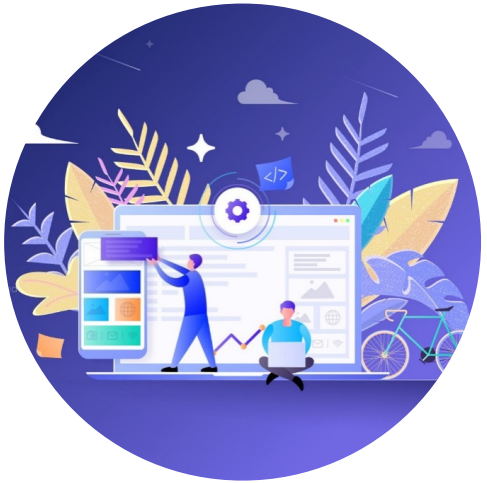
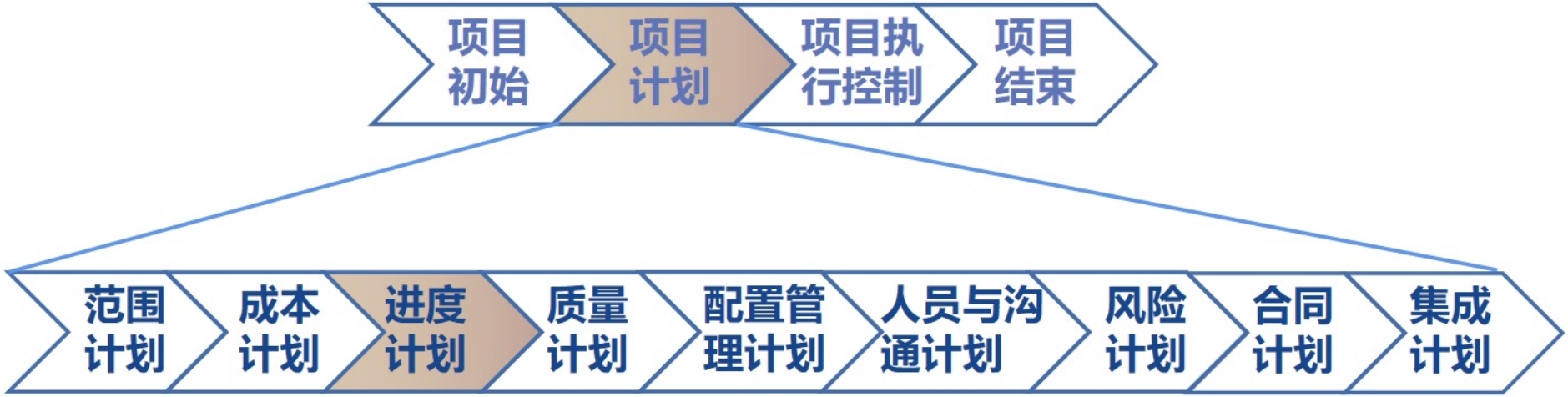
第 二 篇 第 7 章

软件项目进度计划



《软件项目管理》 – 路线图

MIMA



软件项目进度计划 学习要点

MIMA

- **一、进度管理基本概念** 7.1/7.2/7.3/7.4
- 二、任务历时估算
- 三、进度计划编排
- 四、项目进度模型
- 五、案例分析
- 六、课程实践

关于进度

MIMA

- 时间是一种特殊的资源，以其单向性、不可重复性、不可替代性而有别于其他资源
- 进度是对执行的活动和里程碑制定的工作计划日期表
- 一个项目管理者应该定义所有的项目任务，识别出关键任务，跟踪关键任务的进展情况，同时能够及时发现拖延进度的情况。为此，项目管理者必须制定一个足够详细的进度表，以便 监督项目进度并控制整个项目

进度计划

MIMA

- 进度计划的重要性
  - 按时完成项目是项目经理最大的挑战
  - 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

- 主要过程：
  - 根据WBS分解出主要的任务（活动）；
  - 确立任务（活动）之间的关联关系；
  - 然后估计每个任务（活动）需要的资源、时间；
  - 最后编制出项目的进度计划。

7.2 任务定义

MIMA

- WBS的每个工作包需要被划分成所需要的活动（任务），每个被分配的活动（任务）都应该与一个工作包相关，通过任务（活动）定义这一过程可使项目目标体现出来。
- 任务定义：确认和描述一些特定的活动的过程。
- 完成了这些活动意味着完成了WBS中的项目细目和子细目。
- 对WBS做进一步地分解

7.3 项目任务的关联关系

MIMA

- 任务定义之后，接下来需要确定任务之间的关系。为了进一步制定切实可行的进度计划， 必须对活动（任务）进行适当的顺序安排。它是通过分析所有的任务，项目的范围说明以及 里程碑等信息来确定各个任务之间的关系。

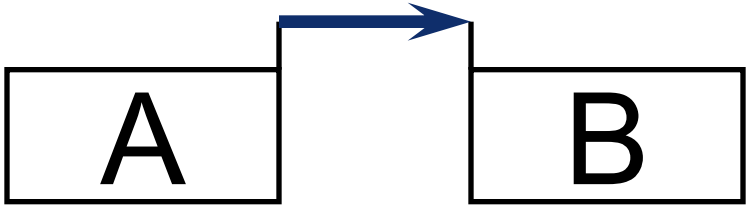
- 项目各项任务之间存在相互联系与相互依赖关系
- 根据这些关系安排任务之间的顺序

- **前置活动（任务）---〉后置活动（任务）**

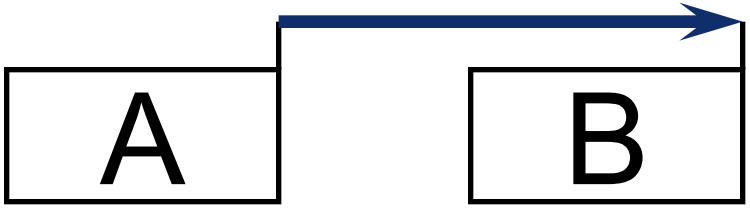
- 排序可由计算机执行（利用计算机软件）或手工排序

7.3.1 任务(活动)之间的关系

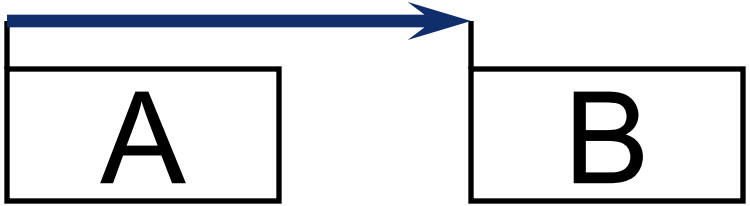
MIMA



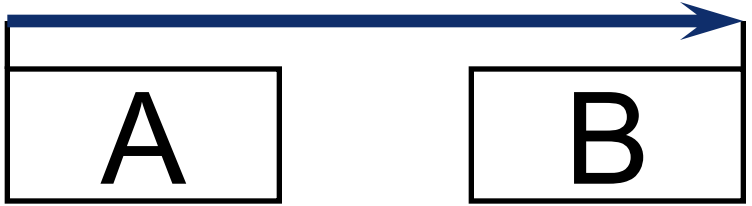
结束-开始 **FS**



结束-结束 **FF**



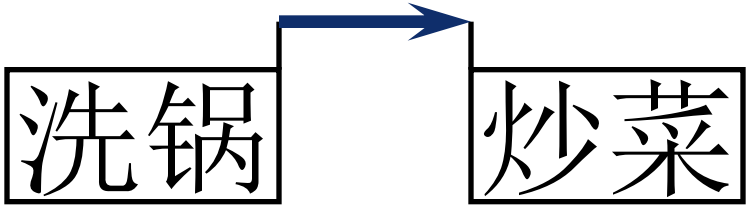
开始-开始 **SS**



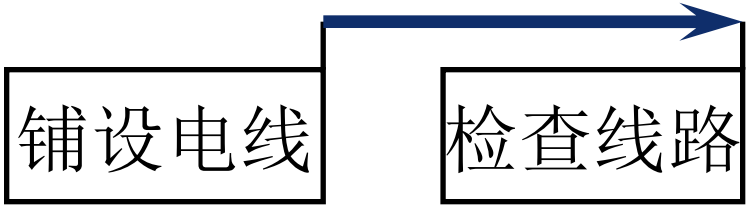
开始-结束 **SF**

7.3.1 任务(活动)之间的关系 - 示例

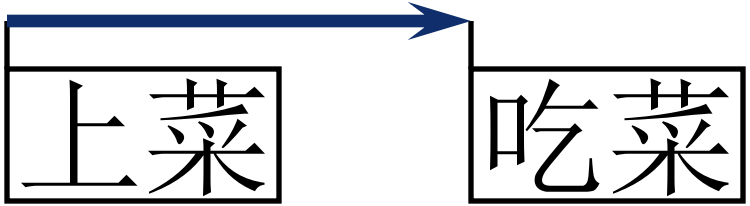
MIMA



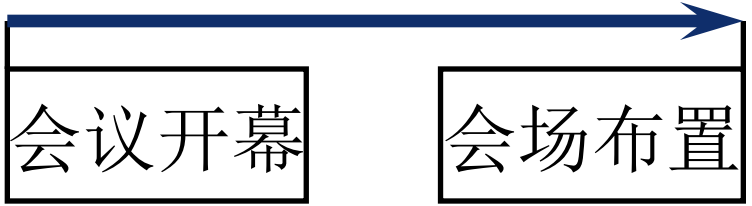
结束-开始 FS



结束-结束 FF



开始-开始 SS



开始-结束 SF

7.3.2 任务(活动)间关系的依据

MIMA

- 确定任务（活动）之间关联关系的依据：
- **硬逻辑关系**　（强制性依赖关系）
  固有的、不可违背的

- **软逻辑关系**
  人为的、主观的

- **外部依赖关系**
  项目活动与非项目活动之间

7.4 进度管理图示

MIMA

- 项目进度管理的图示：
- **网络图** network diagramming
  - 节点法/单代号网络图 PDM (Precedence Diagramming Method)
  - 箭线法/双代号网络图 ADM (Arrow Diagramming Method)
- **甘特图** Gantt图
  - 棒状甘特图
  - 三角形甘特图
- **里程碑图**
- **资源图**

7.4 进度管理图示 → 网络图

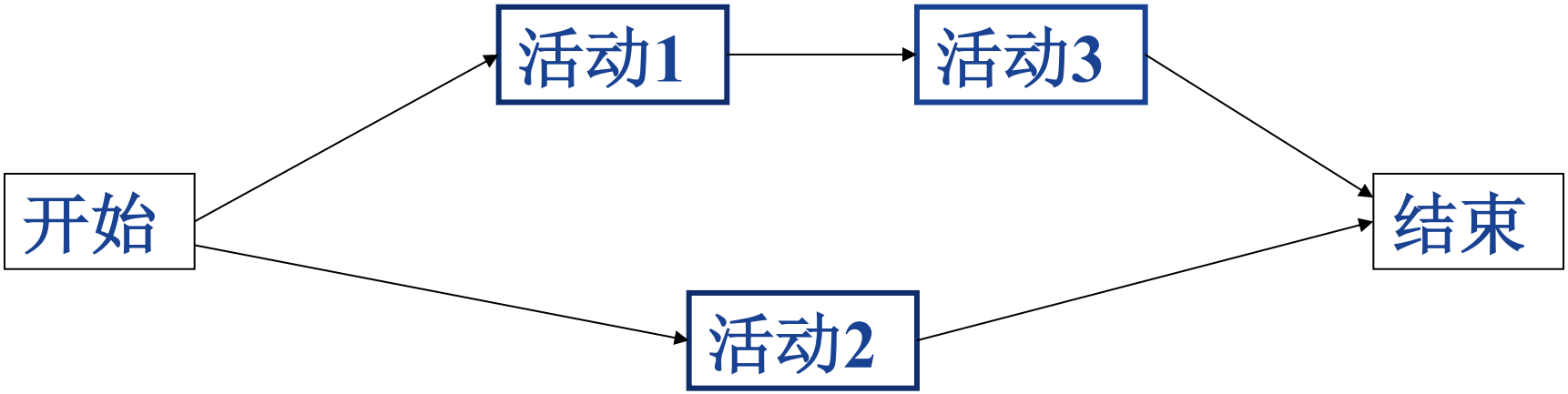
MIMA

- 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

- 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

7.4 进度管理图示 → 网络图 → PDM

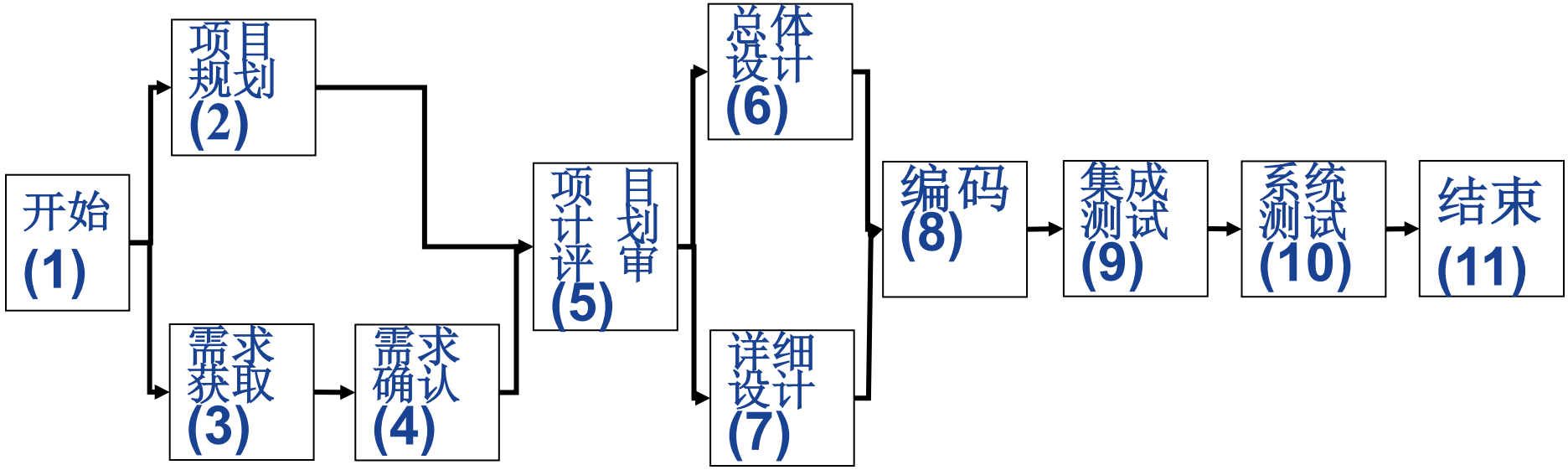
MIMA



- 节点表示活动(任务)
- 箭线表示各活动(任务)之间的逻辑关系

优先图法/节点法/单代号网络图
Precedence Diagramming Method

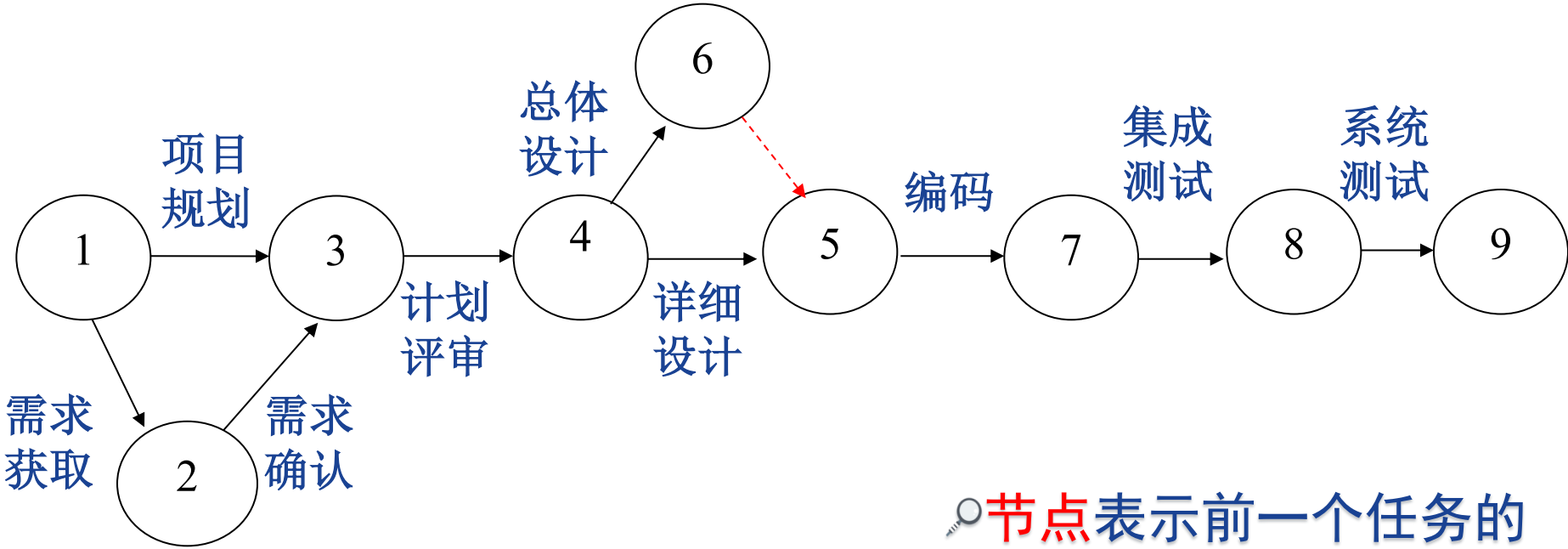
7.4 进度管理图示 → 网络图 → PDM



优先图法/节点法/单代号网络图
Precedence Diagramming Method

- 节点表示活动(任务)
- 箭线表示各活动(任务)之间的逻辑关系

7.4 进度管理图示 → 网络图 → ADM



箭线法/双代号网络图

Arrow Diagramming Method

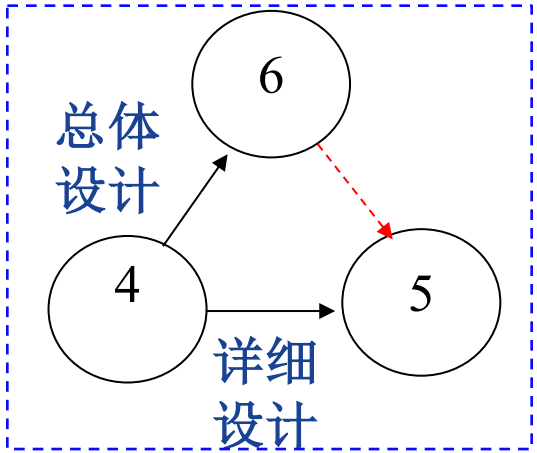
- 节点表示前一个任务的结束以及后一个的开始
- 箭线表示任务

7.4 进度管理图示 → 网络图 → ADM

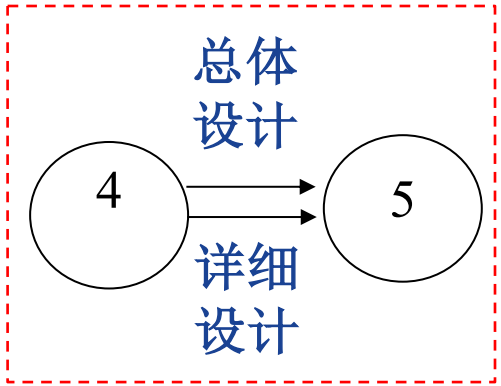
MIMA

- 双代号表示网络图中两个代号**唯一**确定一个任务

- 虚活动
  - 为了定义活动
  - 为了表示逻辑关系
  - 不消耗资源的



有虚活动的ADM图

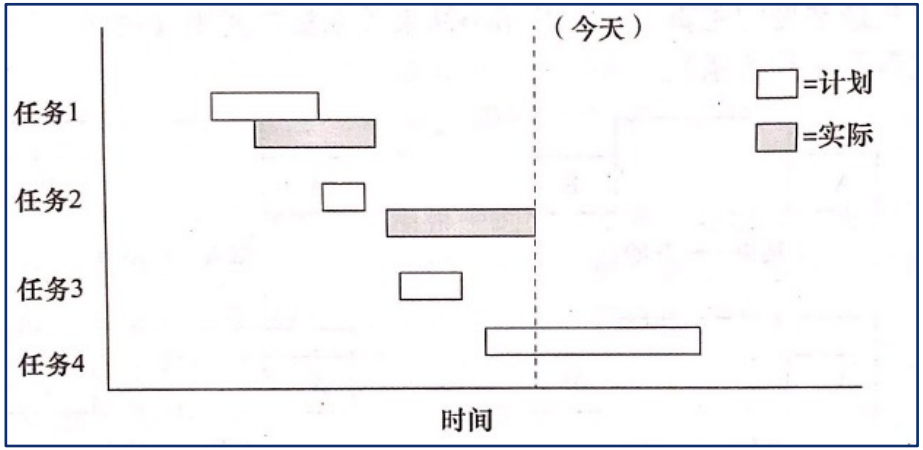


不正确的ADM图

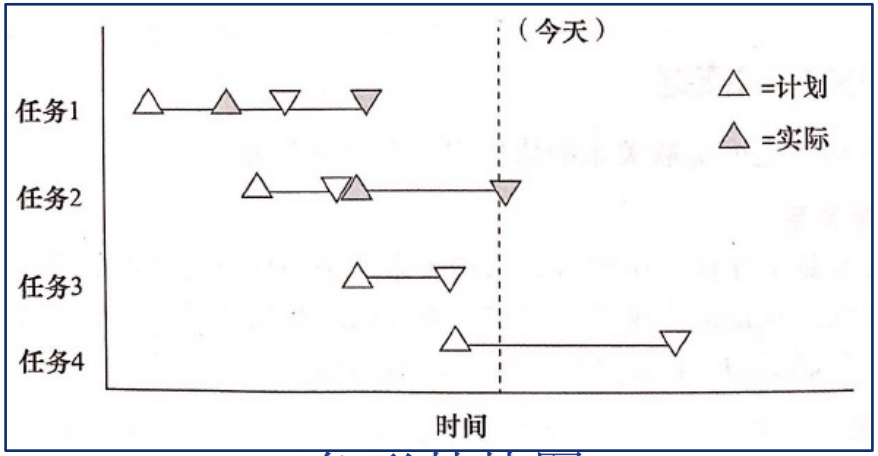
7.4 进度管理图示 → 甘特图

MIMA

- 显示基本的任务信息
- 可以查看任务的工期、开始时间和结束时间以及资源的信息。
- 只有时标，没有活动的逻辑关系



棒状甘特图



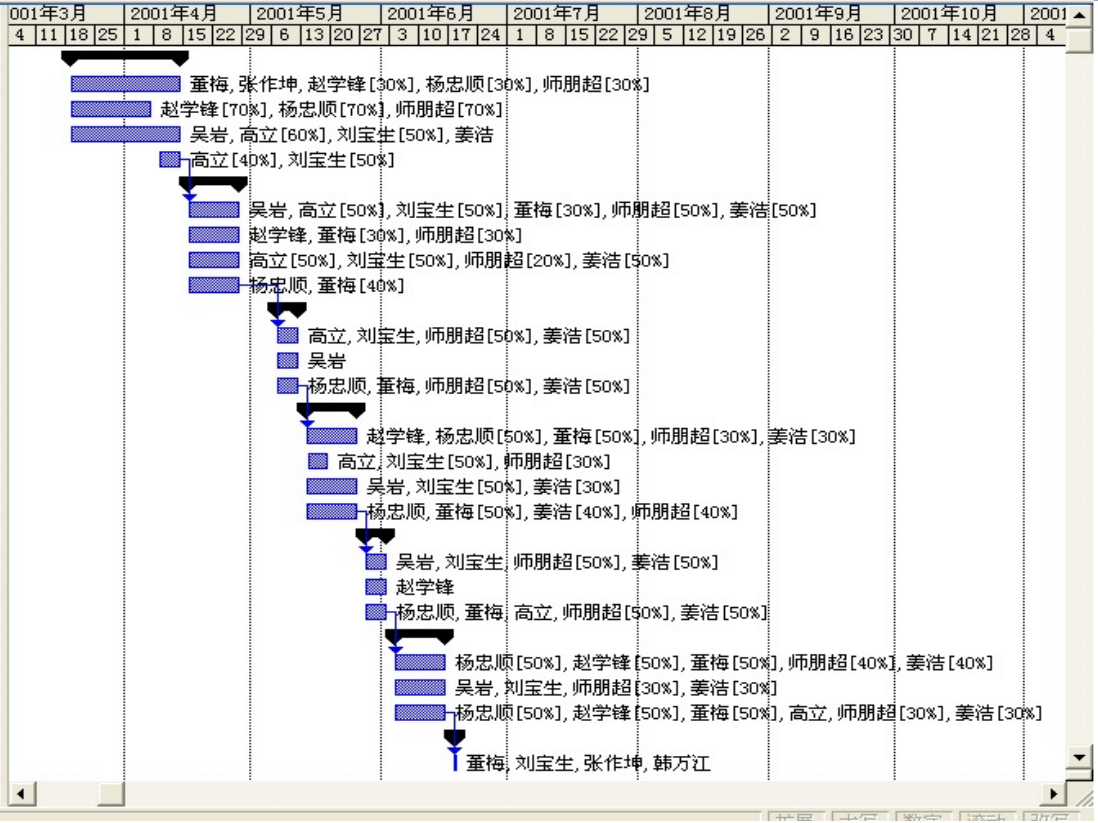
三角形甘特图

7.4 进度管理图示 → 甘特图

MIMA

Phase1

	任务名称
1	Phase1
2	公共控制系统调试，规范代码
3	公共控制系统详细设计
4	中央会计系统详细设计及规范代码
5	客户管理系统详细设计及规范代码-1
6	Phase2
7	中央会计系统调试
8	修改公共控制系统的详细设计并编写其系统设计
9	客户管理系统详细设计及规范代码-2
10	日终，日始批处理详细设计-1
11	Phase3
12	客户管理系统调试
13	修改中央会计系统的详细设计并编写其系统设计
14	日终，日始批处理详细设计-2
15	Phase4
16	日终，日始批处理调试
17	修改客户管理系统详细设计并编写其系统设计
18	转账业务详细设计，规范代码并编写其系统设计
19	贷款业务详细设计，规范代码-1
20	Phase5
21	转账业务调试
22	修改日终，日始批处理详细设计并编写其系统设计
23	贷款业务详细设计，规范代码-2
24	Phase6
25	贷款业务调试
26	修改转账业务详细设计及其系统设计
27	修改贷款业务详细设计并编写其系统设计
28	第一里程碑结束
29	代码提交入产品库



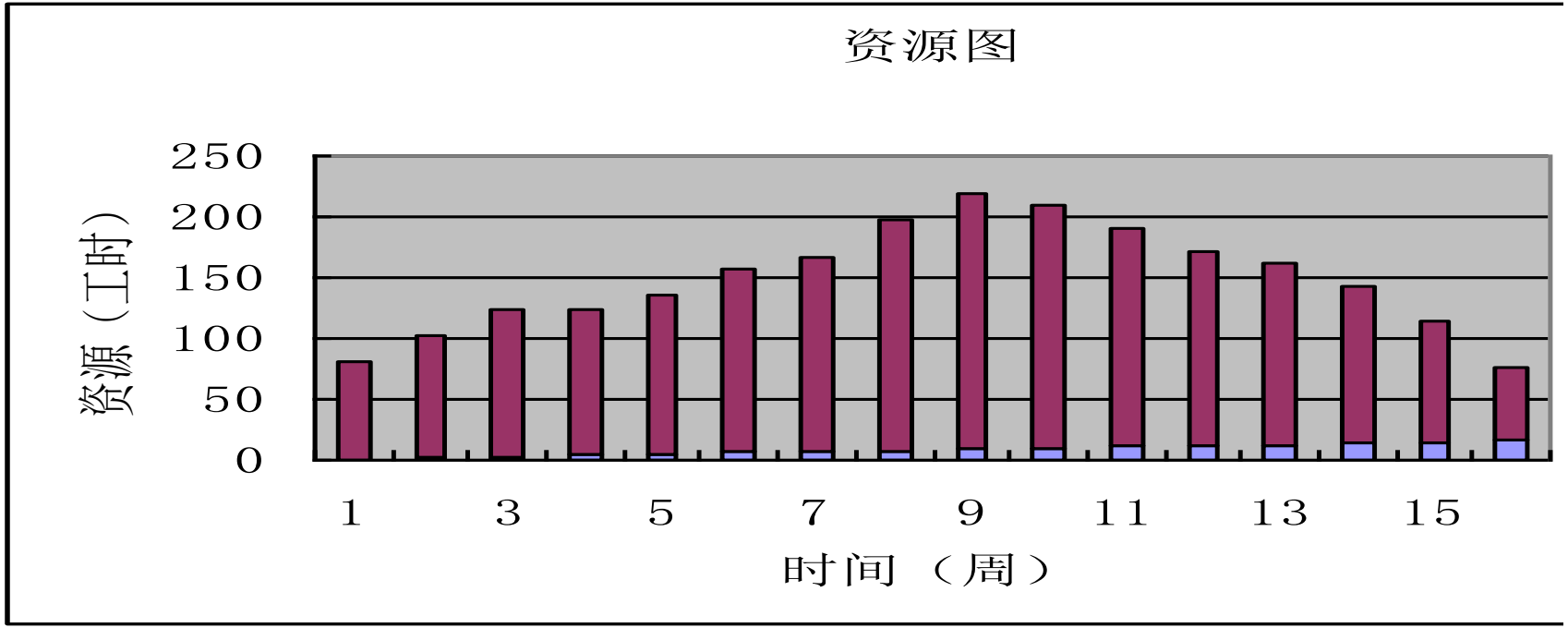
7.4 进度管理图示 → 里程碑图

MIMA

- 是由一系列的里程碑事件组成的
- 里程碑图显示项目进展中的重大工作完成
- 里程碑不同于活动
  - 活动是需要消耗资源的
  - 里程碑仅仅表示事件的标记

7.4 进度管理图示 → 资源图

MIMA



单选题 1分

设置

MIMA

（　　）可以显示任务的基本信息，使用该类图能方便的查看任务的工期、开始时间、结束时间以及资源的信息。

A 甘特图

B 网络图

C 里程碑图

D 资源图

提交

软件项目进度计划 学习要点

MIMA

- 一、进度管理基本概念
- **二、任务历时估算** 7.6
- 三、进度计划编排
- 四、项目进度模型
- 五、案例分析
- 六、课程实践

任务历时估算

MIMA

- 任务历时估计是估计任务的持续时间
  - 项目中每个任务的历时估计：分别估算项目各个活动所需要的时间
  - 项目总历时估计：根据项目活动的排序来确定整个项目所需要的时间

- 过长或过短的估计都是不利的
  - 过短，则会使项目团队处于被动紧张的状态
  - 过长，则会延迟项目的完成，可能使项目失去大好的获利机会

历时估算的基本方法

MIMA

- 定额估算法
- 经验导出模型
- PERT(工程评估评审技术)
- 基于承诺的进度估计
- Jones的一阶估算准则

基本方法 → 定额估算法 7.6.1

MIMA

- 方法比较的简单，容易计算。
- 适合项目的规模比较小
  - 比如，小于10000LOC（代码行）或者说小于6个月的项目

T=Q/(R*S)

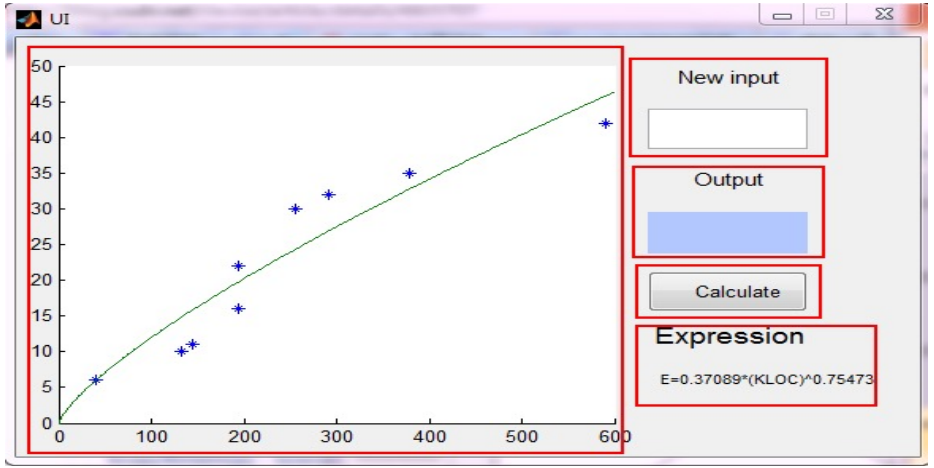
T:活动历时
Q:任务规模(工作量)
R:人力数量
S:工作效率(贡献率)

Q=6人天 ，R=2人，S=1
则： T=3天
Q=6人天 ，R=2人，S=1.5
则： T=2天

基本方法 → 经验导出模型 7.6.2

MIMA

- 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
  - a、b是依赖于项目的自然属性



- **IBM模型/基本COCOMO模型**

  $D=2.4*E^{0.35}$　　$D=2.5*E^b$ (b在0.32-0.38之间)

有机	0.38
半有机	0.35
嵌入式	0.32

基本方法 → 工程评估评审技术 7.6.3

MIMA

- PERT: Program Evaluation and Review Technique
- 于1958年，为适应大型工程的需要，并取得不错效果
- 利用网络顺序图的逻辑关系和加权算法估算任务历时

基本方法 → PERT 7.6.3 → 加权算法

MIMA

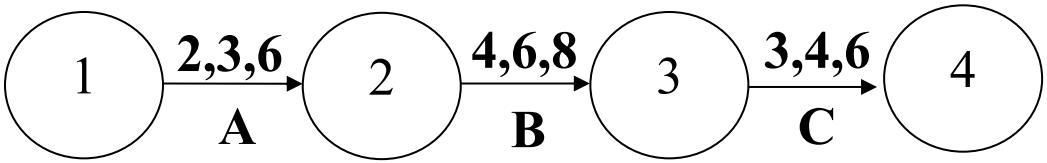
- 它是基于对某项任务的乐观，悲观以及最可能的概率时间估计

- 采用加权平均得到期望值

  PERT历时 = (O+4m+P) / 6

  - O是最小估算值:乐观(Optimistic Time)
  - P是最大估算值:悲观(Pessimistic Time)
  - M是最可能估算(Most Likely Time)

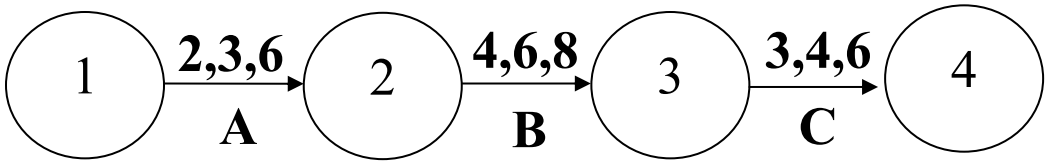
基本方法 → PERT 7.6.3 → 例



PERT历时 = (O+4m+P) / 6

基本方法 → PERT 7.6.3 → 例

MIMA

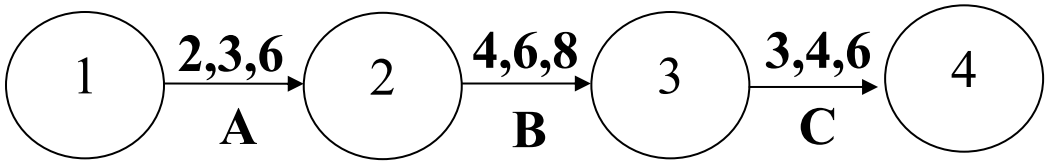


活动 \ 项	O,M,P	PERT估计值
A	2,3,6	3.33
B	4,6,8	6
C	3,4,6	4.17
估计项目总历时		

PERT历时 = (O+4m+P) / 6

PERT历时 = (4+4*6+8) / 6=6

基本方法 → PERT 7.6.3 → 例



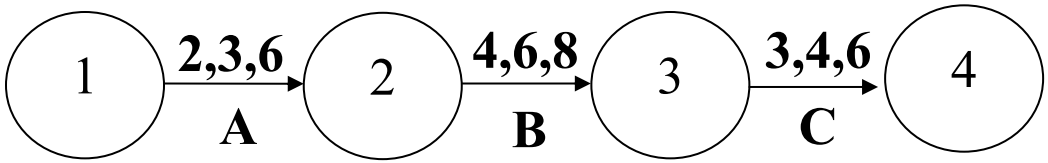
$$\delta = \frac{P-O}{6}$$ 标准差

$$\delta^2 = \left(\frac{P-O}{6}\right)^2$$ 方差

活动 \ 项	O,M,P	PERT估计值	δ	δ^2
A	2,3,6	3.33	4/6	16/36
B	4,6,8	6	4/6	16/36
C	3,4,6	4.17	3/6	9/36
估计项目总历时				

$$\delta_B^2 = \left(\frac{8-4}{6}\right)^2 = \frac{16}{36}$$

基本方法 → PERT 7.6.3 → 例



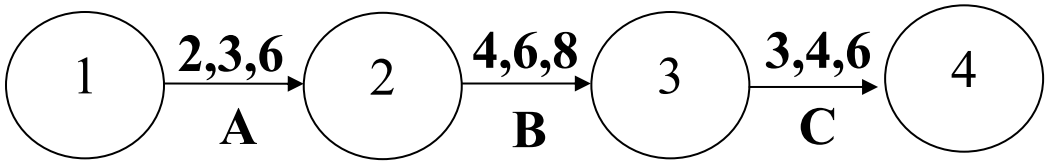
活动 \ 项	O,M,P	PERT估计值	δ	δ^2
A	2,3,6	3.33	4/6	16/36
B	4,6,8	6	4/6	16/36
C	3,4,6	4.17	3/6	9/36
估计项目总历时		13.5	1.07	41/36

$$E = E_A + E_B + E_C$$

$$\delta^2 = \delta_A^2 + \delta_B^2 + \delta_C^3$$

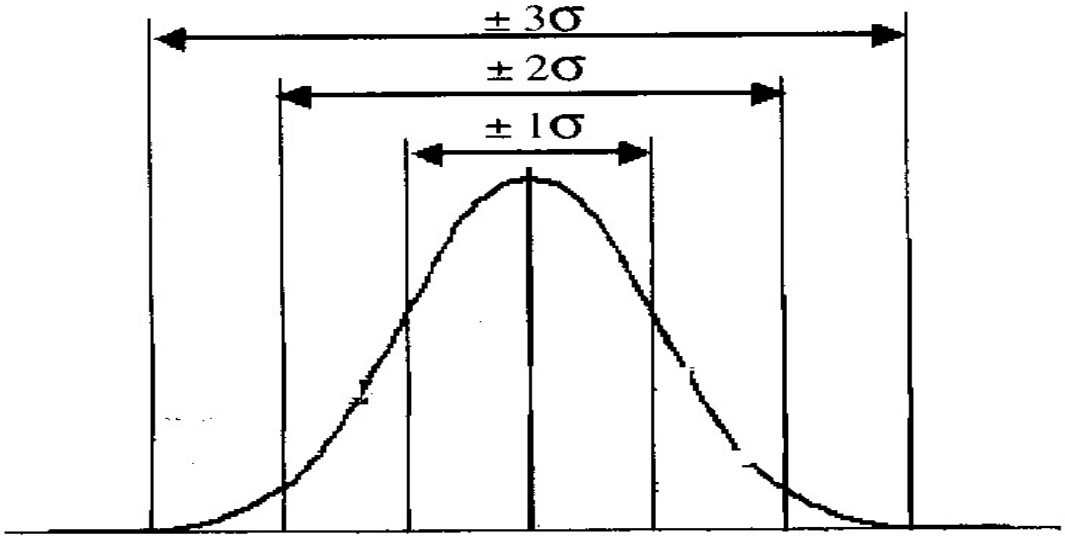
$$\delta = \sqrt{\delta_A^2 + \delta_B^2 + \delta_C^3}$$

基本方法 → PERT 7.6.3 → 例



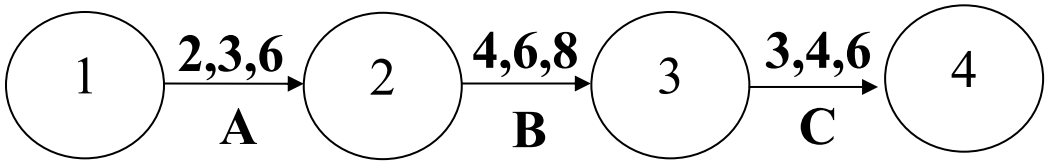
$$E = 13.5$$

$$\delta = 1.07$$

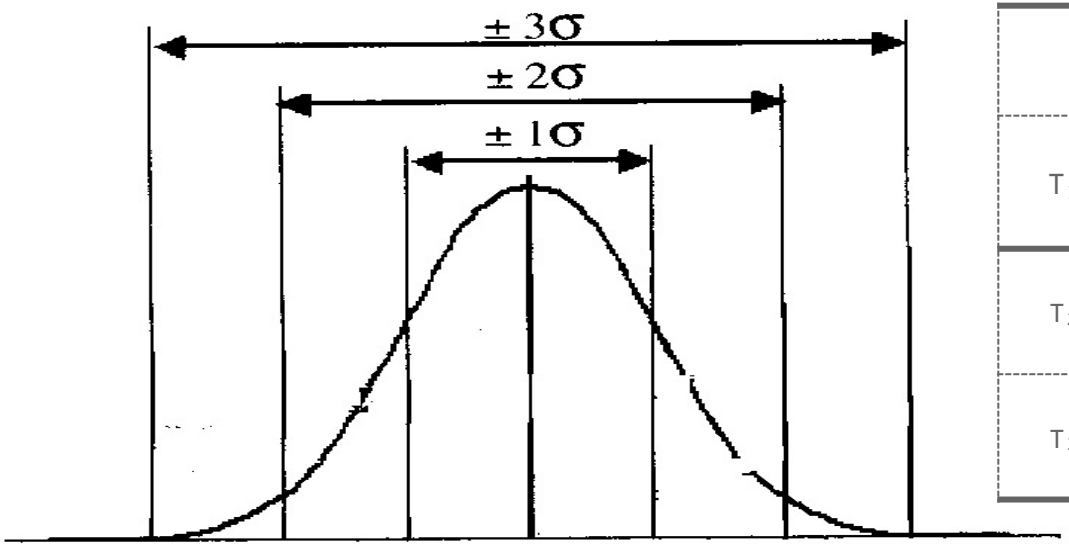


$E \pm \delta$	68.3%
$E \pm 2\delta$	95.5%
$E \pm 3\delta$	99.7%

基本方法 → PERT 7.6.3 → 例



$$E = 13.5 \quad \delta = 1.07$$



范围		概率	从	到
T_1	±δ	68.3%	**12.43**	**14.57**
T_2	±2δ	95.5%	**11.36**	**15.64**
T_3	±3δ	99.7%	**10.29**	**16.71**

基本方法 → 基于承诺的进度估计法 7.6.7

MIMA

- 基于开发人员做出的进度承诺进行
- 不进行中间的工作量（规模）估计

- 优点
  - 有利于开发者对进度的关注
  - 有利于开发者在接受承诺之后的士气高昂
- 缺点
  - 易于产生大的估算误差

基本方法 → Jones的一阶估算准则 7.6.8

MIMA

- 基于估算项目功能点
- 从幂次表中选择合适的幂次将功能点升幂

- 例如，一个商业软件的功能点FP=350，若一个平均水平的软件公司来承担，粗略的进度估算为 $350^{0.43}$=12月。

软件类型	最优级	平均	最差级
系统软件	0.43	0.45	0.48
商业软件	0.41	**0.43**	0.46
封装商品软件	0.39	0.42	0.45

一阶幂次表

软件项目进度计划 学习要点

MIMA

- **一、进度管理基本概念**
- **二、任务历时估算**
- 三、进度计划编排
- 四、项目进度模型
- 五、案例分析
- 六、课程实践

单选题 1分

设置

MIMA

判断题：一个工作包可以通过多个活动完成。

A 对

B 错

提交

单选题 1分

设置

MIMA

判断题：项目各项活动之间不存在相互联系与相互依赖关系。

A 错

B 对

提交

单选题 1分

设置

以下哪一项是项目计划中灵活性最小的因素？

A 时间

B 人工成本

C 管理

D 开发

提交

软件项目进度计划 学习要点

MIMA

- 一、进度管理基本概念
- 二、任务历时估算
- **三、进度计划编排** 7.7
- 四、项目进度模型
- 五、案例分析
- 六、课程实践

项目进度计划编制

MIMA

- 进度计划编制是决定项目开始和结束日期的活动。

- 主要目的是控制和节约项目的时间，保证项目在规定的时间内能够完成。
- 最终目标是建立一个现实的项目进度计划，为监控项目的进度进展情况提供一个基础。

- 项目进度计划过程反复多次，最后才能确定项目进度计划

进度编制的基本方法

MIMA

- 关键路径法 CPM Critical Path Method 正推法/逆推法
- 时间压缩法 应急法/平行作业法
- 资源平衡
- 管理预留
- 敏捷计划

基本方法 → 关键路径法 7.7.1

MIMA

- 关键路径法是根据指定的网络图逻辑关系进行的单一的历时估算。
- 计算每一个活动的单一的、确定的最早和最迟开始和完成日期；计算网络图中最长的路径。以便确定项目完成时间估计。
- 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

基本方法 → 关键路径法 7.7.1

MIMA

- 进度编制相关术语：

  - 最早开始时间(Early Start, ES)　表示一项任务（活动）的最早可以开始执行的时间
  - 最晚开始时间(Late Start, LS)　表示一项任务（活动）的最晚可以开始执行的时间
  - 最早完成时间(Early Finish, EF)　表示一项任务（活动）的最早可以完成的时间
  - 最晚完成时间(Late Finish, LF)　表示一项任务（活动）的最晚可以完成的时间

基本方法 → 关键路径法 7.7.1

MIMA

- 进度编制相关术语：
  - 超前(Lead) 表示两个任务的逻辑关系所允许的提前后置任务的时间
  - 滞后(Lag) 表示两个任务的逻辑关系所允许的推迟后置任务的时间
  - 浮动(Float) 表示一个任务的机动性，是其在不影响项目完成的情况下可以推迟的时间量
  - 自由浮动(Free Float, FF) 在不影响后置任务最早开始时间，本任务可以延迟的时间
  - 总浮动(Total Float, TF) 表示在不影响项目最早完成时间，本任务可以延迟的时间

FF=ES(后置任务)-EF-lag　　TF=LS-ES / TF=LF-EF

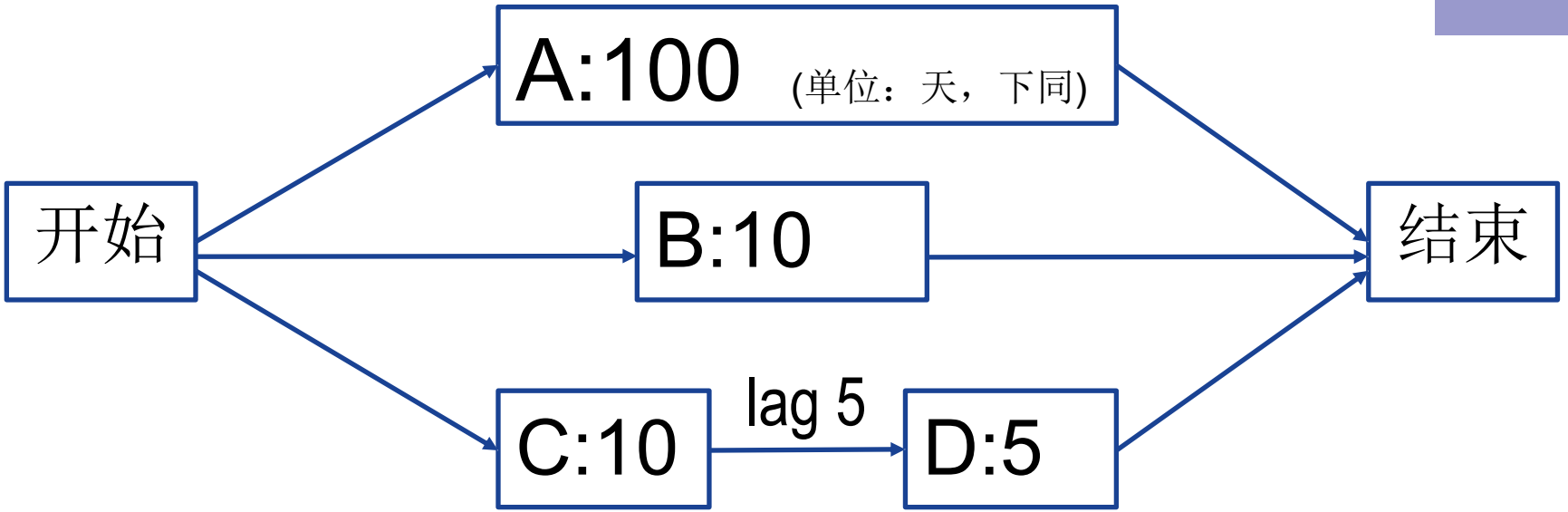
基本方法 → 关键路径法 7.7.1

MIMA

- 进度编制相关术语：

- 关键路径 (Critical Path)
  - 时间浮动为0（Float=0）的路径　　TF=LS-ES / TF=LF-EF
  - 网络图中最长的路径
  - 关键路径是决定项目完成的最短时间
  - 关键路径上的任何活动延迟，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

基本方法 → 关键路径法 7.7.1 → 例

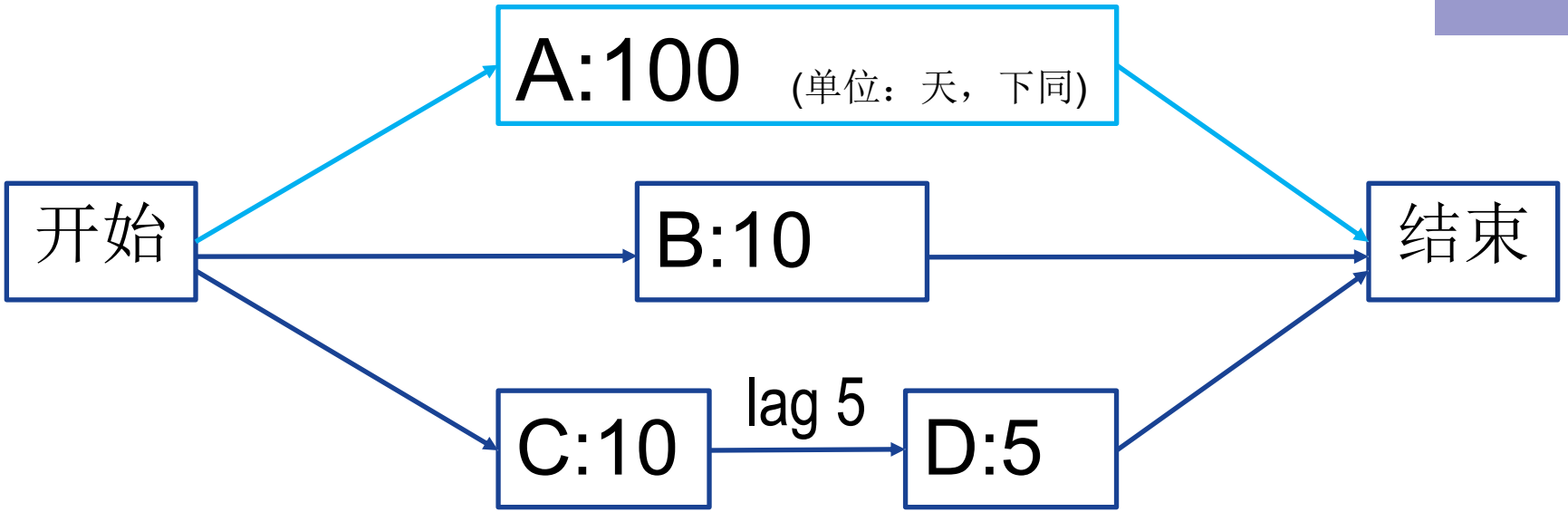


ES(A)=LS(A)=0
EF(A)=LF(A)=100

ES(B)=0 LS(B)=90
EF(B)=10 LF(B)=100

TF(A)=LS(A)-ES(A)=LF(A)-EF(A)=0
TF(B)=LS(B)-ES(B)=LF(B)-EF(B)=90

基本方法 → 关键路径法 7.7.1 → 例



ES(C)=0 ES(D)=15
EF(C)=10 EF(D)=20

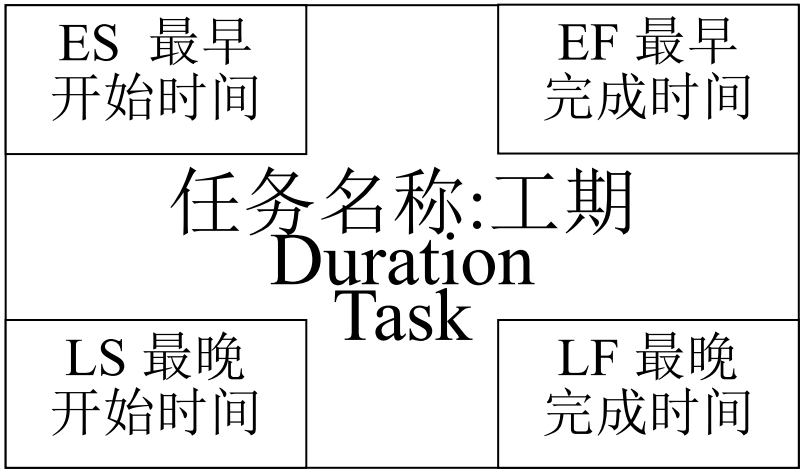
FF(C)=ES(D)-EF(C)-lag=0

LF(D)=100 LS(D)=95
LF(C)=90 LS(C)=80

基本方法 → 关键路径法 7.7.1

MIMA

- 任务图示

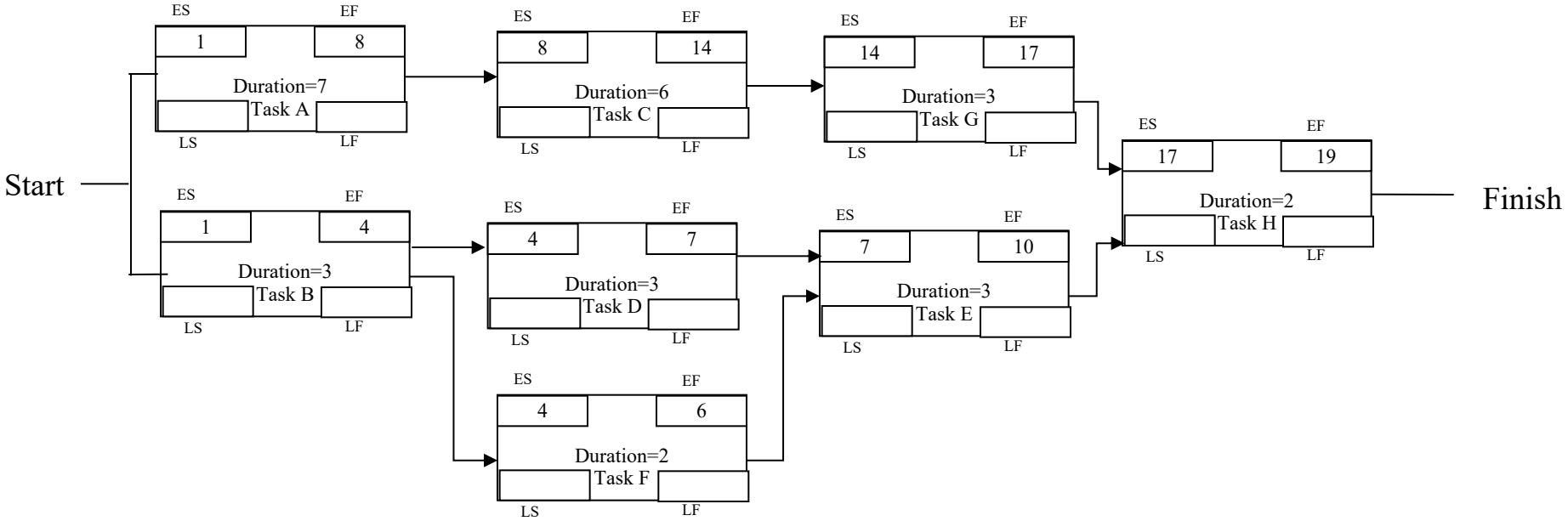


基本方法 → 关键路径法 7.7.1 → 正推法

MIMA

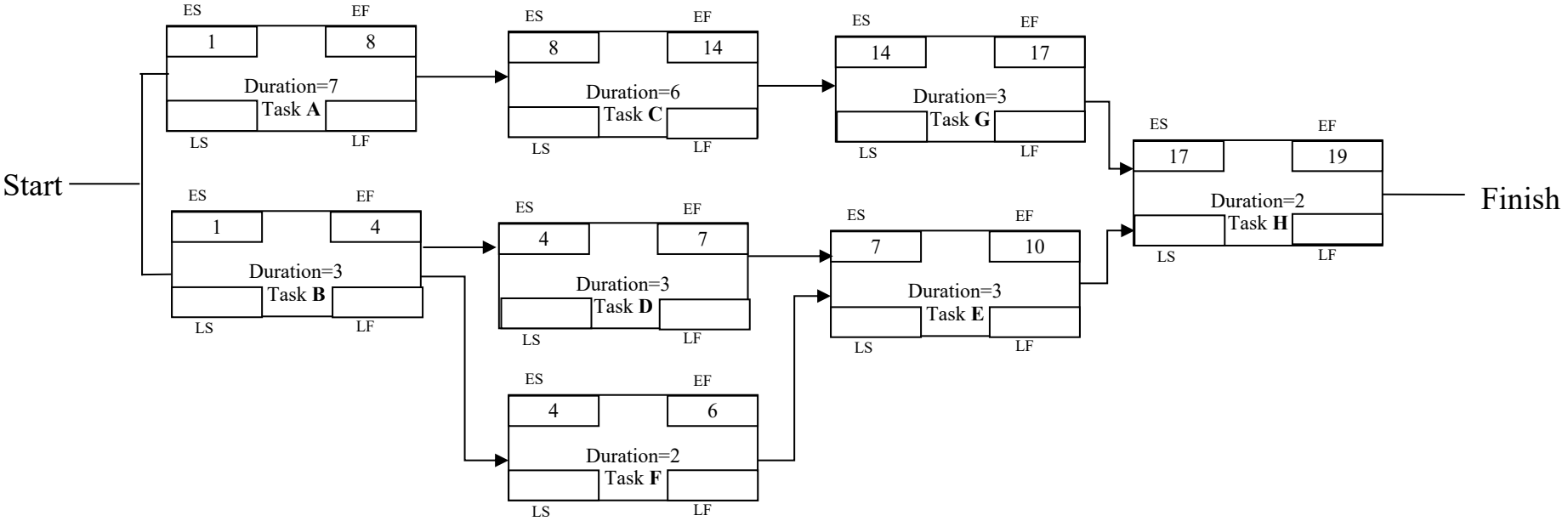
- **正推法**：按照时间顺序计算各个任务（活动）的最早开始时间和最早完成时间的方法
- 过程如下
  - 确定项目的开始时间
  - 从左到右，从上到下进行任务编排
  - 计算每个任务的最早开始时间ES和最早完成时间EF：
    - 网络图中第一个任务的最早开始时间是项目的开始时间
    - ES+Duration=EF
    - EF+Lag=ES(s)
    - 当一个任务有多个前置任务时，选择前置任务中最大的EF加上Lag作为其ES

基本方法 → 关键路径法 7.7.1 → 正推法



基本方法 → 关键路径法 7.7.1 → 正推法

MIMA

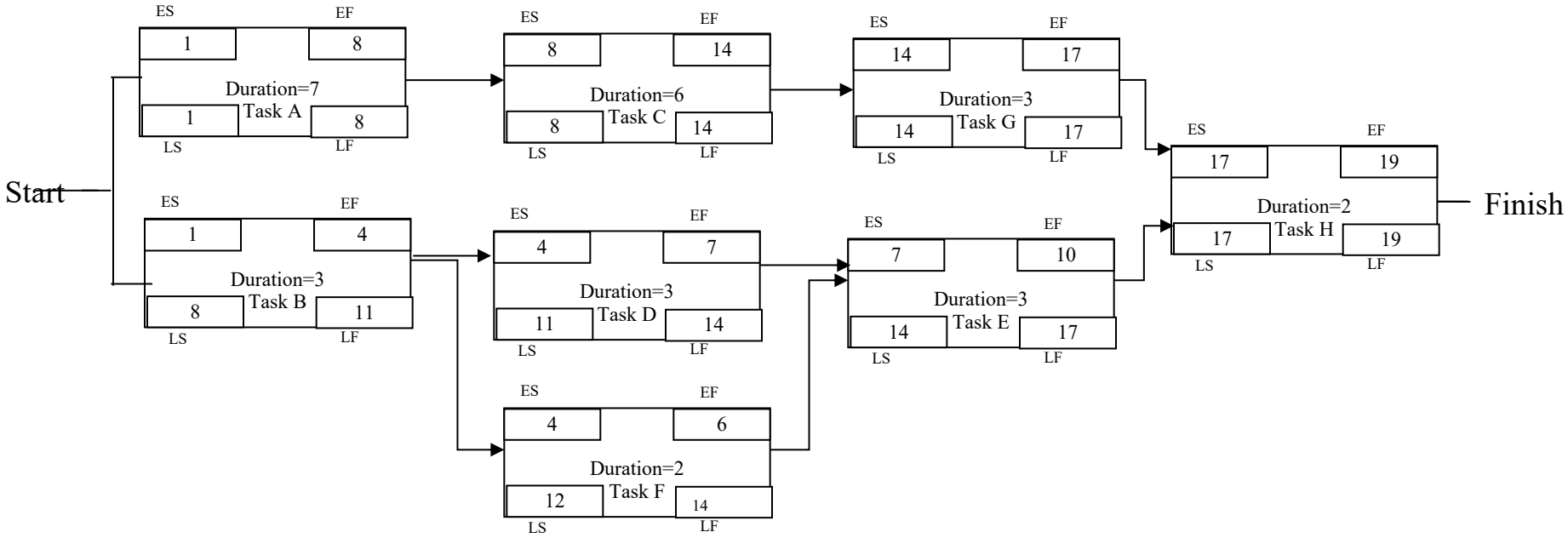


基本方法 → 关键路径法 7.7.1 → 逆推法

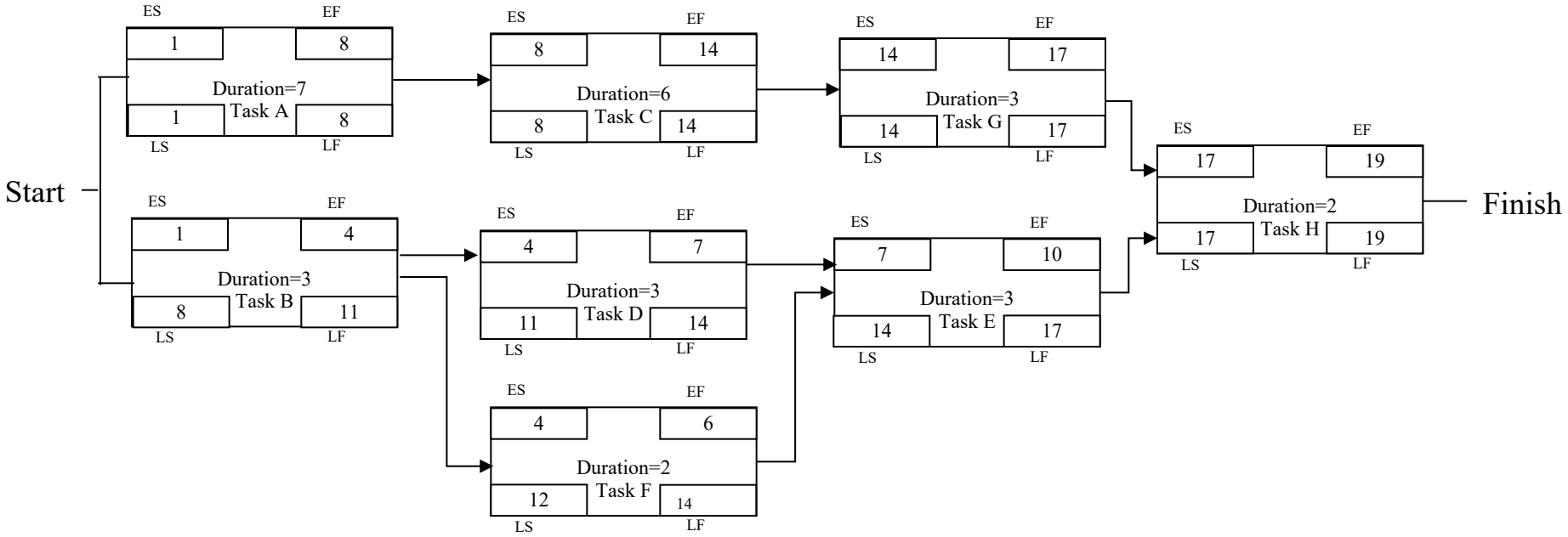
MIMA

- **逆推法**：按照时间顺序计算各个任务（活动）的最晚开始时间和最晚完成时间的方法
- 过程如下
  - 确定项目的结束时间
  - 从右到左，从上到下进行任务编排
  - 计算每个任务的最晚开始时间LS和最晚完成时间LF：
    - 网络图中最后一个任务的最晚完成时间是项目的结束时间
    - LF-Duration=LS
    - LS-Lag=LF(p)
    - 当一个任务有多个后置任务时，选择其后置任务中最小LS减Lag作为其LF

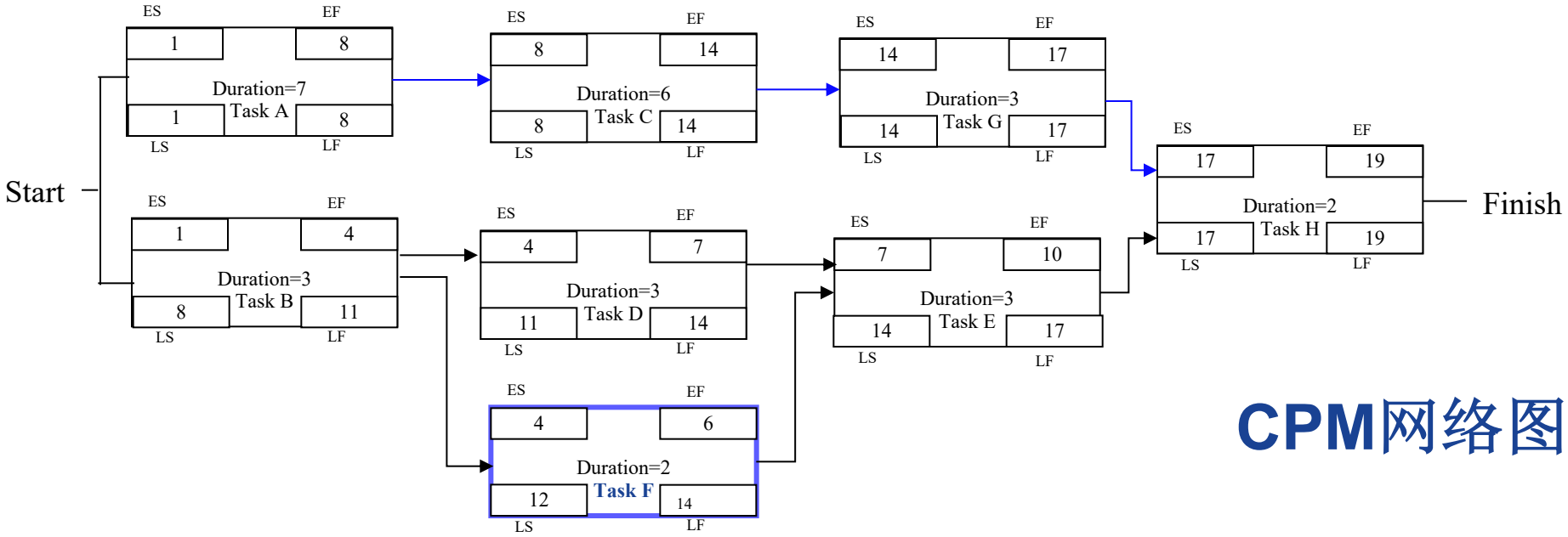
基本方法 → 关键路径法 7.7.1 → 逆推法



基本方法 → 关键路径法 7.7.1 → 逆推法



基本方法 → 关键路径法 7.7.1 → 逆推法



CPM网络图

FF(F)=ES(E)-EF(F)-lag=7- 6-0=1

TF(F)=LS(F)-ES(F)=LF(F)-EF(F)=8

关键路径:A->C->G->H

Cp Path:19

软件项目进度计划 学习要点

MIMA

- **一、进度管理基本概念**
- **二、任务历时估算**
- **三、进度计划编排**
- 四、项目进度模型
- 五、案例分析
- 六、课程实践

单选题 1分

设置

MIMA

“软件编码完成之后，我才可以对它进行软件测试”，这句话说明了哪种依赖关系？

A 强制性依赖关系

B 软逻辑关系

C 外部依赖关系

提交

单选题 1分

设置

MIMA

时间是项目规划中灵活性最小的因素。

A 对

B 不对

提交

单选题 1分

设置

外部依赖关系又称强制性依赖关系，指的是项目活动与非项目互动之间的依赖关系。

A 不对

B 对

提交

单选题 1分

设置

MIMA

浮动是在不增加项目成本的条件下，一个活动可以延迟的时间量。（ ）

A 错

B 对

提交

多选题 1分

设置

MIMA

下面说法中不正确的是（　）

A EF=ES+lag

B LS=LF-duration

C EF=ES+duration

D TF=LS-ES=LF-EF

提交