

《软件开发过程与项目管理》

Software Development Process and Project Management

任课教师： 范 国 祥

电 话： 0451-86418876-811(O)

13199561265(Mobile)

邮 箱： fgx@hit.edu.cn

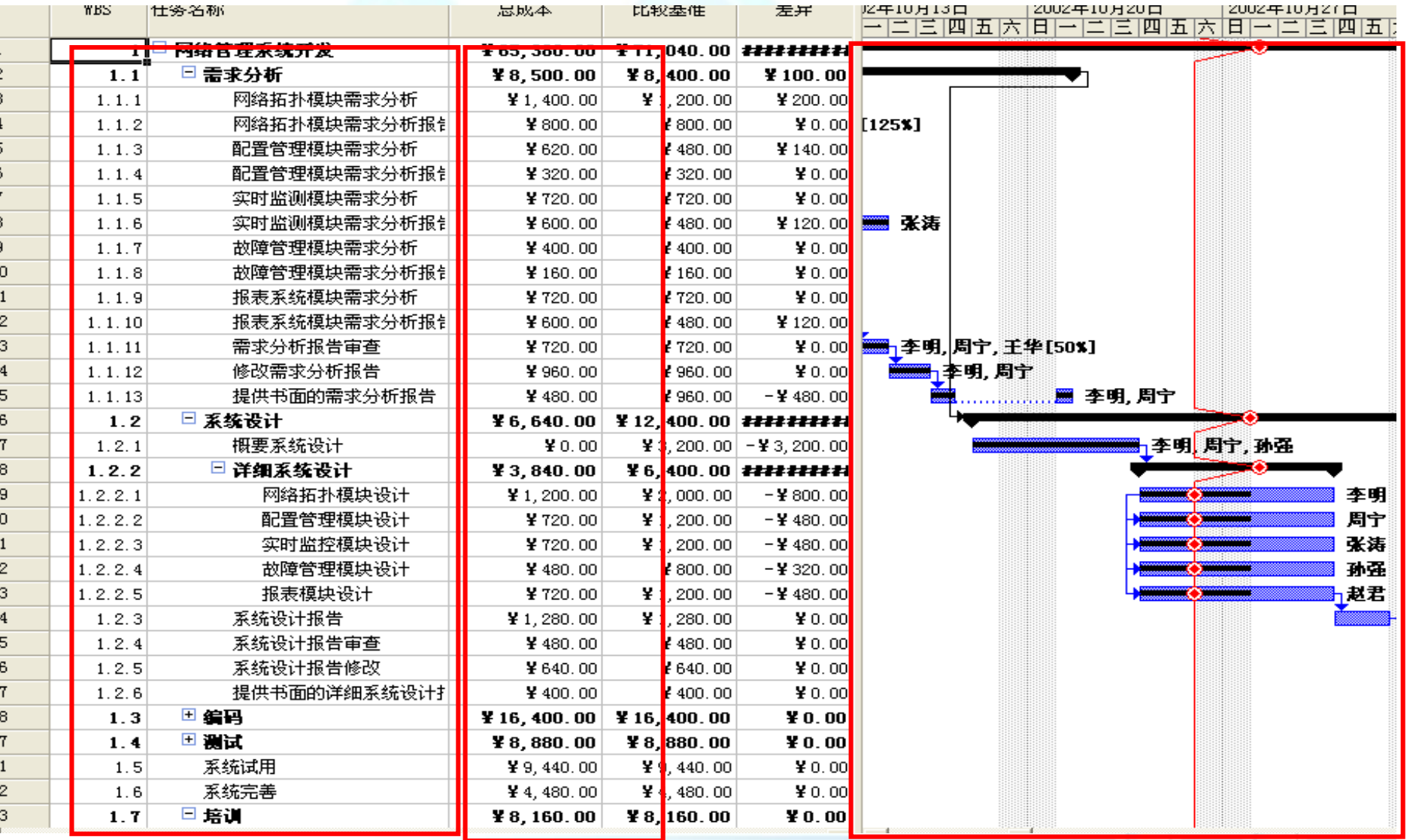
哈工大计算学部

国家示范性软件学院

软件工程教研室

2023. 10

情景引入：进度安排



进度计划的重要性

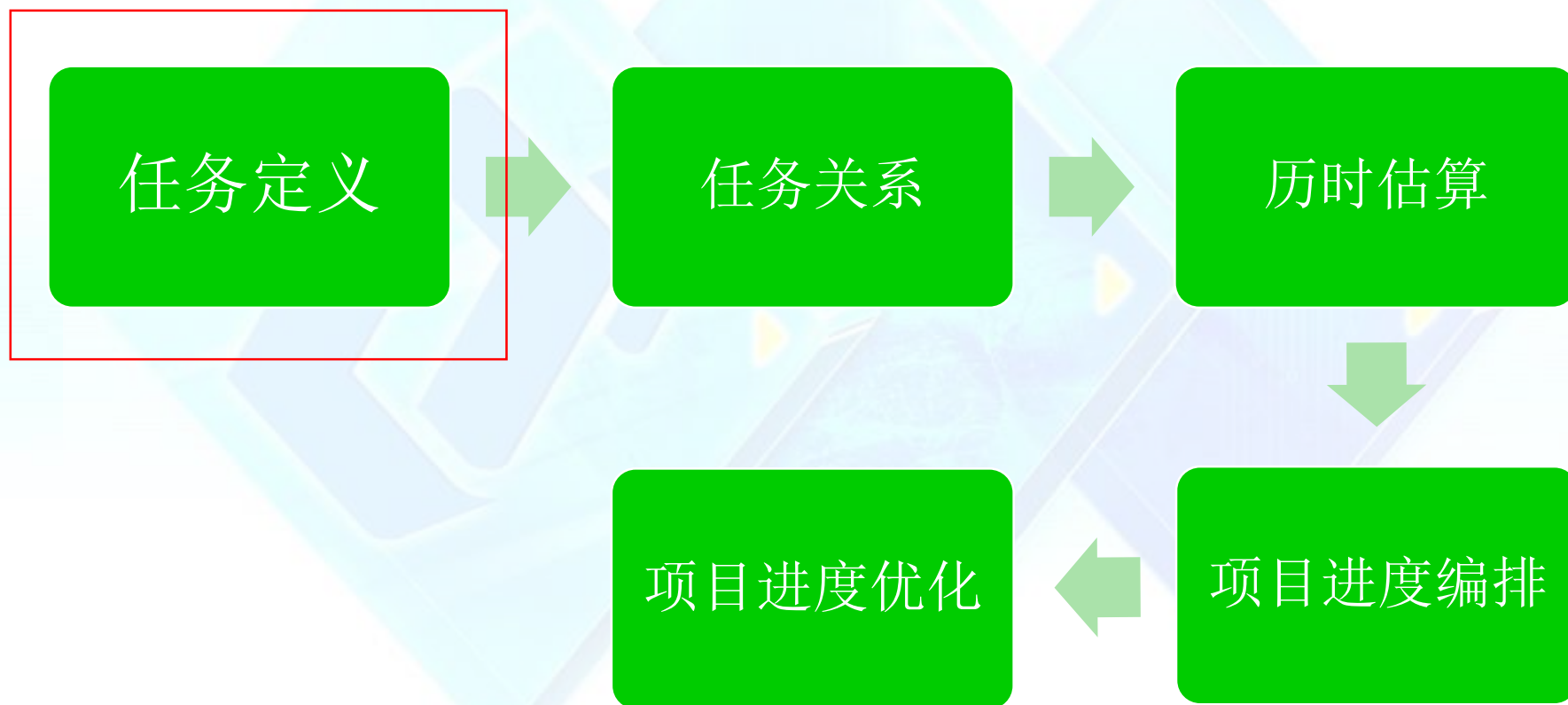
- 按时完成项目是项目经理最大的挑战之一
- 时间是项目规划中灵活性最小的因素
- 进度问题是项目冲突的主要原因

进度的定义

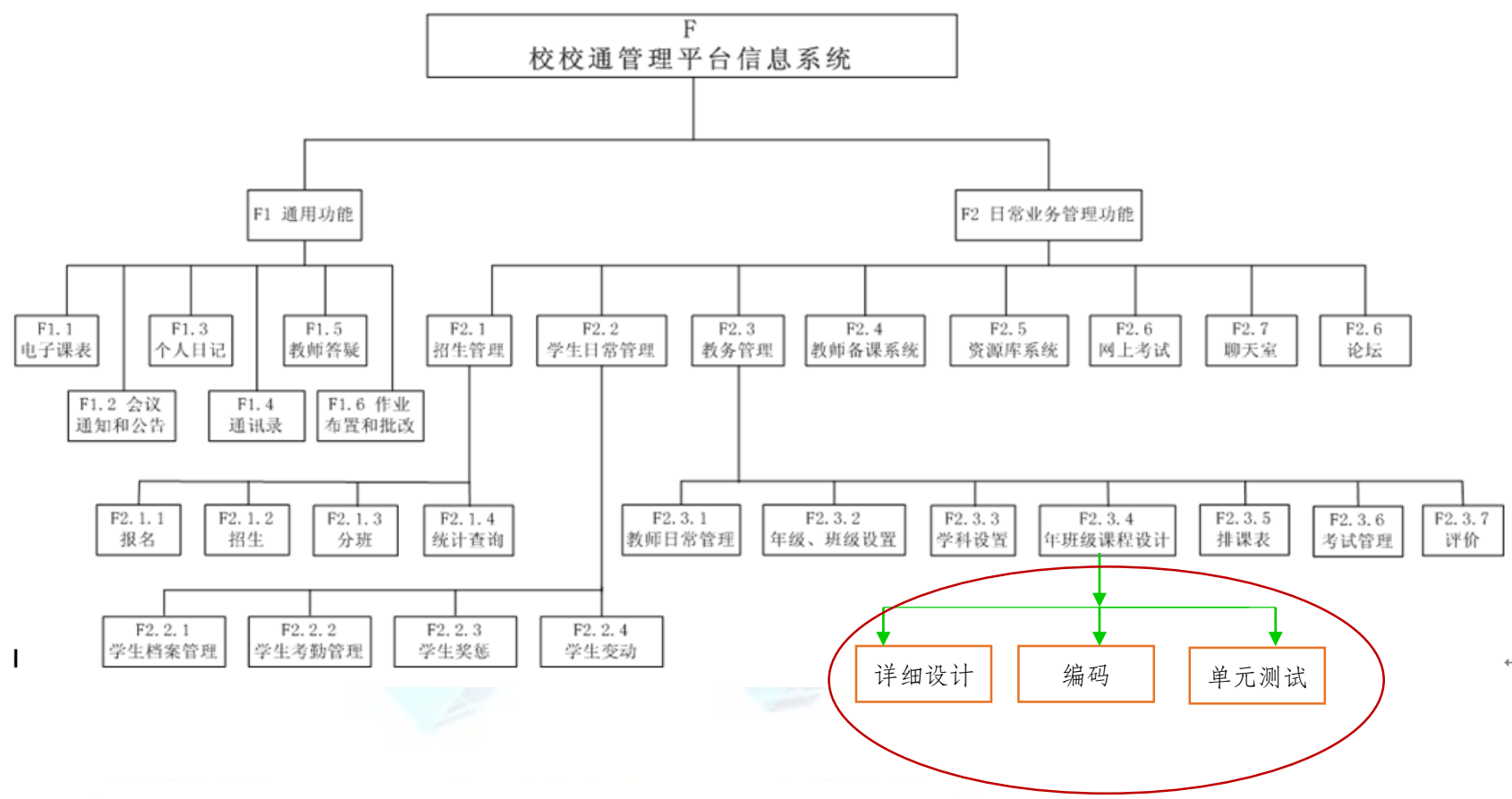
► 进度是对执行的活动和里程碑制定的工作计划日期表



项目进度计划过程



WBS → 任务



任务定义 (Defining Activities)

➤ 确定为完成项目的各个交付成果所必须进行的诸项具体活动

项目进度计划过程



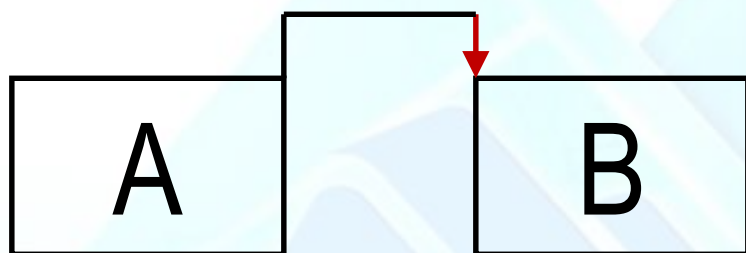
项目任务的关联关系

- 项目各项活动之间存在一定的关联关系
- 根据这些关系安排任务之间的顺序

前置活动（任务）

后置活动（任务）

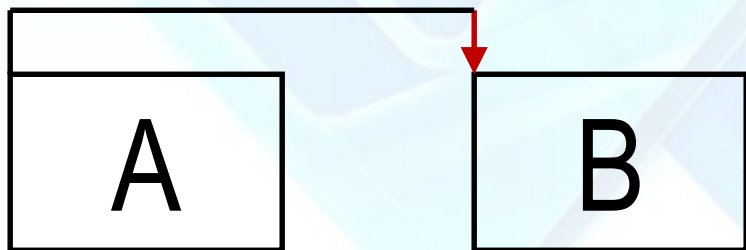
项目任务的关联关系



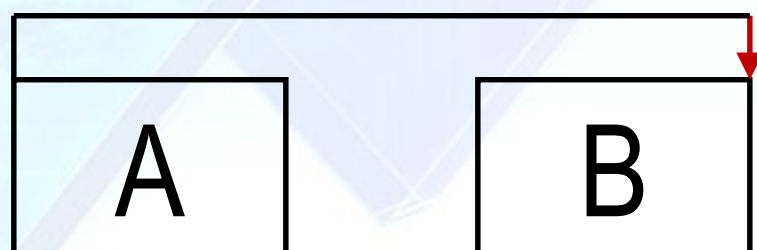
结束-开始



结束-结束



开始-开始



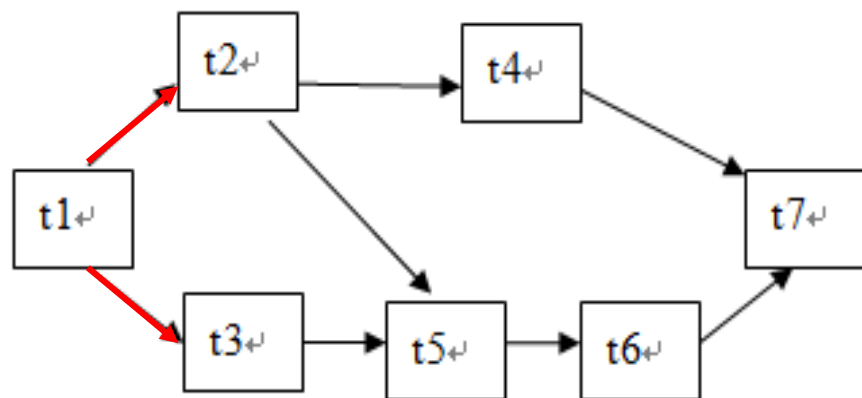
开始-结束

关系依赖矩阵

$$\begin{vmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{vmatrix}$$

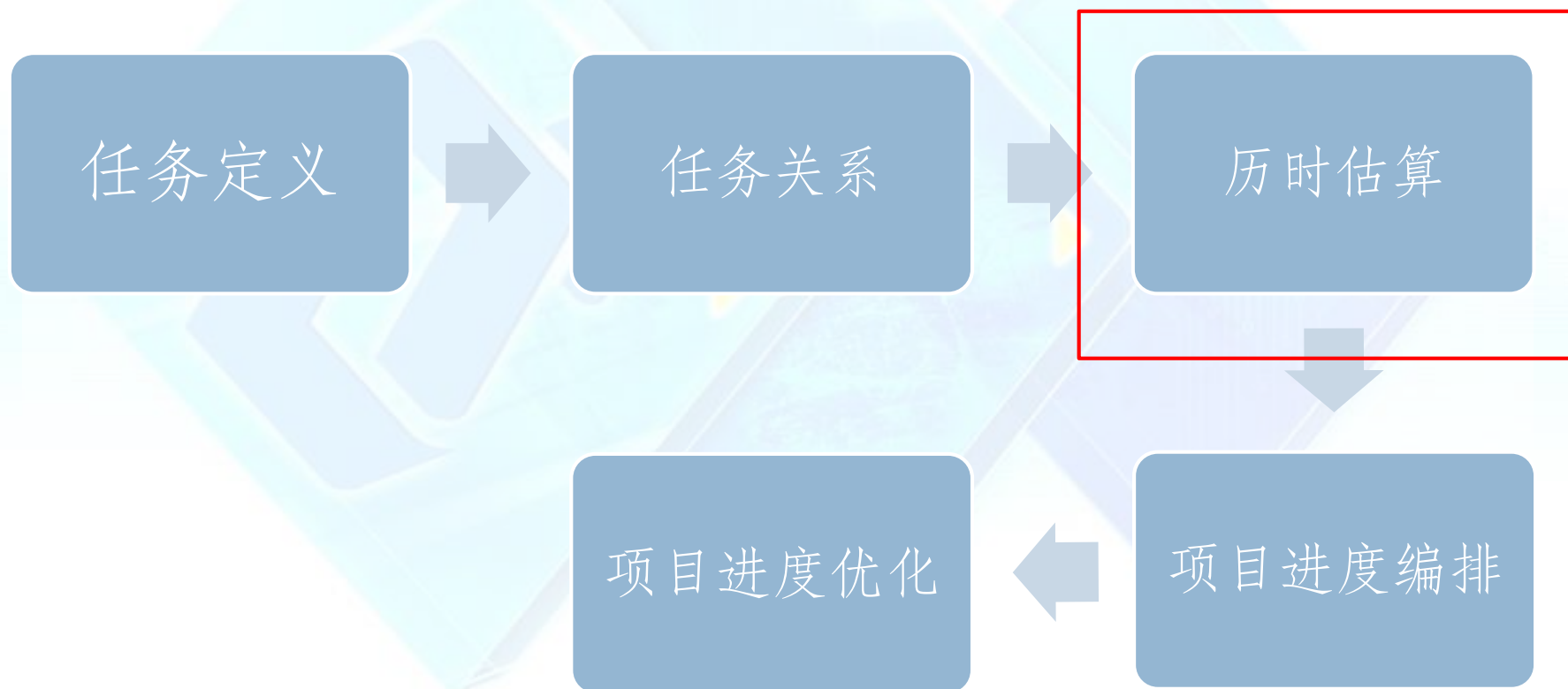
如果 d_i 是 d_j 的前置, 则 $d_{ij} = 1$, 否则 $d_{ij} = 0$

关系依赖矩阵：实例



0	1	1	0	0	0	0
0	0	0	1	1	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0

项目进度计划过程



历时估算

➤ 估计任务、路径、项目的持续时间

历时估算的基本方法

- 定额估算法
- 经验导出模型
- CPM(关键路径法估计)
- 类比估算
- 专家判断
- 基于承诺的估计

定额估算法

$$T=Q/(R*S)$$

- T: 活动历时
- Q: 任务工作量
- R: 人力数量
- S: 工作效率(贡献率)

已知: $Q=6$ 人天
 $R=2$ 人
 $S=0.5(50\%)$
则: $T=6$ 天

已知: $Q=6$ 人天
 $R=2$ 人
 $S=1(100\%)$
则: $T=3$ 天

经验导出模型

$$D = a * E^b$$

- D: 进度(以月单位)
- E: 工作量(以人月单位)
- a: 2-4之间
- b: 1/3左右: 依赖于项目的自然属性

工作量 → 历时估算

举例:

假设: 导出模型 $D = 3 * E^{1/3}$

则: $E = 65$ 人月 $\rightarrow D = 3 * 65^{1/3} = 12$ 月

关键路径法估计

- 确定项目网络图
- 每个任务有单一的历时估算
- 确定网络图中任务的逻辑关系
- 关键路径是网络图中最长的路径
- 关键路径可以确定项目完成时间

专家判断

根据下面专业知识而做出的历时估算：

- 进度计划
- 有关估算经验
- 学科或应用知识

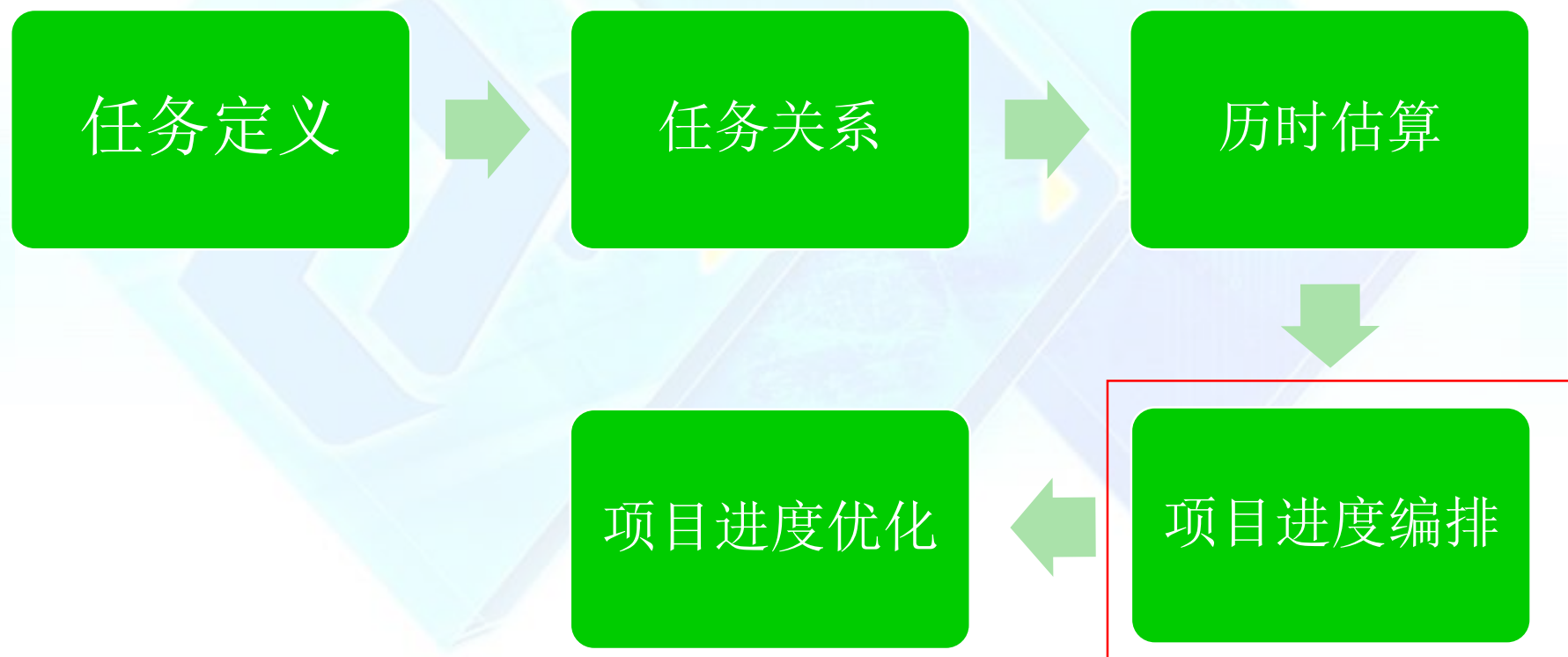
基于承诺的进度估算

要求开发人员做出进度承诺
不进行中间的工作量（规模）估计

主要优点

- 有利于开发者对进度的关注

项目进度计划过程

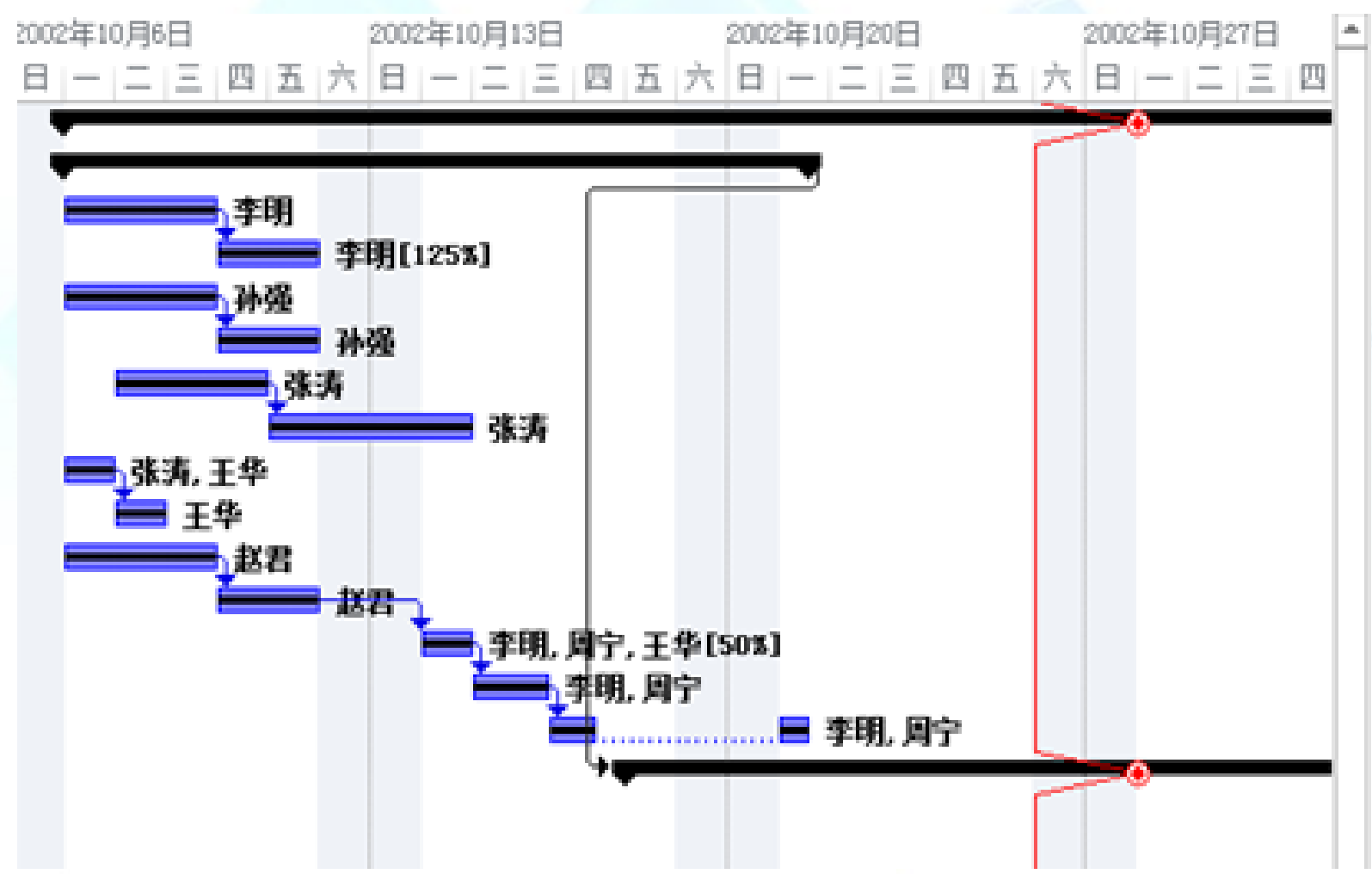


项目进度管理图示

根据项目任务的执行排序、历时估算及所需资源等进行分析，制定计划

进度制定的主要工具：甘特图 里程碑计划 网络图

项目进度管理图示：甘特图



项目进度管理图示：网络图

- 网络图是活动排序的一个输出
- 展示项目中各个活动以及活动之间的逻辑关系

项目进度管理图示：网络图

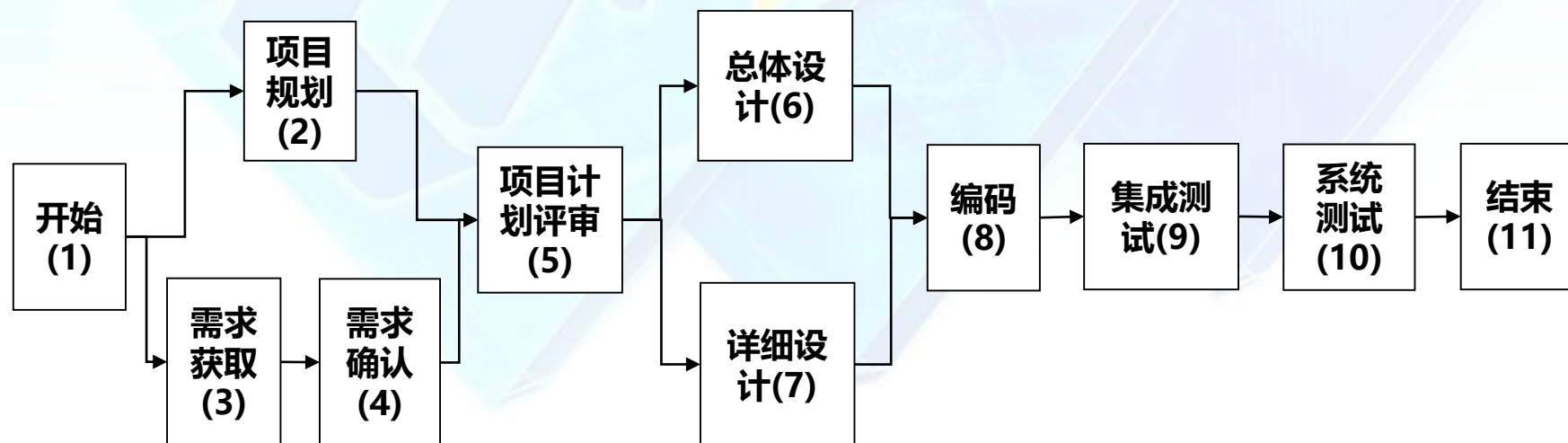
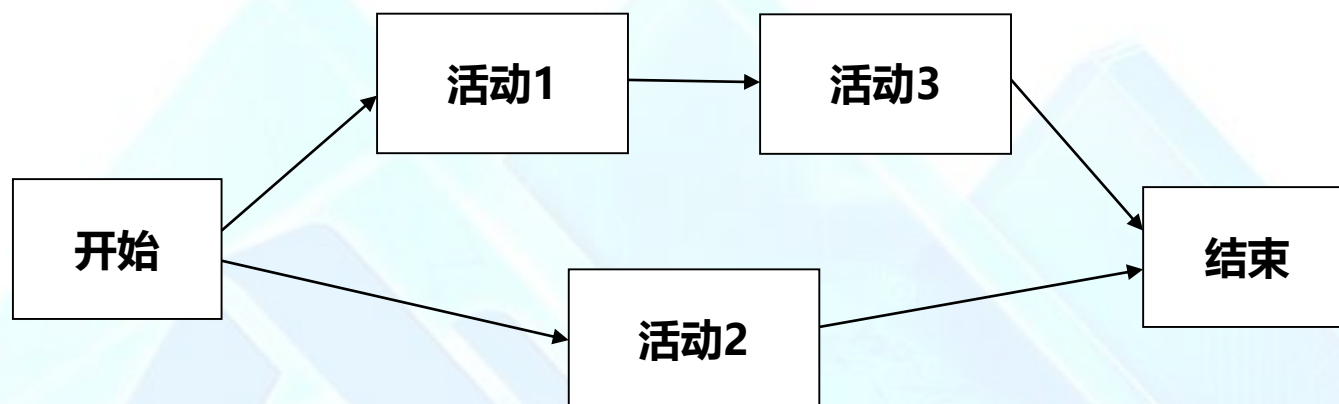
**PDM (Precedence Diagramming
Method)**
优先图法,节点法(单代号)网络图

**ADM (Arrow Diagramming
Method)**
箭线法(双代号)网络图

项目进度管理图示：网络图PDM(Precedence Diagramming Method优先图法/节点法/单代号网络图)

- 构成PDM网络图的基本元素是节点(Box)
- 节点(Box)表示活动（任务）
- 用箭线表示各活动(任务)之间的逻辑关系.
- 可以方便的表示活动之间的各种逻辑关系

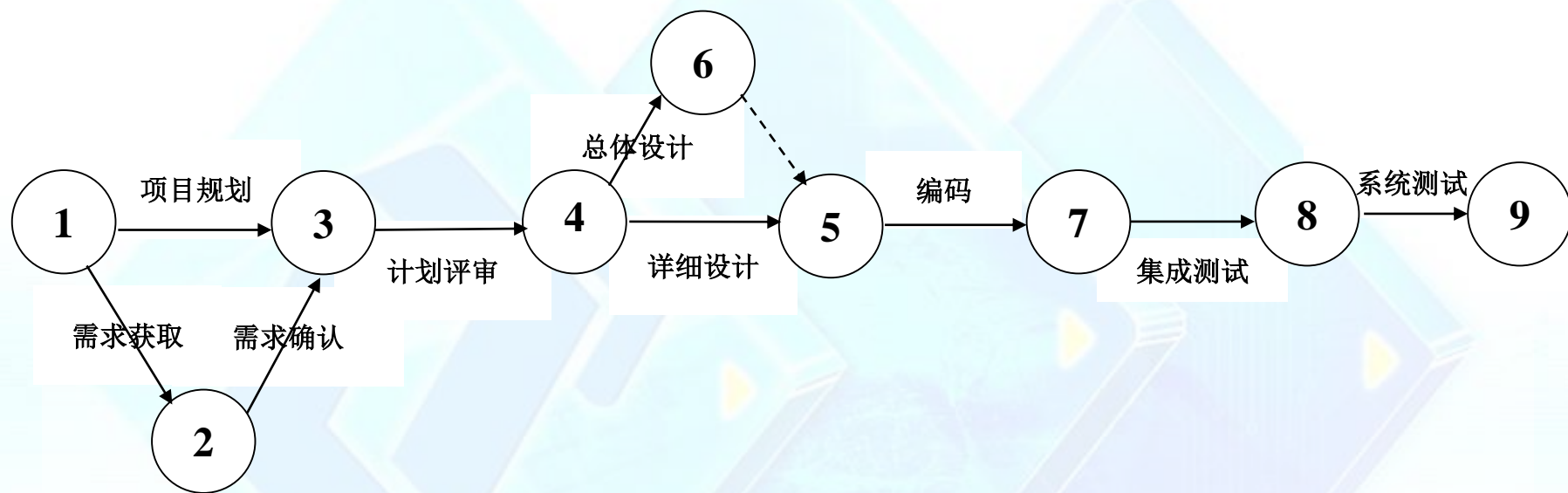
PDM 图例



项目进度管理图示：网络图ADM（Arrow Diagramming Method箭线法/双代号网络图）

- ADM也称为双代号项目网络图，
- 在ADM网络图中，箭线表示活动(任务)
- 两个代号唯一确定一个任务
- 代号表示前一任务的结束，同时也表示后一任务的开始

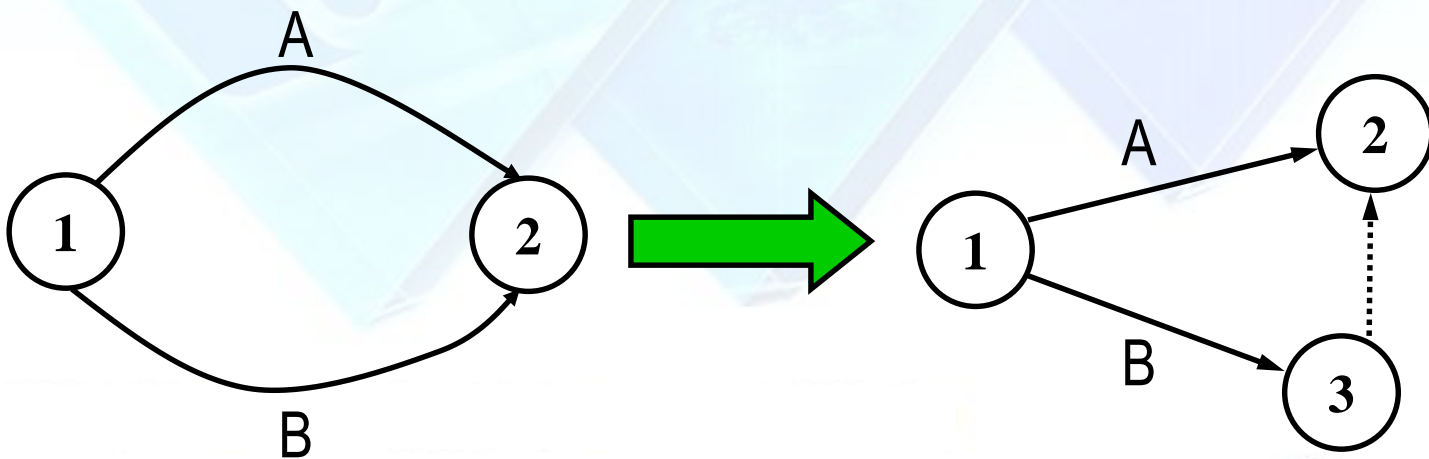
ADM 图例



ADM图例-虚活动

虚活动

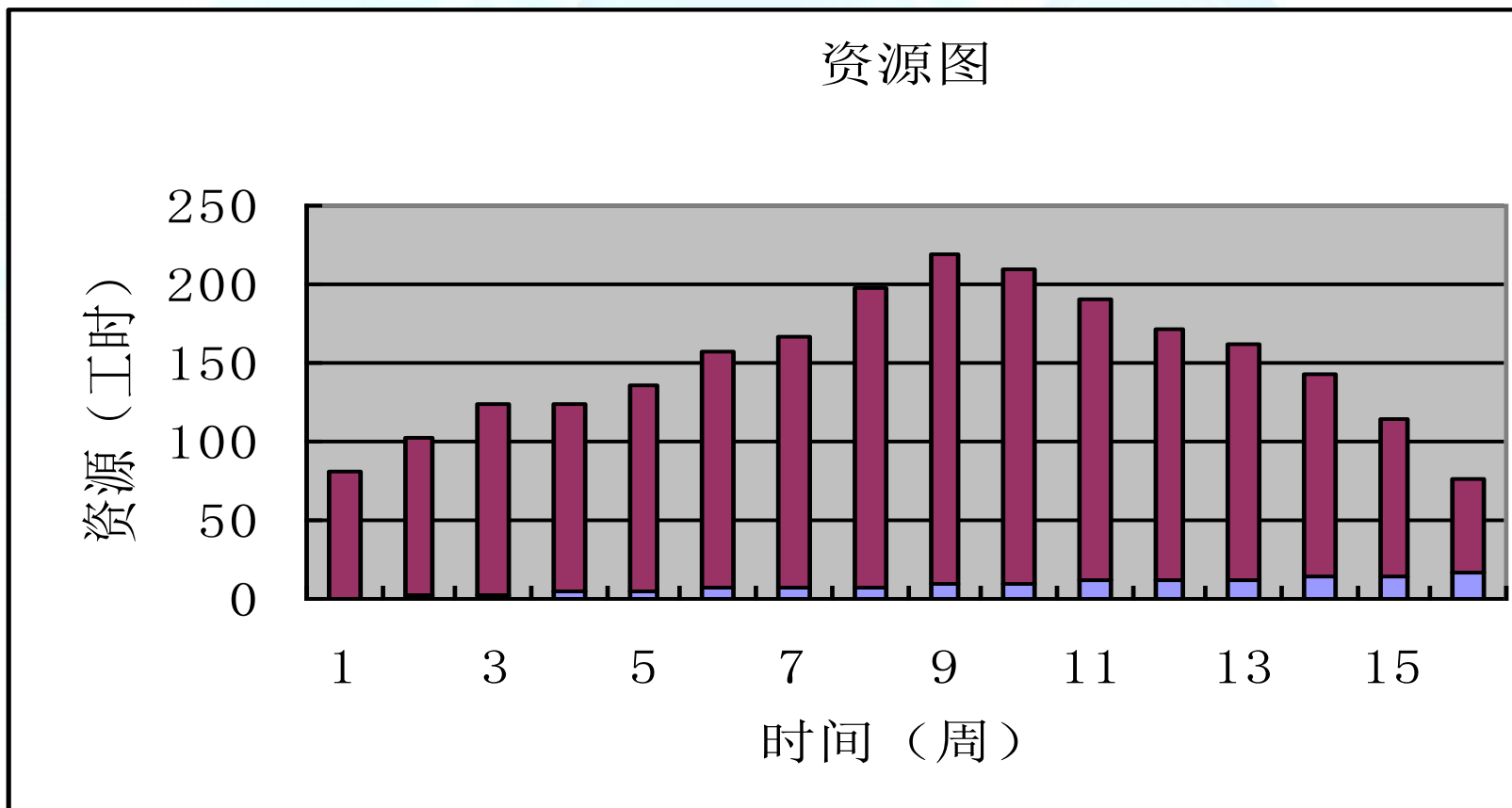
- 为了定义活动
- 为了表示逻辑关系
- 不消耗资源



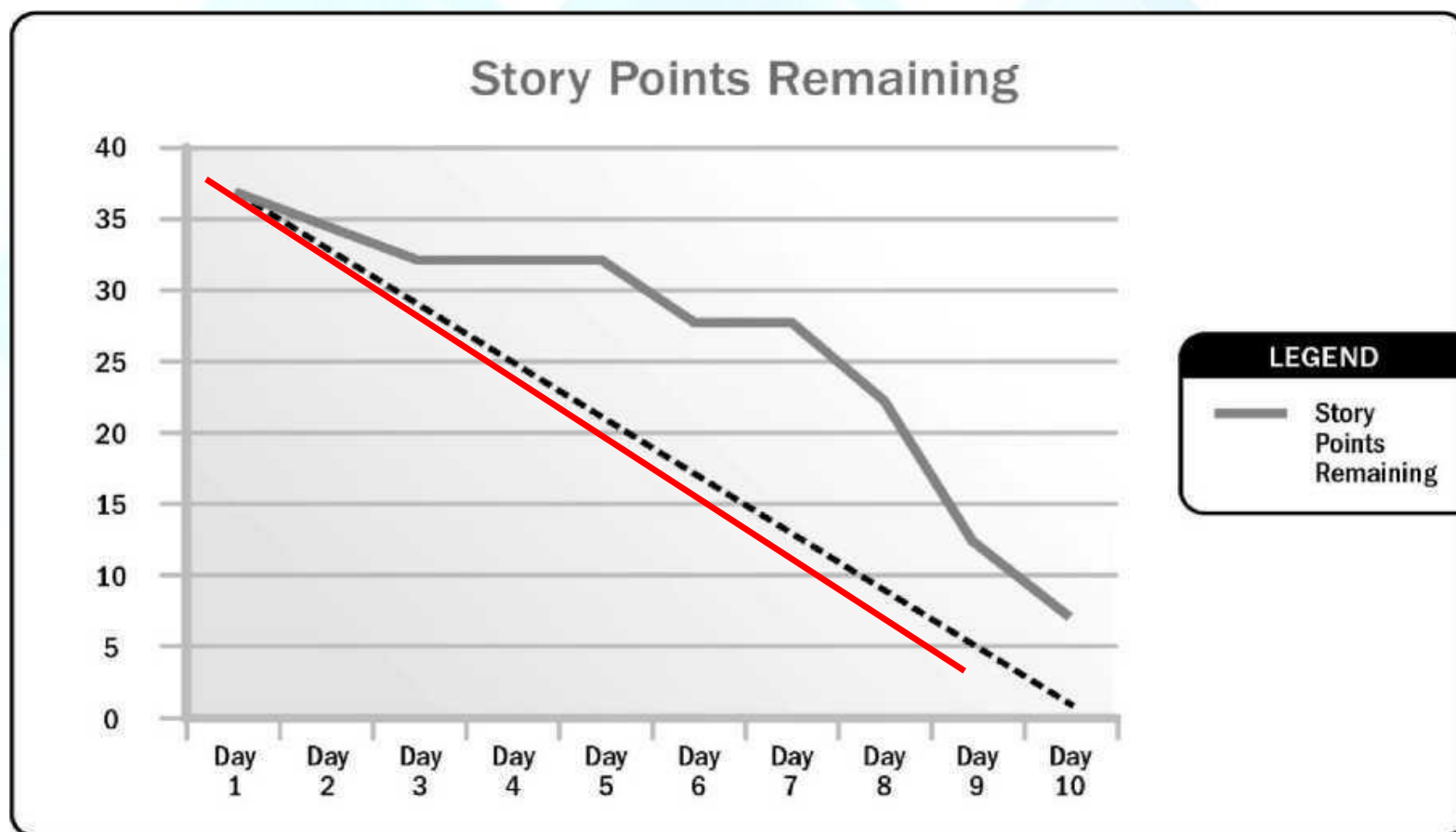
项目进度管理图示：里程碑图示



项目进度管理图示：资源图（人力、设备等）



项目进度管理图示：燃尽图(Burndown Chart)



chapter__7

项目进度管理图示：燃起图(Burnup Chart)

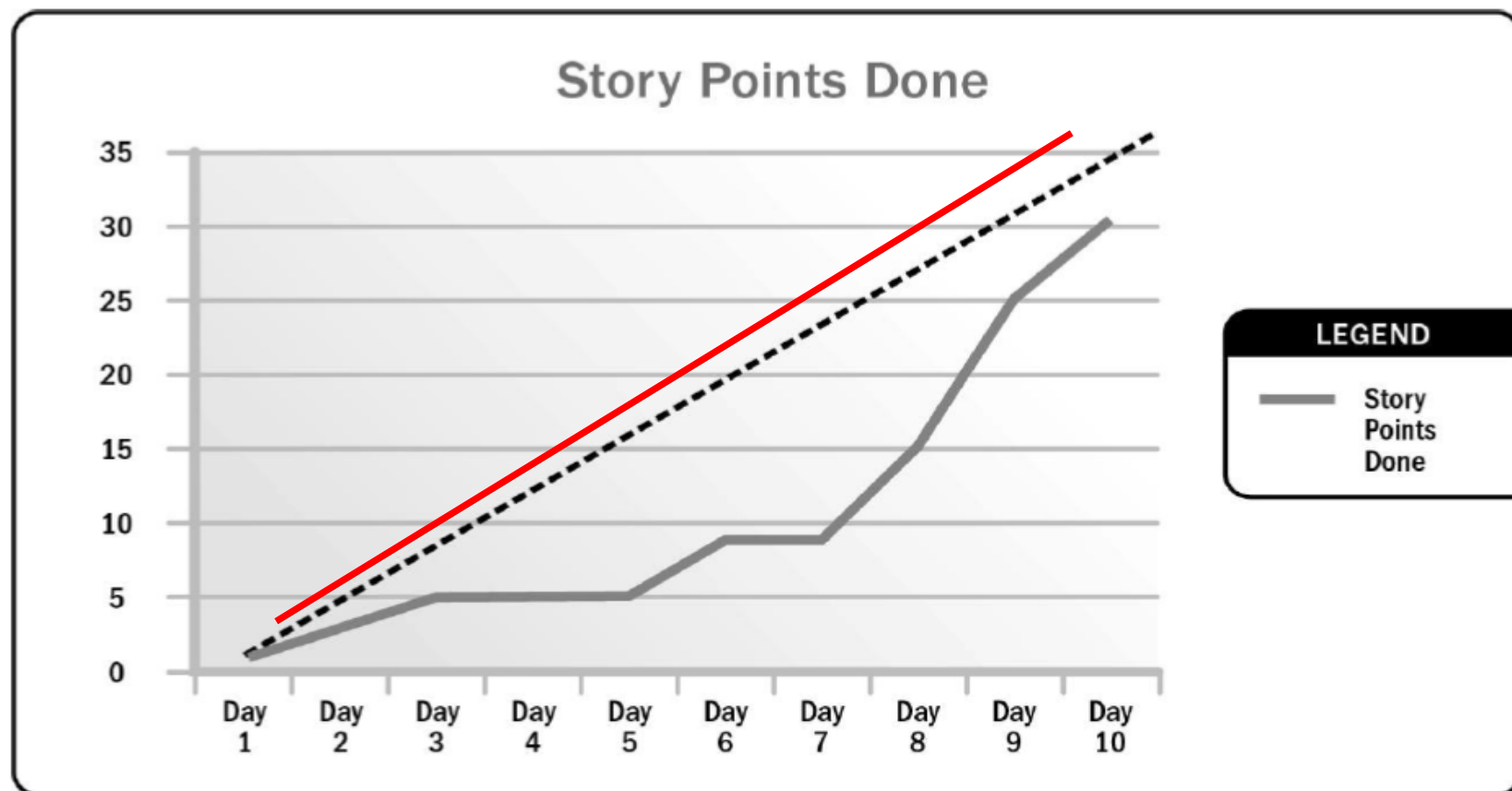
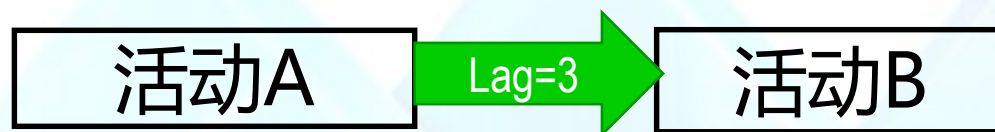


图5-2显示已完成故事点的燃起图

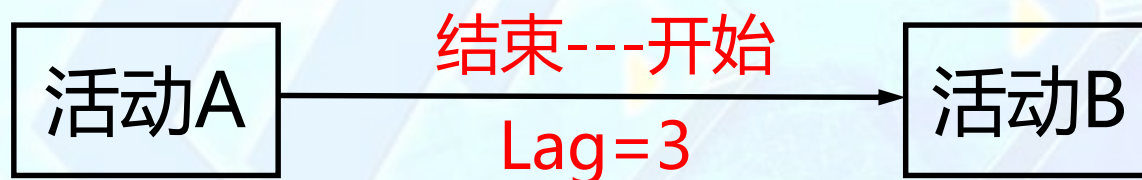
项目进度计划编排

- 超前(Lead)与滞后(Lag)
- 关键路径法
- 时间压缩法
- 资源优化

项目进度计划编排：任务滞后(lag)



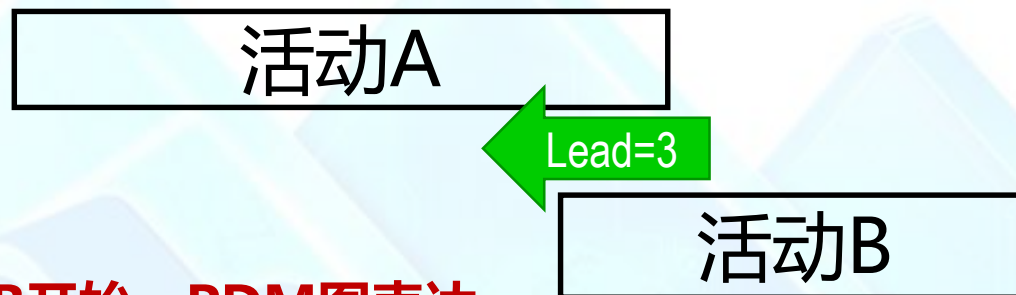
A完成之后3天B开始，PDM图表达



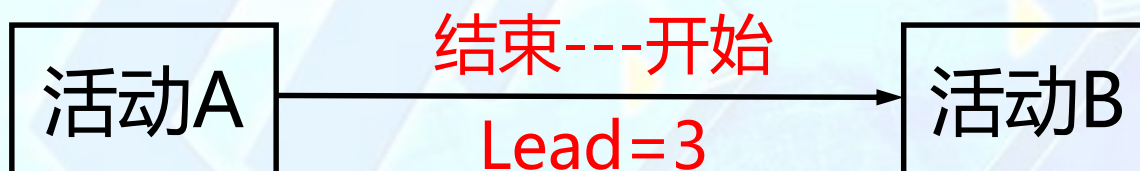
作用：

- 1) 人员休整
- 2) 等待试用验证

项目进度计划编排：任务超前(lead)



A完成之前3天B开始，PDM图表达



作用：

- 1) 解决任务的搭接
- 2) 对任务可以进行合理的拆分
- 3) 缩短项目工期

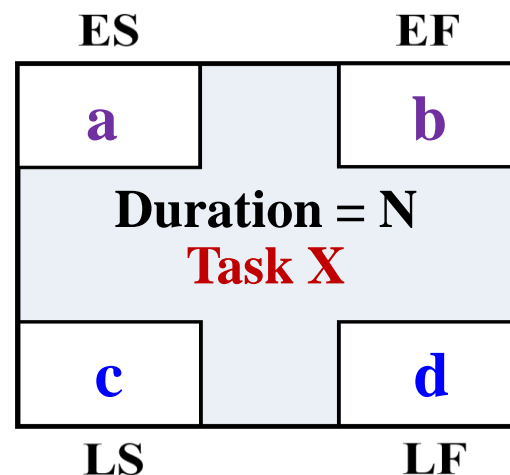
项目进度计划编排：关键路径法(CPM, Critical Path Method)

项目进度计划编排：针对项目开发中的所有活动进行合理的时间安排

□其核心对象是**任务/活动**，以及与其息息相关的任务间关系网络、任务历时估计、资源需求、相关责任人、其他假设条件等

关于“任务/活动”属性的基本概念

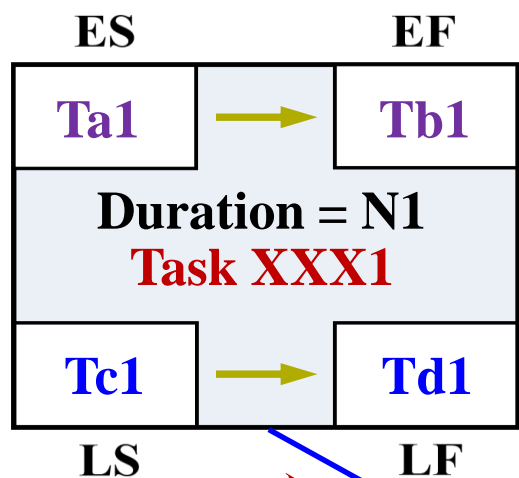
- 活动名称 (**Task**)、历时时间(**Duration**)
- 最早开始时间(Early Start, **ES**)
- 最晚开始时间(Late Start, **LS**)
- 最早完成时间(Early Finish, **EF**)
- 最晚完成时间(Late Finish, **LF**)
- 总浮动时间(Total Float, **TF**)
- 自由浮动时间(Free Float, **FF**)
- 调整超前量(**lead**)、调整滞后量(**lag**)
- 前置活动(Predecessor, **p**)、后置活动(Successor, **s**)



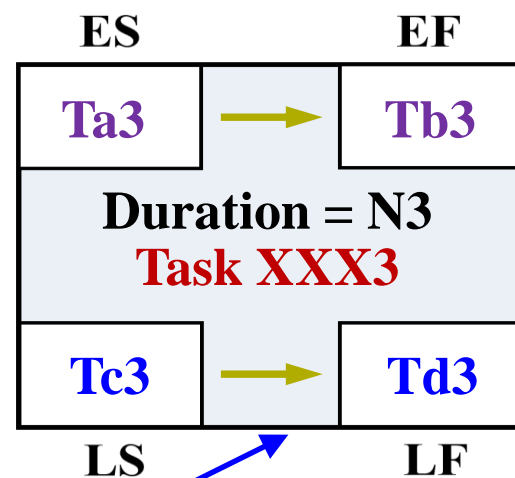
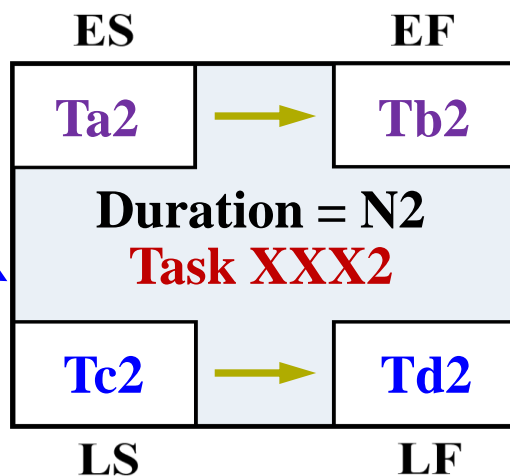
关于“任务/活动”属性的基本概念

- 活动名称 (Task)
- 历时时间 (Duration)
- 最早开始时间 (Early Start, ES)
- 最早完成时间 (Early Finish, EF)
- 最迟开始时间 (Late Start, LS)
- 最迟完成时间 (Late Finish, LF)

- 最晚完成时间 (Late Finish, LF)
- 总浮动时间 (Total Float, TF)
- 自由浮动时间 (Free Float, FF)
- 调整超前量 (lead)、调整滞后量 (lag)
- 前置活动 (Predecessor, p)
- 后置活动 (Successor, s)



Lead = t-lead

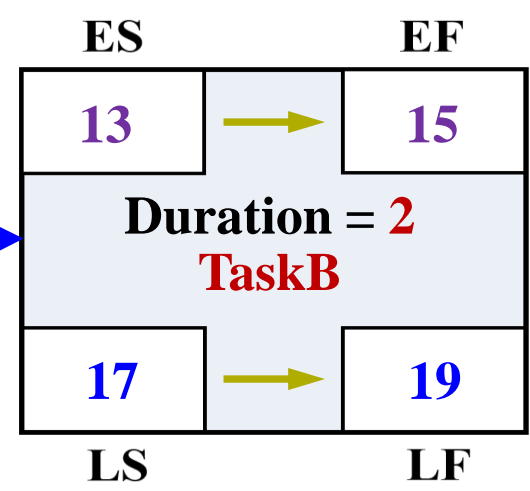
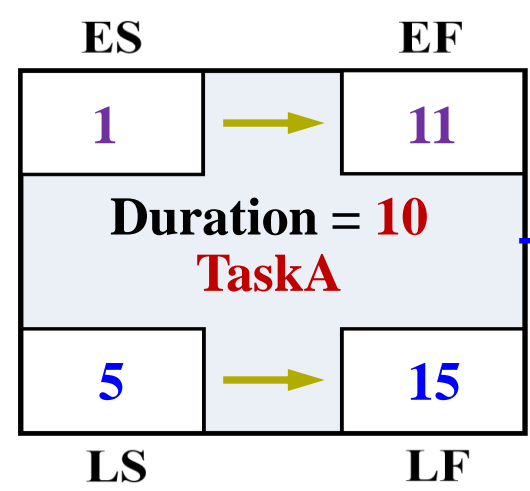


Lag = t-lag

浮动时间(Float)

- **浮动时间(Float)**: 是一个任务/活动的机动性时间, 它是一个任务/活动在不影响其它任务/活动或者项目完成的情况下可以延迟的时间量
- **总浮动时间(Total Float, TF)**: 在不影响**项目最早完成时间**的前提下, 一个任务/活动可以延迟的时间
- **自由浮动时间(Free Float, FF)**: 在不影响**后置任务/活动最早开始时间**的前提下, 一个任务/活动可以延迟的时间

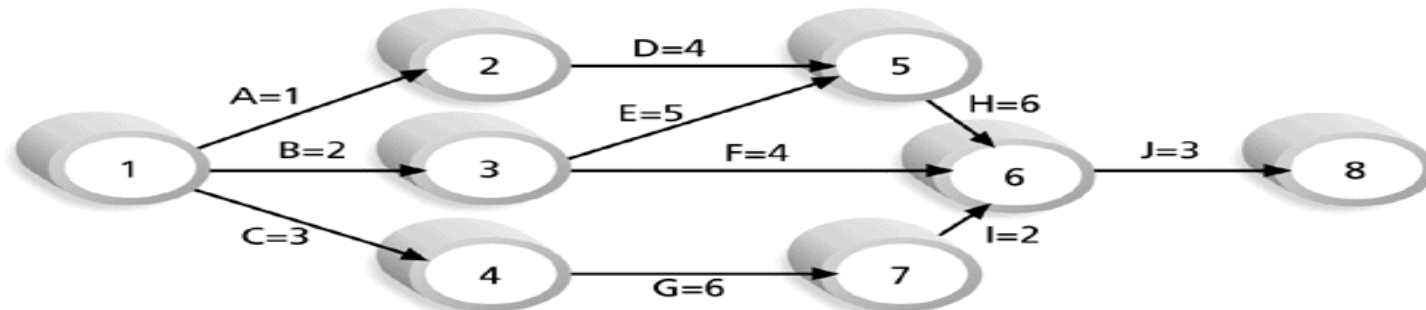
TF = 4
FF = 2



关键路径 (Critical Path)

在一个项目中表达任务/活动关系的网络图 (PDM、ADM) 中:

- 网络图中时间最长的路径称为“关键路径”
- 关键路径是决定项目完成的最短时间
- 时间浮动为0(Float=0)的路径
- 关键路径上任何活动延迟, 都会导致整个项目完成时间的延迟
- 关键路径可能不止一条(长度一定一样!)



Note: Assume all durations are in days.

Path 1:	A-D-H-J	Length = 1+4+6+3 = 14 days
Path 2:	B-E-H-J	Length = 2+5+6+3 = 16 days
Path 3:	B-F-J	Length = 2+4+3 = 9 days
Path 4:	C-G-I-J	Length = 3+6+2+3 = 14 days

Since the critical path is the longest path through the network diagram, Path 2, B-E-H-J, is the critical path for Project X.



关键路径历时估算法

对一个项目进度计划编排，关键问题是要确定项目完成总时间长度

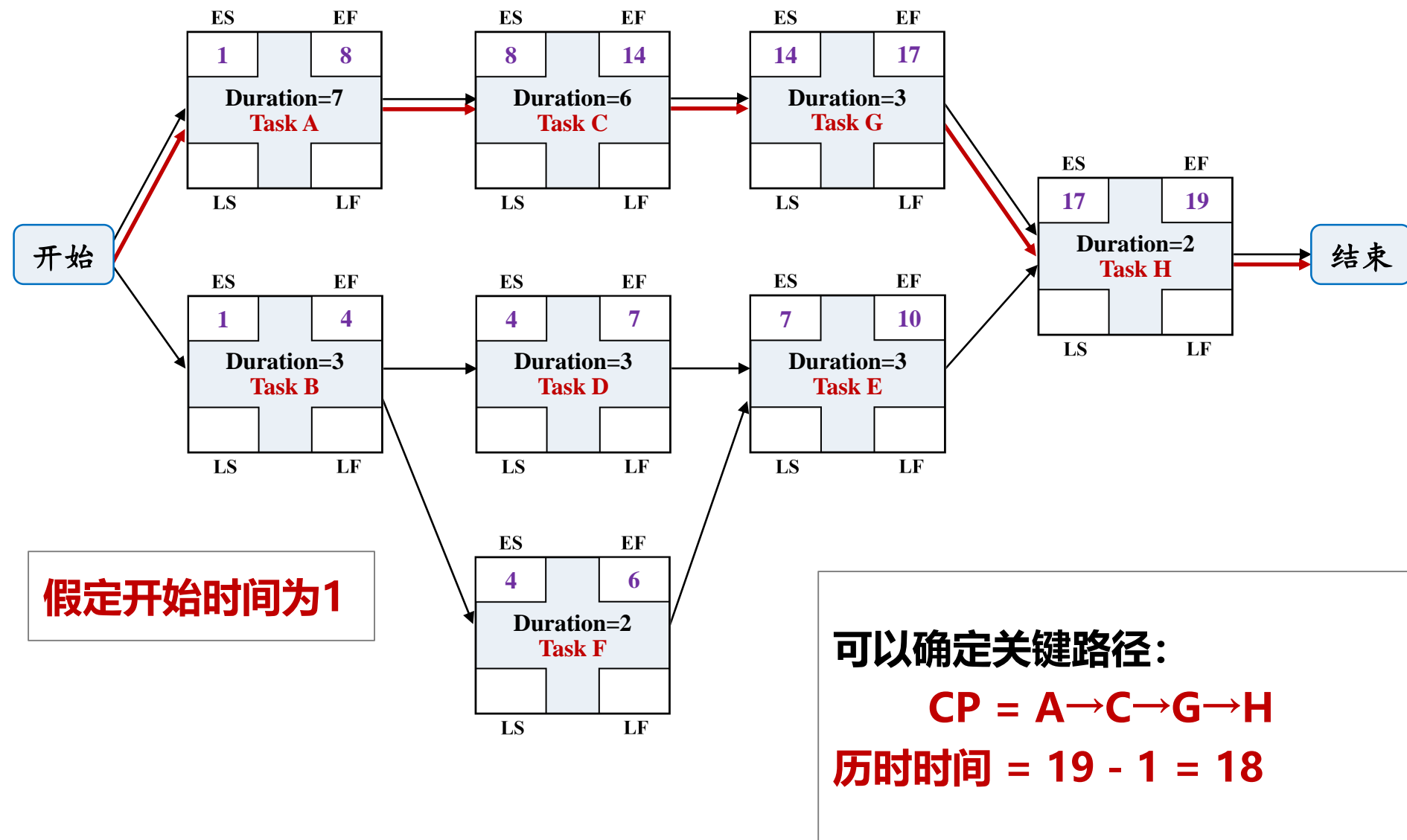
- **针对任务分解结果及活动网络图，找出“关键路径”，即抓住了进度管理的关键点**
- **关键路径检索有2种方法：正推法(Forward pass)和逆推法(Backward pass)**

关键路径历时估算法 — 正推法(Forward pass)

正推法(Forward pass): 按照时间顺序计算网络图中每个活动的**最早开始时间ES**和**最早完成时间EF**，从而找到关键路径的方法

- 首先确定项目的开始时间，网络图中第一个任务的最早开始时间是项目的开始时间，逐个对网络中每个任务节点，计算ES、EF
- $EF = ES + \text{Duration}$
- $ES = EF(p) + \text{Lag} - \text{Lead}$
 - p为前置任务节点，Lag为前置节点与本节点相关的滞后量，Lead为前置节点允许当前节点的超前量
 - 当一个任务有多个前置任务时，选择前置任务中最 $EF(p) + \text{Lag} - \text{Lead}$ 结果最大值作为当前节点的ES
- 遍历网络图，计算每个路径的所有任务的ES和EF
- 最后，根据EF最大值的节点为线索，确定关键路径

关键路径历时估算法 — 正推法实例

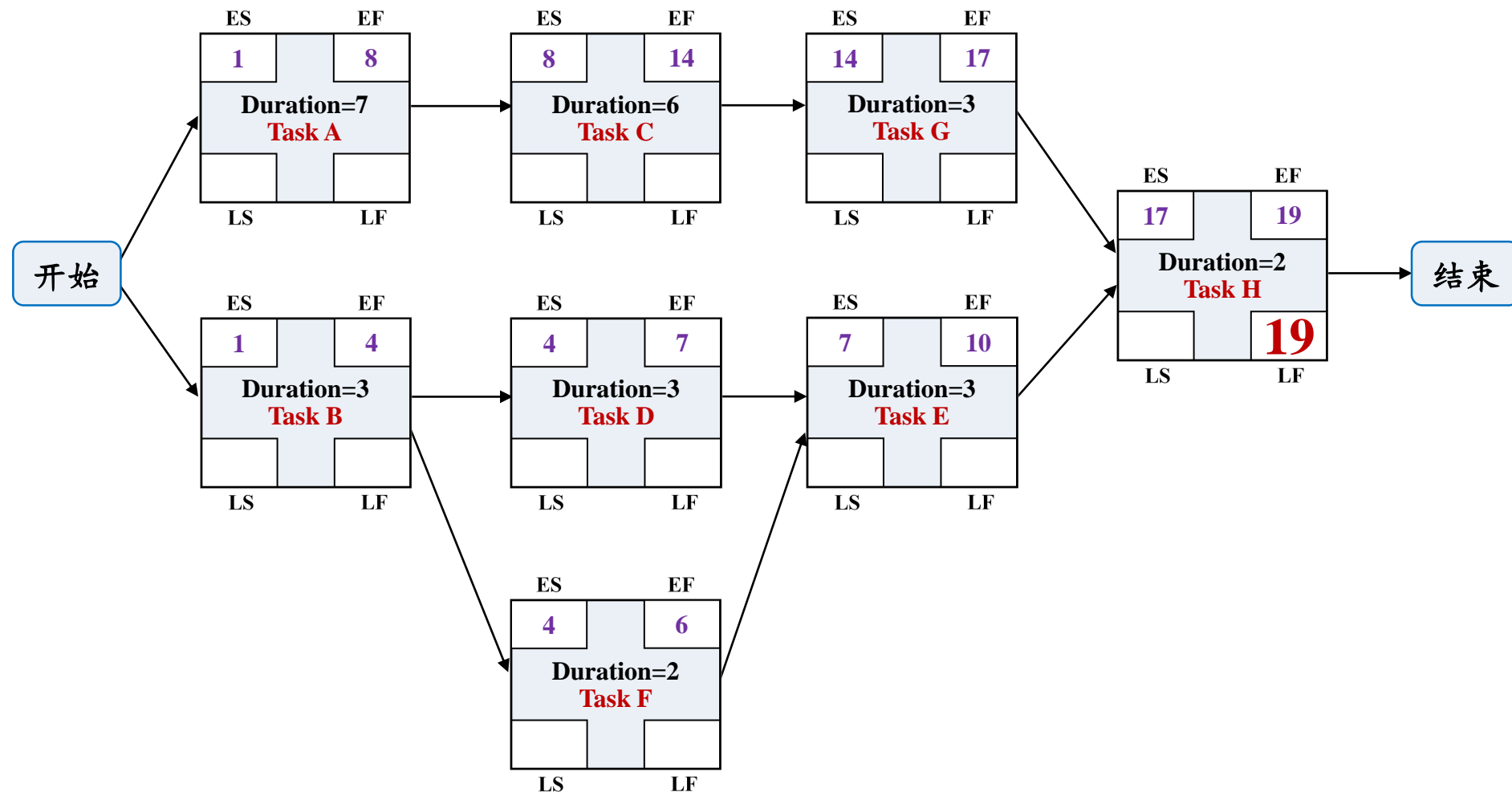


关键路径历时估算法 — 逆推法(Backward pass)

逆推法(Forward pass): 按照时间顺序计算网络图中每个活动的**最晚开始时间LS**和**最晚完成时间LF**，从而找到关键路径的方法

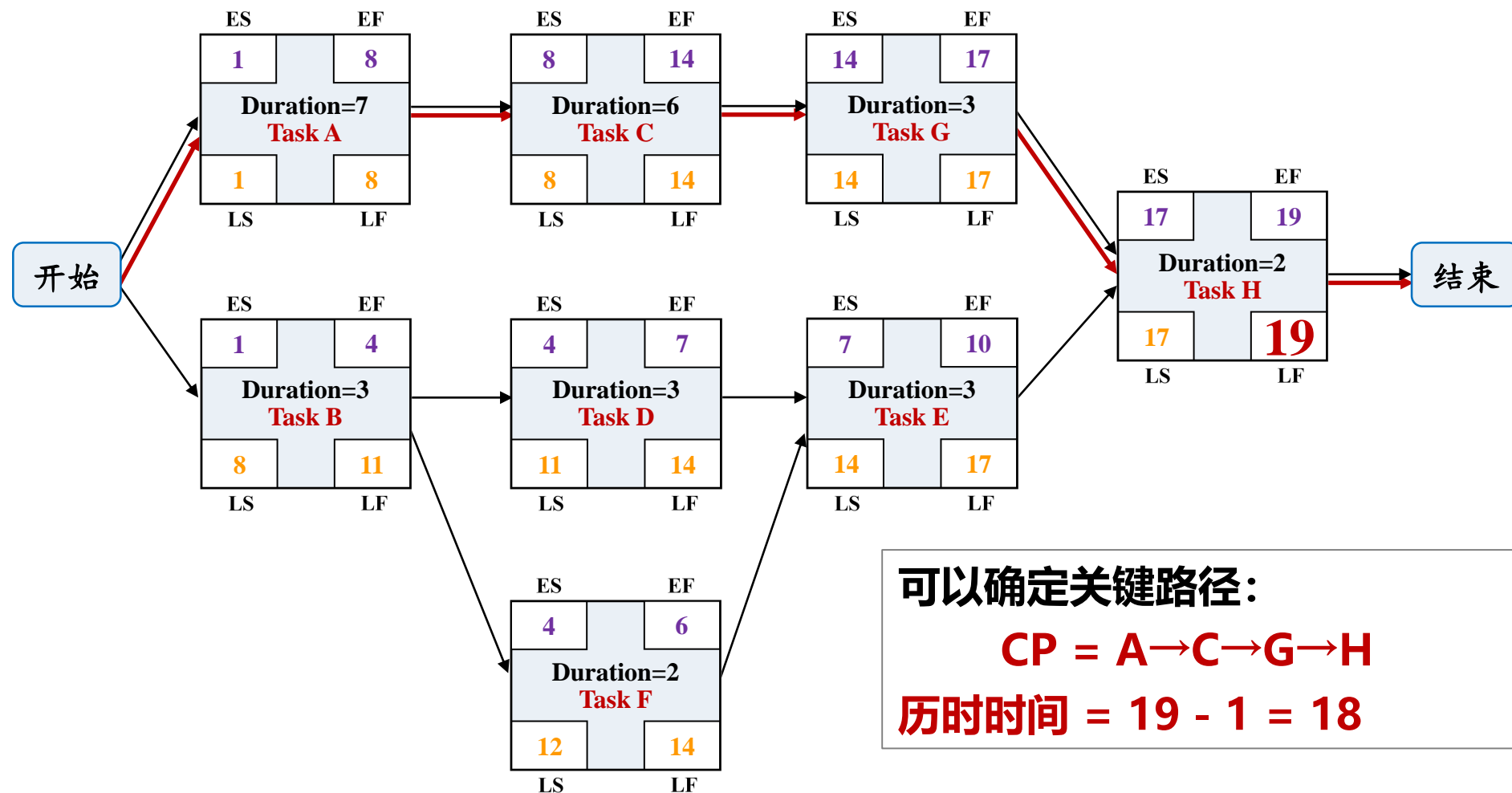
- 首先确定项目的结束时间，网络图中最后一个任务最晚完成时间是项目的结束时间，逐个对网络中每个任务节点，计算LS、LF
- $LS = LF - \text{Duration}$
- $LF = LS(s) - \text{Lag} + \text{Lead}$
 - s为后置任务节点，Lag为后置节点与本节点相关的滞后量，Lead为本节点允许后置节点的超前量
 - 当一个任务有多个后置任务时，选择后置任务中 $LS(s) - \text{Lag} + \text{Lead}$ 结果最小值作为当前节点的LF
- 遍历网络图，计算每个路径的所有任务的LS和LF
- 最后，确定关键路径，方法：找到浮动时间Float为0的路径

关键路径历时估算法 — 逆推法实例



问题: 为什么选择 $LF(H)=19$?

关键路径历时估算法 — 逆推法实例



可以确定关键路径:

CP = A → C → G → H

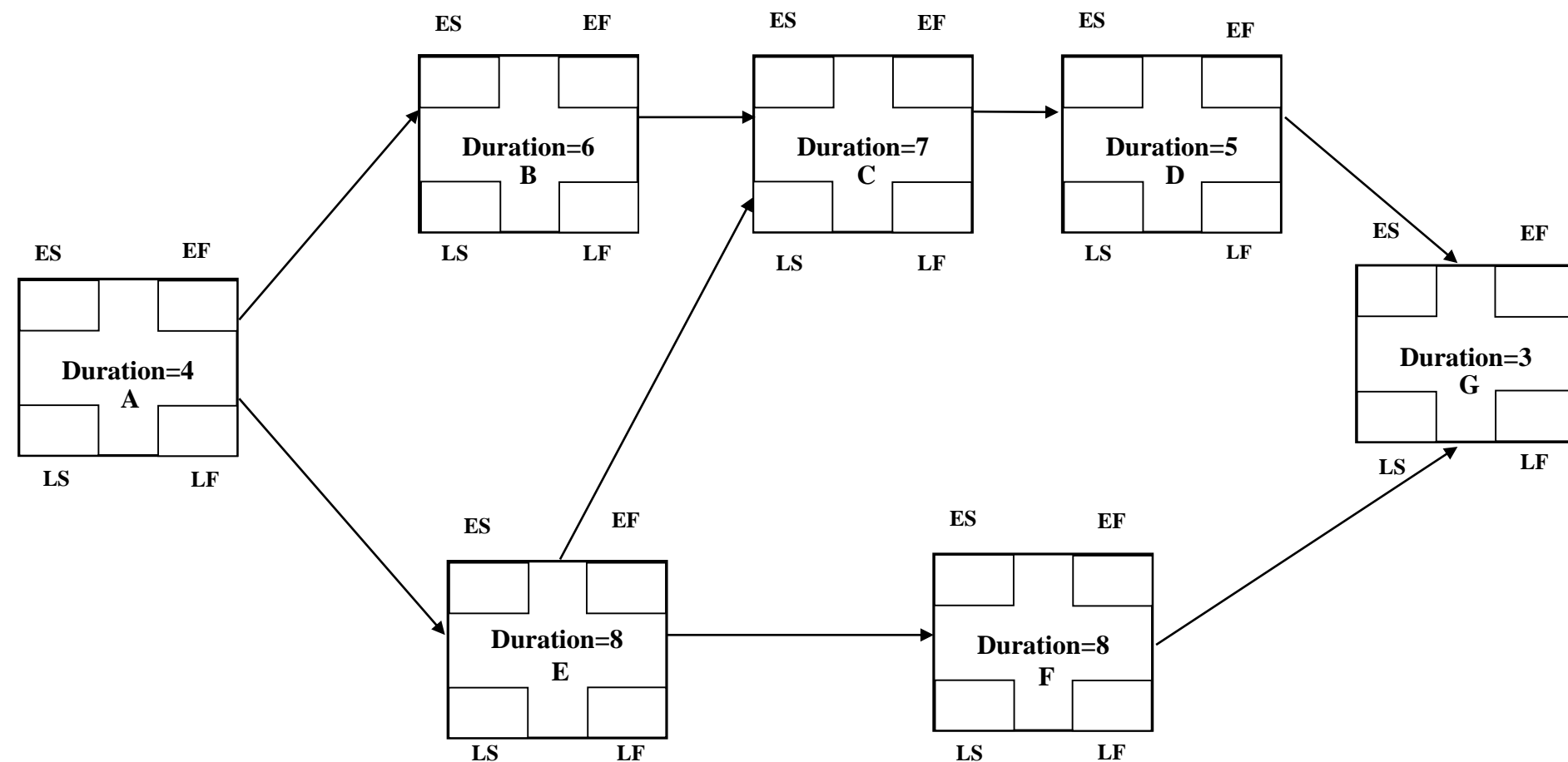
历时时间 = 19 - 1 = 18

注意: 有了正推和逆推结果后, 判定关键路径更加方便, 即路径上所有节点的浮动时间都为0, 或者说 $|ES-LS|$ 且 $|EF-LF|$ =最小恒定值

课堂练习

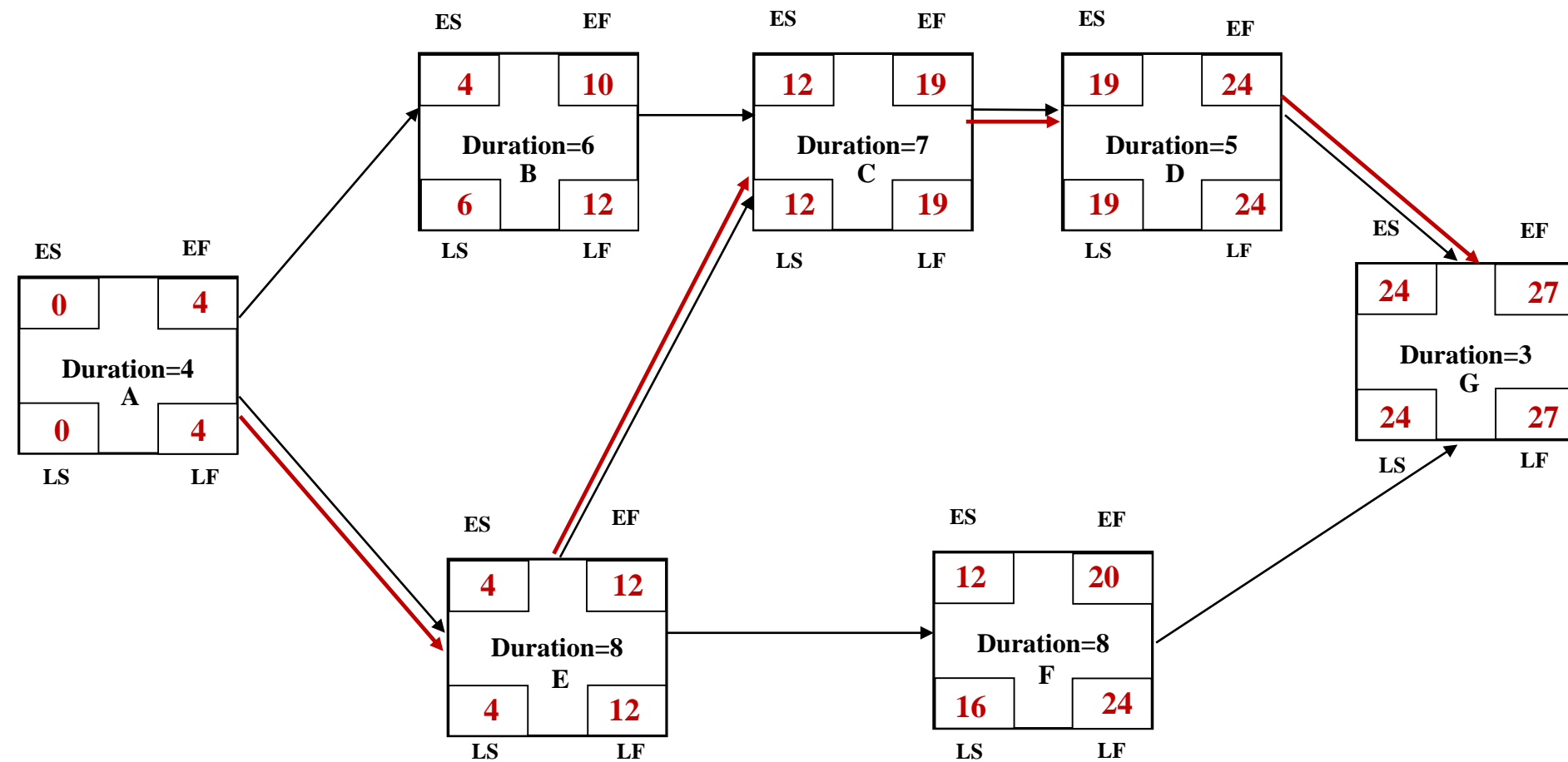
- 作为项目经理，你需要给一个软件项目做计划安排
- 经过任务分解后得到任务A, B, C, D, E, F, G
- 假设各个任务之间没有滞后和超前，下图是这个项目的PDM网络图
- 通过历时估计已经估算出每个任务的工期，现已标识在PDM网络图上
- 假设项目的最早开工日期是第0天，请计算每个任务的最早开始时间，最晚开始时间，最早完成时间，最晚完成时间，同时确定关键路径，并计算关键路径的长度，计算任务F的自由浮动和总浮动

课堂练习



1. 确定所有任务的ES, EF, LS, LF
2. 确定关键路径以及关键路径的长度?
3. 确定 F 的自由浮动和总浮动?

课堂练习-答案



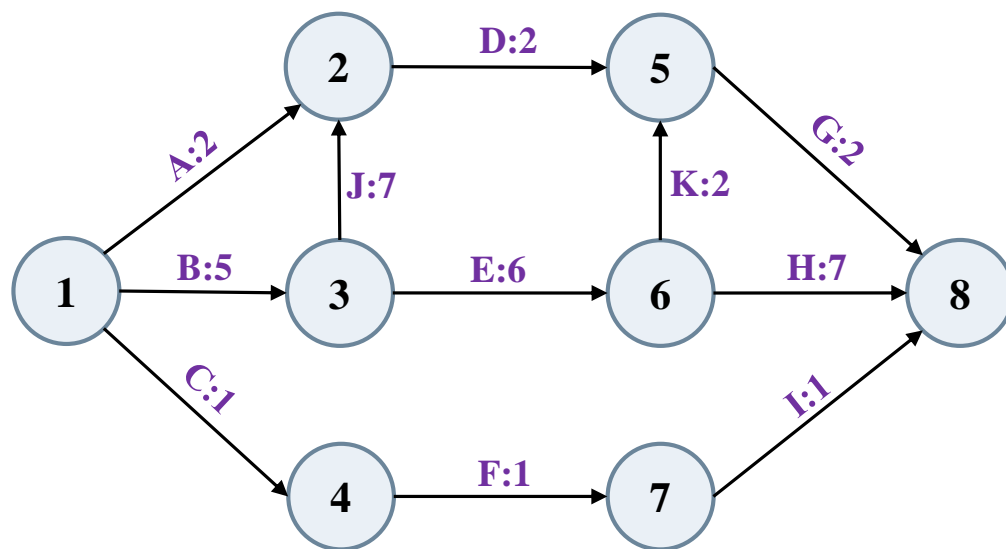
CP:A->E->C->D->G

CP Path:27

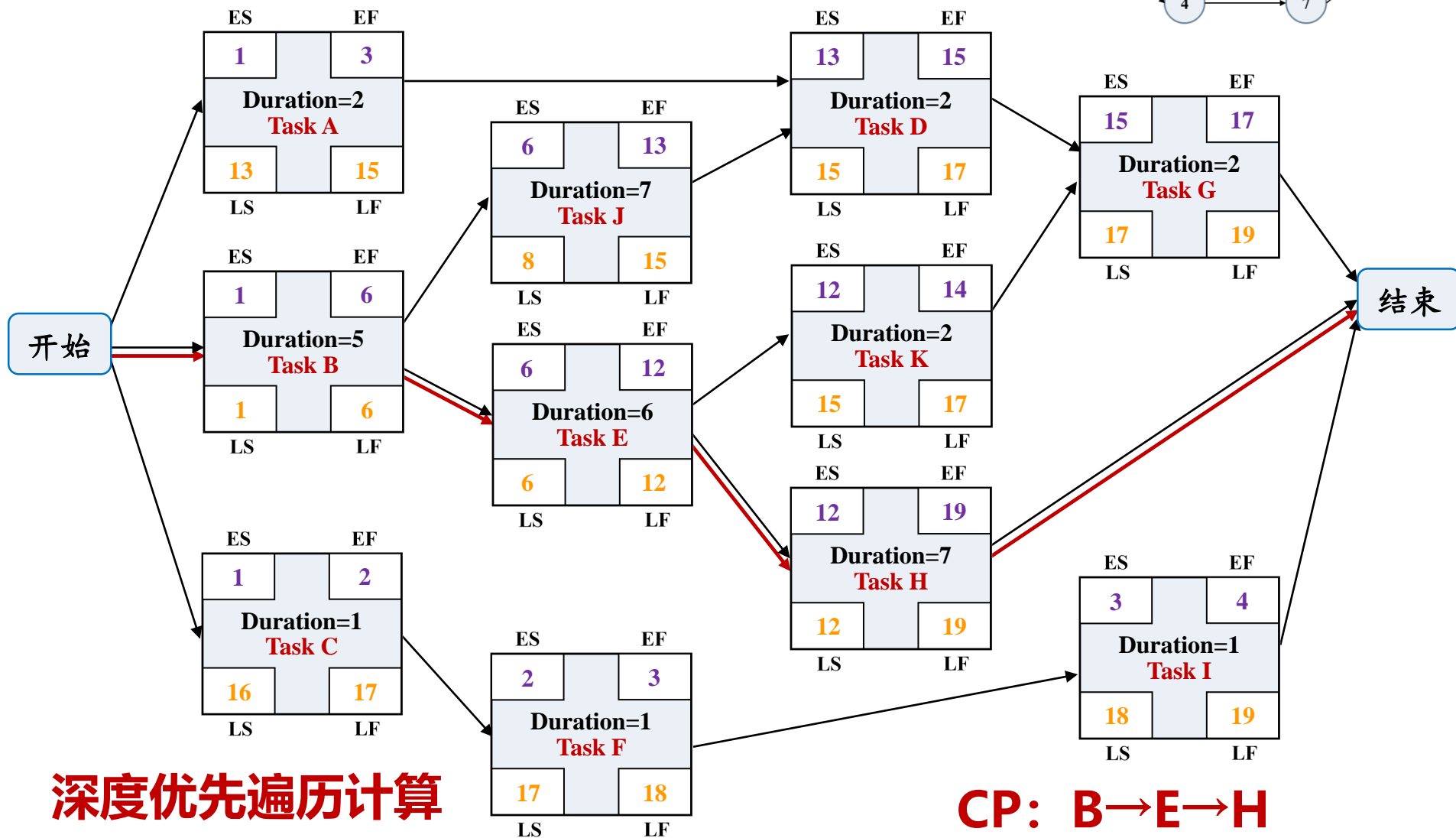
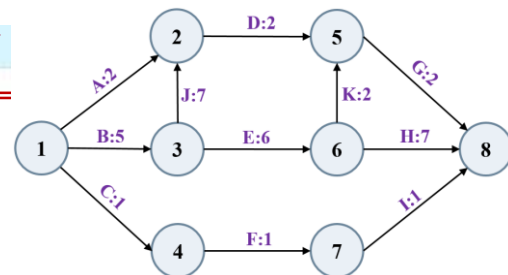
TF(F)=4

FF(F)=4

增加1个练习题



增加1个练习题



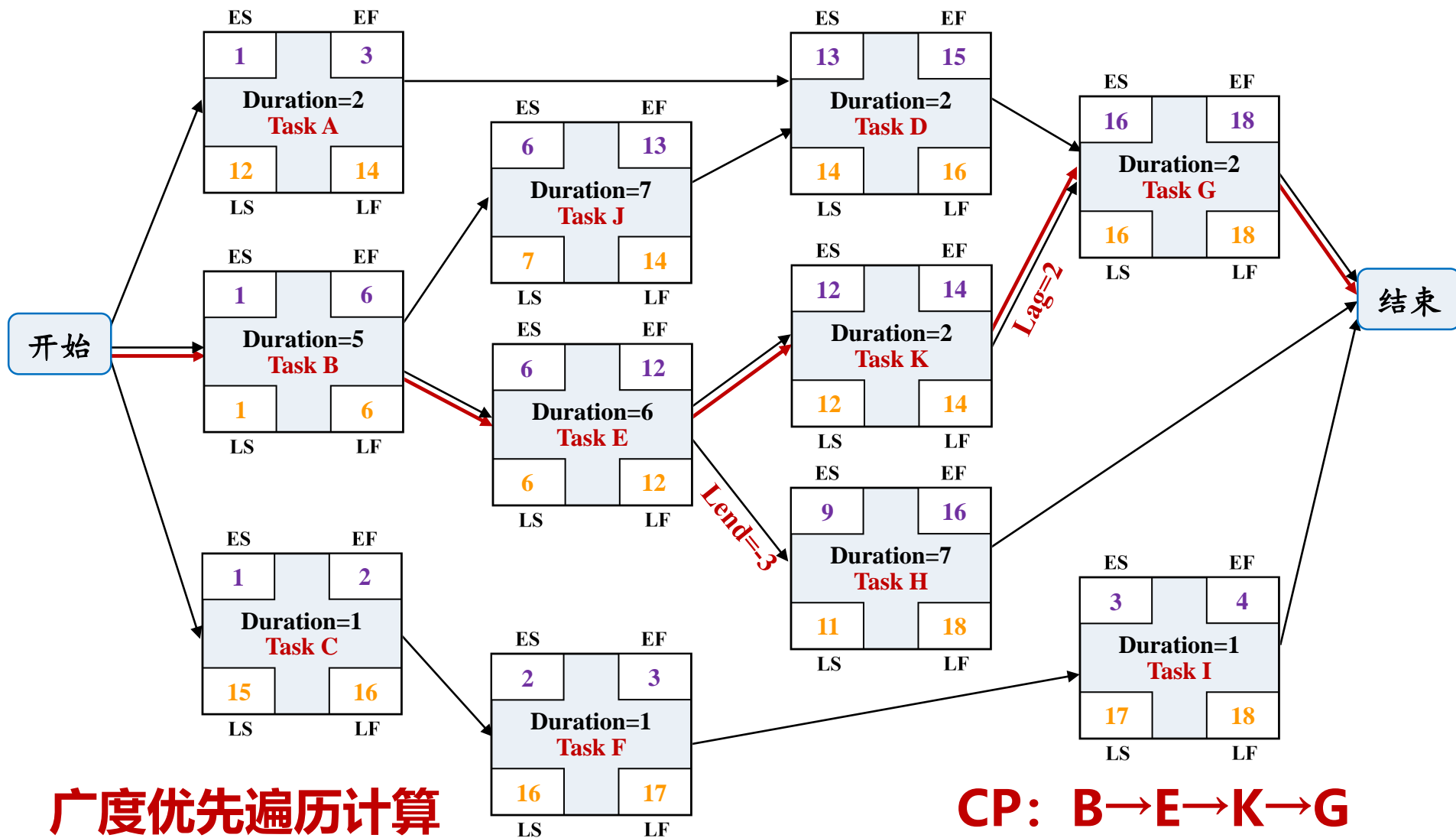
深度优先遍历计算

CP: B → E → H

CP历时 = 19 - 1 = 18

增加1个练习题

假设：Task H可以在E完成时间提前3个时间单位
Task K到G需要滞后2个时间单位



广度优先遍历计算

CP: B→E→K→G

CP历时=18-1=17