

SOFTWARE PROJECT MANAGEMENT (6)

Du Yugen

ygdu@sei.ecnu.edu.cn

Chapter 6 Activity Planning

软件项目策划

OBJECTIVES

When you have completed this chapter you will be able to:

- **Produce an activity plan for a project;**
- **Estimate the overall duration of a project;**
- **Create a critical path and a precedence network for a project.**

6.1 introduction

- 软件项目活动策划**目的**
 - 保证在**正确的时间**有**正确的资源**可用
 - 避免**不同的活动**在**相同的时间**竞争**相同的资源**
 - 为每个人员分配任务
 - 实际的进度可以有**标准**进行衡量
 - 产生成本效益计划,现金流预测
 - 根据实际情况, 调整项目
 - **可行性研究**
 - 协调人员

6.2 when to play何时计划

- **计划是一个渐进、迭代的过程**
- **每次迭代，细节更丰富，也更准确**
- **每一此迭代的目的有所不同：**
 - **可行性研究阶段：时间和成本估计**
 - **项目开始：生成活动计划保证资源分配和资金流**
 - **项目进行中：不断监控和调整项目计划**

6.3 project schedules项目进度表

在项目开始之前,或较大项目的某个阶段开始之前,给出每项活动何时开始和何时完成的日期,并说明每项活动要求多少资源,这样的项目计划就称为项目进度表.创建项目进度表包括四个阶段:

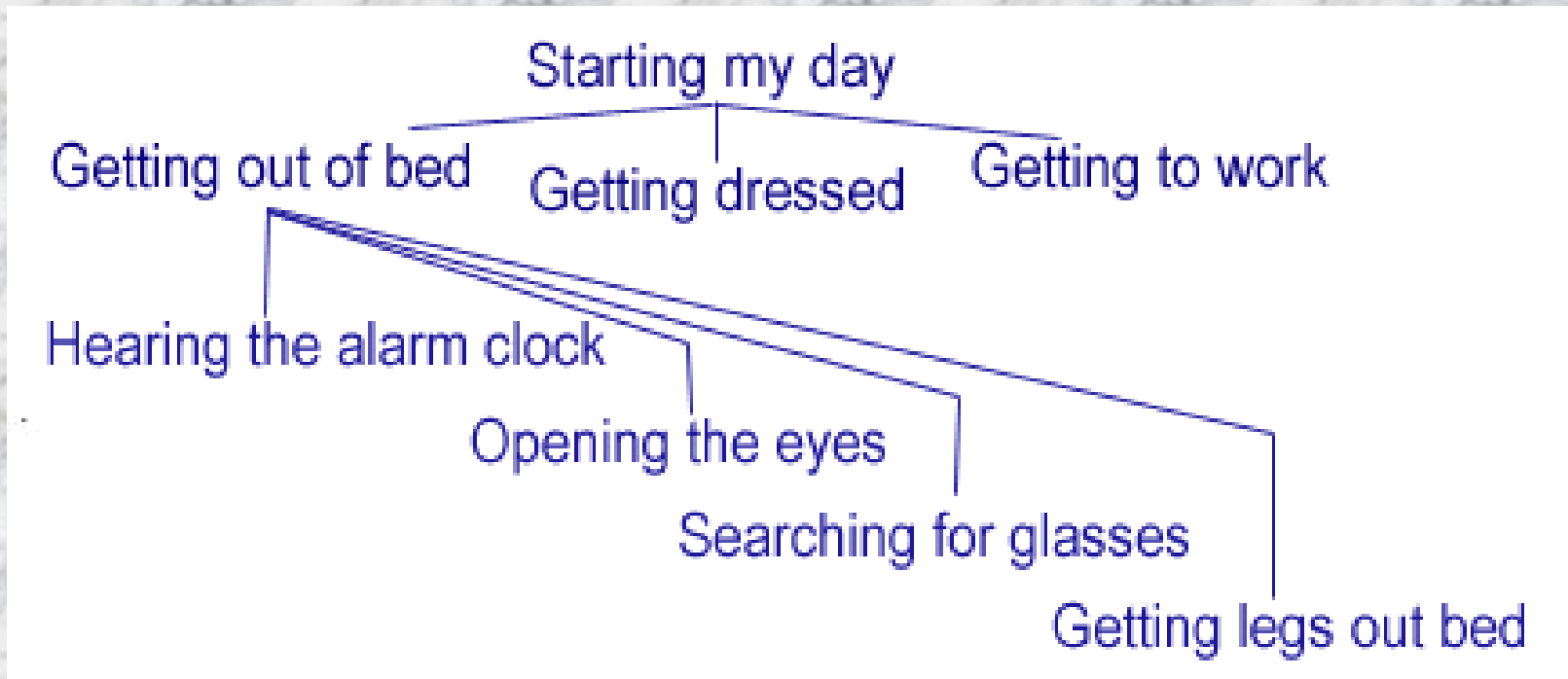
1. 构造一个理想的活动计划
2. 活动的风险分析
3. 资源分配
4. 产生进度表

6.4 projects and activities项目和活动

- **项目由一组相关的活动构成**
- **项目在至少有一个活动准备开始时才开始**
- **项目中所有的活动结束项目才结束**
- **活动必须有一个清晰的开始和清晰的结束点，一般以产生一个可交付物为标志**
- **活动需要资源，并假定在活动中资源的需求水平保持一致**
- **活动的持续时间必须可预测**
- **活动之间有优先顺序**

确定项目中的活动

- **活动确定方法**
 - 自顶向下的分解——WBS
- **问题：请给出每天起床的WBS**



6.5 WBS构造方法

- 识别出项目（包括项目管理活动）中的**主要交付物**。
- 问题：学院工资系统的交付物有哪些？
- 主要交付物总是根据项目如何真正地组织来定义：
 - 项目生命周期的各个阶段可以作为第一层次，并将项目交付物作为第二层次
 - 每个分支的组织方法可以不一样

WBS验证

- 判断对这一层次是否能够**对成本和日期进行评估**，如果不能，继续分解，否则该分枝分解结束
- 识别交付物的组成部分。组成部分必须是**实际的，可验证的**部件
- 验证分解的正确性
 - 是否被分解的条目下对子项目是**充分必要的**？如果不是，则需要增，删或修改
 - 是否每一条目被**清晰准确**的定义
 - 是否每一条目能够被合适的计划，成本规划，被分配给一个合适的组织

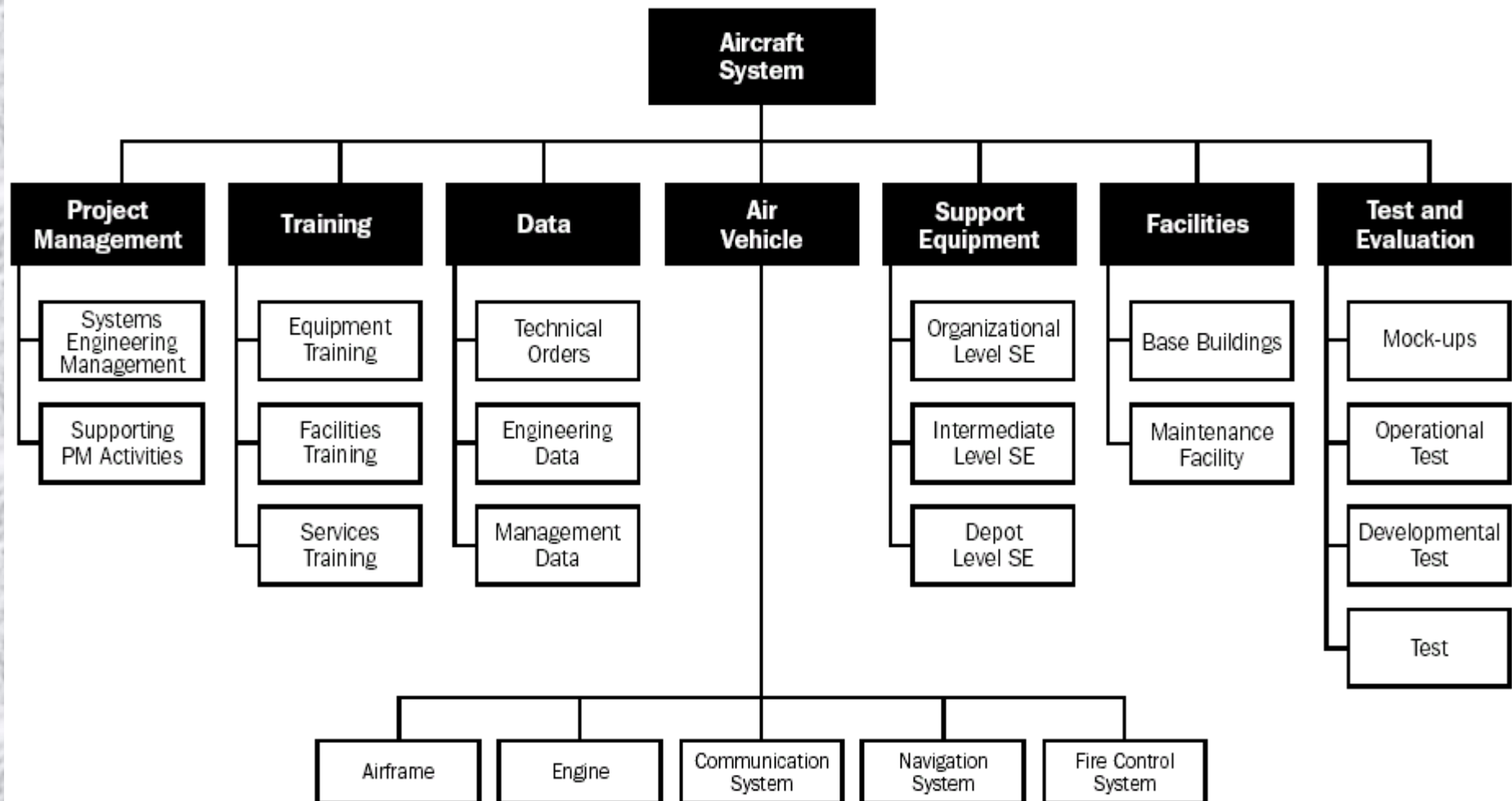
WBS

- **输出**

- **WBS:最底下的一层被称为工作包, Work package**

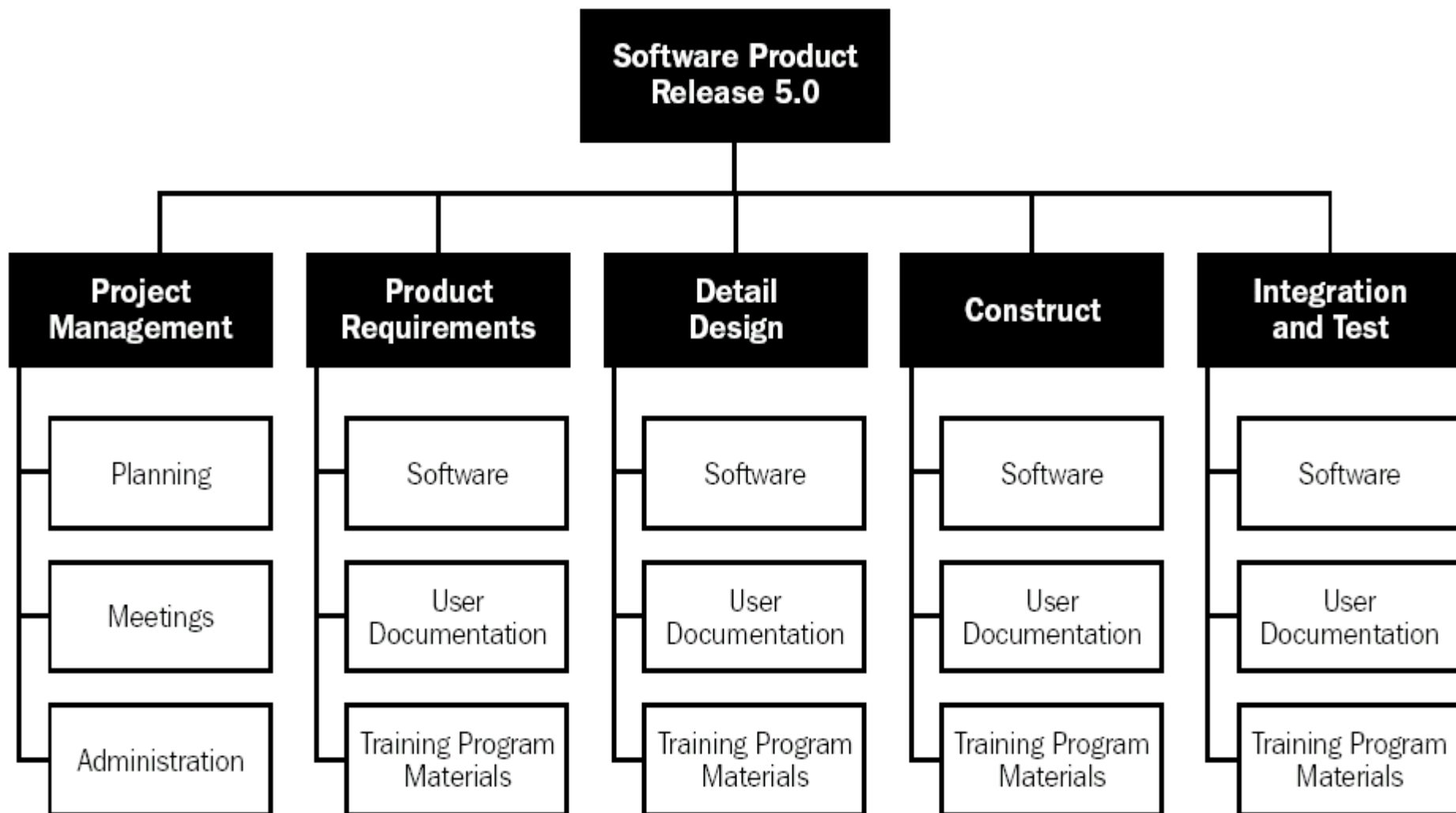
必须与下面概念相区别:

- **Contractual WBS, CWBS(seller to buyer)**
 - **Organizational breakdown structure, OBS**
 - **Resource breakdown structure, RBS**
 - **Bill of materials, BOM**
 - **Project breakdown structure, PBS**

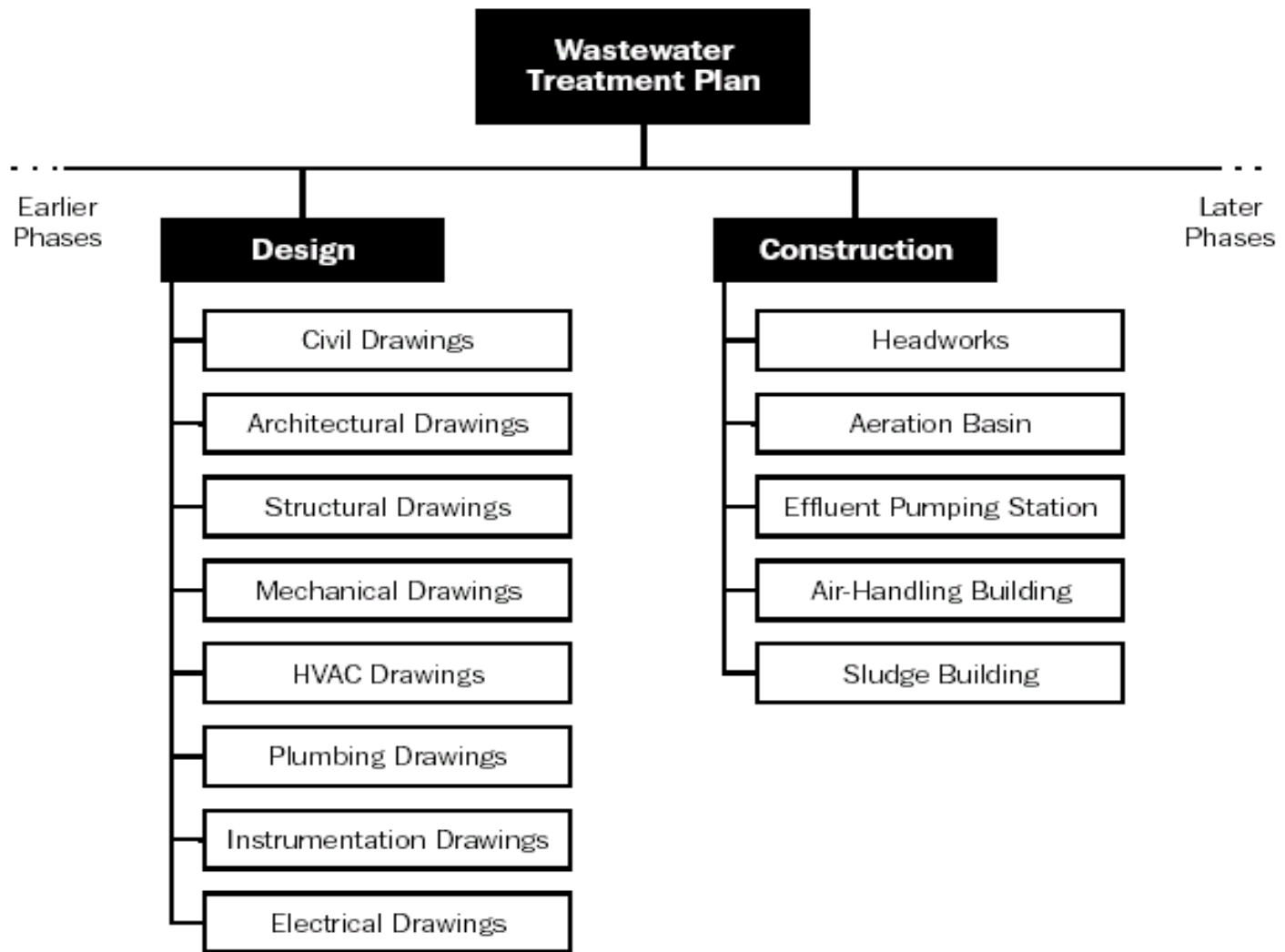


This WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.

Figure 5–2. Sample Work Breakdown Structure for Defense Material Items



This WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.

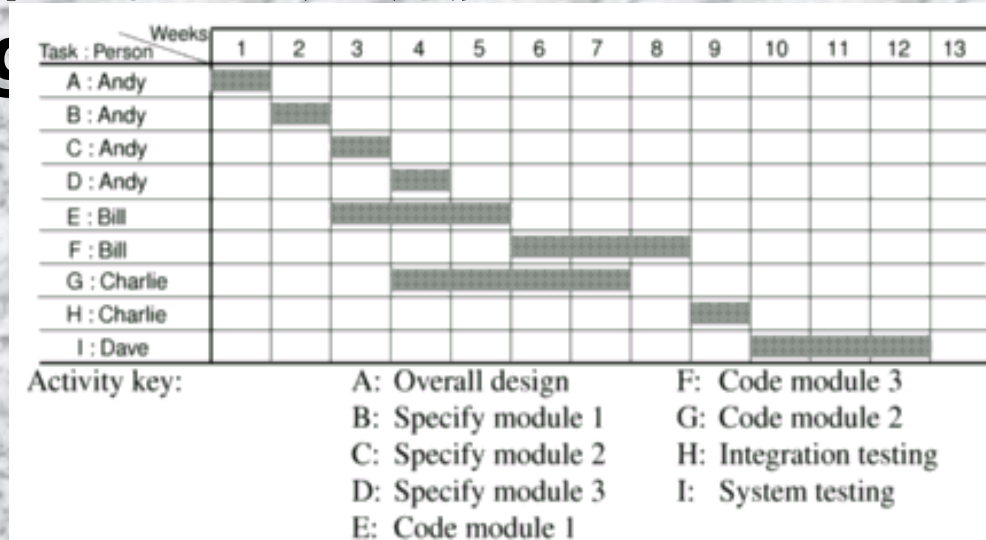


This WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.

Figure 5-4. Sample Work Breakdown Structure for Wastewater Treatment Plant

6.6 Sequencing and scheduling 活动顺序和调度

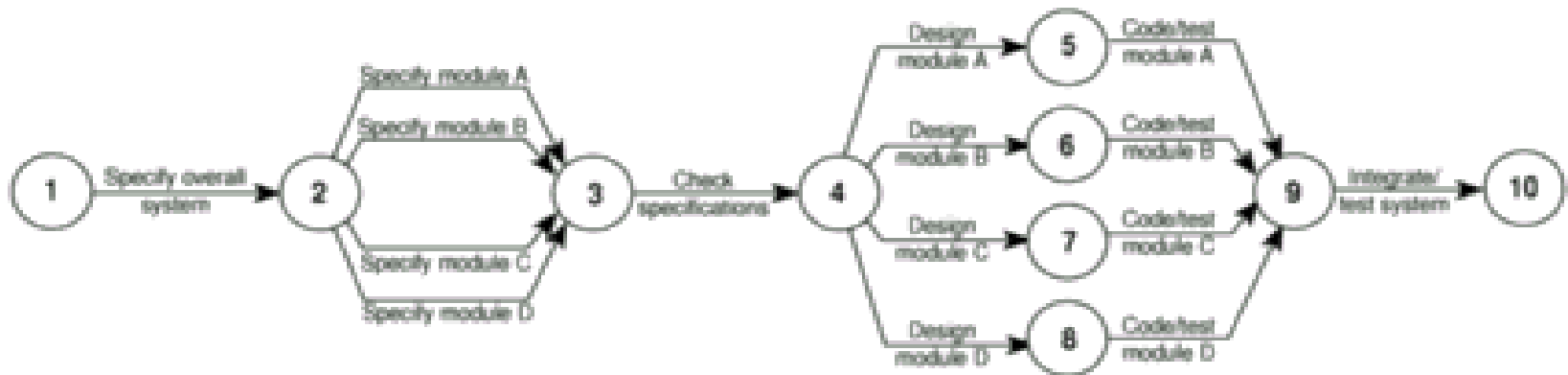
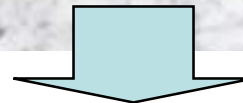
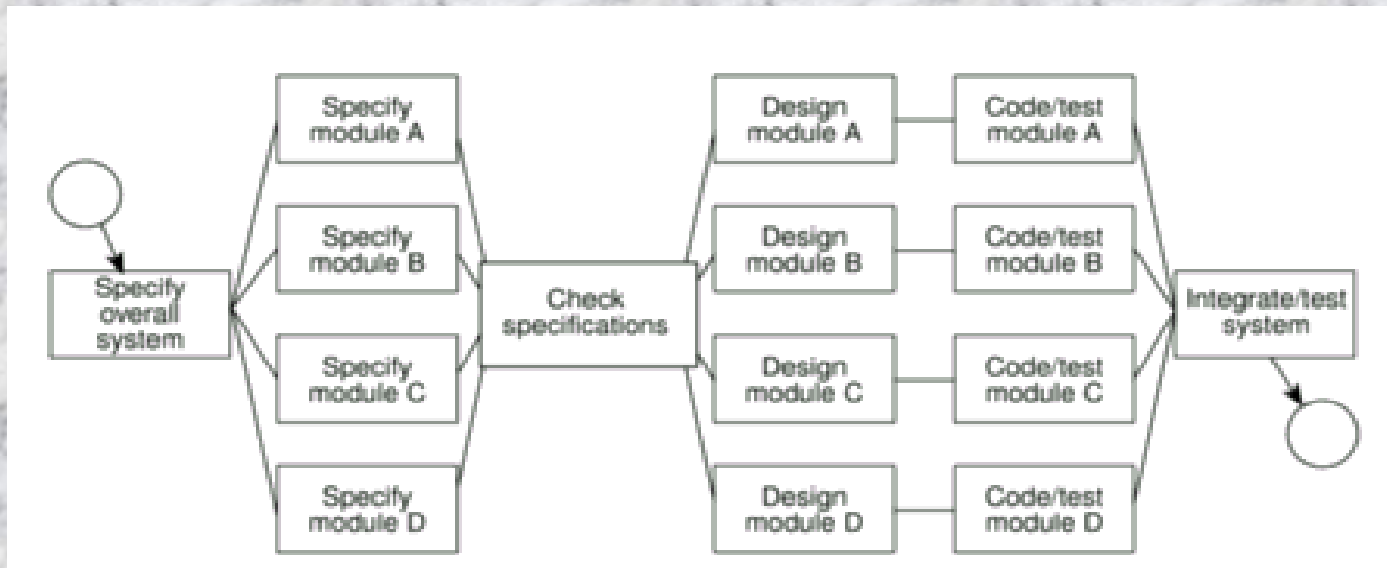
- 在整个项目中，我们需要一份能清晰描述活动发生的时间和所需要的资源的计划。
 - 仅仅考虑活动之间的顺序关系，我们称之为活动排序（Sequencing）——逻辑设计
 - 考虑资源约束，我们称为调度（Scheduling）



6.7 Network planning model 网络计划模型

- 将项目活动及其关系可以表示成网络
- 该项技术起源于二十世纪50年代
- CPM(关键路径法, Critical Path Method)
- PERT(计划评审技术, Program Evaluation Review Technique)
- 采用AOA(Activity on arrow)方法
- 优先网络(Precedence Networks)是上述两种模型的变种,采用AON(Activity on Node)
- 三种模型是类似的, 因而经常被有些人通称为CPM

Activity network→CPM的例子



6.8 CPM的构造方法

- 项目网络只有一个起点only one start node
- 项目网络只有一个终点only one end node
- 连接(表示一个活动)有持续时间a link has duration
- 节点无持续时间nodes have no duration
- 时间从左到右time moves from left to right
- 节点顺序编号nodes are numbered sequentially
- 网络不包含圈a network may not contain loop

CPM构造方法

- 网络不包含悬点(dangle)

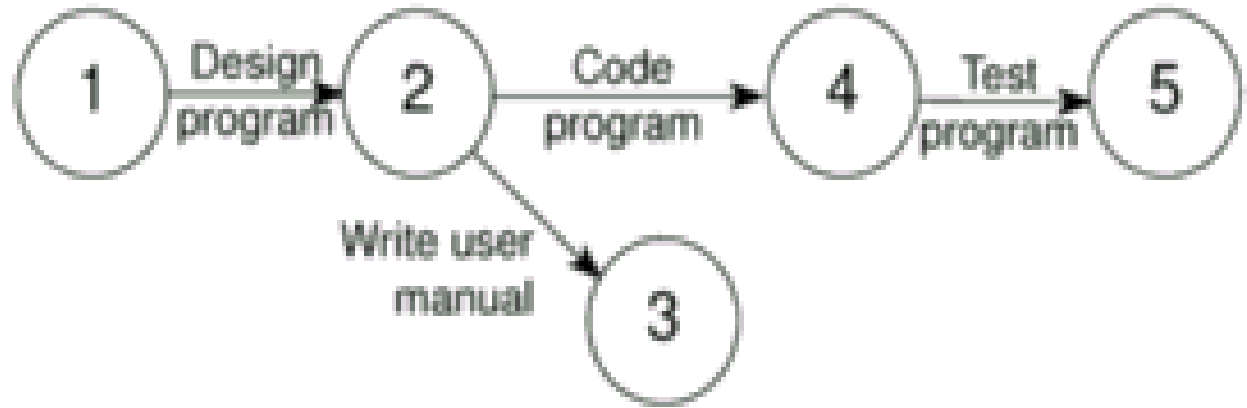
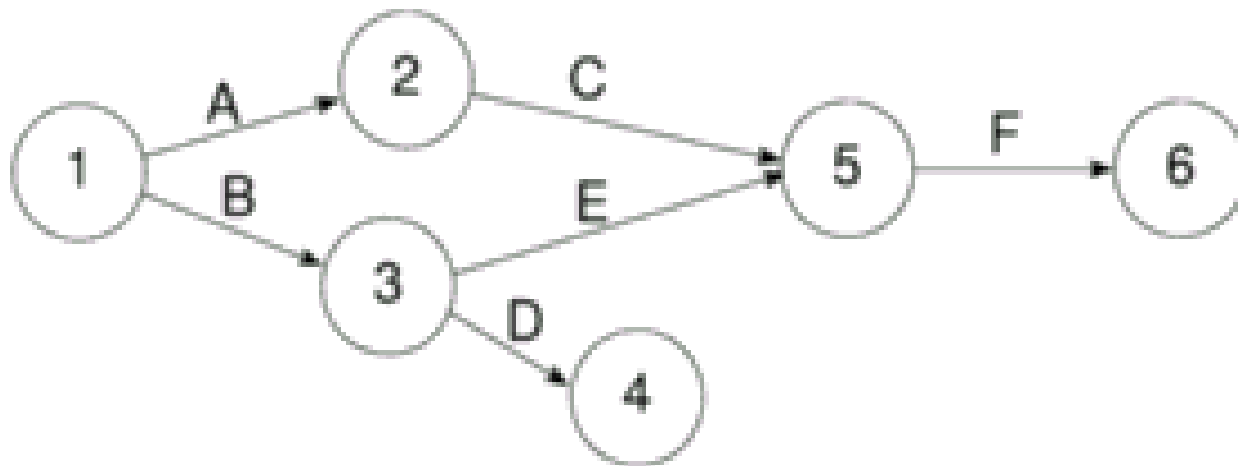


Figure 6.11 *A dangle.*

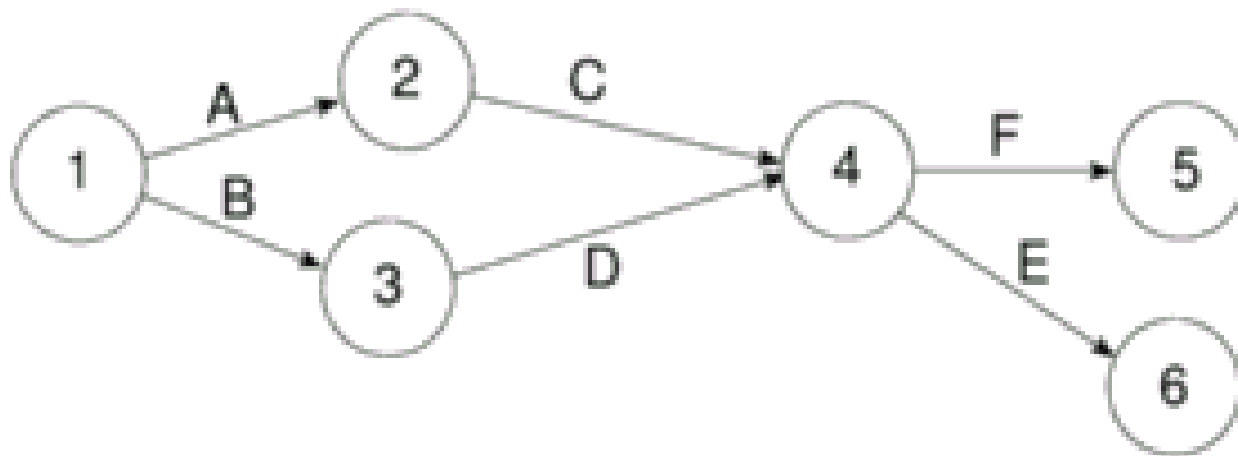
- 前置是指某活动紧前活动Precedents are the immediate preceding activities

练习下列网络图中哪些存在错误

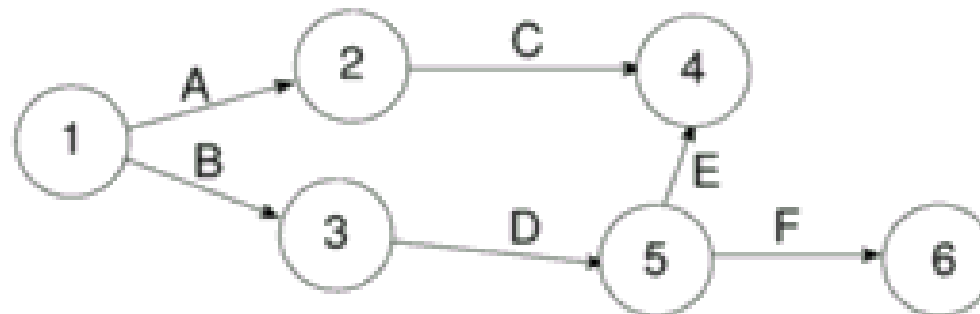
(a)



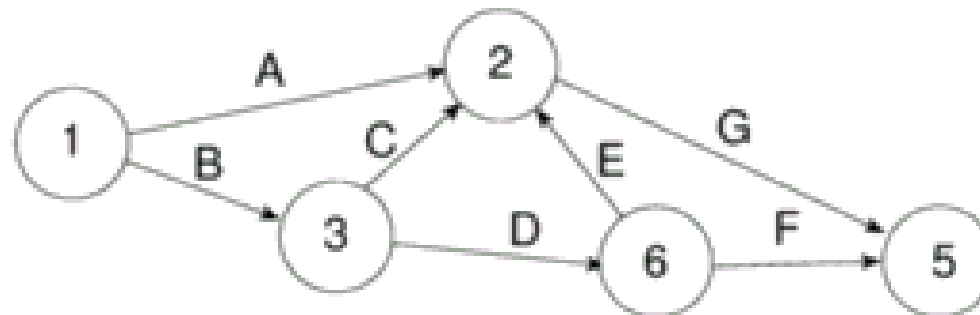
(b)



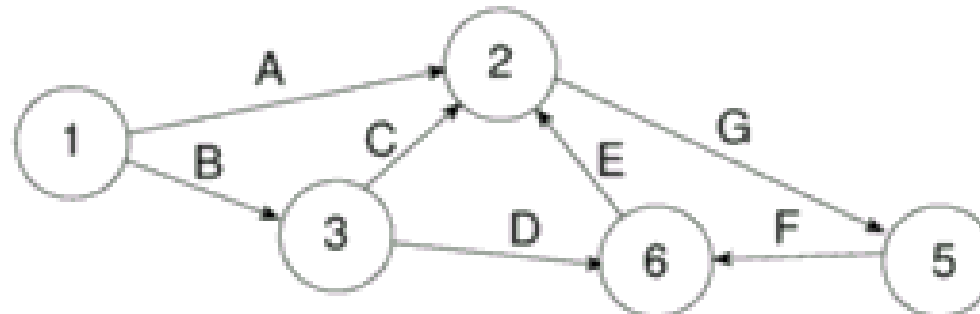
(c)



(d)



(e)



6.9 虚拟作业,哑活动(dummy activity)

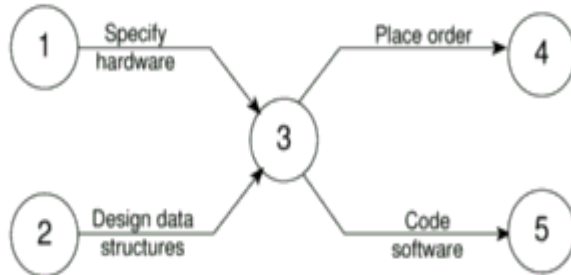


Figure 6.13 Two paths with a common node.

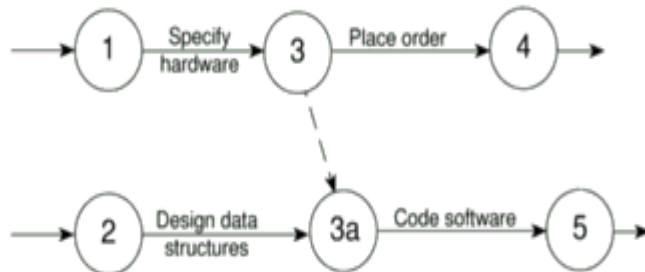
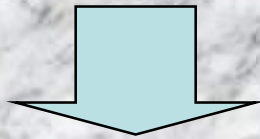


Figure 6.14 Two paths linked by a dummy activity.

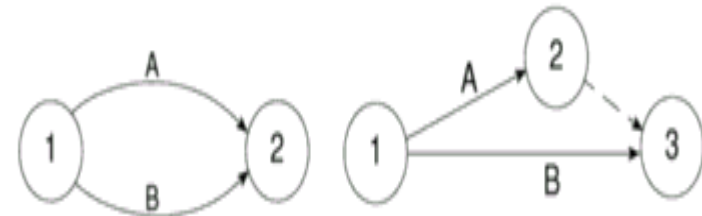
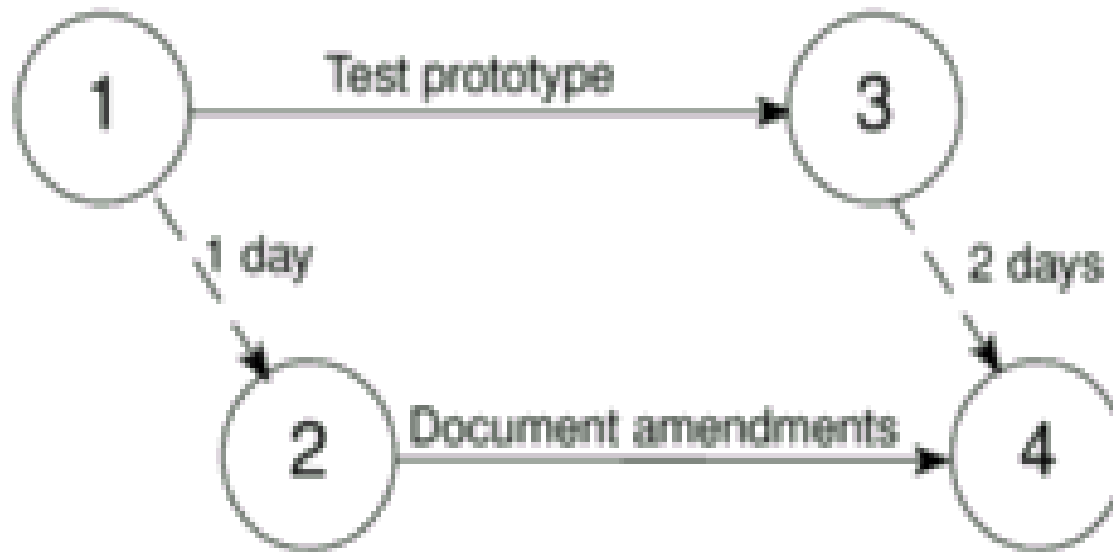


Figure 6.15 Another use of a dummy activity.

6.10 representing lagged activities 延时活动的表达

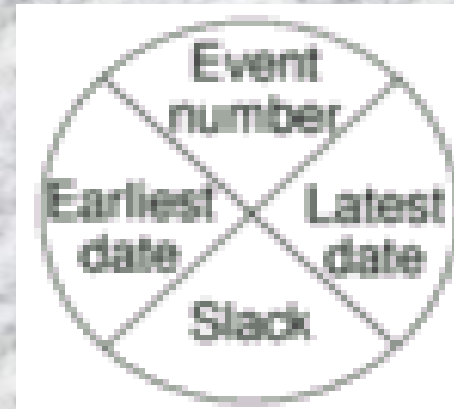


6.11 adding the time dimension加入时间因素

- **关键路径法主要关注两个目标：**
 - **尽快完成整个项目**
 - **识别那些一旦延期将对整个项目周期产生影响的**活动
- **对每一个活动赋予持续时间后，我们可以采用前向路径(forward pass)计算项目和各个活动的最早发生(完成/开始)时间，采用反向路径法(backward pass)计算项目和活动的最晚发生(完成/开始)时间**

CPM中节点表示

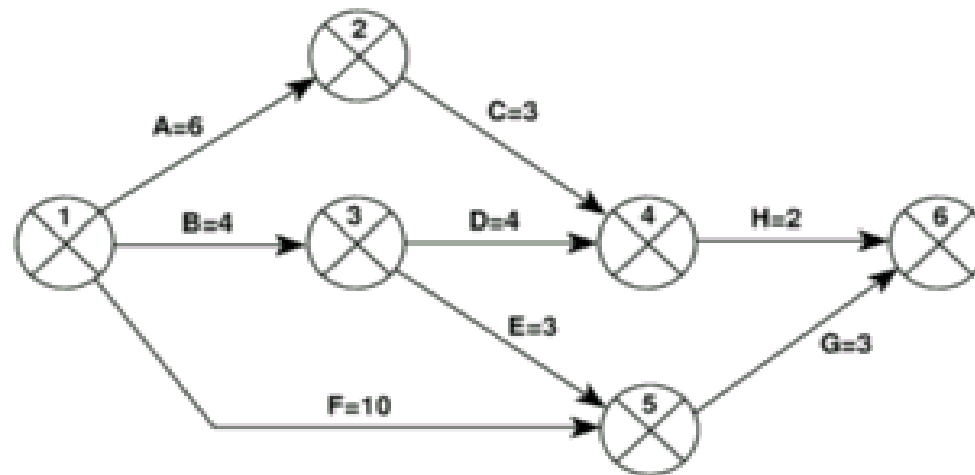
- **Event number: 事件编号**
- **Earliest Date: 最早(开始/完成)时间**
- **Latest Date: 最晚(开始/完成)时间**
- **Slack: 松弛时间**



例子

Table 6.1 *An example project specification with estimated activity durations and precedence requirements*

Activity		Duration (weeks)	Precedents
A	Hardware selection	6	
B	Software design	4	
C	Install hardware	3	A
D	Code & test software	4	B
E	File take-on	3	B
F	Write user manuals	10	
G	User training	3	E, F
H	Install & test system	2	C, D

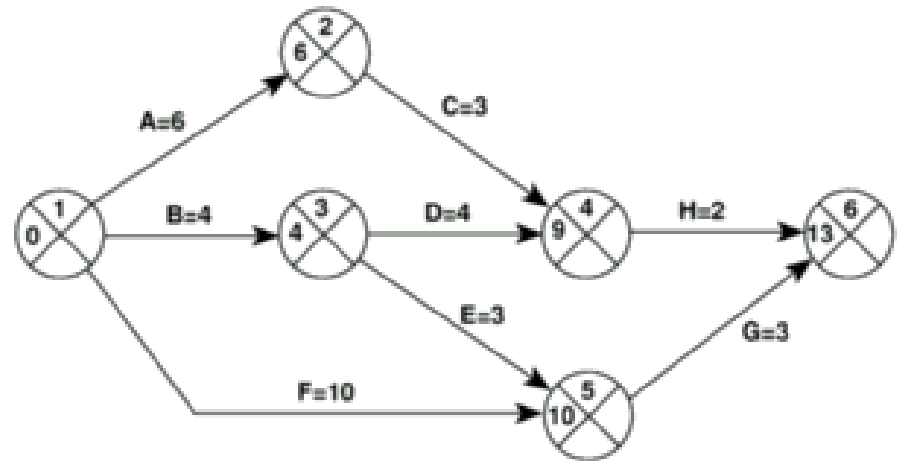


6.12 forward pass 前向路径计算

$$EFT_1 = 0$$

$$EFT_j = \max_i (EFT_i + D_{ij})$$

节点*i*是以*j*为结束节点的所有活动的开始节点



- 一个活动的最早开始时间(EST)就是它的开始节点的最早完成时间

- 一个活动的最早完成时间(EFT)就是它的结束节点的最早完成时间

Table 6.2 The activity table after the forward pass

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish date	Latest finish date	Total float
A	6	0		6		
B	4	0		4		
C	3	6		9		
D	4	4		8		
E	3	4		7		
F	10	0		10		
G	3	10		13		
H	2	9		11		

6.13 backward pass后向路径计算

$$LFT_{last} = EFT_{last}$$

$$LFT_i = \min_j (LFT_j - D_{ij})$$

节点*j*是以*i*为开始节点的所有活动的结束节点

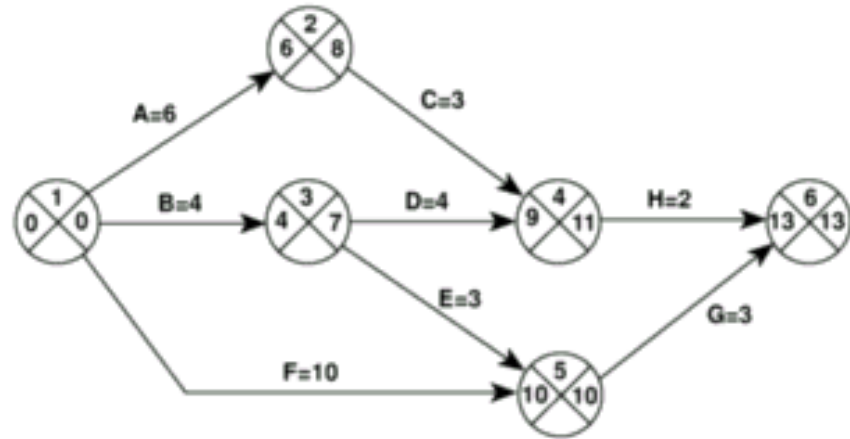
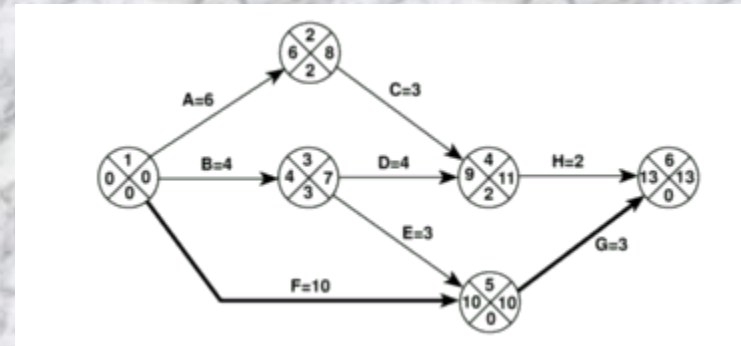


Table 6.3 The activity table following the backward pass

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish date	Latest finish date	Total float
A	6	0	2	6	8	
B	4	0	3	4	7	
C	3	6	8	9	11	
D	4	4	7	8	11	
E	3	4	7	7	10	
F	10	0	0	10	10	
G	3	10	10	13	13	
H	2	9	11	11	13	

6.14 关键路径识别

- 最早开始时间和最晚开始时间的差称为**松弛时间**(Slack), 它表示一个事件推迟多少时间可以不影响项目的结束。Slack为0的事件为关键事件, 将关键事件连接起来的最长路径为关键路径



- 关键路径的意义:
 - 必须保证关键路径上的资源和关键路径活动顺利执行
 - 要缩短整个项目周期, 必须缩短关键路径

6.15 Activity float活动的浮动时间

- 具有slack事件的活动具有浮动时间。总浮动是活动的最早结束时间和最迟结束时间的差。

Table 6.4 *The activity schedule showing total float for each activity*

<i>Activity</i>	<i>Duration (weeks)</i>	<i>Earliest start date</i>	<i>Latest start date</i>	<i>Earliest finish fate</i>	<i>Latest finish date</i>	<i>Total float</i>
A	6	0	2	6	8	2
B	4	0	3	4	7	3
C	3	6	8	9	11	2
D	4	4	7	8	11	3
E	3	4	7	7	10	3
F	10	0	0	10	10	0
G	3	10	10	13	13	0
H	2	9	11	11	13	2

活动的浮动时间

- 各个活动的浮动时间是相关的，如果某个活动用了浮动时间，则后续的活动可能就没有了浮动时间。其它浮动时间：
 - **自由浮动(free float)**: 活动的最早结束时间和紧接活动的最早开始时间的差为自由浮动时间，它不影响其它活动。
 - **阻碍浮动(interfering float)**: 活动的自由浮动与总浮动之差，它反映了总浮动使用后，活动还能被延时多少而不影响整个项目的结束时间

6.16 缩短项目时间和识别关键活动

- **缩短项目时间**
 - 活动持续时间的缩短
 - 整个项目过程的重组
- **识别关键活动**
 - 关键路径上的活动
 - 近似关键路径near-critical path: 浮动时间是整个关键路径的10%的路径, 近似关键路径很容易变为关键路径

6.17 优先网络

- 优先网络(precedence networks)
- 优先网络使用节点代表活动，而连接代表依赖关系

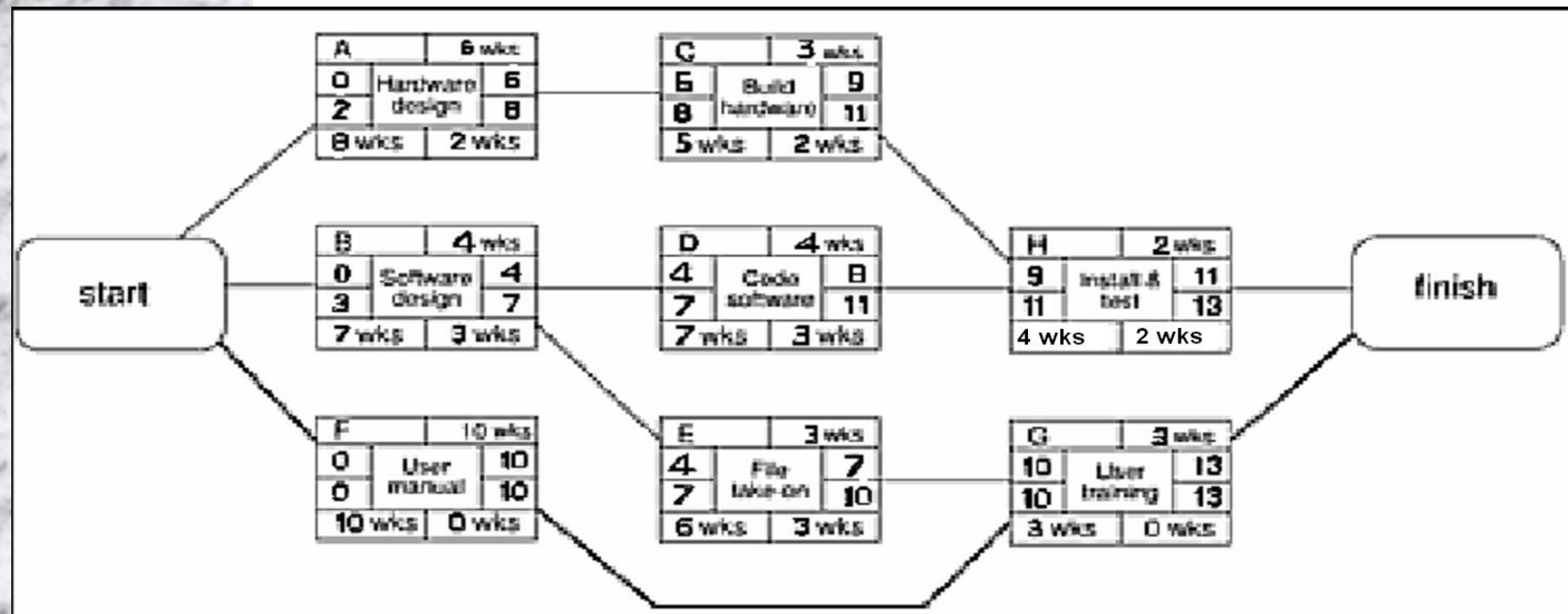
Activity label		Duration	
Earliest start	Activity description	Earliest finish	
Latest start		Latest finish	
Activity span		Float	

例子

Table 6.1

An example project specification with estimated activity durations and precedence requirements

Activity	Duration (weeks)	Precedents
A Hardware selection	6	
B Software design	4	
C Install hardware	3	A
D Code & test software	4	B
E File take-on	3	B
F Write user manuals	10	
G User training	3	E, F
H Install & test system	2	C, D

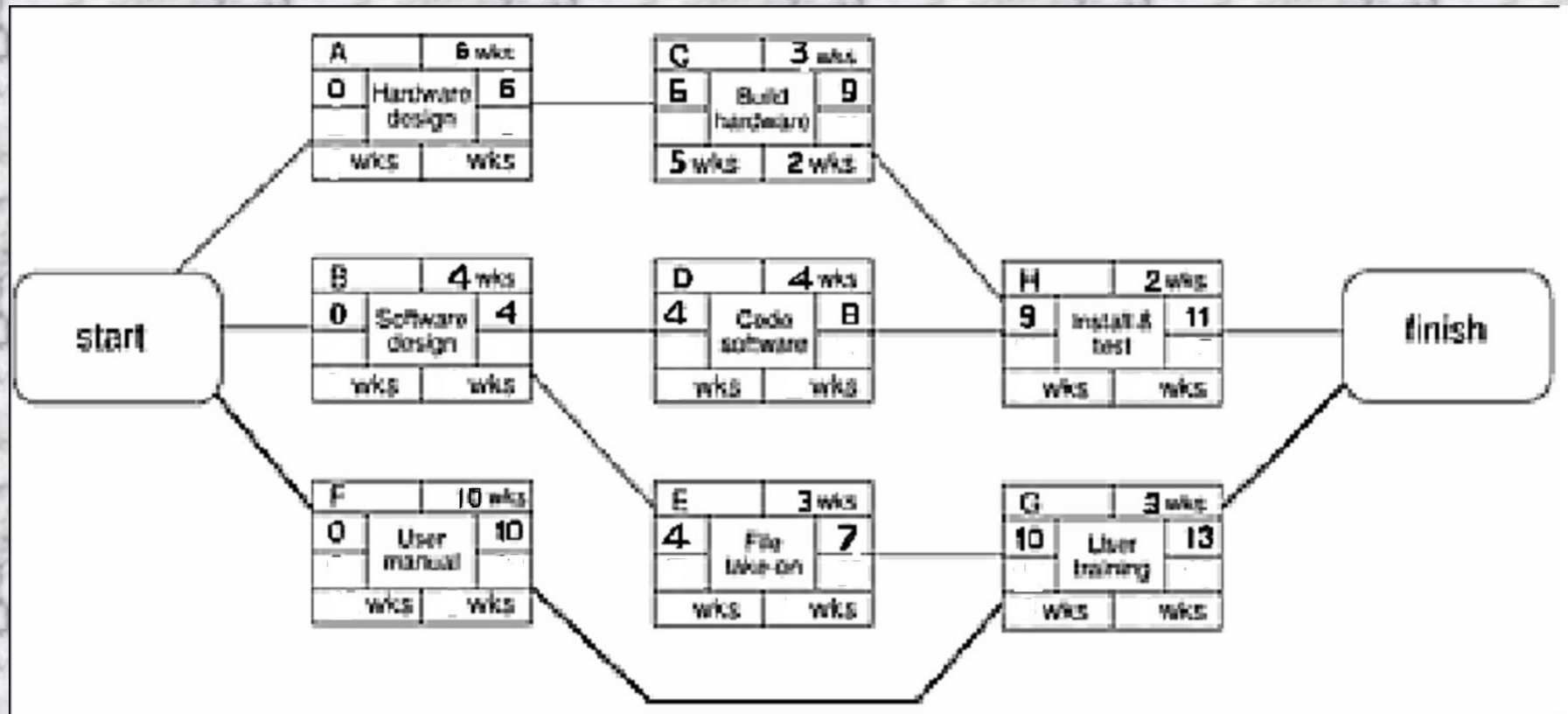


6.17.1 forward pass 前向路径计算

$$EST_{start} = 0, EFT_{start} = D_{start}$$

$$EFT_j = \max_i (EFT_i + D_i), EST_j = \max_i EFT_i$$

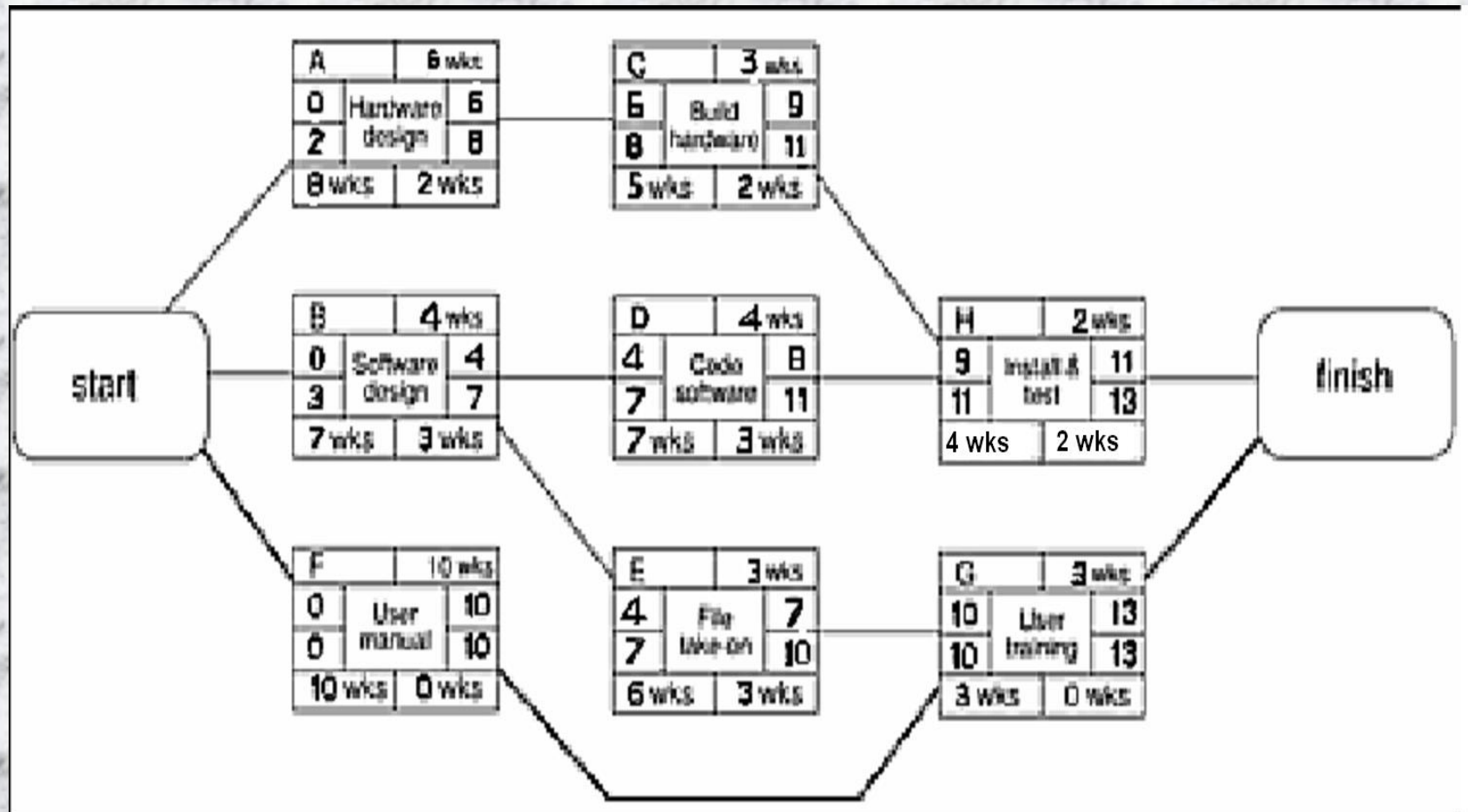
活动是活动的所有前置活动



6.17.2 backward pass后向路径计算

$$LFT_{last} = \max_{last} EFT_{last}, LFT_i = \min_j (LFT_j - D_{ij})$$

活动*j*是所有以为前置的活动

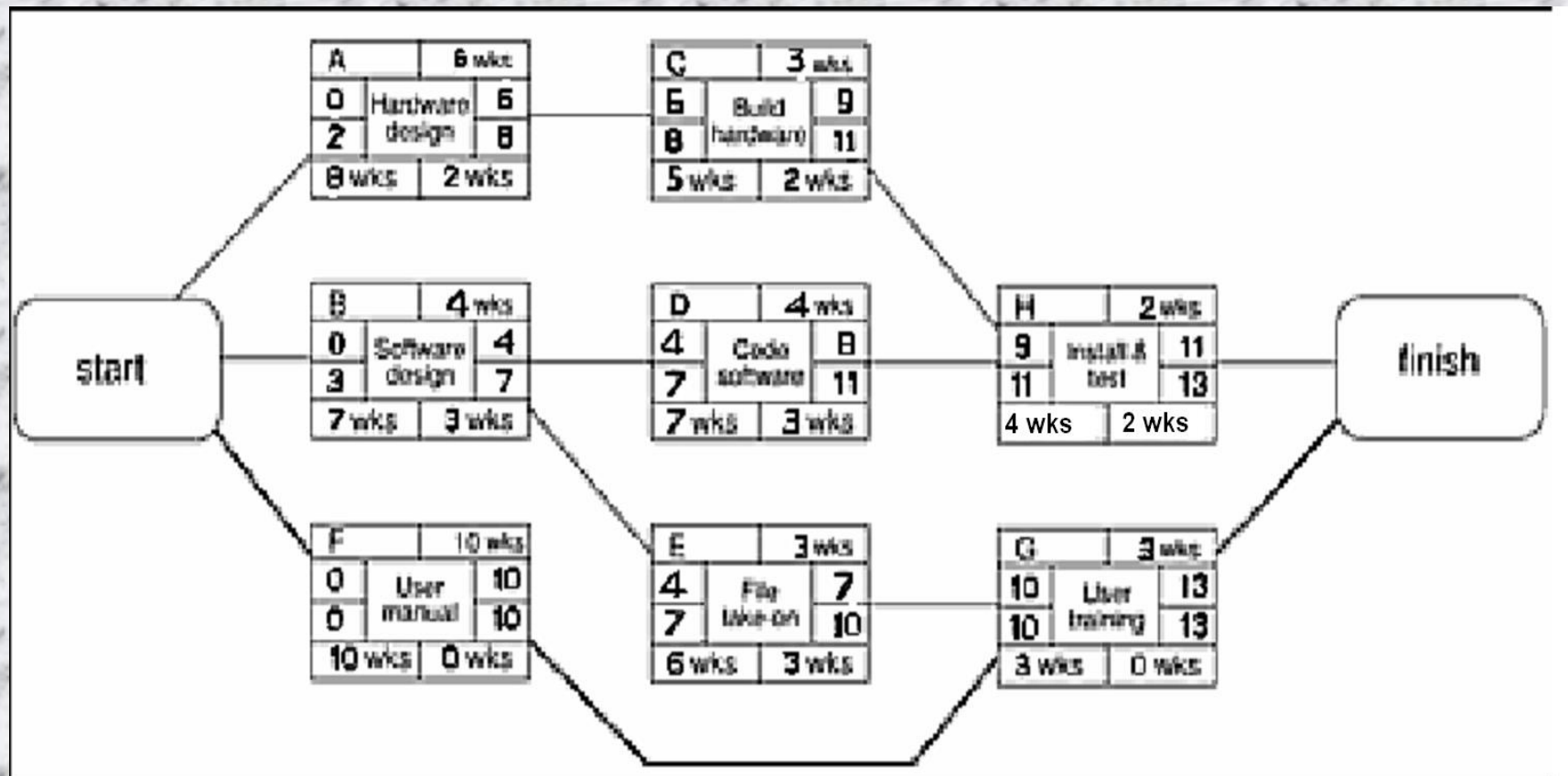


6.17.3 关键路径识别

$$Float_i = LFT_i - EFT_i = LST_i - EST_i$$

$$span_i = Float_i + D_i$$

i 是活动



练习

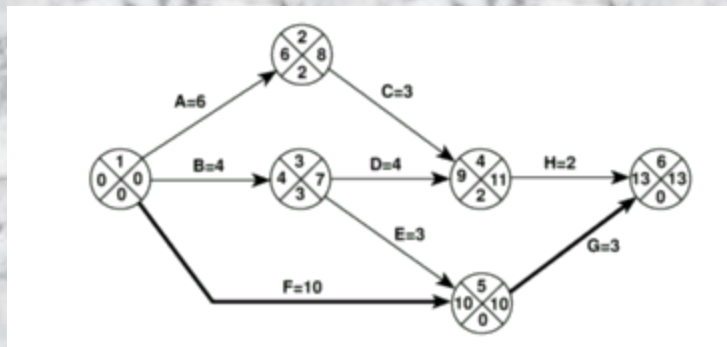
• 计算自由浮动和阻碍浮动时间

Table 6.4 The activity schedule showing total float for each activity

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish date	Latest finish date	Total float
A	6	0	2	6	8	2
B	4	0	3	4	7	3
C	3	6	8	9	11	2
D	4	4	7	8	11	3
E	3	4	7	7	10	3
F	10	0	0	10	10	0
G	3	10	10	13	13	0
H	2	9	11	11	13	2

■ **自由浮动(free float)**: 活动的最早结束时间和紧接活动的最早开始时间的差为自由浮动时间, 它不影响其它活动。

■ **阻碍浮动(interfering float)**: 活动的自由浮动与总浮动之差, 它反映了自由浮动使用后, 活动还能被延时多少而不影响整个项目的结束时间



Activity	Total float	Free float	Interf. float
A	2	0	2
B	3	0	3
C	2	0	2
D	3	1	2
E	3	3	0
F	0	0	0
G	0		
H	2		