数学建模
（Mathematic Modeling）

国防科技大学

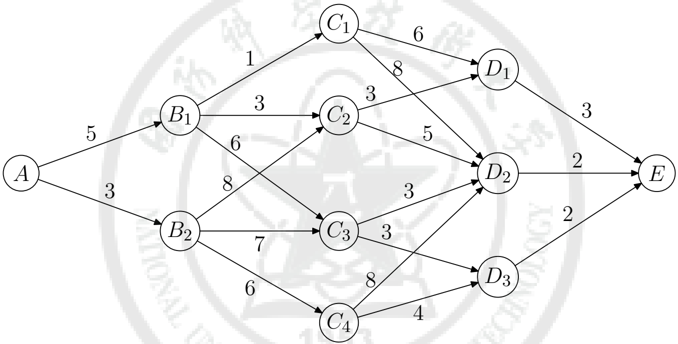
第五讲

第五讲

- 主要内容：介绍网络模型和统筹模型。

最短路问题及其算法

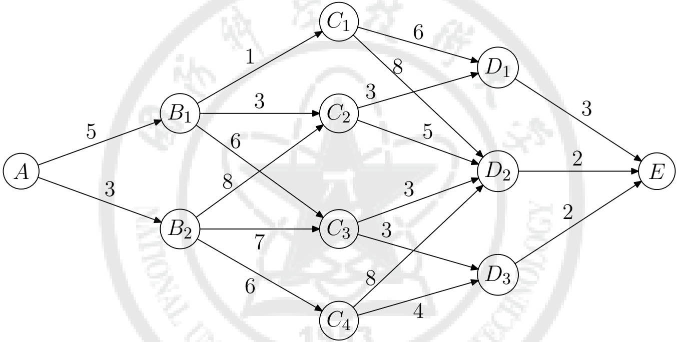
在动态规划模型中我们讲了一个最短路线问题的例子。



最短路径：$A \to B_1 \to C_2 \to D_1 \to E$ 路径长为：14

最短路问题及其算法

在动态规划模型中我们讲了一个最短路线问题的例子。



最短路径：$A \to B_1 \to C_2 \to D_1 \to E$ 路径长为：14

事实上，最短路问题的应用背景很广，如网络设计、运输方案、工作计划、设备更新等。

例 1 （设备更新问题）设某公司需使用某种设备一套，设备购买价格及维修费用见表。 现设该公司在第一年开始时新购入一套设备，问今后 5 年的设备更新方案如何，才能使得总费用最省？

年	1	2	3	4	5
价格	11	11	12	12	13

使用年限	0-1	1-2	2-3	3-4	4-5
维修费用	5	6	8	11	18

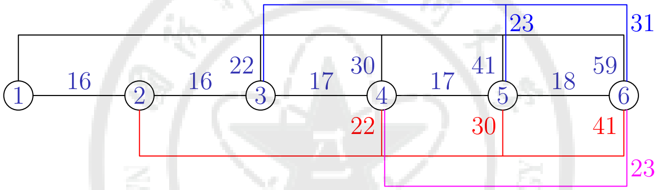
例 1 （设备更新问题）设某公司需使用某种设备一套，设备购买价格及维修费用见表。现设该公司在第一年开始时新购入一套设备，问今后 5 年的设备更新方案如何，才能使得总费用最省？

年	1	2	3	4	5
价格	11	11	12	12	13

使用年限	0-1	1-2	2-3	3-4	4-5
维修费用	5	6	8	11	18

若穷举，有 $2^4 = 16$ 种情形，但若年限更大，穷举不是办法。

建立网络模型



结点 i 表示第 i 年开始时购买一套设备，结点 6 为虚设结点。

用 p_i 表示第 i 年的购买费，m_k 表示 k 个使用年限的维修费。

令弧 (i,j) 的长度 d_{ij} 为第 i 年的购买费与 $j-i$ 年里的维修费之和，即

$$b_{ij} = p_i + \sum_{k=1}^{j-i} m_k$$

求①→⑥的最短路。

此问题无法用动态规划法求解。

Dijkstra 算法——标号法

为了算法的简便，将图改为完全图，令虚设的弧的长度为 ∞.

$T(j)$——第 j 个点的临时标号

$P(j)$——第 j 个点的永久标号，表示 $1 \to j$ 的最短路长。

Dijkstra 算法——标号法

为了算法的简便，将图改为完全图，令虚设的弧的长度为 ∞.

$T(j)$——第 j 个点的临时标号

$P(j)$——第 j 个点的永久标号，表示 $1 \to j$ 的最短路长。

基本思想：从起点 S 沿一切可能的弧派遣使者，这些使者均以相同的速度匀速前进，最早有使者到达的顶点作上记号（临时标号），记下历经的路的长度，然后从这点沿所有可能的弧再派出使者，这些使者与原来尚未到达顶点的使者一起以相同速度匀速前进，$\ldots$，重复以上过程，直到有人到达终点为止。

算法步骤:

算法步骤：

Step 1：令 $P(1)=T(1)=0, T(j)=\infty, \qquad j=2,3,\ldots,N$

算法步骤：

Step 1：令 $P(1) = T(1) = 0, T(j) = \infty, \qquad j = 2, 3, \ldots, N$

Step 2：计算 $T(j) = \min\{T(j), P(1) + b_{1j}\},$
$j = 2, 3, \ldots, N$
如果结点 j 的临时标号发生了变化，就令 $PRIOR(j) = 1$。

算法步骤：

Step 1：令 $P(1) = T(1) = 0, T(j) = \infty, \qquad j = 2, 3, \ldots, N$

Step 2：计算 $T(j) = \min\{T(j), P(1) + b_{1j}\}$, $j = 2, 3, \ldots, N$
如果结点 j 的临时标号发生了变化，就令 $PRIOR(j) = 1$。

Step 3：在所有临时标号中，取出最小的一个标号（若有多个，任取其一），改为永久性标号，即若 k 是临时标号最小的一个结点，则令 $P(k) = T(k)$。

若无临时标号点或 $k = N$，则转 Step 5；否则，转 Step 4。

Step 4：设刚获得永久性标号的点为 k，对每个具有临时标号的点 j，计算

$$T(j) = \min\{T(j), P(k) + b_{kj}\}$$

对于 $T(j)$ 发生了变化的每个结点 j，令 $PRIOR(j) = k$.然后转 Step 3。

Step 4：设刚获得永久性标号的点为 k，对每个具有临时标号的点 j，计算

$$T(j) = \min\{T(j), P(k) + b_{kj}\}$$

对于 $T(j)$ 发生了变化的每个结点 j，令 $PRIOR(j) = k$.然后转 Step 3。

Step 5：$P(N)$ 即为 $1 \to N$ 的最短路的长。
设 $PRIOR(k_i) = k_{i-1}, \quad i = 1, 2, \ldots, n; \quad k_0 = 1, k_n = N$
则最短路径为：$1 = k_0 \to k_1 \to k_2 \to \cdots \to k_n = N$.

回到例 1.

回到例 1.

① $P(1)=0, \quad T(j)=\infty, \quad j=2,3,4,5,6$

②

$$\begin{array}{ll} T(2)=0+16=16 & T(3)=0+22=22 \\ T(4)=0+30=30 & T(5)=0+41=41 \\ T(6)=0+59=59 & \end{array}$$

$$PRIOR(j)=1 \qquad j=2,3,4,5,6$$

③ 令 $P(2)=16$，即给结点 2 永久性标号为 16.

④ $T(3)=\min\{T(3),P(2)+b_{23}\}=\min\{22,16+16\}=22$

$T(4)=\min\{T(4),P(2)+b_{24}\}=\min\{30,16+22\}=30$

$T(5)=\min\{T(5),P(2)+b_{25}\}=\min\{41,16+30\}=41$

$T(6)=\min\{T(6),P(2)+b_{26}\}=\min\{59,16+41\}=57$

只有结点 6 改变了临时标号，令 $PRIOR(6)=2$.

⑤ 令 $P(3)=22$，即给结点 3 永久性标号为 22.

⑥ $T(4) = \min\{T(4), P(3) + b_{34}\} = \min\{30, 22 + 17\} = 30$

$T(5) = \min\{T(5), P(3) + b_{35}\} = \min\{41, 22 + 23\} = 41$

$T(6) = \min\{T(6), P(3) + b_{36}\} = \min\{57, 22 + 31\} = 53$

结点 6 改变了临时标号，$PRIOR(6) = 3$.

⑦ 令 $P(4) = 30$，即给结点 4 永久性标号 30.

⑧ $T(5)=\min\{T(5),P(4)+b_{45}\}=\min\{41,30+17\}=41$

$$T(6)=\min\{T(6),P(4)+b_{46}\}=\min\{53,30+23\}=53$$

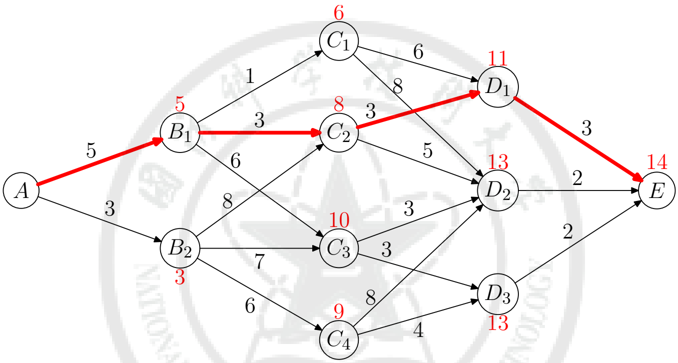
$$PRIOR(6)=3 \quad 或 \quad 4$$

⑨ 令 $P(5)=41$，即给结点 5 永久性标号 41.

⑩ $T(6)=\min\{T(6),P(5)+b_{56}\}=\min\{53,41+18\}=53$
所以 $P(6)=53$，即最短路径长为 53，最短路径为

$$1\rightarrow 3\rightarrow 6 \quad 或 \quad 1\rightarrow 4\rightarrow 6$$

再来看开始的最短路线问题的例子。



获得永久性标号的顶点顺序为：

$B2(A), B1(A), C1(B1), C2(B1), C4(B2), C3(B2),$
$D1(C2), D2(C2), D3(C3\backslash C4), E(D1)$

故最短路径为 $A \to B1 \to C2 \to D1 \to E$

Ford算法

Ford算法

动态规划法与 Dijkstra 算法，都假定弧的长度是非负的，但在某些问题中，有时会出现长度为负值的情况，可采用 Ford 算法。

Ford算法

动态规划法与 Dijkstra 算法，都假定弧的长度是非负的，但在某些问题中，有时会出现长度为负值的情况，可采用 Ford 算法。

称临时标号为未着色标号，永久性标号为着色标号。所谓 Ford 算法，只要对 Dijkstra 算法做两点改变。

1. 计算 $T(j) = \min\{T(j), P(k) + b_{kj}\}$ 时，不仅对未着色点进行，对着色点也要进行。已着色标号也可以减小。
2. 仅在所有顶点都已着色，而且下一步不能使任一顶点的标号减小时，算法才终止。

Ford算法

动态规划法与 Dijkstra 算法，都假定弧的长度是非负的，但在某些问题中，有时会出现长度为负值的情况，可采用 Ford 算法。

称临时标号为未着色标号，永久性标号为着色标号。所谓 Ford 算法，只要对 Dijkstra 算法做两点改变。

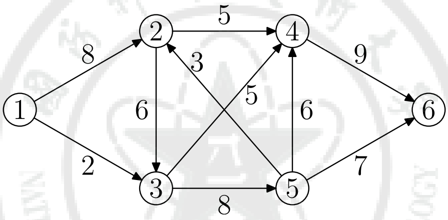
1. 计算 $T(j) = \min\{T(j), P(k) + b_{kj}\}$ 时，不仅对未着色点进行，对着色点也要进行。已着色标号也可以减小。
2. 仅在所有顶点都已着色，而且下一步不能使任一顶点的标号减小时，算法才终止。

作业 205 页 11*

最大流问题及其算法

最大流问题及其算法

最大流问题是网络理论中的一个经典问题。



有向图 $N=(V,A,C)$，$C_{ij}>0$ 为弧 (i,j) 上的容量（道路最大通过量， 管道最大流量， 送电线路最大送电量） 。假定在网络中一点 s 处有大批货物要通过网络运送到另一点 t。 s 称为网络的源，t 称为网络的汇。

一个物质运输方案，为定义在弧集合 A 上的一个函数 f

$$f_{ij} = f(i,j)$$

满足：（1）$\forall i \in V, \quad \sum_j f_{ij} = \sum_j f_{ji}$ （流平衡条件）

（2） $0 \le f_{ij} \le C_{ij}$

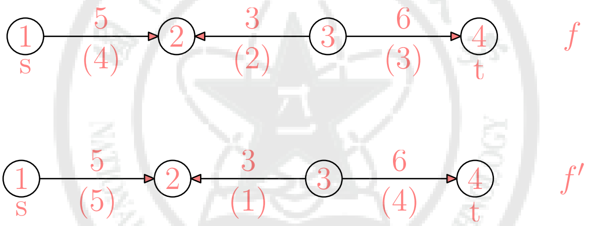
此时称 f 为一个可行流。$\sum_j f_{sj} = \sum_j f_{jt}$ 称为 f 的流量。

N 上流量最大的可行流称为 N 的最大流。

设 f 为一个可行流，若 $f_{ij} = 0$，称 (i,j) 为 f 零弧；
若 $f_{ij} = C_{ij}$， 称 (i,j) 为 f 饱和弧。

一条以 s 为起点，以 t 为终点的路，如果它的前向弧都不是 f 饱和弧，后向弧都不是 f 零弧，则称它为 f 增广路。

如果可行流 f 有增广路，那么 f 不是最大流。



f' 的流量大于 f 的流量。故得证。

若 P 是 f 增广路，记 P 的前向弧集合为 A_1，后向弧集合为 A_2.

令 $\delta_1 = \min\{C_{ij} - f_{ij} \,|(i,j) \in A_1\}$

$$\delta_2 = \min\{f_{ij} \,|(i,j) \in A_2\}$$

$$\delta = \min\{\delta_1, \delta_2\}$$

由增广路的定义知，$\delta > 0$.

定理 1 （增广路定理）设 f 是网络的可行流，则 f 是最大流，当且仅当不存在关于 f 的增广路。

定理 2 （整数定理）若弧容量 C_{ij} 都是正整数，则一定存在一个整数最大流。

最大流的 Ford-Fulkerson 算法

最大流的 Ford-Fulkerson 算法

基本步骤：

最大流的 Ford-Fulkerson 算法

基本步骤:

Step 0 ： 求出 N 的一个可行流 f.

最大流的 Ford-Fulkerson 算法

基本步骤:

Step 0： 求出 N 的一个可行流 f.

Step 1： 求一条关于 f 的增广路，转 Step 2;

若没有增广路，则 f 是最大流，停止。

最大流的 Ford-Fulkerson 算法
基本步骤:

Step 0 : 求出 N 的一个可行流 f.

Step 1 : 求一条关于 f 的增广路，转 Step 2;

若没有增广路，则 f 是最大流，停止。

Step 2 : 求出增广路的 δ 值，并对 f 进行增广得到新的可行流，转 Step 1.

根据整数定理，当 C_{ij} 都是正整数时，上述算法一定能在有限步内求出网络 N 的最大流。算法的关键是求增广路。下面用标号法求增广路。

根据整数定理，当 C_{ij} 都是正整数时，上述算法一定能在有限步内求出网络 N 的最大流。 算法的关键是求增广路。下面用标号法求增广路。

Step 0：任意求出 N 的一个可行流 $f=\{f_{ij}\}$，通常取零流。

根据整数定理，当 C_{ij} 都是正整数时，上述算法一定能在有限步内求出网络 N 的最大流。算法的关键是求增广路。下面用标号法求增广路。

Step 0：任意求出 N 的一个可行流 $f=\{f_{ij}\}$，通常取零流。

Step 1：（求 f 增广路）

<u>1.0</u> 给 s 以标号 $P_s=0$，且 $\delta(s)=\infty$，规定 s 为未检查顶点。

1.1 如果所有已标号顶点都已检查过，转 Step 3;
否则，任取一个已标号未检查顶点 v_i，检查所有与 v_i 关联的弧。

对 (i,j)，如果 v_j 未标号，且 $f_{ij} < C_{ij}$，给 v_j 标号 $P_j = i$，并令

$$\delta(j) = \min\{\delta(i), C_{ij} - f_{ij}\} \text{（弧}(i,j)\text{上可增加的流量）}$$

对 (j,i)，如果 v_j 未标号，且 $f_{ji} > 0$，给 v_j 标号 $P_j = -i$，并令

$$\delta(j) = \min\{\delta(i), f_{ji}\} \text{（弧}(j,i)\text{上可减少的流量）}$$

当所有与 v_i 关联的弧都检查完毕，称 v_i 已检查。

1.1 如果所有已标号顶点都已检查过，转 Step 3；
否则，任取一个已标号未检查顶点 v_i，检查所有与 v_i 关联的弧。

对 (i, j)，如果 v_j 未标号，且 $f_{ij} < C_{ij}$，给 v_j 标号 $P_j = i$，并令

$$\delta(j) = \min\{\delta(i), C_{ij} - f_{ij}\} \text{（弧}(i,j)\text{上可增加的流量）}$$

对 (j, i)，如果 v_j 未标号，且 $f_{ji} > 0$，给 v_j 标号 $P_j = -i$，并令

$$\delta(j) = \min\{\delta(i), f_{ji}\} \text{（弧}(j,i)\text{上可减少的流量）}$$

当所有与 v_i 关联的弧都检查完毕，称 v_i 已检查。

1.2 如果 t 已经得到标号，则一条增广路已经找到，转 Step 2；否则，返回 1.1。

Step 2: （修改增广路上的流值）

Step 2: （修改增广路上的流值）

2.0 取 $v_j = t$;

Step 2: （修改增广路上的流值）

2.0 取 $v_j = t$;

2.1 若 v_j 的标号 $P_j = 0$，v_j 就是 s，增广结束，去掉 N 上所有顶点的标号返回 Step 1； 否则，转 2.2;

Step 2: （修改增广路上的流值）

2.0 取 $v_j = t$;

2.1 若 v_j 的标号 $P_j = 0$，v_j 就是 s，增广结束，去掉 N 上所有顶点的标号返回 Step 1; 否则，转 2.2;

2.2 若 $P_j = i$， 令 $f_{ij} = f_{ij} + \delta(j), v_j = v_i$，返回 2.1.

若 $P_j = -i$，令 $f_{ji} = f_{ji} - \delta(j), v_j = v_i$，返回 2.1.

Step 2: （修改增广路上的流值）

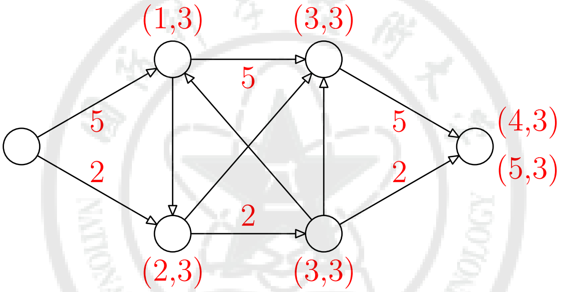
2.0 取 $v_j = t$;

2.1 若 v_j 的标号 $P_j = 0$，v_j 就是 s，增广结束，去掉 N 上所有顶点的标号返回 Step 1； 否则，转 2.2;

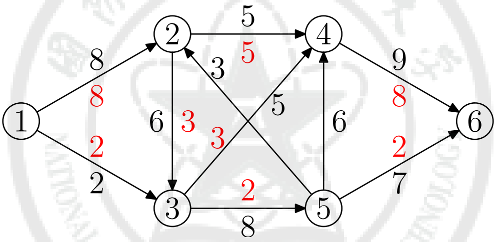
2.2 若 $P_j = i$， 令 $f_{ij} = f_{ij} + \delta(j), v_j = v_i$，返回 2.1.

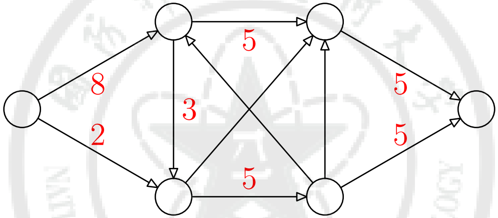
若 $P_j = -i$，令 $f_{ji} = f_{ji} - \delta(j), v_j = v_i$，返回 2.1.

Step 3: （结束） $f = \{f_{ij}\}$ 是 N 中的最大流。


(1,3)
(3,3)
5
5
5
(4,3)
2
2
(5,3)
2
(2,3)
(3,3)

标号顺序：$s \to v_2 \to v_3 \to \begin{cases} v_4 \\ v_5 \end{cases} \to t$




8
2
3
5
5
5
5

统筹模型

统筹模型

统筹方法是运筹学的重要内容。所谓统筹，就是对工业、农业、科学研究等各项实际活动，进行统一筹划，合理安排，使得预定任务能最有效的完成，例如完成最快，开支最省等。

PERT网络 —规划评审技术（Program Evaluation and Review Technique）

PERT网络 —规划评审技术（Program Evaluation and Review Technique）

一项任务通常可分成若干个独立的子任务，这些子任务称为工序。一般情况下，不同工序都存在先后顺序， 每道工序所需的时间也未必相同。可用一个有向图来描述完成任务的过程:

PERT网络 —规划评审技术（Program Evaluation and Review Technique）

一项任务通常可分成若干个独立的子任务，这些子任务称为工序。一般情况下，不同工序都存在先后顺序，每道工序所需的时间也未必相同。可用一个有向图来描述完成任务的过程：

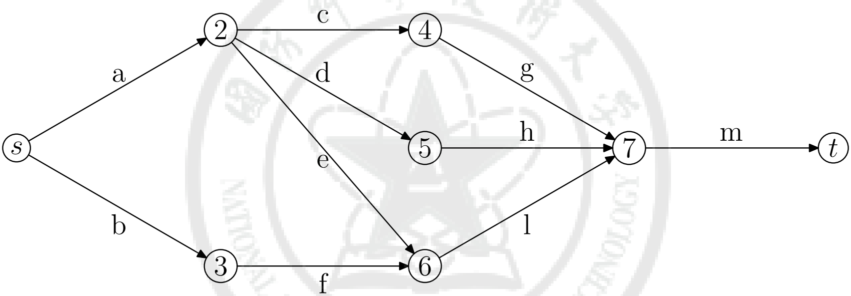
① 以一条有向边来表示一道工序，有向边上的权为此工序的（时间）长度；

② 有向边的起点与终点分别表示相应工序的开工与完工时间结点，称为事项；

③ 前一工序的完工时间即为下一工序的开工时间。

PERT 网络是有向图$G(V,E)$:
① V 中存在起始顶点 s 与终止顶点 t;
② G 中无有向回路;
③ $\forall v \in V-\{s,t\}$，v 在某条从 s 到 t 的有向道路上。

例 1　十项工序 a,b,. . ., m



为叙述方便，引入几个参数:

ET_j—结点 j 的最早可能实现时间，即

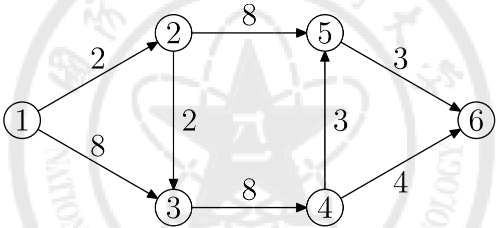
$$ET_j = \max\{ET_i + t_{ij} \,|(i,j) \in E\}$$

ES_{ij}—工序 (i,j) 的最早可能开始时间，即 $ES_{ij} = ET_i$

EF_{ij}—工序(i,j)的最早可能结束时间，即

$EF_{ij} = ES_{ij} + t_{ij} = ET_i + t_{ij}$

例 2　一任务如图，求完成整个任务的最短时间。



[解] （1）$ET_1=0$， $ES_{12}=ES_{13}=ET_1=0$，

$$EF_{12}=ET_1+t_{12}=2 \quad EF_{13}=ET_1+t_{13}=8$$

（2）$ET_2=EF_{12}=2$，

$$EF_{23}=ET_2+t_{23}=2+2=4,$$

$$EF_{25}=ET_2+t_{25}=2+8=10$$

（3）$ET_3=\max\{EF_{13},EF_{23}\}=\max\{8,4\}=8$，

$$EF_{34}=ET_3+t_{34}=8+8=16$$

（4） $ET_4 = EF_{34} = 16$,

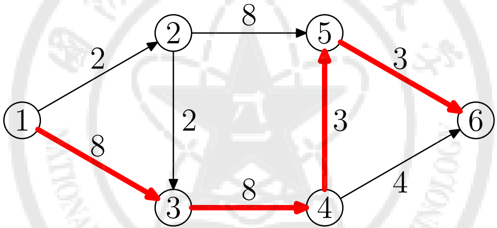
$$EF_{45} = ET_4 + t_{45} = 16+3 = 19 \quad EF_{46} = ET_4 + t_{46} = 16+4 = 20$$

（5） $ET_5 = \max\{EF_{25}, EF_{45}\} = \max\{10, 19\} = 19$

$$EF_{56} = ET_5 + t_{56} = 19 + 3 = 22$$

（6） $ET_6 = \max\{EF_{46}, EF_{56}\} = \max\{20, 22\} = 22$

完成任务最短时间为 22
最长路径为 $1 \to 3 \to 4 \to 5 \to 6$，称为**关键轨道**。



关键轨道不一定唯一。

关键轨道不一定唯一。

欲缩短工期，必须把每条关键轨道上至少一条边的长度缩短。 此方法也称为关键轨道方法 CPM (Critical Path Method)。

例 3 （上海市 91 年竞赛题）现有 14 件工件等待在一台机车上加工， 某些工件的加工必须安排在另一些工件完工以后才能开始，各工件的加工时间及先期必须完成的工件号由下表给出：

1	2	3	4	5	6	7	8	9	10	11	12	13	14
20	28	25	16	42	12	32	10	24	20	40	24	36	16
3 4	5 7 8	5 9	-	10 11	3 8 9	4	3 5 7	4	-	4 7	6 7 11	5 12	1 2 6

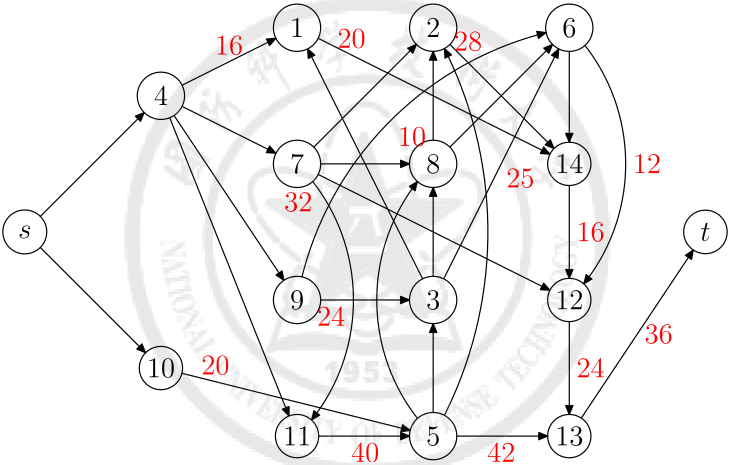
（1）若给出一个加工顺序，则确定了每个工件的完工时间（包括等待与加工两个阶段），试设计一个满足条件的加工顺序，使各个工件的完工时间之和最小。

（2）若第 j 号工件紧接着第 i 号工件完工后开工，机车需要花费的准备时间是 t_{ij}

$$t_{ij} = \begin{cases} i+j, & i<j \\ 2(i-j), & i>j \end{cases}$$

试设计一个满足条件的加工顺序，使机车花费的总时间最小。

[解]　（1）构造 PERT 图 $G(V,E)$ 如下



利用关键轨道算法 CPM 求最长关键轨道

$ET_4 = ET_S + EF_{S4} = 0,$

$EF_{41} = EF_{47} = EF_{49} = EF_{4,11} = 16$

$ET_{10} = ET_S + EF_{S,10} = 0,$

$EF_{10,5} = 20$

$ET_7 = ET_4 + EF_{47} = 16(4),$

$EF_{72} = EF_{78} = EF_{7,11} = EF_{7,12} = 32$

$ET_9 = ET_4 + EF_{49} = 16(4),$

$EF_{93} = EF_{96} = 24$

$ET_{11} = \max\{ET_4 + EF_{4,11}, ET_7 + EF_{7,11}\} =$
$\max\{16, 16+32\} = 48(7),$

$EF_{11,5} = 40,$

$ET_5 = \max\{ET_{10} + EF_{10,5}, ET_{11} + EF_{11,5}\} =$
$\max\{20, 48+40\} = 88(11)$

$EF_{52} = EF_{53} = EF_{58} = EF_{5,13} = 42$

$ET_3 = \max\{ET_9 + EF_{93}, ET_5 + EF_{53}\} =$
$\max\{16+24, 88+42\} = 130(5)$

$EF_{31} = EF_{36} = EF_{38} = 25$

$ET_1 = \max\{ET_4 + EF_{41}, ET_3 + EF_{31}\} =$
$\max\{16, 130+25\} = 155(3)$

$EF_{1,14} = 20$

$ET_8 = \max\{ET_7 + EF_{78}, ET_5 + EF_{58}, ET_3 + EF_{38}\} =$
$\max\{48, 130, 155\} = 155(3)$

$EF_{82} = EF_{86} = 10$

$ET_2 = \max\{ET_7 + EF_{72}, ET_8 + EF_{82}, ET_5 + EF_{52}\} = \max\{42, 165, 130\} = 165(8)$

$EF_{2,14} = 28$

$ET_6 = \max\{ET_9 + EF_{96}, ET_8 + EF_{86}, ET_3 + EF_{36}\} = \max\{40, 165, 155\} = 165(8)$

$EF_{6,12} = EF_{6,14} = 12$

$ET_{14} = \max\{ET_1 + EF_{1,14}, ET_2 + EF_{2,14}, ET_6 + EF_{6,14}\} = 165 + 28 = 193(2)$

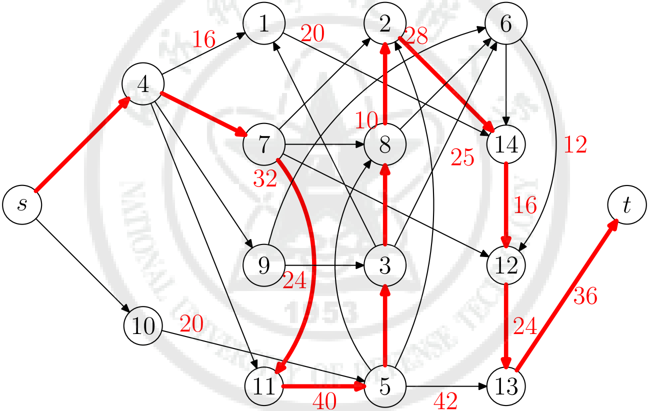
$EF_{14,12} = 16$

$ET_{12} = \max\{ET_7 + EF_{7,12}, ET_6 + EF_{6,12}, ET_{14} + EF_{14,12}\} = 193 + 16 = 209(14)$

$EF_{12,13} = 24$
$ET_{13} = \max\{ET_5 + EF_{5,13}, ET_{12} + EF_{12,13}\} = 209 + 24 = 233(12)$
$EF_{13,T} = 36$
$ET_T = 233 + 36 = 269(13)$

于是，我们利用关键轨道算法得到了一条最长关键轨道 $P(4,13)$

4 7 11 5 3 8 2 14 12 13



显然，$P(4,13)$ 上的工件加工顺序不能改变，还有 ①⑥⑨⑩ 四点要插入上述序列中。把$P(4,13)$ 截成两段：
$P_1 = 4\ 7\ 11\ 5$, $P_2 = 3\ 8\ 2\ 14\ 12\ 13$。

由于工件 10 必须在工件 5 之前加工，而工件 9 必须在工件 4 之后、工件 3 之前加工，所以工件 9，10 插入序列 P_1.而 1，6 都必须在工件 3 之后，故应插入序列 P_2.

若不考虑等待时间，则按任意允许方式插入均可，机车工作时间相同。但若考虑等待时间，显然，要保证完工时间最短，应尽量将加工时间少的工件先加工以减少等待时间。

9，10 插入序列 P_1 后的加工时间序列由小到大排列为

$$t_4\ t_{10}\ t_9\ t_7\ t_{11}\ t_5 = 16\ 20\ 24\ 32\ 40\ 42$$

1，6 插入序列 P_2 后的加工时间序列由小到大排列为

$$t_8\ t_6\ t_{14}\ t_1\ t_{12}\ t_3\ t_2\ t_{13} = 10\ 12\ 16\ 20\ 24\ 25\ 28\ 36$$

在第一个时间单调序列中，P_1 中元素的先后顺序未改变，故插入后顺序为

$$4\ 10\ 9\ 7\ 11\ 5$$

在第二个时间单调序列中，将 P_2 中元素恢复它们原有的相对位置，得到顺序为

$$3\ 8\ 6\ 1\ 2\ 14\ 12\ 13$$

将两段合并为一段，得到最终加工顺序为

$$4\ 10\ 9\ 7\ 11\ 5\ 3\ 8\ 6\ 1\ 2\ 14\ 12\ 13$$

（2）考虑加工准备时间，相当于同一个节点出发的弧上的权不再是相同的，重新求最长关键轨道。

$ET_4 = ET_S + EF_{S4} = 0,$
$EF_{41} = EF_{47} = EF_{49} = EF_{4,11} = 16$
$ET_{10} = ET_S + EF_{S,10} = 0, EF_{10,5} = 20$
$ET_7 = ET_4 + EF_{47} + t_{47} = 16 + 11 = 27(4),$
$EF_{72} = EF_{78} = EF_{7,11} = EF_{7,12} = 32$
$ET_9 = ET_4 + EF_{49} + t_{49} = 16 + 13 = 29(4),$
$EF_{93} = EF_{96} = 24$

$ET_{11} = \max\{ET_4 + EF_{4,11} + t_{4,11}, ET_7 + EF_{7,11} + t_{7,11}\} = \max\{16 + 15, 27 + 32 + 18\} = 77(7)$

$EF_{11,5} = 40$

$ET_5 = \max\{ET_{10} + EF_{10,5} + t_{10,5}, ET_{11} + EF_{11,5} + t_{11,5}\} = max\{20 + 10, 48 + 40 + 12\} = 100(11)$

$EF_{52} = EF_{53} = EF_{58} = EF_{5,13} = 42$

$ET_3 = \max\{ET_9 + EF_{93} + t_{93}, ET_5 + EF_{53} + t_{53}\} = \max\{16 + 24 + 12, 88 + 42 + 8\} = 138(5)$

$EF_{31} = EF_{36} = EF_{38} = 25, \ldots\ldots$

对此题数据而言，最长关键轨道不变，仍为

$$4\ 7\ 11\ 5\ 3\ 8\ 2\ 14\ 12\ 13$$

仍将该轨道分为两段： $P_1 = 4\ 7\ 11\ 5$， $P_2 = 3\ 8\ 2\ 14\ 12\ 13$
将 9,10 插入序列 P_1，共有 19 种排列，计算各种排列下的加工准备时间之和，得到加工准备时间最少的一种排列为

$$P_1^* = 4\ 7\ 11\ 10\ 9\ 5$$

将 1,6 插入序列 P_2，共有 8 种排列，加工准备时间最少的一种排列为

$$P_2^* = 3\ 8\ 6\ 2\ 1\ 14\ 12\ 13$$

故考虑加工准备时间的最终加工顺序为

$$4\ 7\ 11\ 10\ 9\ 5\ 3\ 8\ 6\ 2\ 1\ 14\ 12\ 13$$

作业 P207 13*