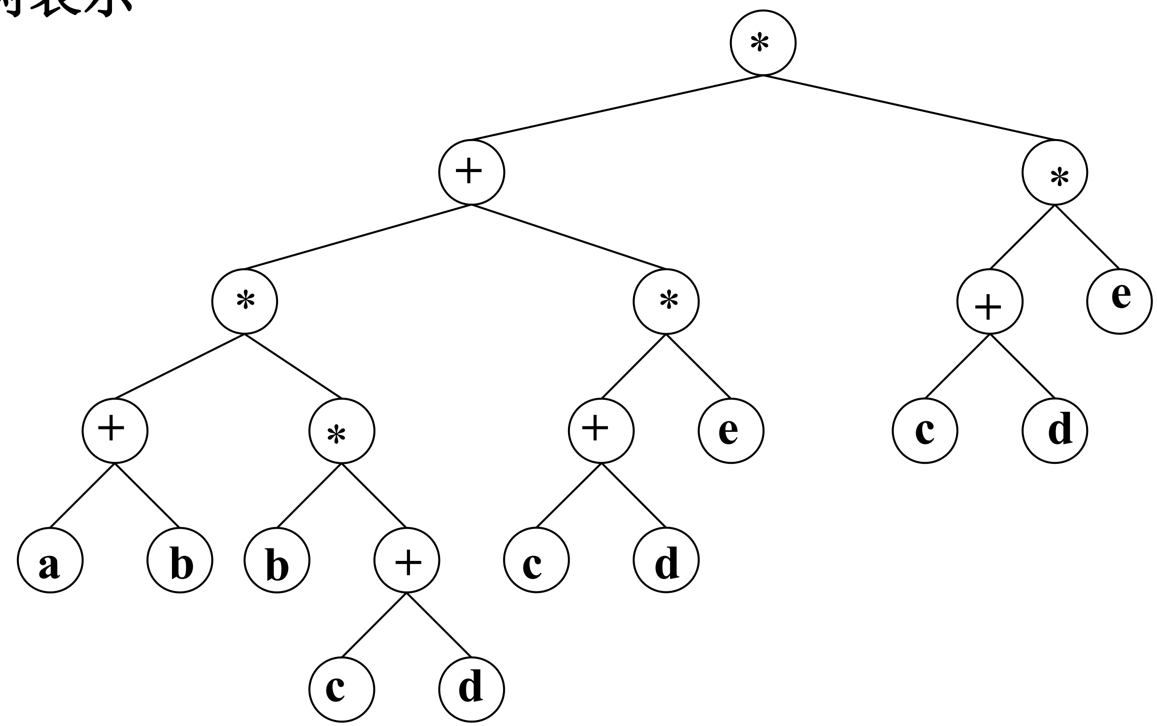
一. 有向无环图

- **问题：** 如何检查有向图中是否有回路呢？
- **解决方法：**
  - **深度优先搜索**
  - **拓扑排序(本节)**

7.5 有向无环图及其应用

一. 有向无环图

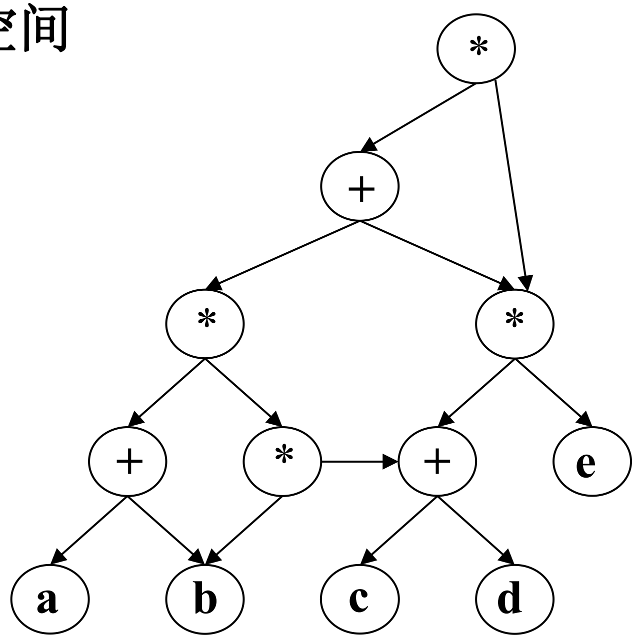
- DAG图用于表达式的共享
  - 表达式((a +b) * (b * (c + d)) + (c + d) * e) * ((c + d) * e)，用二叉树表示



7.5 有向无环图及其应用

一. 有向无环图

- DAG图用于表达式的共享
  - 表达式((a +b) * (b * (c + d)) + (c + d) * e) * ((c + d) * e)，**DAG**图表示.
  - 可以节省存储空间



7.5 有向无环图及其应用

一. 有向无环图

- 有向无环图主要用于研究工程项目的工序问题、工程时间进度问题等。

  一个工程(project)都可分为若干个称为活动(active)的子工程(或工序)，各个子工程受到一定的条件约束：

  - 某个子工程必须开始于另一个子工程完成之后；
  - 整个工程有一个开始点(起点)和一个终点。

- 一个工程活动可以用有向无环图来描述。

7.5 有向无环图及其应用

二. 拓扑排序

- 偏序关系

  - 集合上的关系：集合A上的关系是从A到A的关系(A×A)
  - 关系的自反性：若∀a∈A有(a，a)∈R，称集合A上的关系R是自反的。
  - 关系的对称性：如果对于a，b∈A，只要有(a，b)∈R就有(b，a)∈R，称集合A上的关系R是对称的。
  - 如果对于a，b∈A，仅当a=b时有(a，b)∈R和(b，a)∈R，称集合A上的关系R是反对称的。
  - 关系的传递性：若a，b，c∈A，若(a，b)∈R，并且(b，c)∈R，则(a，c)∈R，称集合A上的关系R是传递的。

7.5 有向无环图及其应用

二. 拓扑排序

- 偏序关系

  - **偏序**：若集合**A**上的关系**R**是自反的，反对称的和传递的，则称**R**是集合**A**上的偏序关系。

    偏序指集合中仅有部分成员之间可比较。

  - **全序**：设**R**是集合**A**上的偏序关系，∀a，**b∈A**，必有**aRb**或**bRa**，则称**R**是集合**A**上的全序关系。

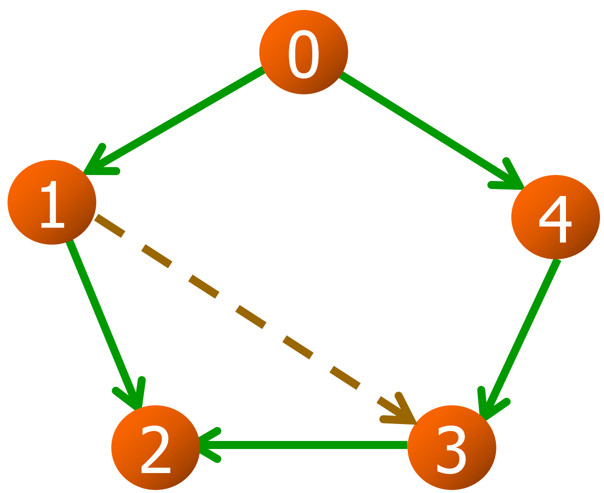
    全序指集合中全体成员之间均可比较。

7.5 有向无环图及其应用

二. 拓扑排序

- 从偏序得到的全序，称为**拓扑有序**。

例如：



7.5 有向无环图及其应用

二. 拓扑排序

- **拓扑排序：**实现拓扑有序的操作。
- **拓扑有序序列**：设G={V，E}是一个具有n个顶点的有向图，V中的顶点序列v1，v2，...，vn，满足：若从顶点vi到vj有一条路径，则在顶点序列中顶点vi必在vj之前，即vi是vj的前驱，vj是vi的后继，这样的顶点序列为一个拓扑有序序列。

7.5 有向无环图及其应用

二. 拓扑排序

- 拓扑排序是对一个有向图构造拓扑有序序列的过程。构造时会有两个结果：
  - 如果此时网的全部顶点都被输出，则说明它不存在环（回路）；
  - 否则，说明这个网存在环（回路）。

7.5 有向无环图及其应用

二. 拓扑排序

- **拓扑排序算法：**

① 在有向图中选一个没有前驱（入度为0 ）的顶点且输出之

② 从图中删除该顶点和所有以它为尾的弧（**该弧的弧头顶点的入度减1**）

③ 重复(1)、(2)两步，直到所有顶点输出为止（或者图中不再有入度为0的顶点为止）

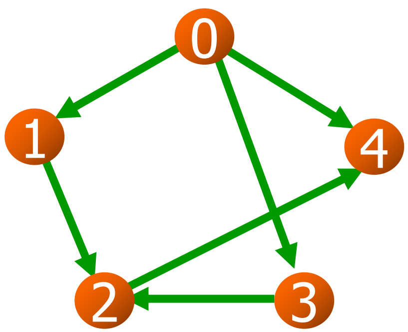
- 如何选择入度为零的顶点呢？ **== 用栈或队列保存入度为0的顶点**

InDegree 记录各顶点的入度

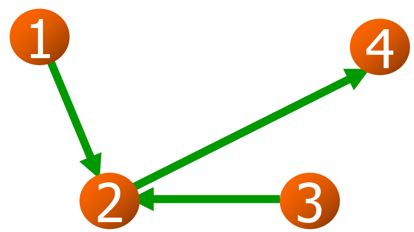
7.5 有向无环图及其应用

二. 拓扑排序

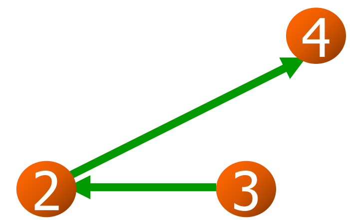
- 拓扑排序举例，最后输出拓扑排序结果：0, 1, 3, 2, 4



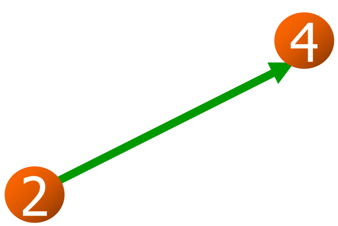
原图



输出0之后



输出0,1之后



输出0,1,3之后

输出0,1,3,2之后

最后的排序结果：

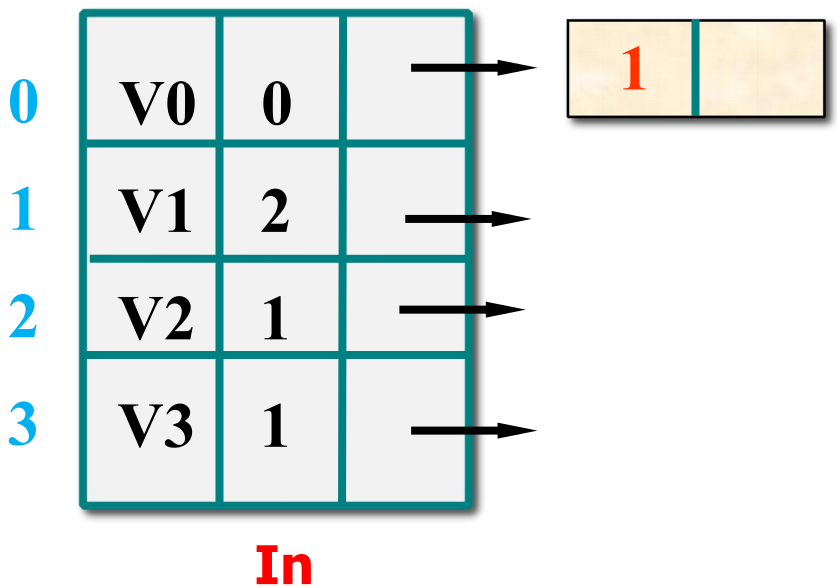
0 1 3 2 4

7.5 有向无环图及其应用

二. 拓扑排序

■ 拓扑排序算法：

由于拓扑排序的过程中，始终要查找入度为0的顶点，因此在原来的顶点表结构中，增加一个入度域in； 还要查找邻接点，显然用邻接表（逆邻接表）更加方便。



7.5 有向无环图及其应用

二. 拓扑排序

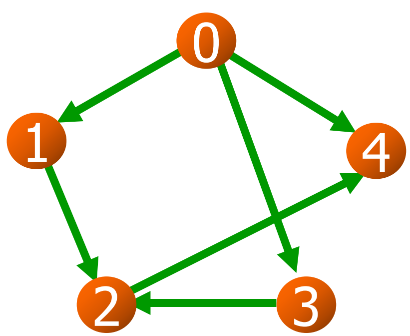
- 拓扑排序程序算法：给出有向图邻接矩阵，求拓扑有序序列

(1) 逐列扫描矩阵，找出入度为0且编号最小的顶点v

(2) 输出v，并标识v已访问（即删除该顶点）

(3) 把矩阵第v行全清0（即删除所有以该顶点为尾的弧）

重复上述步骤，直到所有顶点输出为止或再没有入度为0顶点



$$
\begin{array}{c c}
 & \begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix} \\
\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} &
\begin{bmatrix}
0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

7.5 有向无环图及其应用

二. 拓扑排序

- 拓扑排序的应用
  - 可以检查有向网中是否有环
  - 方法：

    对有向图的顶点进行拓扑排序，若所有顶点都在其拓扑有序序列中，则无环，否则，有环。

7.5 有向无环图及其应用

二. 拓扑排序

- AOV网(Activity On Vertex Network)：对工程的活动加以抽象，图中顶点表示活动，有向边表示活动之间的优先关系，这样的有向图称为顶点表示活动的网。
  - 在AOV网中不能有环，否则，某项活动能否进行是以自身的完成作为前提条件。

7.5 有向无环图及其应用

二. 拓扑排序

- 拓扑排序的应用

  - 可求得一个项目**AOV**网中各个子活动之间的次序关系

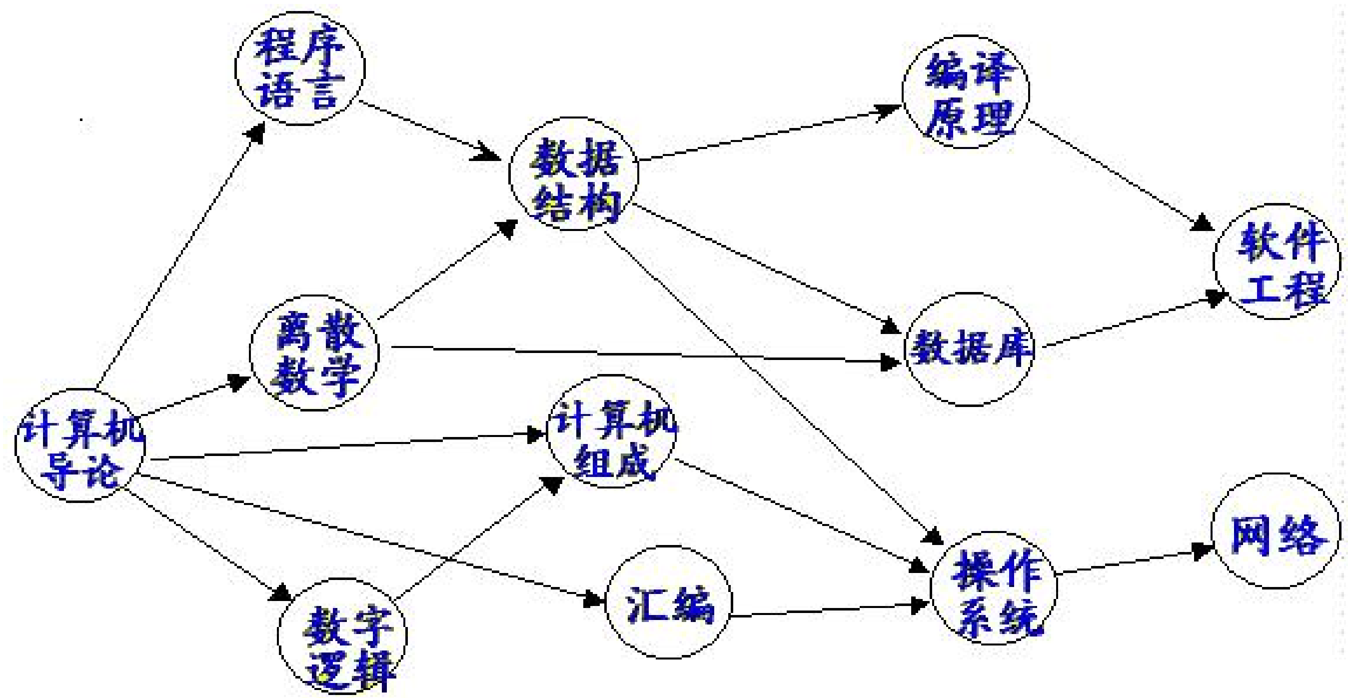
    构造**AOV**网中顶点的一个拓扑有序序列($v_1, v_2, \cdots, v_n$)，使得该序列不仅保持原来有向图中顶点之间的优先关系，而且对原图中没有优先关系的顶点之间也建立一种(人为的)优先关系。

7.5 有向无环图及其应用

二. 拓扑排序

- 拓扑排序的应用

例如，修课顺序：



二. 拓扑排序

- 软件专业的学生必须学习一系列课程，某些课程是其他课程的先修课，即课程之间有先后关系，求学生修课顺序。

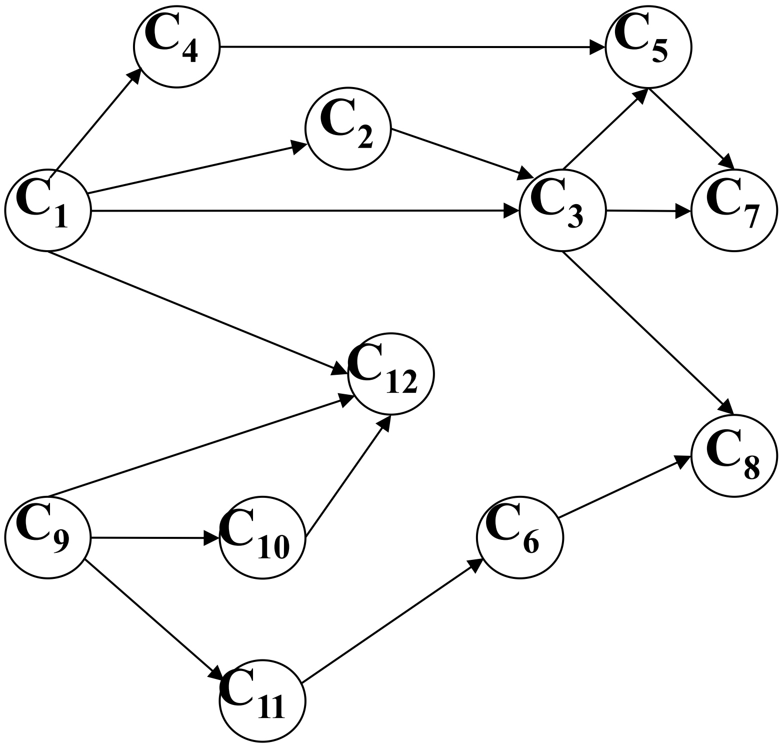
课程编号	课程名称	先决条件
C_1	程序设计基础	无
C_2	离散数学	C_1
C_3	数据结构	C_1，C_2
C_4	汇编语言	C_1
C_5	语言的设计和分析	C_3，C_4
C_6	计算机原理	C_{11}
C_7	编译原理	C_3，C_5
C_8	操作系统	C_3，C_6
C_9	高等数学	无
C_{10}	线性代数	C_9
C_{11}	普通物理	C_9
C_{12}	数值分析	C_9，C_{10}，C_1

7.5 有向无环图及其应用

二. 拓扑排序

- 拓扑排序的应用

根据表格得到课程之间优先关系的DAG图，进行拓扑排序，可得到修课顺序。

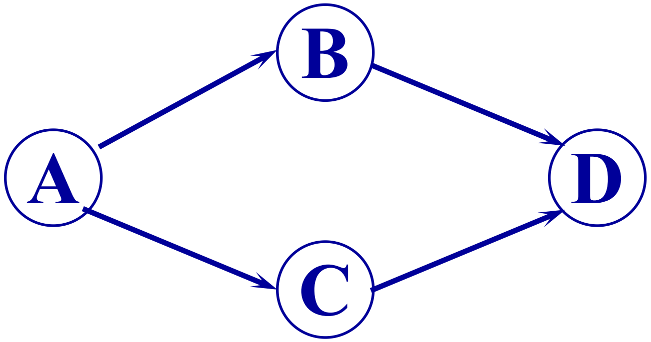


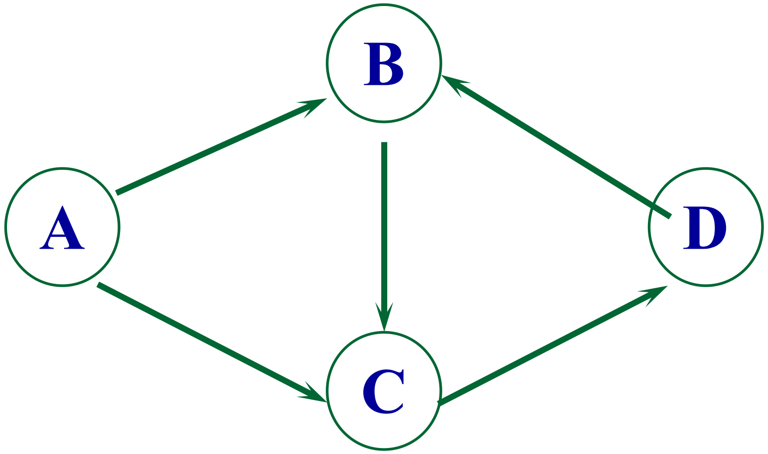
(C1, C2, C3, C4, C5, C7, C9, C10, C11, C6, C12, C8)

或 (C9, C10, C11, C6, C1, C12, C4, C2, C3, C5, C7, C8)

7.5 有向无环图及其应用

练习：写出下图的拓扑排序序列。



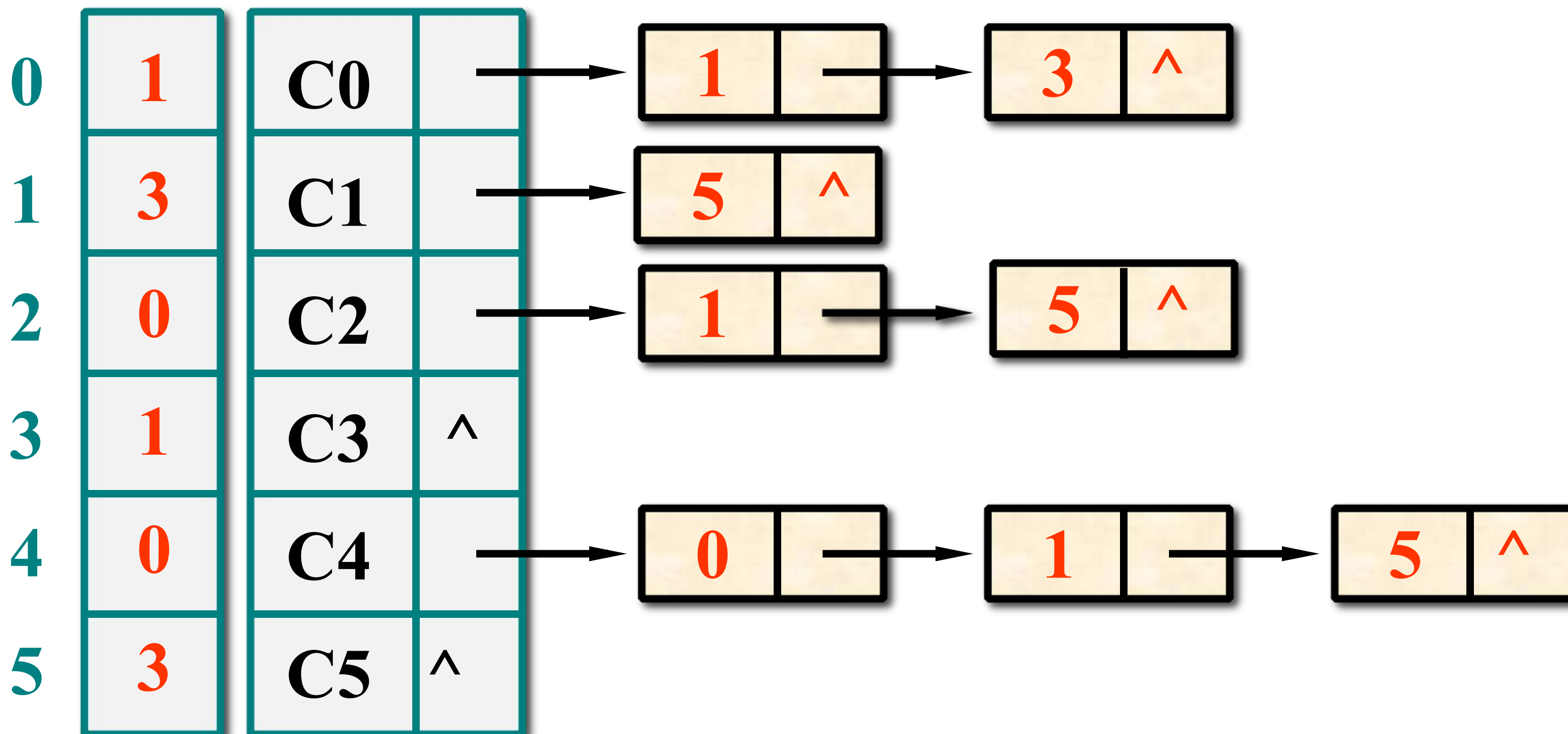


A B C D
或
A C B D

无拓扑有序序列，因为图中存在环 {B, C, D}。

7.5 有向无环图及其应用

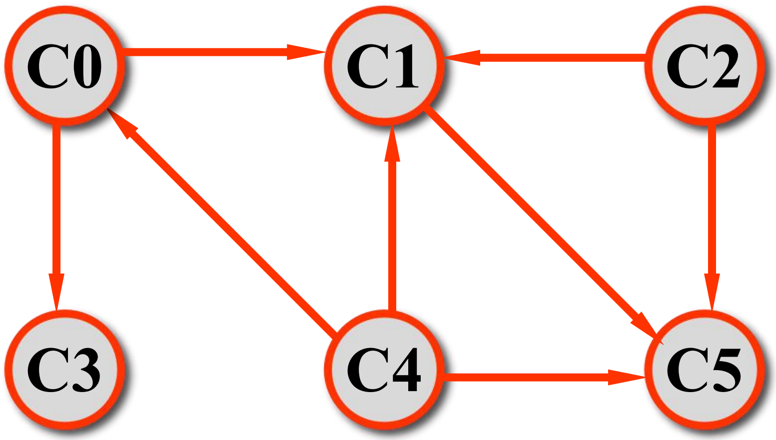
练习：某AOV网的邻接表存储结构如下，画出该AOV网，并写出拓扑排序序列。



Indgree

7.5 有向无环图及其应用

练习参考答案：



拓扑排序：C4、C0、C3、C2、C1、C5（栈）

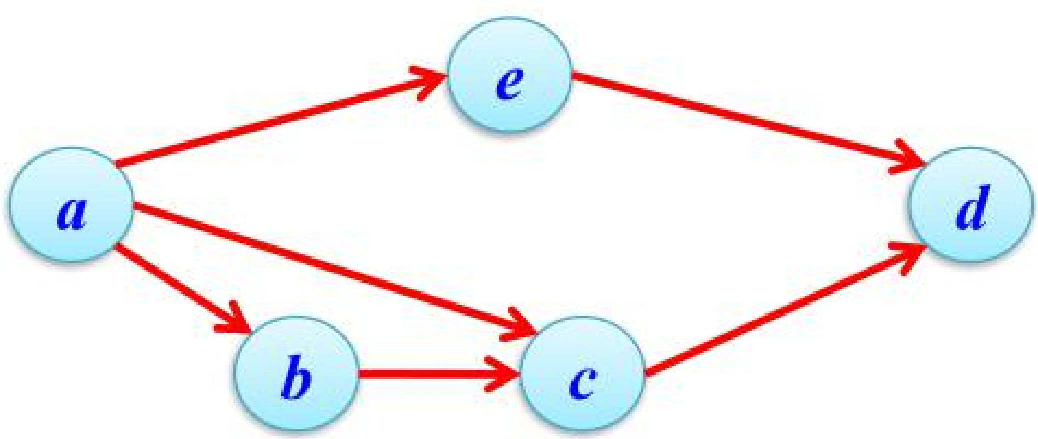
C2、C4、C0、C1、C3、C5（队列）

7.5 有向无环图及其应用

练习： 对如图所示的图进行拓扑排序，可以得到不同的拓扑序列个数是（）。

A. 4　　B. 3　　C. 2　　D. 1

注：2010 年全国考研题



不同的拓扑序列有：aebcd 、abced 、abecd 。答案为B 。

7.5 有向无环图及其应用

三. 关键路径

- 对于工程活动，人们不仅关心工程能否顺利完成，还关心：
  - 估算整个工程完成所必须的最短时间是多少?
  - 影响工程的关键活动是什么?哪些活动的延期将会影响整个工程的进度，而加速这些活动是否会提高整个工程的效率？

7.5 有向无环图及其应用

三. 关键路径

- AOE网(Activity On Edge Network)，以顶点表示事件(Event)，以有向边表示活动，边上的权值表示活动持续的时间，这样的有向图称为边表示活动的网。
- AOE网可以描述工程的预计进度

图中入度为0 的顶点表示工程的开始事件（如开工仪式），出度为0 的顶点表示工程结束事件。

7.5 有向无环图及其应用

三. 关键路径

- 常采用AOE网估算工程的进度

在**AOE**网中列出完成预定工程要发生哪些事件（顶点**v**）、计划所需要进行的活动（边**e**，每个活动预计完成的时间即**e**的权值），以及这些事件与活动之间的关系，其中入度为**0** 的顶点表示工程的开始事件（如开工仪式）称为源点，出度为**0** 的顶点表示工程结束事件称为汇点。

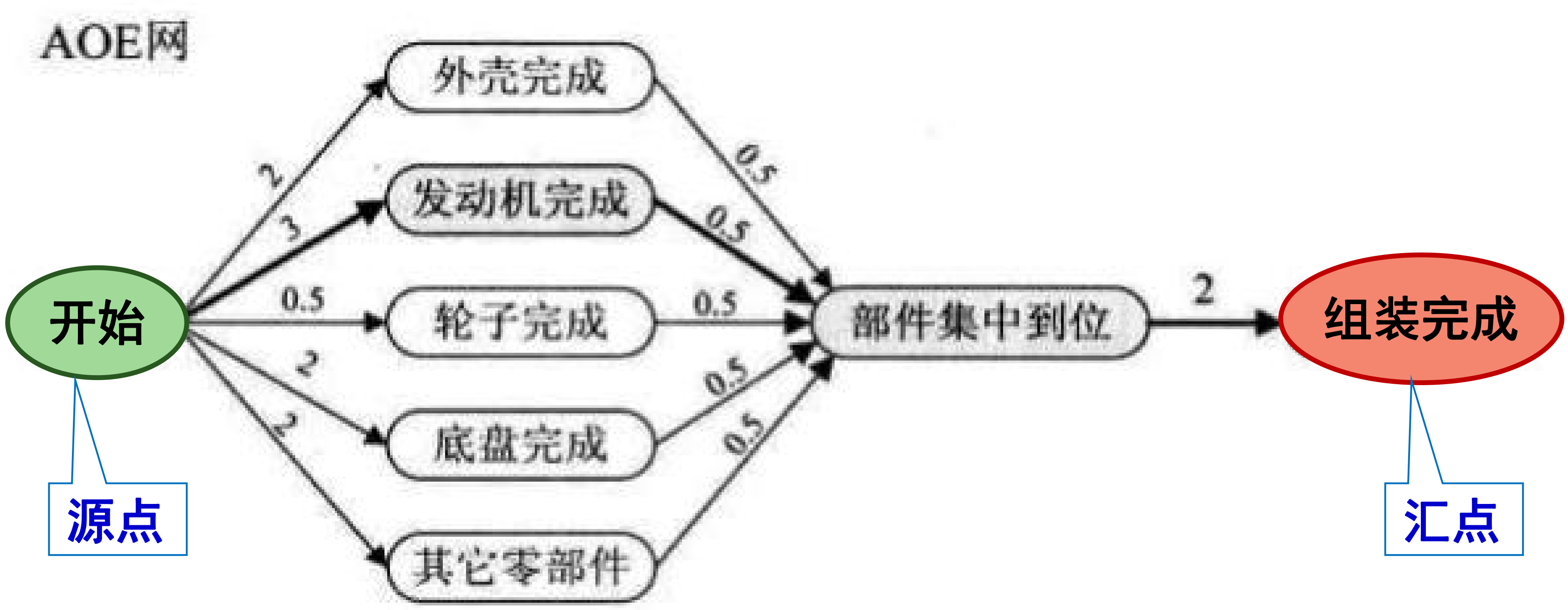
从而可以确定该项工程是否可行，包括：

- ✓ 估算工程完成的时间
- ✓ 确定哪些活动是影响工程进度的关键

7.5 有向无环图及其应用

三. 关键路径

- **AOE网**：边上的权值表示活动的开销（该活动持续的时间）。



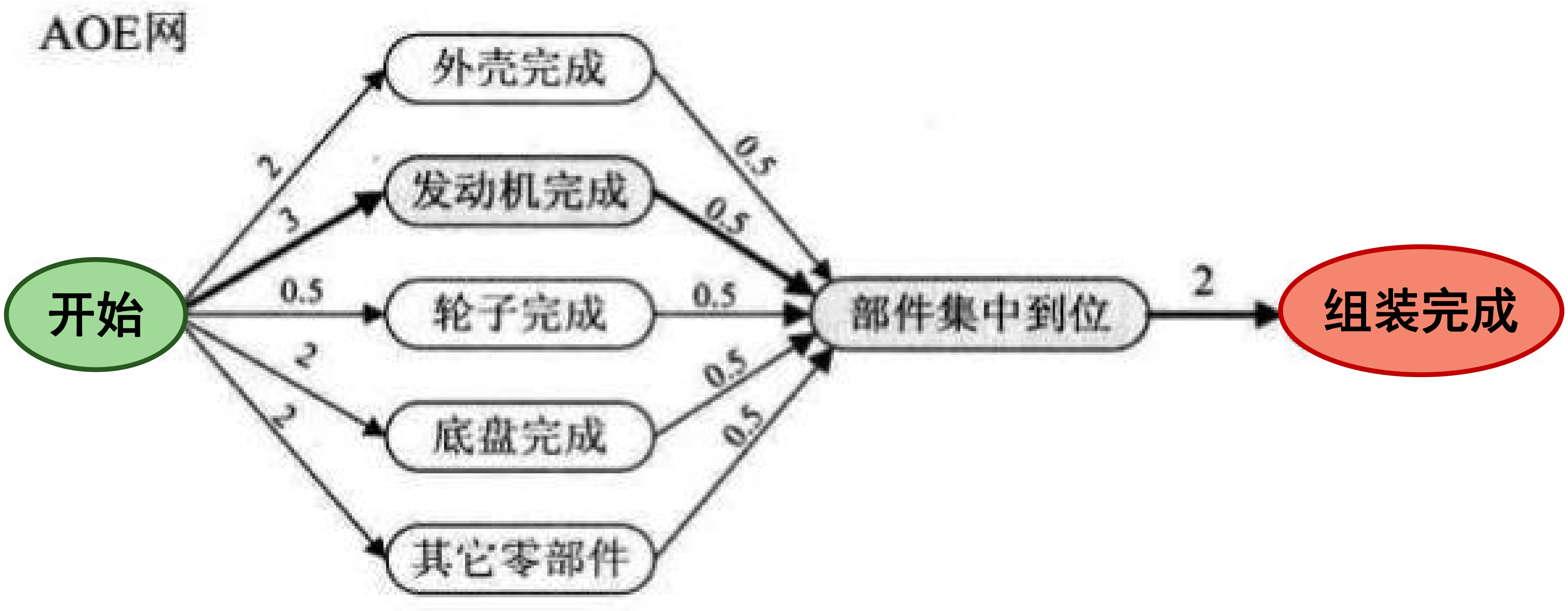
7.5 有向无环图及其应用

三. 关键路径

- **AOE网的性质**：

① 只有在某顶点事件发生后，从该顶点出发的各有向边活动才能开始。

② 只有在进入一某顶点所代表的事件的各有向边所代表的活动都已经结束，该顶点事件才能发生。



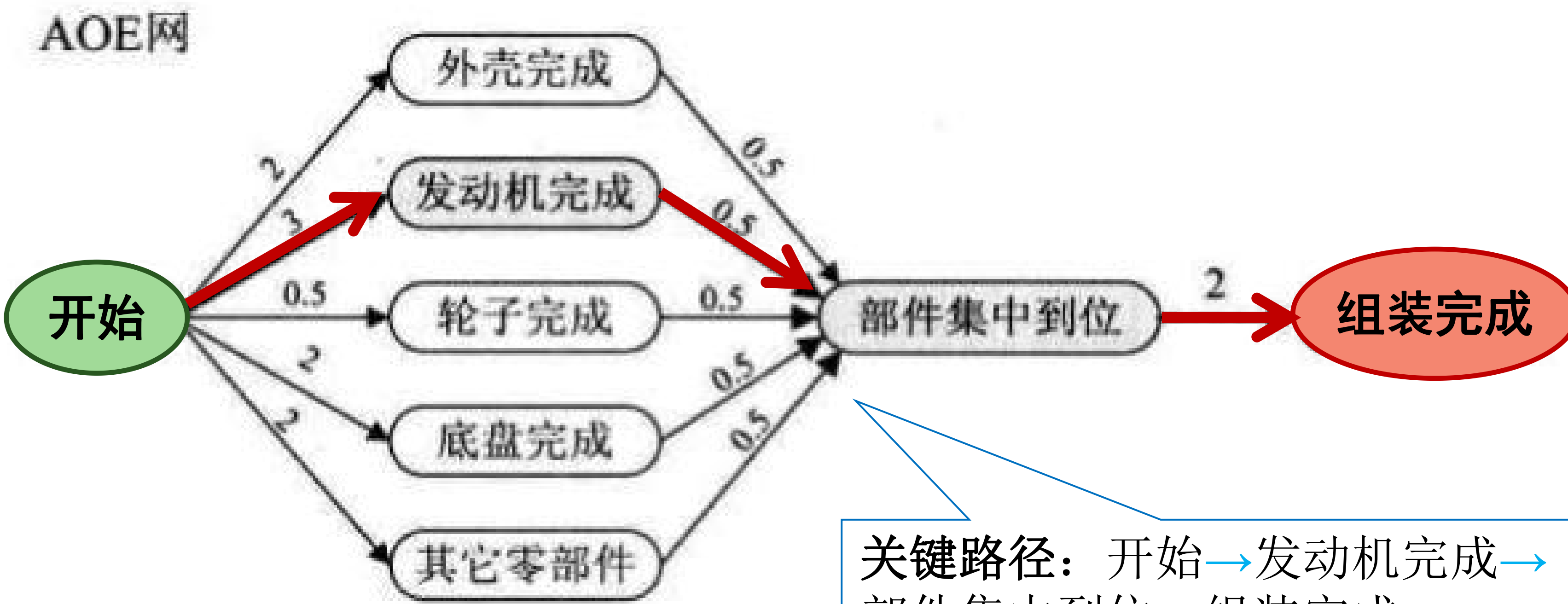
7.5 有向无环图及其应用

三. 关键路径

- 把路径上各个活动所持续时间之和称为**路径长度**
- 关键路径：从源点到汇点的最大长度的路径为关键路径，其长度（路径上各活动持续时间之和）为工程完成最短时间。
- 关键路径上的活动称为关键活动，关键活动是影响整个工程的关键所在。关键路径可能不唯一。

7.5 有向无环图及其应用

三. 关键路径

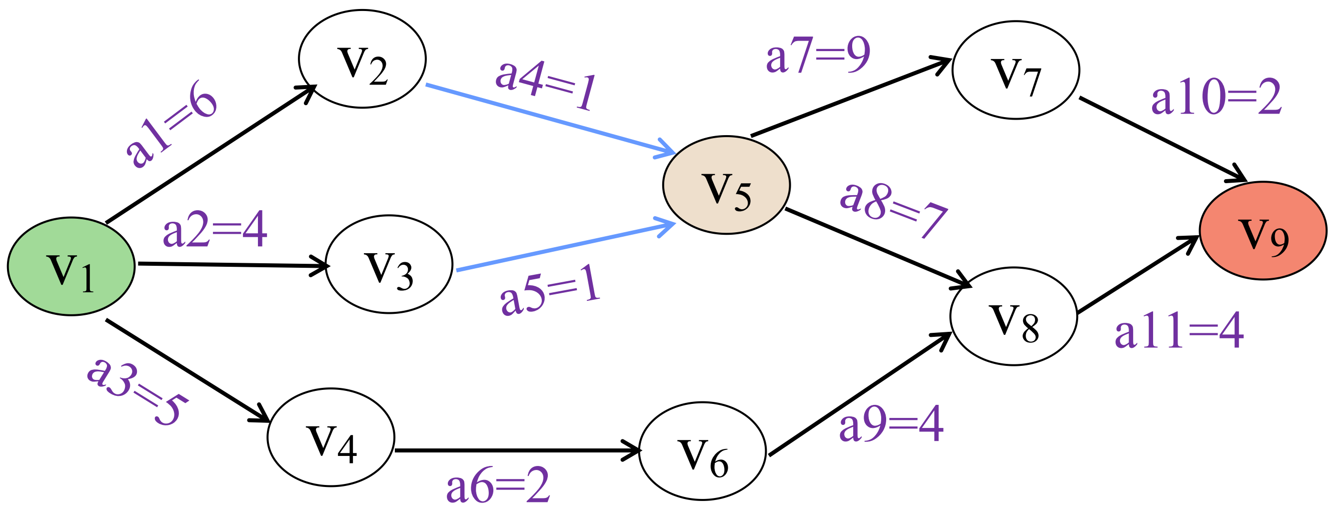


关键路径： 开始→发动机完成→部件集中到位→组装完成

工程完成最短时间，即关键路径长度： 3+0.5+2=5.5（时间单位）

7.5 有向无环图及其应用

三. 关键路径

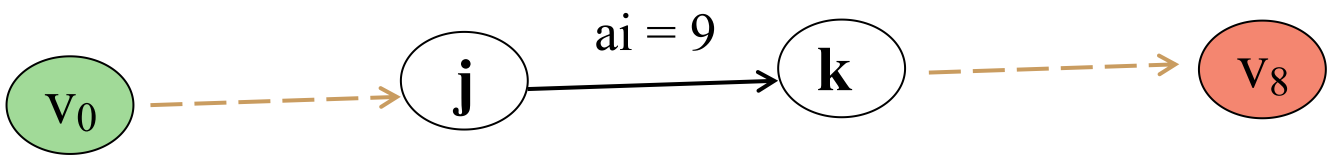


7.5 有向无环图及其应用

三. 关键路径

■ 算法的思路

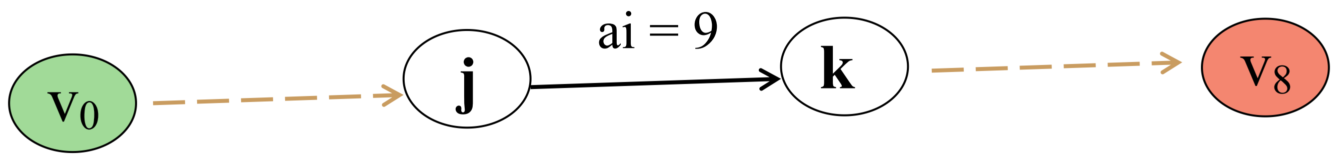
❑ 几个重要参数：



- **Ve(j)**：表示事件j的最早发生时间，即从源点到顶点j的最长路径长度；
- **V*l*(k)**：表示事件k的最晚发生时间，也就是说事件k再不发生，将会延误整个工期。
- **e(ai)**：表示活动ai的最早开始时间
- ***l*(ai)**：在不影响进度的前提下，表示活动ai的最晚开始时间
- ***l*(ai)-e(ai)**表示活动i的时间余量，若*l*(ai)-e(ai)=0，表示活动ai是关键活动，即在关键路径上。

7.5 有向无环图及其应用

三. 关键路径



- e(ai)、*l*(ai)、Ve(j)、V*l*(k)的关系：
  - e(ai) = Ve(j)（即活动***ai***的弧尾事件***j***的最早发生时间——只有事件***j***发生了，活动***ai***才可以开始）
  - l(ai) = Vl(k)- dut(<j, k>) （即活动***ai***的弧头事件***k***的最晚发生时间减去活动***ai***的持续时间）

说明：找到所有活动的最早开始时间e(ai)和最晚开始时间l(ai)，并且相比较，如果e(ai)= l(ai)，则此活动ai是关键活动

7.5 有向无环图及其应用

三. 关键路径

- 以顶点k为弧头的多条弧〈x, k〉, x=0, 1, ……, 存在公式

$$
Ve(k) = \begin{cases} 0 & \text{若 } k=0\text{，表示}k\text{是源点} \\ \mathrm{Max}\{ Ve(x) + dut(\langle x, k\rangle) \mid \langle x, k\rangle\text{是网中的弧}\} & \end{cases}
$$

含义：源点事件的最早发生时间设为0；除源点外，只有进入顶点k的所有弧所代表的活动全部结束后，事件 k才能发生。即只有k的所有前驱事件x的最早发生时间Ve(x)计算出来后，才能计算Ve(k) 。

方法：对所有事件进行拓扑排序，然后依次按拓扑顺序计算每个事件的最早发生时间。

7.5 有向无环图及其应用

三. 关键路径

- 以顶点j为弧尾的多条弧< j, y>， k=0, 1, ……， 存在公式

$$Vl(j) = \begin{cases} Ve(n-1) & \text{若}j=n-1\text{，表示}j\text{是汇点} \\ \mathrm{Min}\{ Vl(y) - dut(\langle j, y\rangle) \mid \langle j, y\rangle\text{是网中的弧}\} & \end{cases}$$

含义：只有 j 的所有后继事件y的最晚发生时间Vl(y)计算出来后，才能计算Vl(j)。

方法：按拓扑排序的逆顺序，依次计算每个事件的最晚发生时间。

7.5 有向无环图及其应用

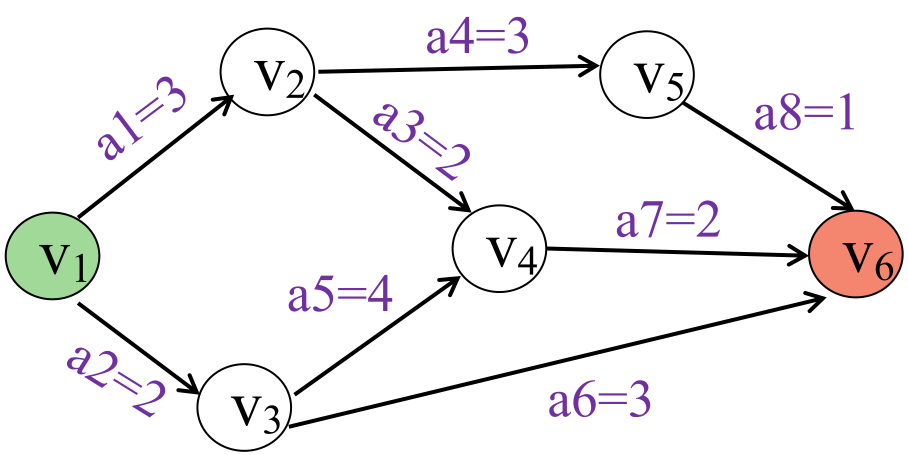
三. 关键路径

- 求完成最短时间和关键路径的方法
  - 利用拓扑排序求出**AOE**网的一个拓扑序列；
  - 从拓扑序列的第一个顶点(源点)开始，按拓扑顺序依次计算每个事件的最早发生时间***Ve(j)***；
  - 从拓扑序列的最后一个顶点(汇点)开始，按逆拓扑顺序依次计算每个事件的最晚发生时间***Vl(j)***；
  - 计算每个活动ai的最早开始时间*e(ai)*和最晚开始时间*l(ai)*，若e(ai)= l(ai)，则该活动ai为关键活动，该弧所依附的顶点序列为关键路径，关键路径上的权值之和为完成工程的最短时间。

7.5 有向无环图及其应用

三. 关键路径

- 求完成最短时间和关键路径
  - 拓扑排序序列：(v_1, v_2, v_3 , v_4, v_5 , v_6)
  - 计算各个事件的ve(i)和v*l*(i)值



顶点	ve	vl
v1	0	0
v2	3	4
v3	2	2
v4	6	6
v5	6	7
v6	8	8

7.5 有向无环图及其应用

三. 关键路径

- 求完成最短时间和关键路径
  - 计算各个活动的e(ai)和*l*(ai)值

顶点	ve	vl
v1	0	0
v2	3	4
v3	2	2
v4	6	6
v5	6	7
v6	8	8

活动	弧	e	dut	l	l-e
a1	<1,2>	0	3	1	
a2	<1,3>	0	2	0	0
a3	<2,4>	3	2	4	
a4	<2,5>	3	3	4	
a5	<3,4>	2	4	2	0
a6	<3,6>	2	3	5	
a7	<4,6>	6	2	6	0
a8	<5,6>	6	1	7	

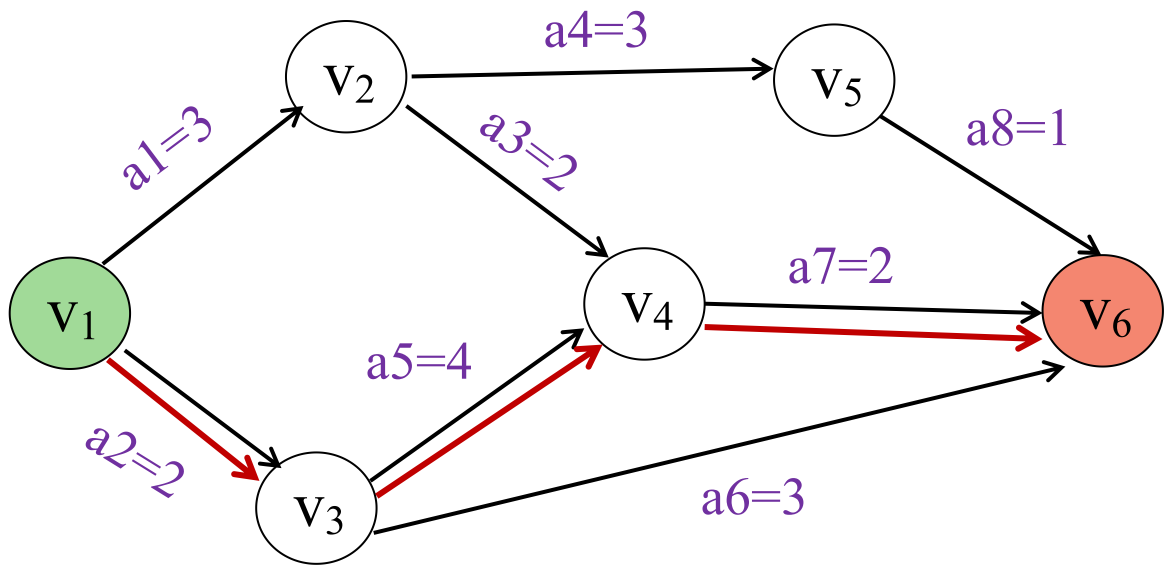
关键活动：**a2** <v1, v3>，**a5** <v3, v4>，**a7** <v4, v6>，
关键路径：**(v_1, v_3, v_4, v_6)**

7.5 有向无环图及其应用

三. 关键路径

- 求完成最短时间和关键路径

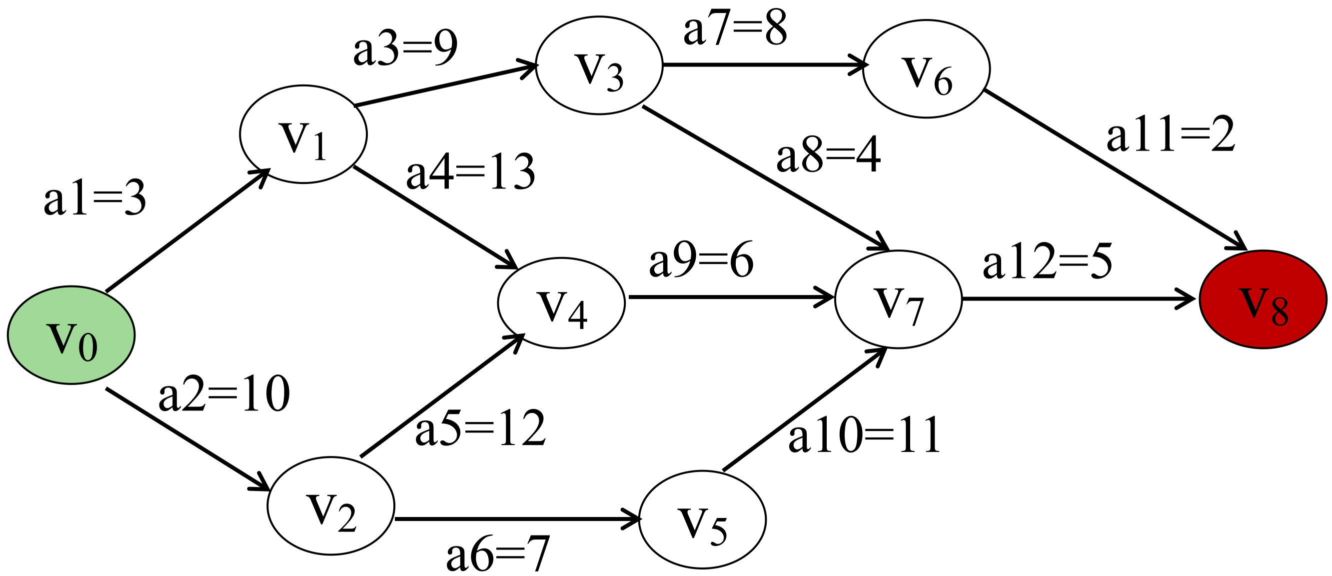
关键活动：**a2** <v1, v3>，**a5** <v3, v4>，**a7** <v4, v6>，
关键路径：**(v_1, v_3, v_4, v_6)**



7.5 有向无环图及其应用

三. 关键路径

- 求完成最短时间和关键路径



7.5 有向无环图及其应用

三. 关键路径

- 求完成最短时间和关键路径

顶点	ve	vl
v0	0	0
v1	3	9
v2	10	10
v3	12	23
v4	22	22
v5	17	17
v6	20	31
v7	28	28
v8	33	33

活动	弧	dut	e	l	l-e
a1	<0,1>	3	0	6	
a2	<0,2>	10	0	0	0
a3	<1,3>	9	3	14	
a4	<1,4>	13	3	9	
a5	<2,4>	12	10	10	0
a6	<2,5>	7	10	10	0
a7	<3,6>	8	12	23	
a8	<3,7>	4	12	24	
a9	<4,7>	6	22	22	0
a10	<5,7>	11	17	17	0
a11	<6,8>	2	20	31	
a12	<7,8>	5	28	28	0

关键活动：**a2** <v0, v2>， **a5** <v2, v4>， **a6** <v2, v5>， **a9** <v4, v7>， **a10** <v5, v7>， **a12** <v7, v8>

关键路径： **(v_0, v_2, v_4, v_7 , v_8)** 和 **(v_0, v_2, v_5 , v_7 , v_8)**

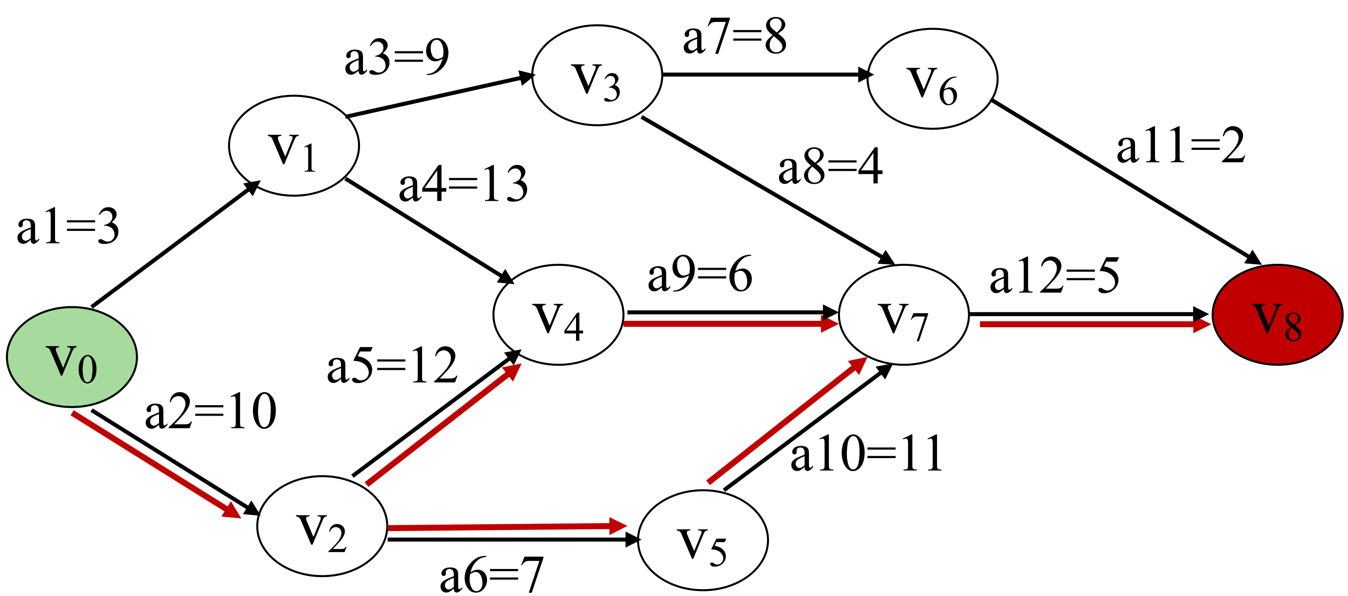
7.5 有向无环图及其应用

三. 关键路径

- 求完成最短时间和关键路径

关键活动：**a2<v0, v2>，a5<v2, v4>，a6<v2, v5>，a9<v4, v7>，a10<v5, v7>，a12<v7, v8>**

关键路径：**(v_0, v_2, v_4, v_7 , v_8) 和(v_0, v_2, v_5 , v_7 , v_8)**



7.5 有向无环图及其应用

三. 关键路径

- 用AOE网估算工程完成时间
  - 求关键路径必须在拓扑排序的前提下进行，有环图不能求关键路径;
  - 只有在不改变关键路径的前提下，缩短关键活动时间才能缩短整个工程工期;
  - 一个项目可以有多个、并行的关键路径。若一个关键活动不在所有的关键路径上，减少它并不能减少工期。

练习

- 下表给出了某工程各工序之间的优先关系和各工序所需的时间（其中“-”无先驱工序）。

工序代号	A	B	C	D	E	F	G	H
所需时间	3	2	2	3	4	3	2	1
先驱工序	-	-	A	A	B	B	C,E	D

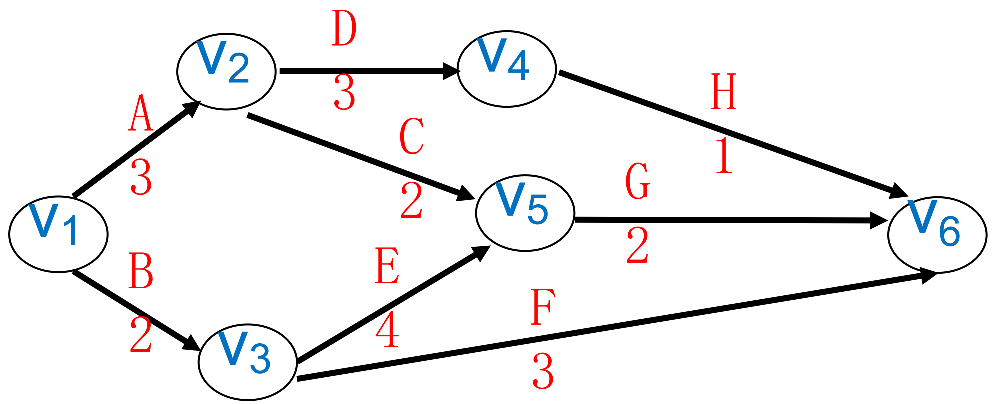
问:（**1**）画出相应的**AOE**网；

（**2**）如果该工程是否能够顺利进行，请问要花多长时间?

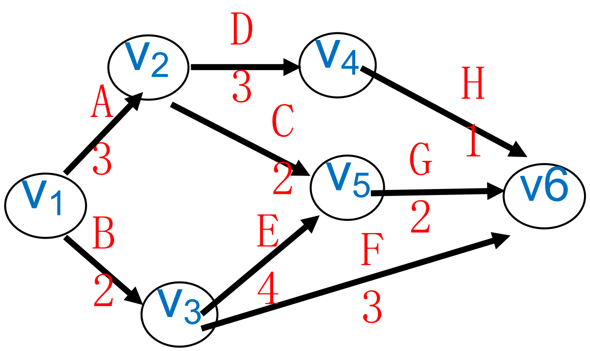
（**3**）缩短哪些工序可以缩短整个工程的完工时间?

练习（参考答案）

工序代号	A	B	C	D	E	F	G	H
所需时间	3	2	2	3	4	3	2	1
先驱工序	-	-	A	A	B	B	C,E	D



练习（参考答案）



(2) 所有事件的最早发生时间ve，如下所示：

ve(v1)= 0　　ve(v2)= 3　　ve(v3)= 2

ve(v4)= Max{ ve(v2)+2，ve(v3)+4}= 6

ve(v5)= ve(v2)+3 = 6

ve(v6)= Max{ ve(v3)+3，ve(v4)+2，ve(v5)+1}= 8

所有事件的最迟发生时间vl，如下所示：

vl(v6)= 8　vl(v5)=vl(v6)−1= 7　vl(v4)=vl(v6)−2 = 6

vl(v3)= Min{ vl(v4)−4，vl(v6)−3}= 2

vl(v2)= Min{ vl(v4)−2，vl(v5)−3}= 4

vl(v1)= Min{ vl(v2)−3，vl(v3)−2}= 0

(3)　求所有活动的最早发生时间e、最迟发生时间l和时间余量l-e。

e(A)=ve(v1)= 0　l(A)=vl(v2)−3= 1　l(A)−e(A)= 1

e(B)=ve(v1)= 0　l(B)=vl(v3)−2= 0　l(B)−e(B)= 0

e(C)=ve(v2)= 3　l(C)=vl(v4)−2= 4　l(C)−e(C)= 1

e(D)=ve(v2)= 3　l(D)=vl(v5)−3= 4　l(D)−e(D)= 1

e(E)=ve(v3)= 2　l(E)=vl(v4)−4= 2　l(E)−e(E)= 0

e(F)=ve(v3)= 2　l(F)=vl(v6)−3= 5　l(F)−e(F)= 3

e(G)=ve(v4)= 6　l(G)=vl(v6)−2= 6　l(G)−e(G)= 0

e(H)=ve(v5)= 6　l(H)=vl(v6)−1= 7　l(H)−e(H)= 1

所以，关键路径为：B、E、G。

完成该工程最少需要8天时间。