

## 6.8 关键路径

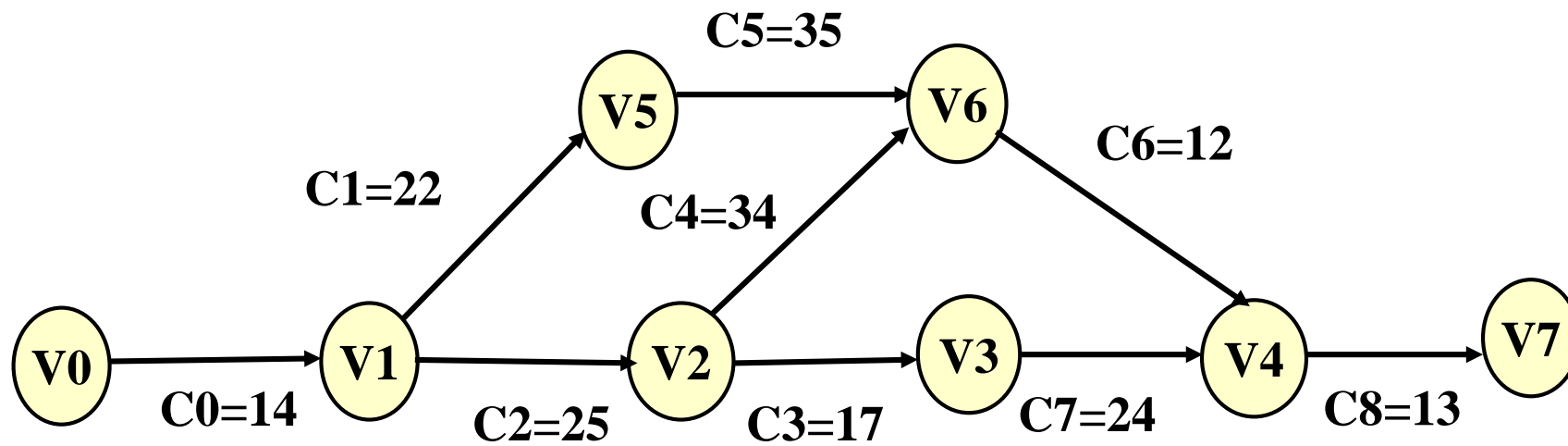
### AOE网(Activity On Edge network)

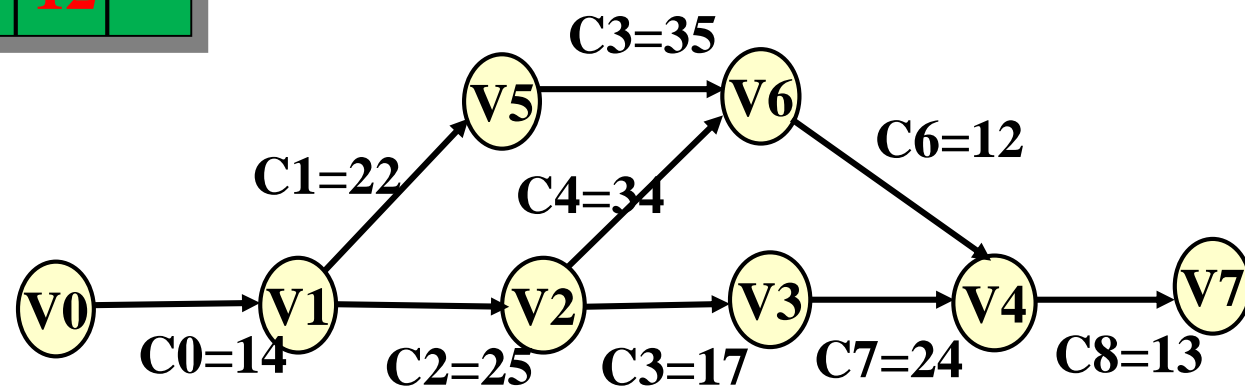
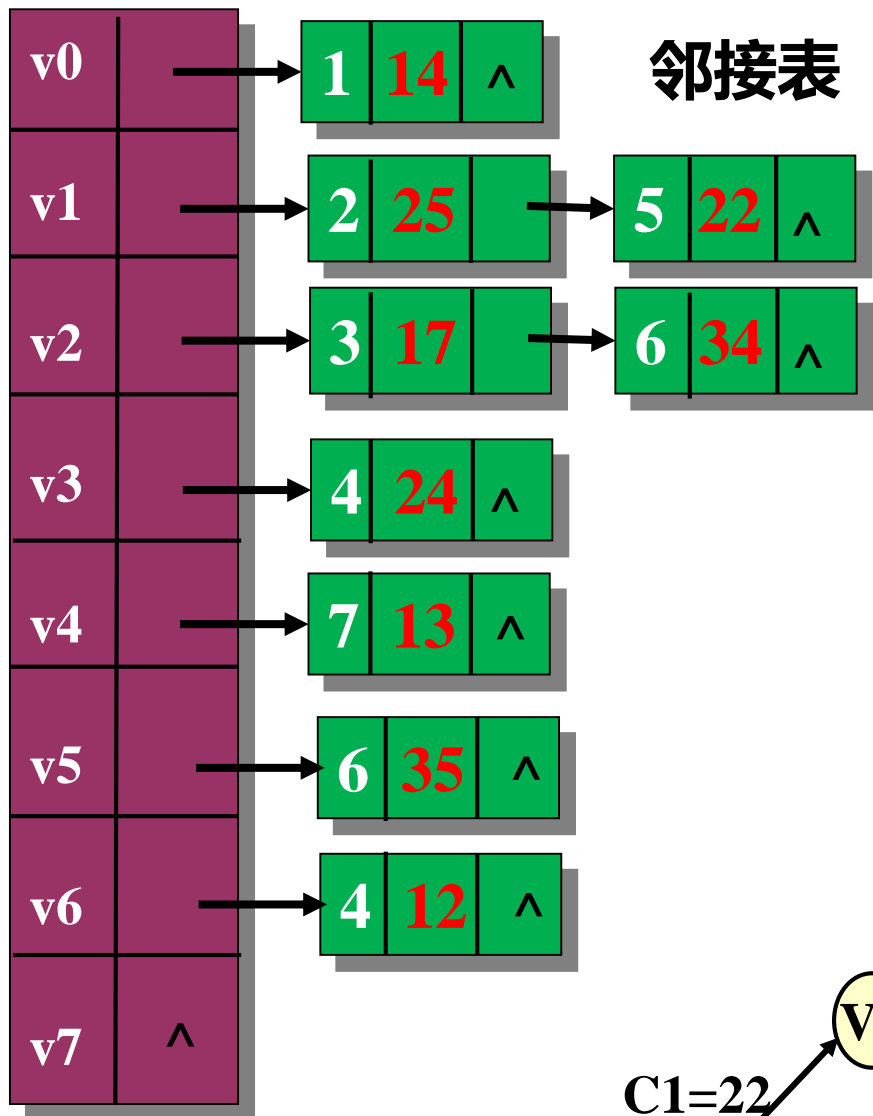
- 带权的有向图;
- 顶点表示事件, 有向边表示活动;
- 边上的权值表示活动持续的时间;
- 顶点所表示的事件实际上就是它的入边所表示的活动都已完成, 它的出边所表示的活动可以开始这样一种状态;
- 只有一个入度为0的顶点
- 只有一个出度为0的顶点



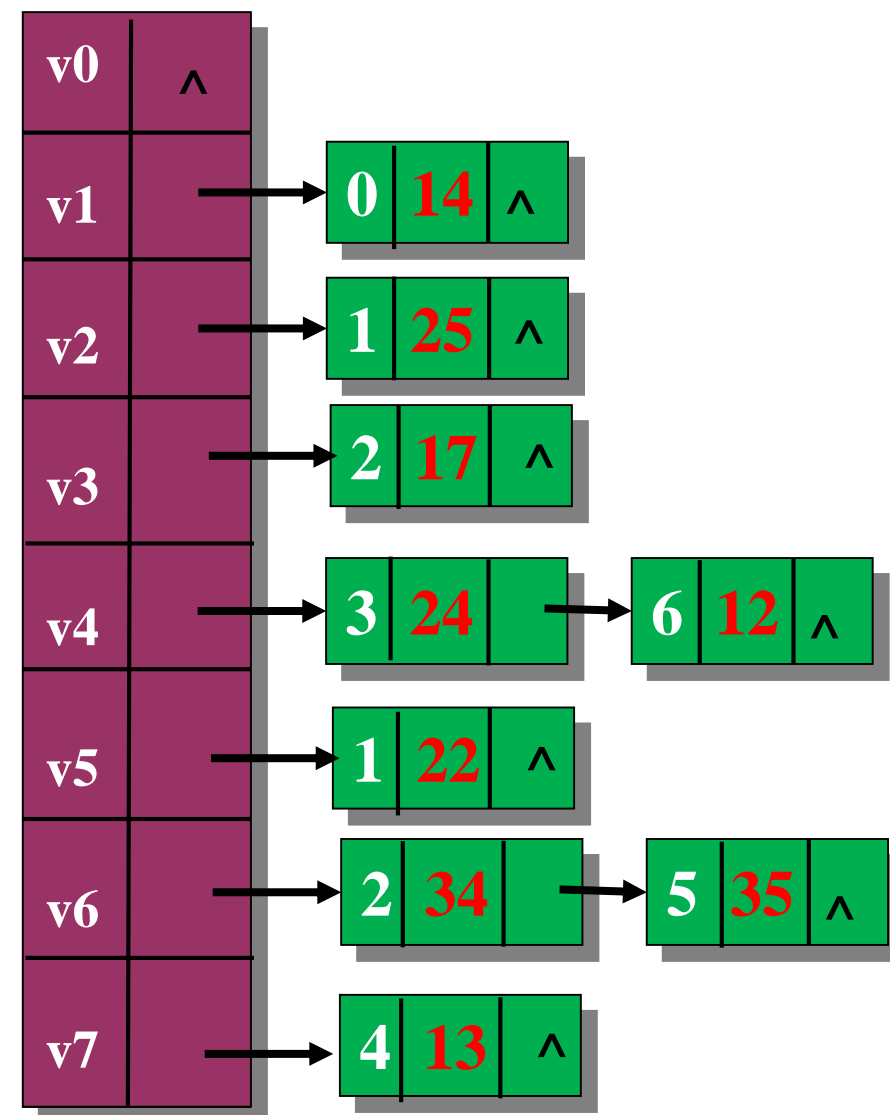
表 6-8 活动计划表

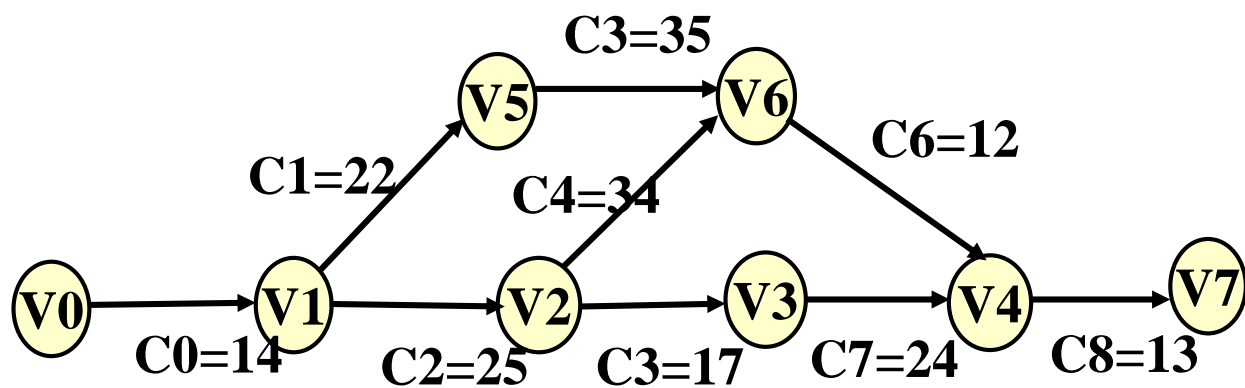
活动名称	符号	活动时间 (天)	依赖活动
框架	C0	14	---
屋面	C1	22	C0
外墙	C2	25	C0
门窗	C3	17	C2
卫生管道	C4	34	C2
各种电气	C5	35	C1
内部装修	C6	12	C4,C5
外部粉刷	C7	24	C3
工程验收	C8	13	C6,C7





**逆邻接表**





## 逆邻接表

v0	^	
v1		→ 0 14 ^
v2		→ 1 25 ^
v3		→ 2 17 ^
v4		→ 3 24 → 6 12 ^
v5		→ 1 22 ^
v6		→ 2 34 → 5 35 ^
v7		→ 4 13 ^

## 6.8 关键路径

例 设一个工程有9项活动C0.C1,...C8 , 8个事件

事件V0——表示整个工程开始事件

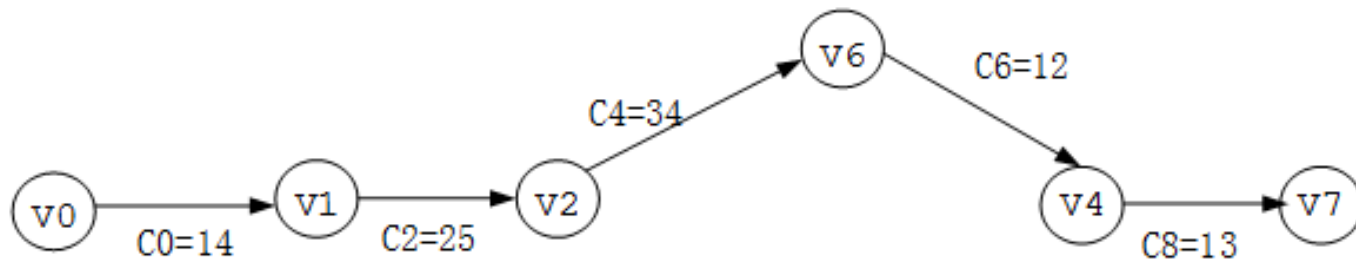
事件V7——表示整个工程完成事件

问题: (1) 完成整项工程至少需要多少时间?

(2) 哪些活动是影响工程进度的关键?

AOE网中有些活动可以并行进行, 所以完成整个工程的最短时间是从开始顶点到完成顶点的最长路径长度 (路径长度为路径上各边的权值之和)

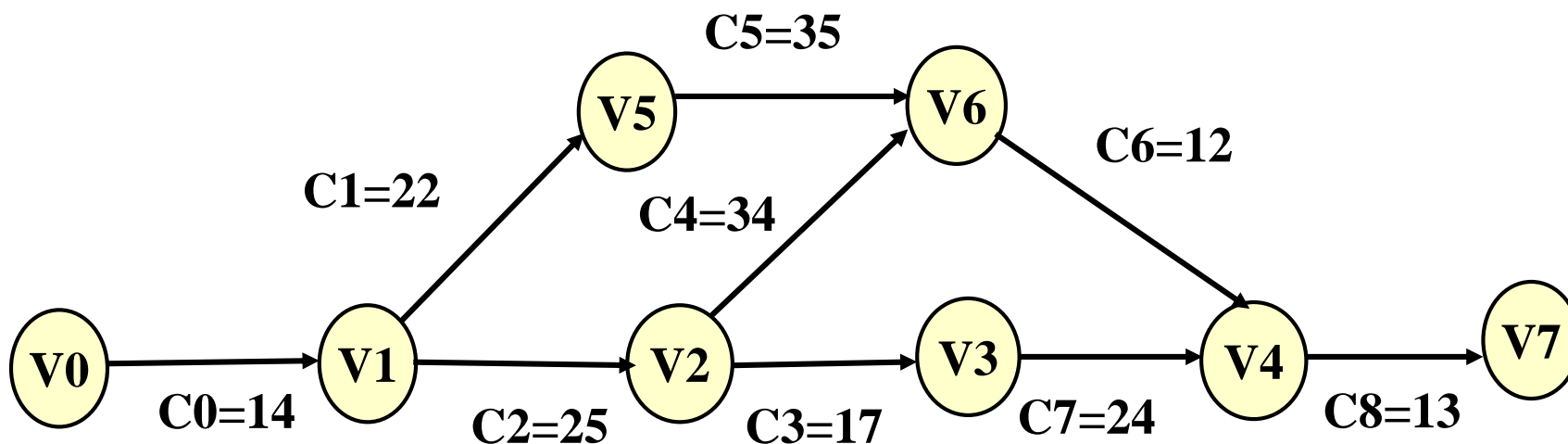
把开始顶点到完成顶点的最长路径称为关键路径



## 6.8 关键路径

### ●求关键路径步骤

- 求事件的最早发生时间 $\text{earliestTime}(j)$
- 求事件的最迟发生时间 $\text{latestTime}(j)$
- 求活动的最早发生时间 $\text{activityEarliestTime}(k)$
- 求活动的最迟发生时间 $\text{activityLatestTime}(k)$
- 计算活动的时差 $\text{reminder} = \text{activityLatestTime}(k) - \text{activityEarliestTime}(k)$
- $\text{Reminder} = 0$  的即为关键活动



## 6.8 关键路径

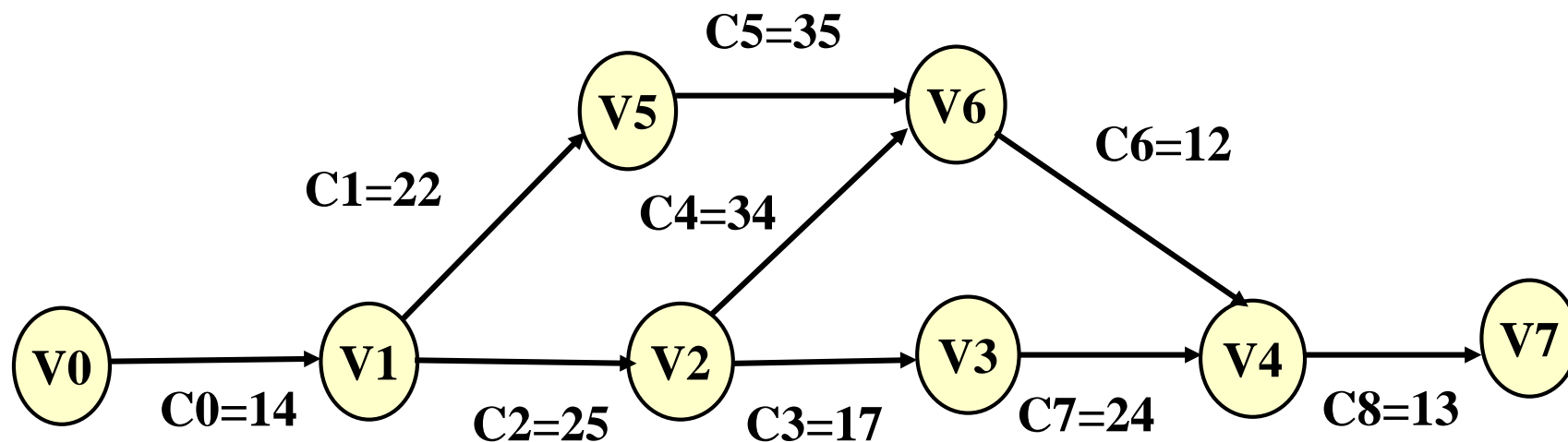
■ **事件的最早发生时间(earliestTime):** 事件 $V_i$ 的最早可能的开始时间, 是从开始顶点 $V_0$ 到顶点 $V_i$ 的最长路径的长度。计算事件的最早发生时间。

采用正向递推方式: 初始 $\text{earliestTime}[0] = 0$ ;

$\text{earliestTime}[j] = \max\{ \text{earliestTime}[i] + \text{weight}\langle v_i, v_j \rangle \}$

其中,  $\langle v_i, v_j \rangle$  是以顶点 $v_j$ 为终点的所有有向边

	V0	V1	V2	V3	V4	V5	V6	V7
eT	0	14	39	56	85	36	73	98



## 6.8 关键路径

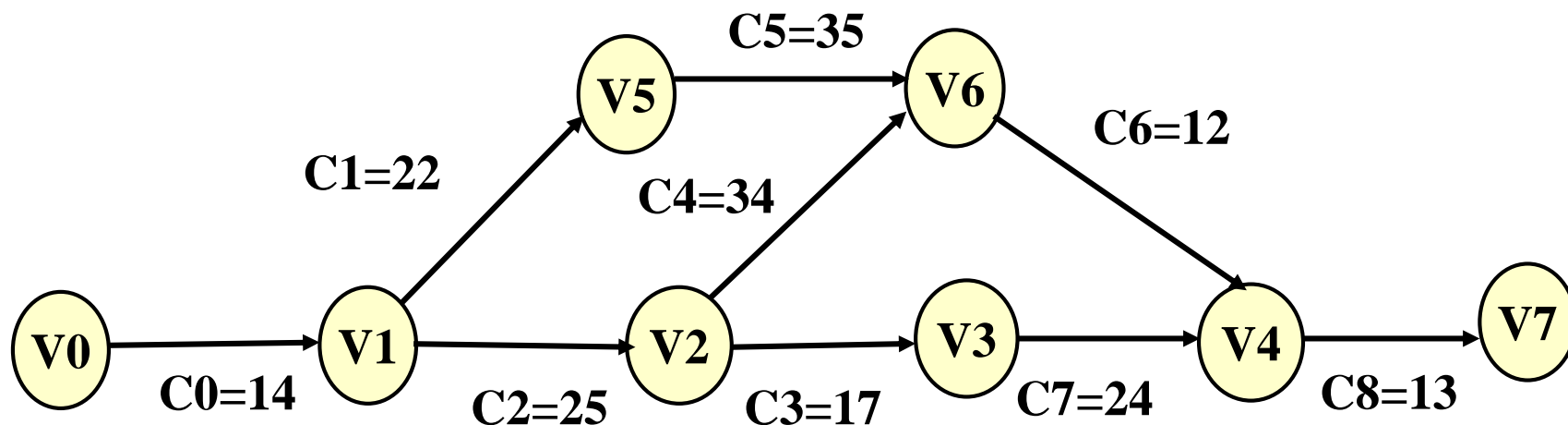
■ **事件的最迟发生时间(latestTime)**: 事件 $V_i$ 最迟允许的开始时间, 是指在不推迟整个工期的前提下, 事件 $V_i$ 允许的最晚时间。

采用反向递推方式: 初始 $\text{latestTime}[n-1] = \text{earliestTime}[n-1]$

$\text{latestTime}[j] = \min\{ \text{latestTime}[i] - \text{weight}\langle v_j, v_i \rangle \}$

其中,  $\langle v_j, v_i \rangle$  是以顶点 $v_j$ 为起点的所有有向边

	V0	V1	V2	V3	V4	V5	V6	V7
eT	0	14	39	56	85	36	73	98
IT	0	14	39	61	85	38	73	98



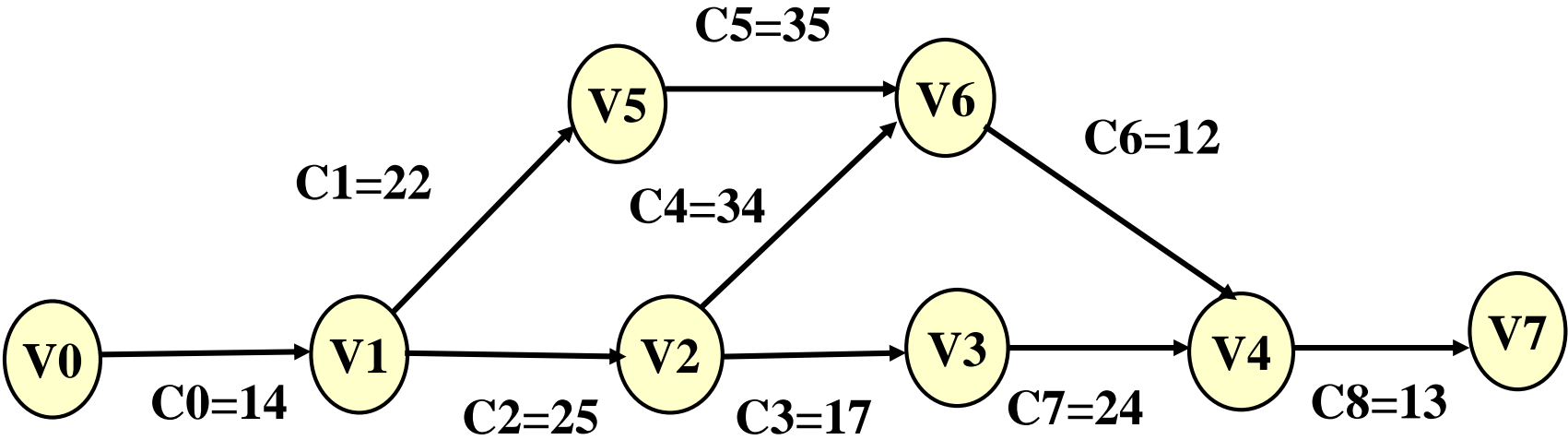


# 6.8 关键路径

■ **活动的最早发生时间计算activityEarliestTime:** 设 $C_k$ 是边 $\langle v_i, v_j \rangle$ 上的活动, 则activityEarliestTime是源点 $v_0$ 到起始顶点 $v_i$ 的最长路径长度, 即为:  
**activityEarliestTime[k] = earliestTime[i]**

	C0	C1	C2	C3	C4	C5	C6	C7	C8
aET	0	14	14	39	39	36	73	56	85

	V0	V1	V2	V3	V4	V5	V6	V7
eT	0	14	39	56	85	36	73	98

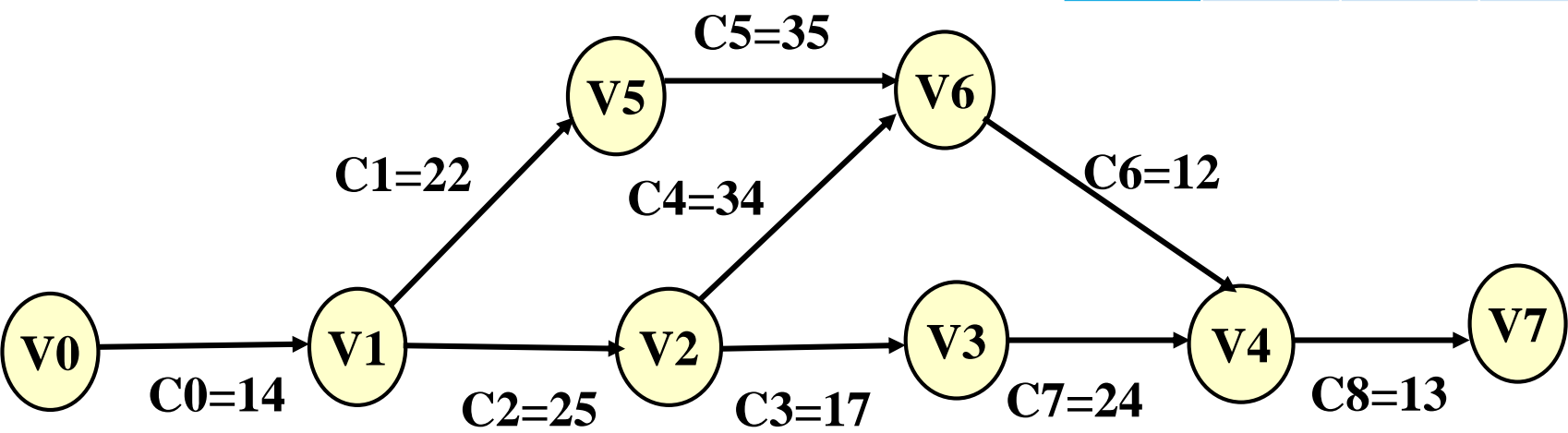


# 6.8 关键路径

■ **活动的最迟发生时间计算activityLatestTime:**设 $C_k$ 是边 $\langle v_i, v_j \rangle$ 上的活动， $activityLatestTime[k]$ 是在不引起时间延误的前提下，活动 $C_k$ 允许的最迟时间，也就是顶点事件 $v_j$ 的最迟发生时间减去活动 $C_k$ 持续的时间 $weight\langle v_i, v_j \rangle$ ，即为：  
 **$activityLatestTime[k] = latestTime[j] - weight\langle v_i, v_j \rangle$**

	C0	C1	C2	C3	C4	C5	C6	C7	C8
aLT	0	16	14	44	39	38	73	61	85

	V0	V1	V2	V3	V4	V5	V6	V7
IT	0	14	39	61	85	38	73	98

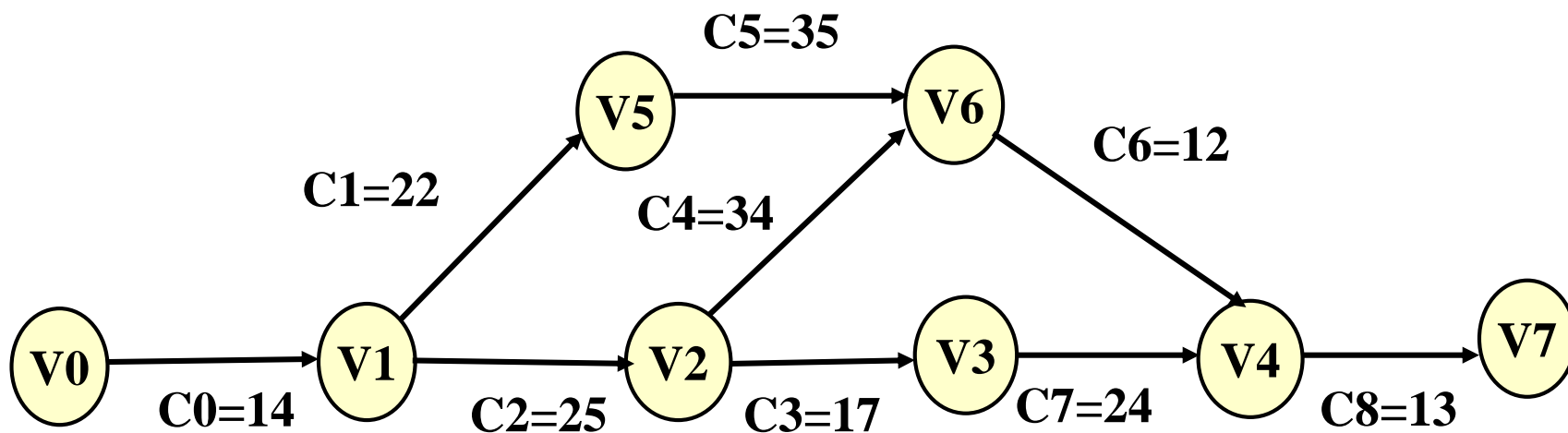


## ■ 活动的时间余量reminder

reminder[k]表示活动Ck的最早发生时间和最迟发生时间的时间余量，  
即为： $\text{reminder}[k] = \text{activityLatestTime}[k] - \text{activityEarliestTime}[k]$

当 $\text{activityLatestTime}[k] = \text{activityEarliestTime}[k]$ 时， $\text{reminder}[k] = 0$   
表示该活动Ck的时间余量为0，即该活动为关键活动。

	C0	C1	C2	C3	C4	C5	C6	C7	C8
aET	0	14	14	39	39	36	73	56	85
aLT	0	16	14	44	39	38	73	61	85
reminder	0	2	0	5	0	2	0	5	0
关键活动	是	否	是	否	是	否	是	否	是



- 求关键路径步骤

- 求earliestTime (j)

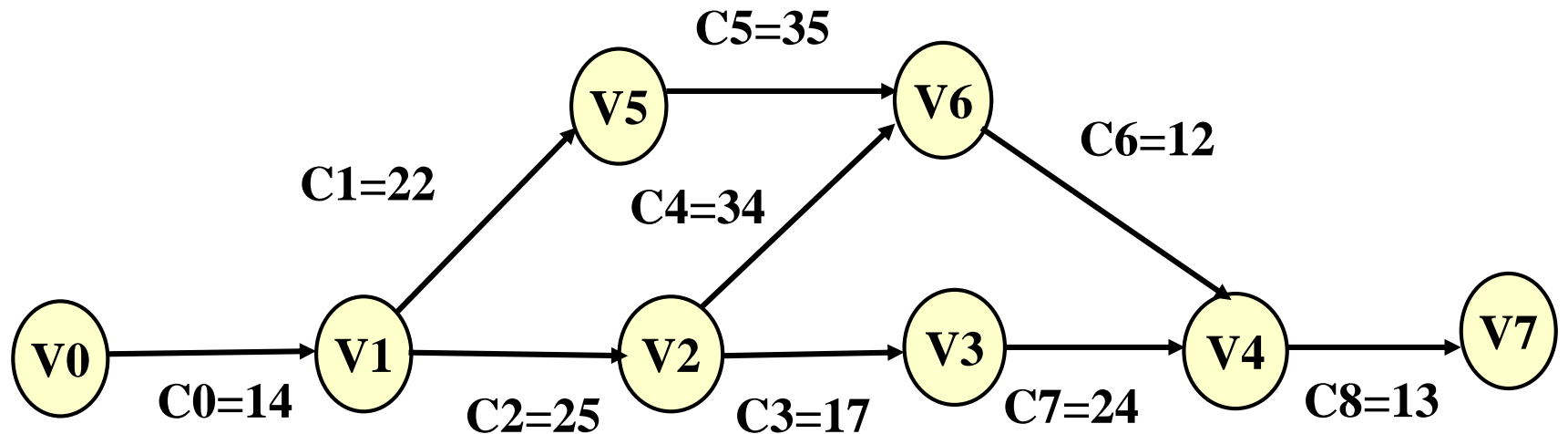
- 求latestTime (i)

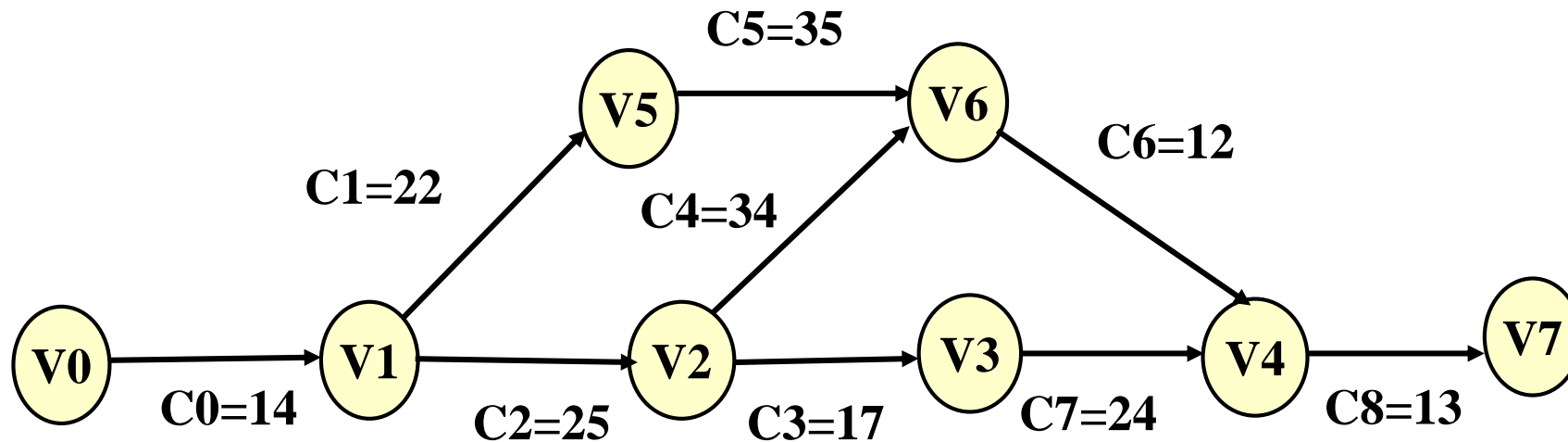
- 求activityEarliestTime (k)

- 求activityLatestTime (k)

- 计算activityLatestTime(k)- activityEarliestTime(k)

顶点	eT	lT
V0	0	0
V1	14	14
V2	39	39
V3	56	61
V4	85	85
V5	36	38
V6	73	73
V7	98	98





顶点	eT	lT
V0	0	0
V1	14	14
V2	39	39
V3	56	61
V4	85	85
V5	36	38
V6	73	73
V7	98	98

◆ 事件 $v_j$ 可能的最早发生时间earliestTime (j)

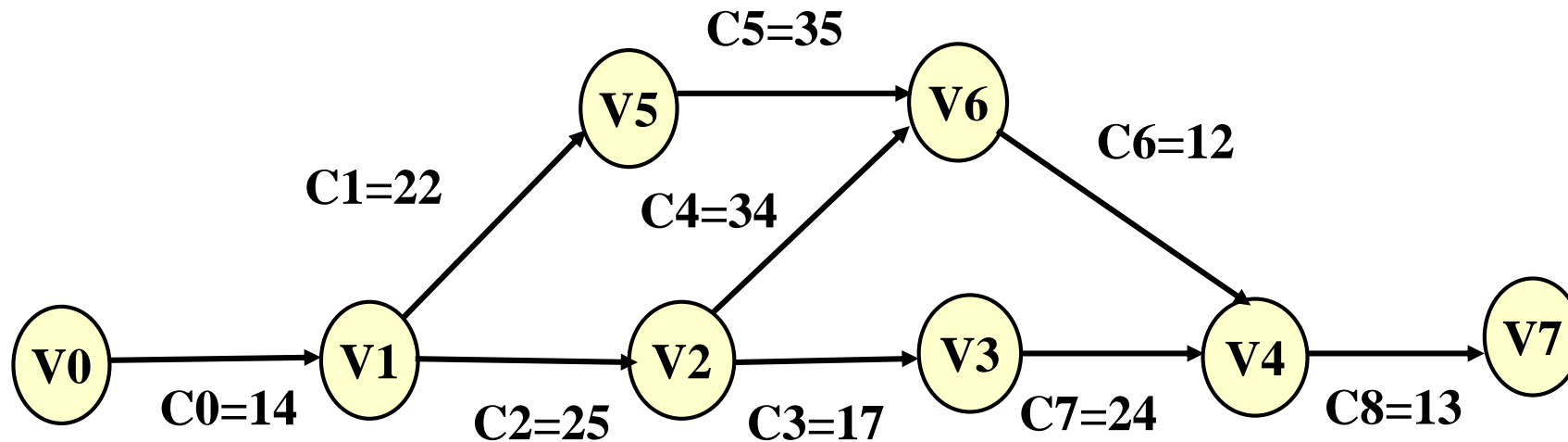
初始earliestTime[0] = 0;

$\text{earliestTime}[j] = \max\{ \text{earliestTime}[i] + \text{weight}\langle v_i, v_j \rangle \}$

◆ 事件 $v_j$ 允许的最迟发生时间earliestTime (j)

初始latestTime[n-1] = earliestTime[n-1]

$\text{latestTime}[j] = \min\{ \text{latestTime}[i] - \text{weight}\langle v_j, v_i \rangle \}$



顶点	eT	lT
V0	0	0
V1	14	14
V2	39	39
V3	56	61
V4	85	85
V5	36	38
V6	73	73
V7	98	98

活动	aET	lET	lET-aET
C0	0	0	0 ✓
C1	14	16	2
C2	14	14	0 ✓
C3	39	44	5
C4	39	39	0 ✓
C5	36	38	2
C6	73	73	0 ✓
C7	56	61	5
C8	85	85	0 ✓

- ◆ 活动 $a_k = \langle v_i, v_j \rangle$ 的最早开始时间  
 $\text{activityEarliestTime}[k] = \text{earliestTime}[i]$
- ◆ 活动 $a_k = \langle v_i, v_j \rangle$ 的最晚开始时间  
 $\text{activityLatestTime}[k] = \text{latestTime}[j] - \text{weight}\langle v_i, v_j \rangle$   
 $\text{weight}\langle v_i, v_j \rangle$ 为边 $\langle v_i, v_j \rangle$ 的权值

表 6-10 活动的最早发生和最迟发生时间以及关键活动

	C0	C1	C2	C3	C4	C5	C6	C7	C8
activityEarliestTime	0	14	14	39	39	36	73	56	85
activityLatestTime	0	16	14	44	39	38	73	61	85
reminder	0	2	0	5	0	2	0	5	0
关键活动	是	否	是	否	是	否	是	否	是

表 6-9 各个顶点的最早和最迟发生时间

	V0	V1	V2	V3	V4	V5	V6	V7
earliestTime	0	14	39	56	85	36	73	98
latestTime	0	14	39	61	85	38	73	98

