

CHAPTER 9

Software Project Management

Outline

- **Estimating Cost**
- **Planning**
- **Team organization**
- **Configure management**
- **Management standards**

何为项目 (Project)?

□ 项目是指为创建一个唯一的产品，或者提供唯一的服务而进行的努力活动。

□ 项目示例

- “阿波罗登月”项目
- “**Windows 2000**”项目
- “载人飞船”项目
- “同济大学学生管理信息系统”项目

项目特点

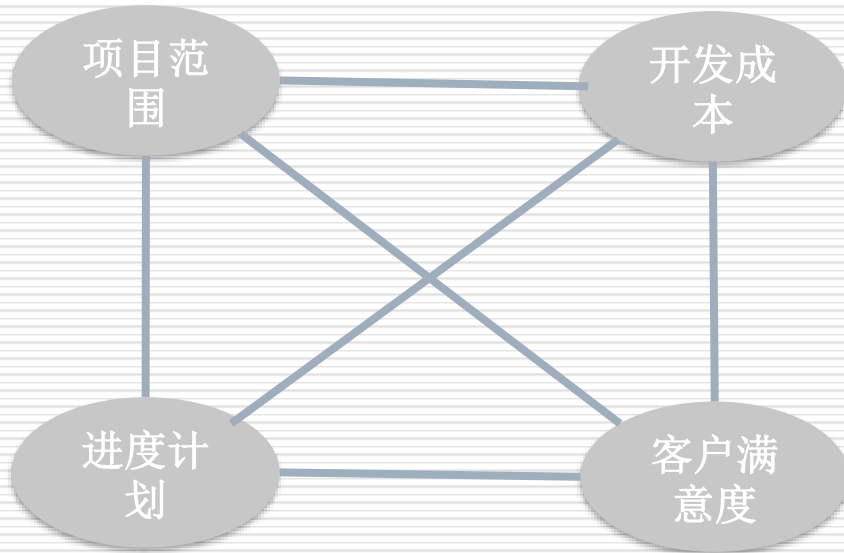
- 目标性：获得预期的结果
- 周期性：在限定时间完成
- 约束性：有限资源如人员、成本、工具等
- 不确定性：项目的实施及其结果不确定

软件项目的特点

- 对象：作为逻辑产品的软件
- 过程：不是以制造为主，无重复生产过程
- 属性：成本、进度、质量难以度量和估算
- 易变性：软件需求通常难以确定且经常变
- 复杂性：作为逻辑产品，复杂性非常高

软件项目管理

□ 项目成功的制约因素



□ **软件项目管理**：是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对成本、人员、进度、质量、风险等进行分析和管理的活动。

□ 对软件项目开发过程中所涉及的过程、人员、产品、成本和进度等要素进行度量、分析、规划、组织和控制的过程，以确保软件项目按照预定的成本、进度、质量要求顺利完成。

软件项目管理的任务

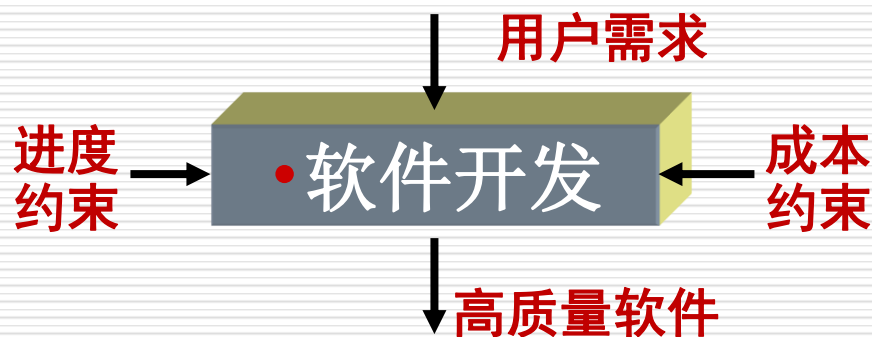
□ 按照预定的进度、成本和质量，开发出满足用户要求的软件产品

■ 用户需求

■ 确保质量

■ 成本限制

■ 进度限制



Goal of Project Management

To complete a project:

- ✓ **On time**
- ✓ **On budget**
- ✓ **With required functionality**
- ✓ **To the satisfaction of the client**
- ✓ **Without exhausting the team**

Why is management important?

- ✓ Software engineering is an **economic activity** and therefore is subject to economic, non-technical constraints.
- ✓ Well-managed projects sometimes fail. Badly managed projects inevitably fail.
- ✓ The objective of this lecture is to introduce **management activities** rather than teach you to be managers.
- ✓ You can only learn to manage by managing.

软件项目管理重要性

- 大型软件工程项目失败 -- 使人们逐渐认识到软件项目管理的重要性和特殊性。
- 失败的原因 -- 不是软件开发工程师无能，而主要是管理不善。
- 所谓管理 -- 就是通过计划、组织和控制等一系列活动，合理地配置和使用各种资源，以达到既定目标的过程。
- 软件项目管理 -- 先于任何技术活动之前开始，并且贯穿于软件的整个生命周期之中。
- 软件项目管理 -- 从一组项目计划活动开始，其基础是工作量估算和完成期限估算。

软件管理的困难性

- **The product is intangible.**
- **The product is flexible.**
- **The software development process is not standardized.**
- **Most software projects are one-off projects.**

Management activities

- ✓ **Proposal writing**
- ✓ **Project costing**
- ✓ **Project planning and scheduling**
- ✓ **Project monitoring and reviews**
- ✓ **Personnel selection and evaluation**
- ✓ **Report writing and presentations**

知识领域--管理工具和技术

| 知识领域 | 工具和技术 |
|--------|--|
| 集成管理 | 项目挑选方法、项目管理方法论、利益相关者分析、项目章程、项目管理技术、项目管理软件、变更请求、变更控制委员会、项目评审会议、经验教训会议 |
| 范围管理 | 范围说明、工作分解结构、工作说明、需求分析、范围管理计划、范围验证技术、范围变更控制 |
| 时间管理 | 甘特图、项目网络图、关键路径分析、赶工、快速跟踪、进度绩效测量 |
| 成本管理 | 净现值、投资回报率、回收分析、挣值管理、项目组合管理、成本估算、成本管理计划、成本基线 |
| 质量管理 | 质量控制、核减清单、质量控制图、帕累托图、鱼骨图、成熟度模型、统计方法 |
| 人力资源管理 | 激励技术、同理聆听、责任分配矩阵、项目组织图、资源柱状图、团队建设练习 |
| 沟通管理 | 沟通管理技术、开工会议、冲突管理、传播媒体选择、现状和进程报告、虚拟沟通、模版、项目网站 |
| 风险管理 | 风险管理计划、风险登记册、概率/影响矩阵、风险分级 |
| 采购管理 | 自制-购买分析、合同、需求建议书、资源选择、供应商评价矩阵 |

项目管理

- 成本估计
- 进度计划
- 人员组织
- 质量保证
- 配置管理
- 管理标准和规范

Cost Estimation

1. Static and single variable method

Target code

source code

$MM=4.7901I^{0.991}$ $MM=5.258I^{1.057}$, 对于所有应用

$MM=4.573I^{1.228}$ $MM=4.089I^{1.263}$, 对于控制类应用

$MM=4.495I^{1.068}$ $MM=7.054I^{1.019}$, 对于科学计算

$MM=2.895I^{0.784}$ $MM=4.495I^{0.781}$, 对于商业应用

$MM=12.039I^{0.719}$ $MM=10.078I^{0.811}$, 对实用程序

MM 为人月数, I 为代码的千行数

Cost Estimation

2. Static and multi-variable method

$$MM = C1 * MM1 + C2 * MM2 + \dots$$

Cost Estimation

3. Dynamic and multi-variable method

$$K(\text{ man-year})=L^3 C_k^{-3} t_d^{-4}$$

K: 开发需要的人力, 单位为人年

L: 为源代码行数

t_d: 开发需用的时间, 单位为年

C_k: 为技术水平常数

=2500, for bad development env.

=10000, for good development env.

=12500, for perfect development

Cost Estimation

4. Standard method(expert)

$$L = \frac{a + 4m + b}{6}$$

- 代码行技术的主要**优点**：代码是所有软件开发项目都有的“产品”，而且很容易计算代码行数。
- 代码行技术的**缺点**：源程序仅是软件配置的一个成分，用它的规模代表整个软件的规模似乎不太合理；用不同语言实现同一个软件所需要的代码行数并不相同。

Cost Estimation

5. COCOMO Model

$$MM = C_1 \times KLOC^a \times \prod_{i=1}^{15} f_i$$

$$MM = 3.2 \times KLOC^{1.05} \times \prod_{i=1}^{15} f_i \quad \text{有组织式}$$

$$MM = 3.0 \times KLOC^{1.12} \times \prod_{i=1}^{15} f_i \quad \text{半独立}$$

$$MM = 2.8 \times KLOC^{1.20} \times \prod_{i=1}^{15} f_i \quad \text{嵌入式}$$

MM 是开发工作量（以人月为单位）， C_1 是模型系数，
KLOC 是估计的源代码行数（以千行为单位），**a** 是模型指数，
 $f_i (i=1 \sim 15)$ 是成本因素。

Cost Factors

| Cost Factor | Very Low | Low | Nominal | High | Very High | Extra High |
|-------------|----------|------|---------|------|-----------|------------|
| RELY | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 | |
| DATA | | 0.94 | 1.00 | 1.08 | 1.16 | |
| CPLX | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| RUSE | | | 1.00 | 1.10 | 1.30 | 1.50 |
| TIME | | | 1.00 | 1.11 | 1.30 | 1.66 |
| STOR | | | 1.00 | 1.06 | 1.21 | 1.56 |
| VMVH | | 0.92 | 1.00 | 1.09 | 1.17 | |
| VMVT | | 0.93 | 1.00 | 1.07 | 1.16 | |
| TURN | | 0.87 | 1.00 | 1.07 | 1.15 | |
| ACAP | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 | |
| AEXP | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 | |
| PCAP | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 | |
| VEXP | 1.21 | 1.10 | 1.00 | 0.90 | | |

Cost Factors

| Cost Factor | Very Low | Low | Nominal | High | Very High | Extra High |
|-------------|----------|------|---------|------|-----------|------------|
| LEXP | 1.14 | 1.07 | 1.00 | 0.95 | | |
| MODP | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 | |
| TOOL | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 | |
| SCHED | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 | |
| SECU | | | 1.00 | 1.10 | | |

计划进度

- **项目管理者的目标**是定义全部项目任务，识别出关键任务，跟踪关键任务的进展状况，以保证能及时发现拖延进度的情况。为达到上述目标，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。
- **软件项目的进度安排**是通过把工作量分配给特定的软件工程任务，并规定完成各项任务的起止日期，从而将估算出的项目工作量分布于计划好的项目持续期内。

软件项目进度计划

- 制定和文档化软件项目计划，确保软件开发计划是可行、科学、符合实际的
 - 要对软件开发过程中的那些方面制定计划？
 - 制定软件项目的计划的基础和依据是什么？
 - 要考虑哪些方面的问题？
 - 如何确保计划是科学的和可行的？（软件度量）
 - 如何描述计划？
 - 利用哪些工具可辅助计划的制定？

Project scheduling

- **Split project into tasks and estimate time and resources required to complete each task.**
- **Organize tasks concurrently to make optimal use of workforce.**
- **Minimize task dependencies to avoid delays caused by one task waiting for another to complete.**
- **Dependent on project managers intuition and experience.**

Scheduling problems

- ❑ **Estimating the difficulty of problems and hence the cost of developing a solution is hard.**
- ❑ **Productivity is not proportional to the number of people working on a task.**
- ❑ **Adding people to a late project makes it later because of communication overheads.**
- ❑ **The unexpected always happens. Always allow contingency in planning.**

计划进度, 方法

➤ Gantt图(横道图)

➤ 工程网络

- ✓ Graphical notations used to illustrate the project schedule.
- ✓ Show project breakdown into tasks. Tasks should not be too small. They should take about a week or two.
- ✓ Activity charts show task dependencies and the the critical path.
- ✓ Bar charts show schedule against calendar time. (Gantt)

Gantt图

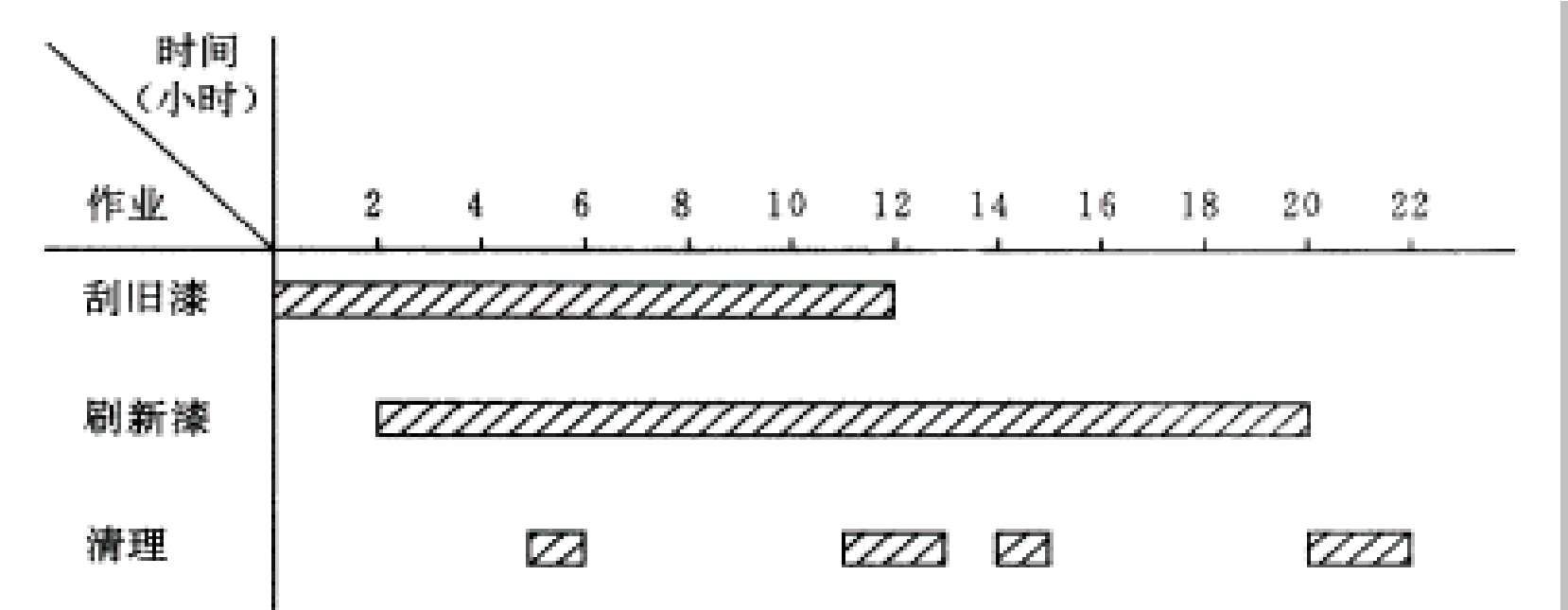
Gantt图（甘特图）是在1917年由亨利·甘特提出的，其思想简单，基本是一条线条图，横轴表示时间，纵轴表示活动(项目)，线条表示在整个期间上计划和实际的活动完成情况。

[illegible]

Gantt图

表 各道工序估计需用的时间(小时)

| 工序 墙壁 | 刮旧漆 | 刷新漆 | 清理 |
|----------|-----|-----|----|
| 1 或 3 | 2 | 3 | 1 |
| 2 或 4 | 4 | 6 | 2 |



Gantt图

Gantt图 (横道图)：Gantt图能很形象地描绘任务分解情况，以及每个子任务(作业)的开始时间和结束时间，因此是进度计划和进度管理的有力工具。它具有直观简明和容易掌握、容易绘制的优点，但是Gantt图也有三个**主要缺点**：

- (1) 不能显式地描绘各项作业彼此间的依赖关系；
- (2) 进度计划的关键部分不明确，难于判定哪些部分应当是主攻和主控的对象；
- (3) 计划中有潜力的部分及潜力的大小不明确，往往造成潜力的浪费。

工程网络图

工程网络图是一种有向图，该图中用圆表示事件，有向弧或箭头表示子任务的进行，箭头上的数字称为权，该权表示此子任务的持续时间，箭头下面括号中的数字表示该任务的机动时间，图中的圆表示与某个子任务开始或结束事件的时间点。

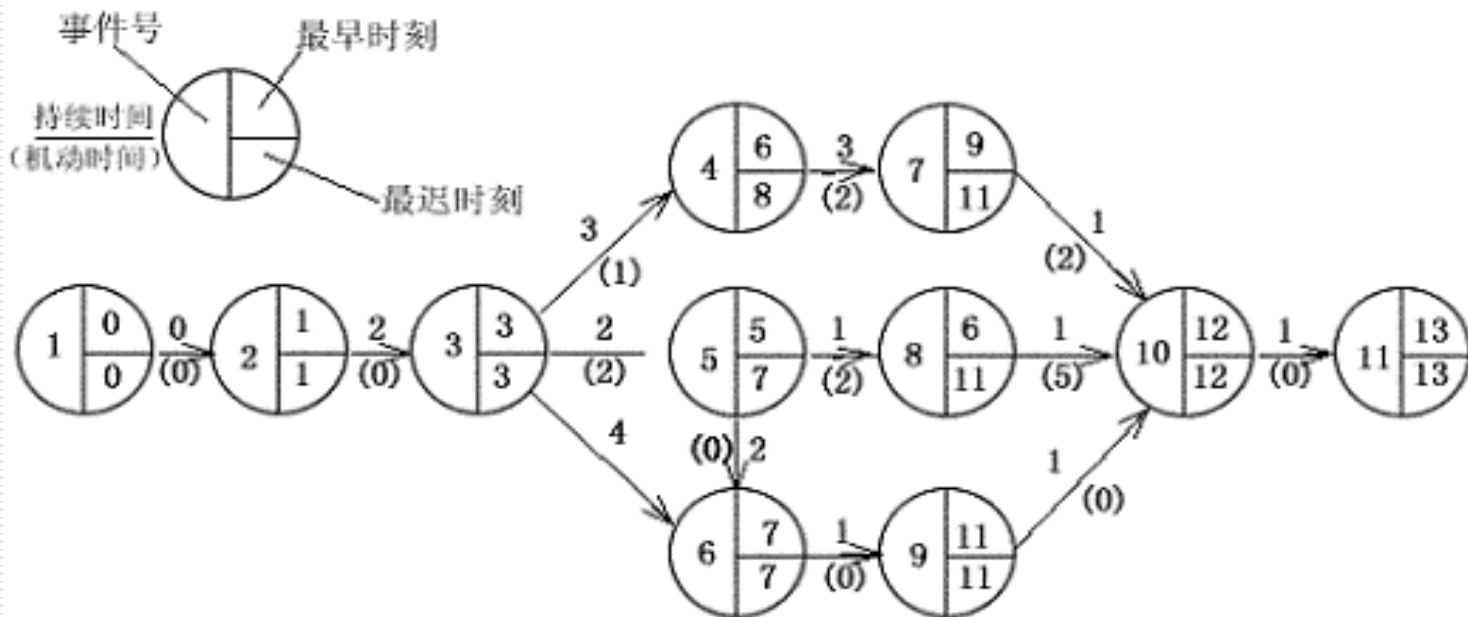
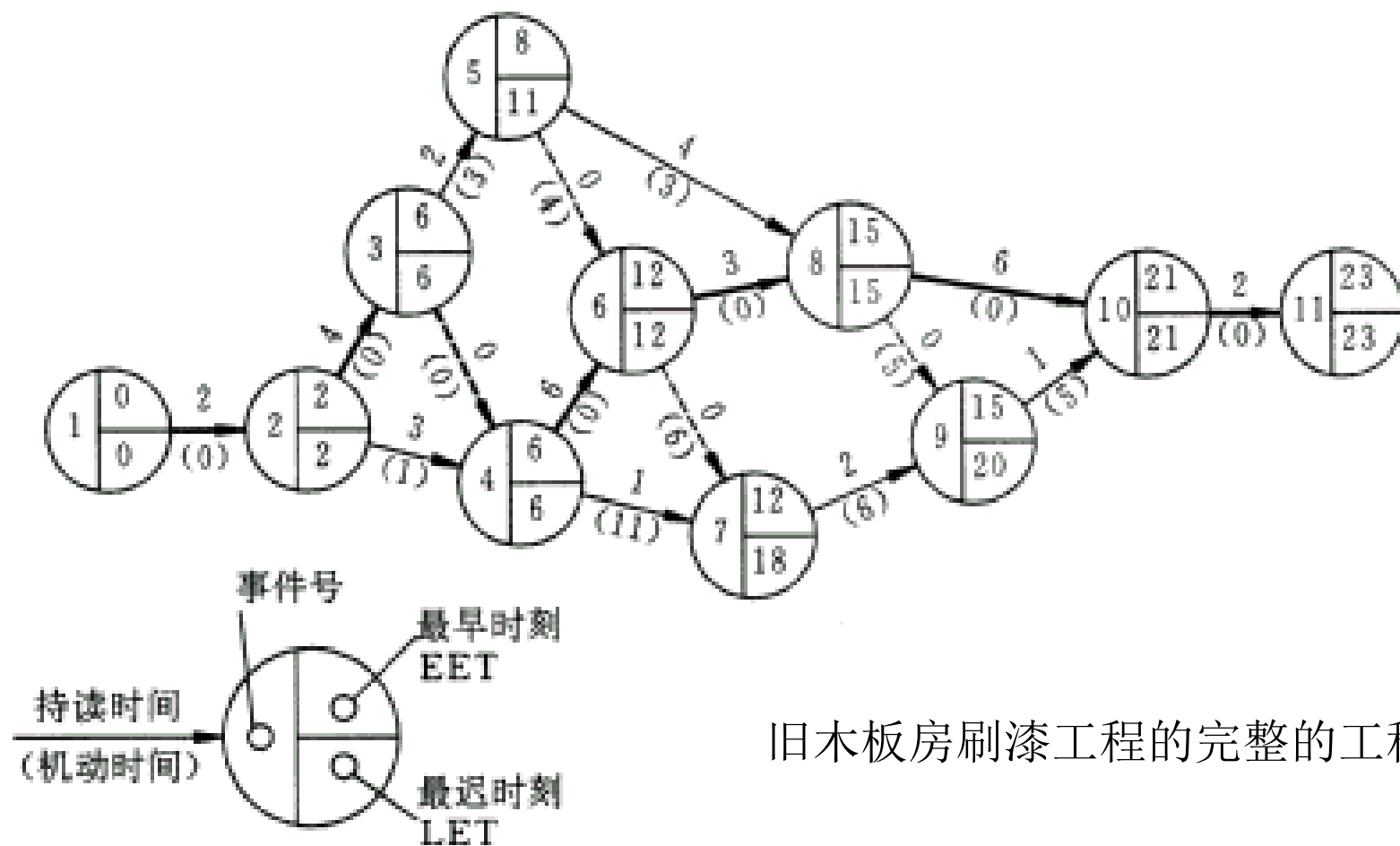


图12-2 工程网络图示例

工程网络图

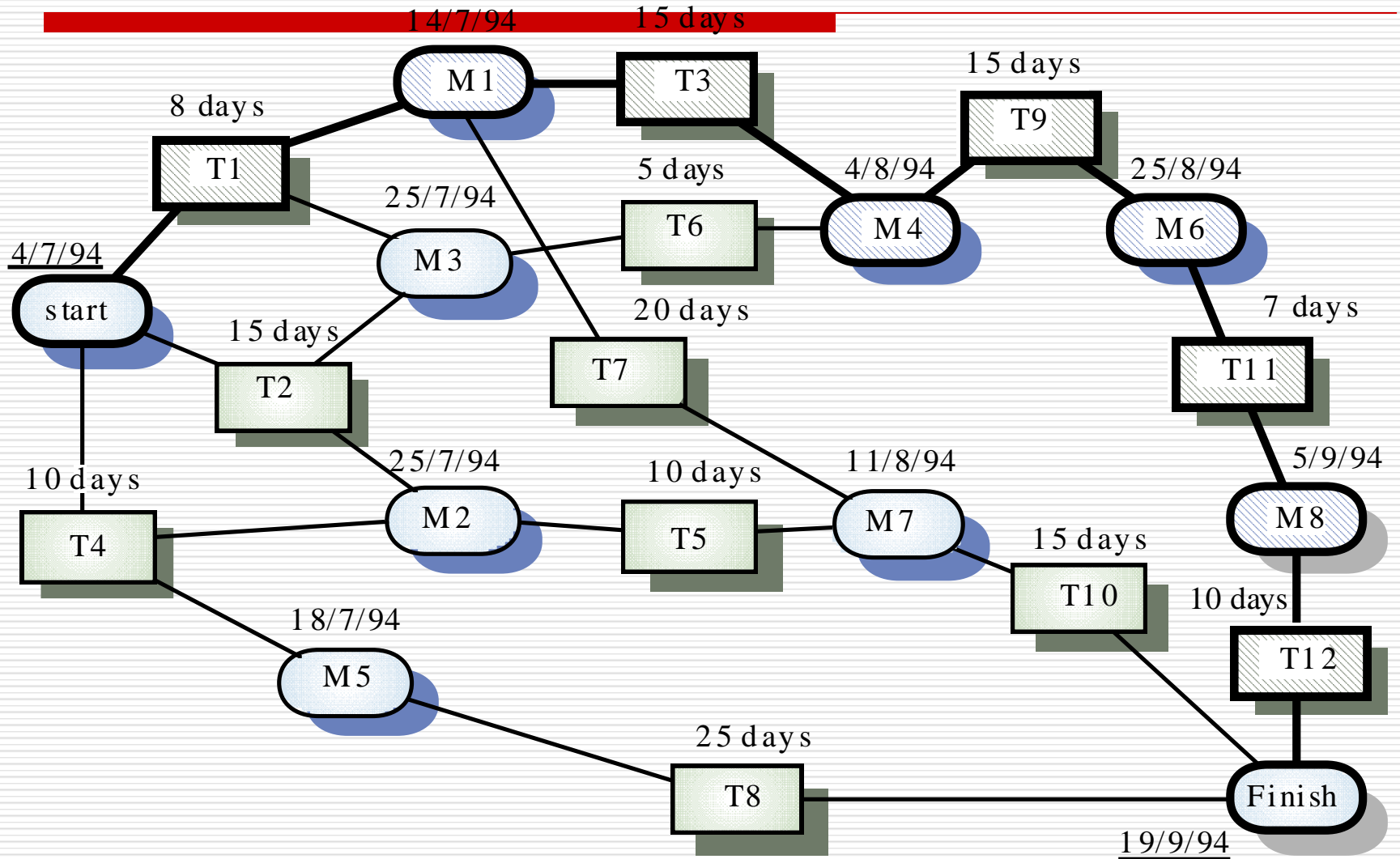


旧木板房刷漆工程的完整的工程网络

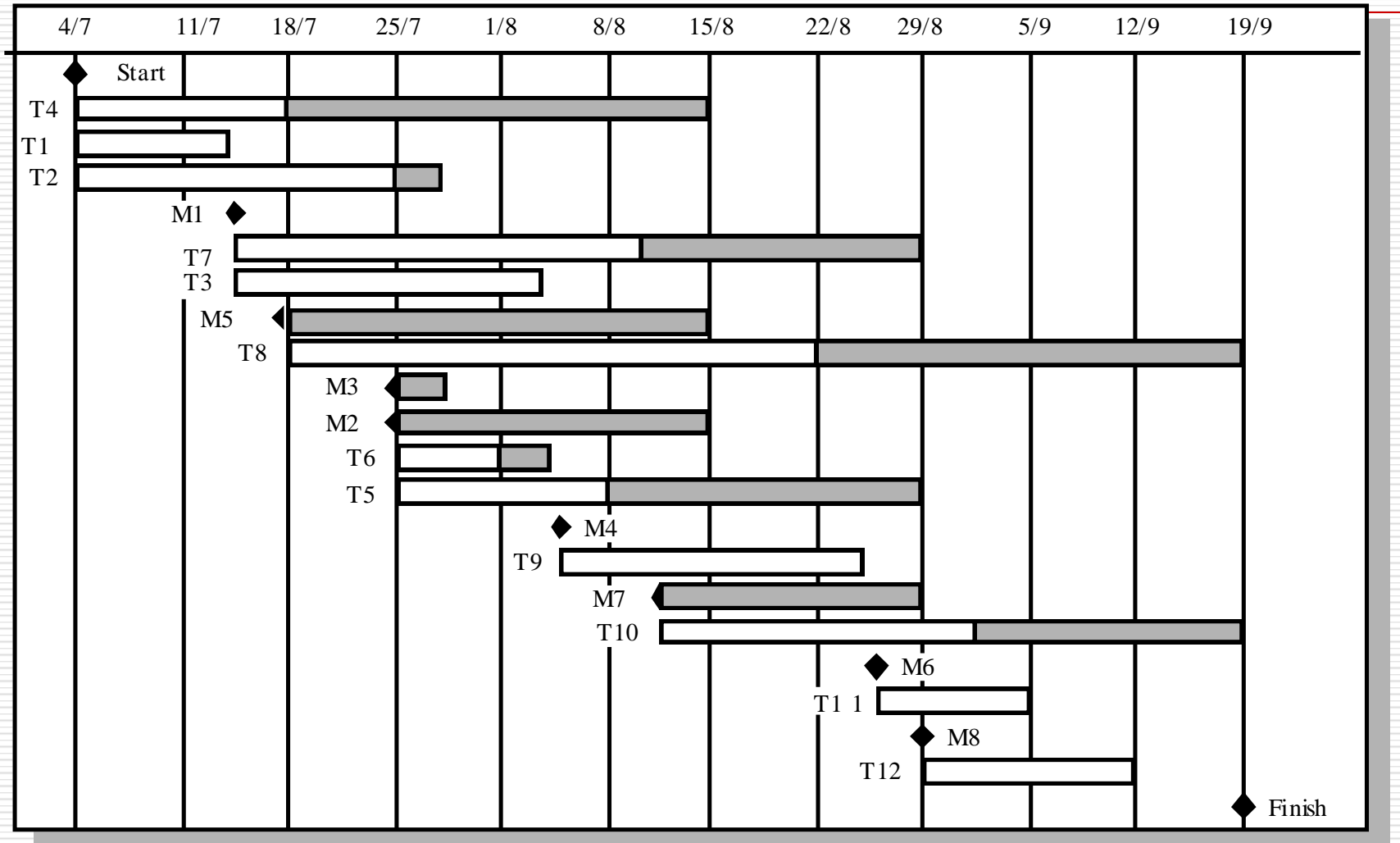
Task durations and dependencies

| Task | Duration (days) | Dependencies |
|------|--------------------|--------------|
| T1 | 8 | |
| T2 | 15 | |
| T3 | 15 | T1 |
| T4 | 10 | |
| T5 | 10 | T2, T4 |
| T6 | 5 | T1, T2 |
| T7 | 20 | T1 |
| T8 | 25 | T4 |
| T9 | 15 | T3, T6 |
| T10 | 15 | T5, T7 |
| T11 | 7 | T9 |
| T12 | 10 | T11 |

Activity network

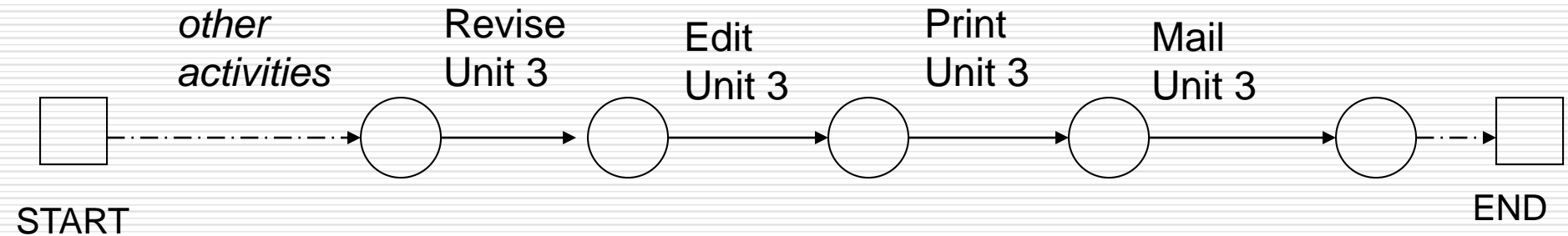


Activity time-line

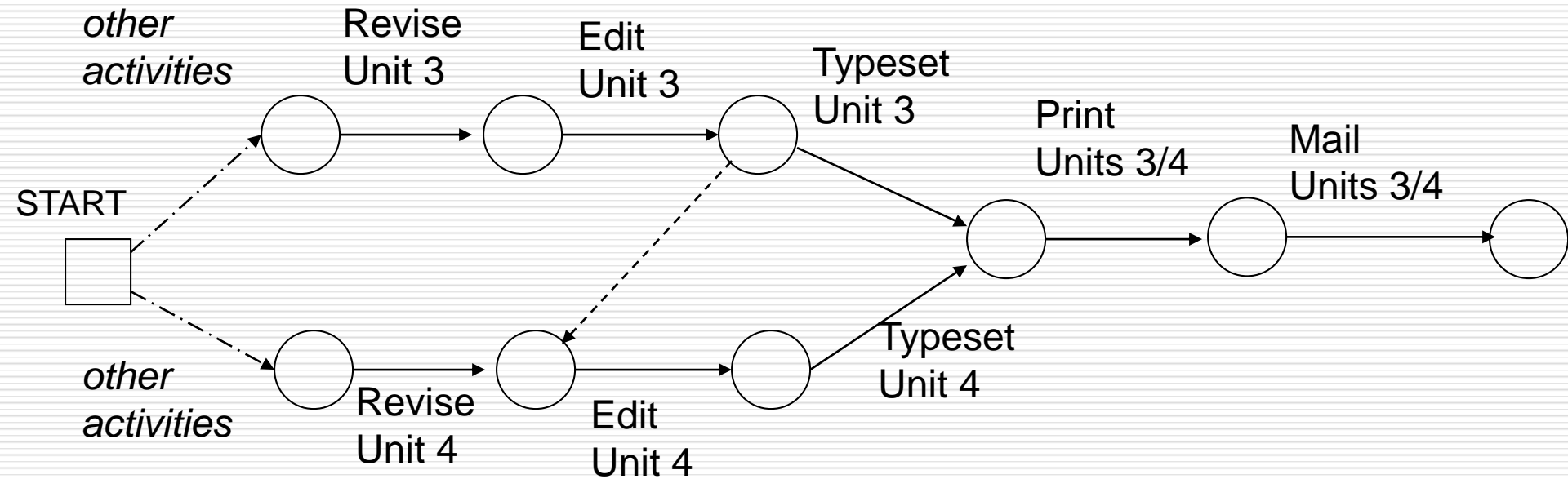


Critical Path

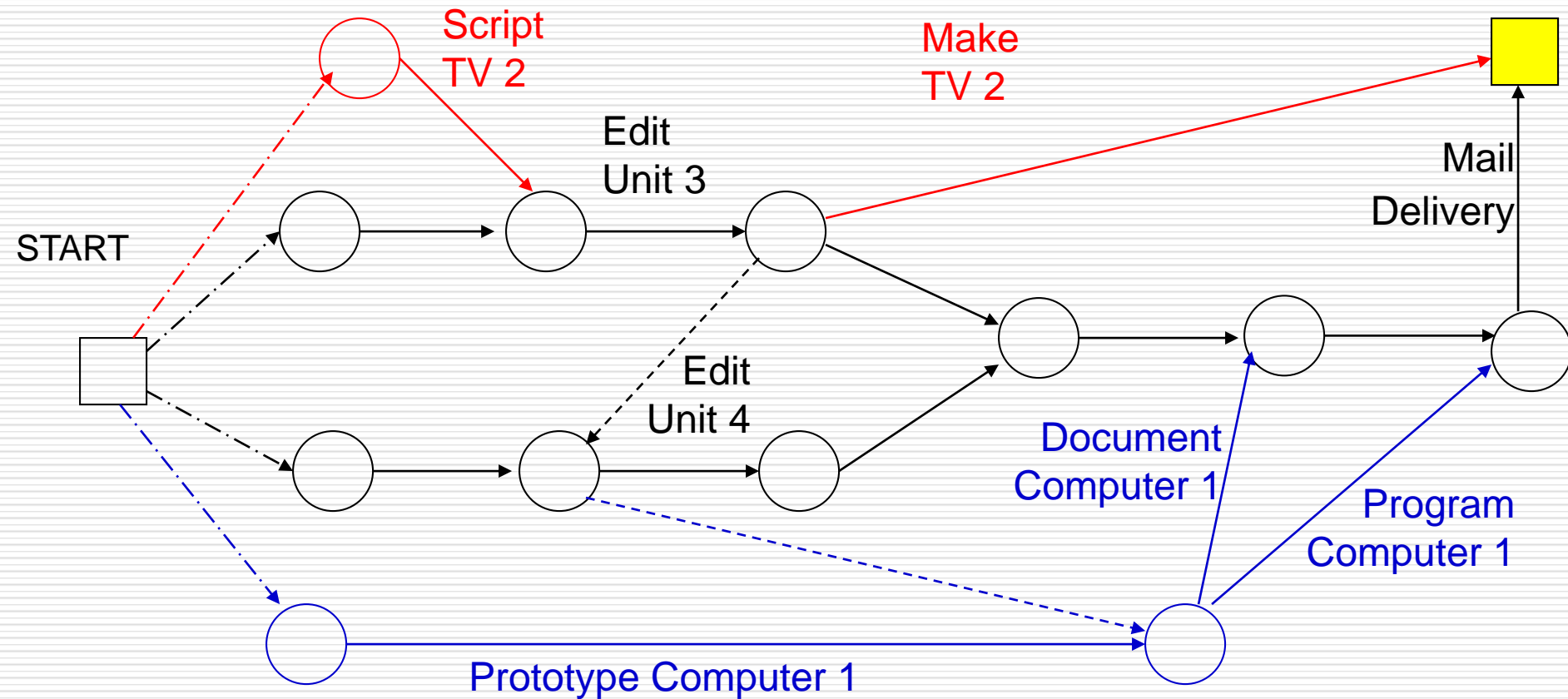
最早时刻和最迟时刻相同



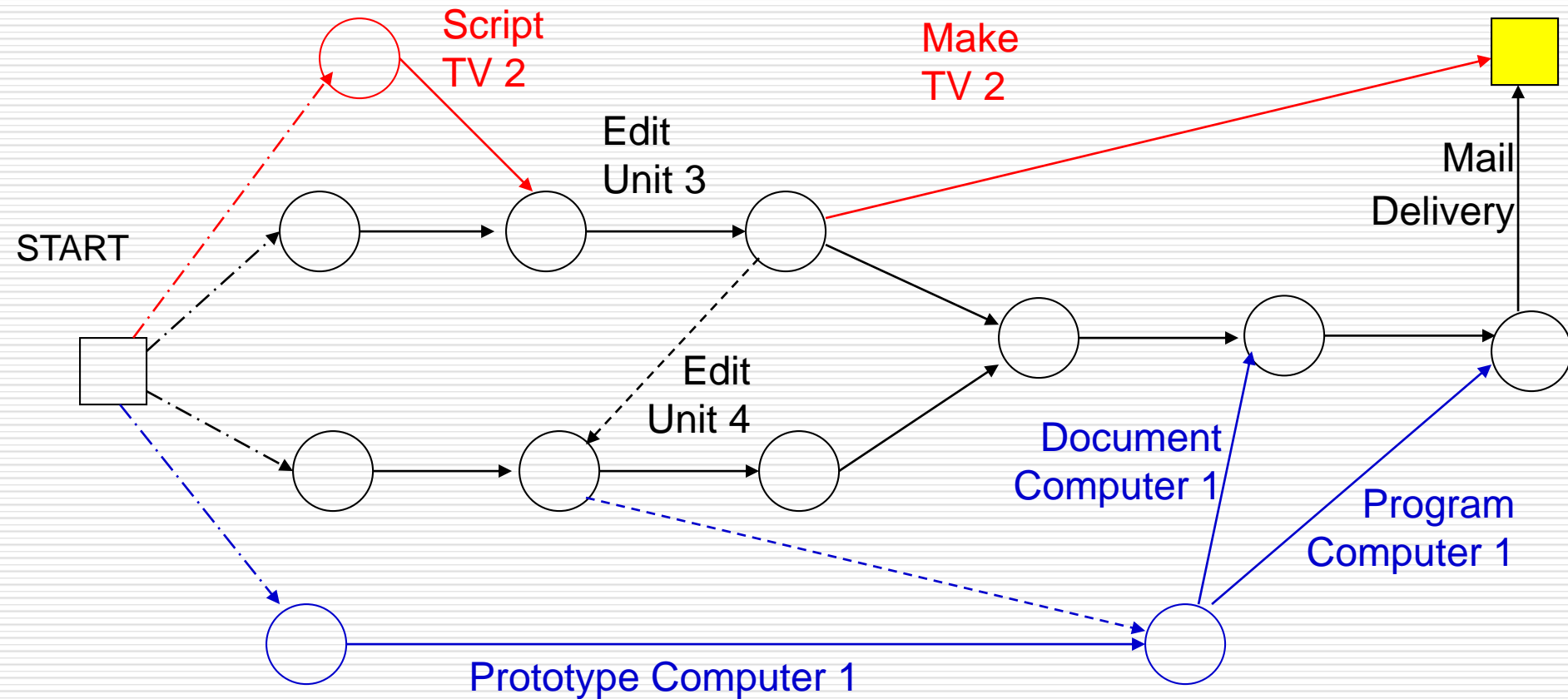
Critical Path Method



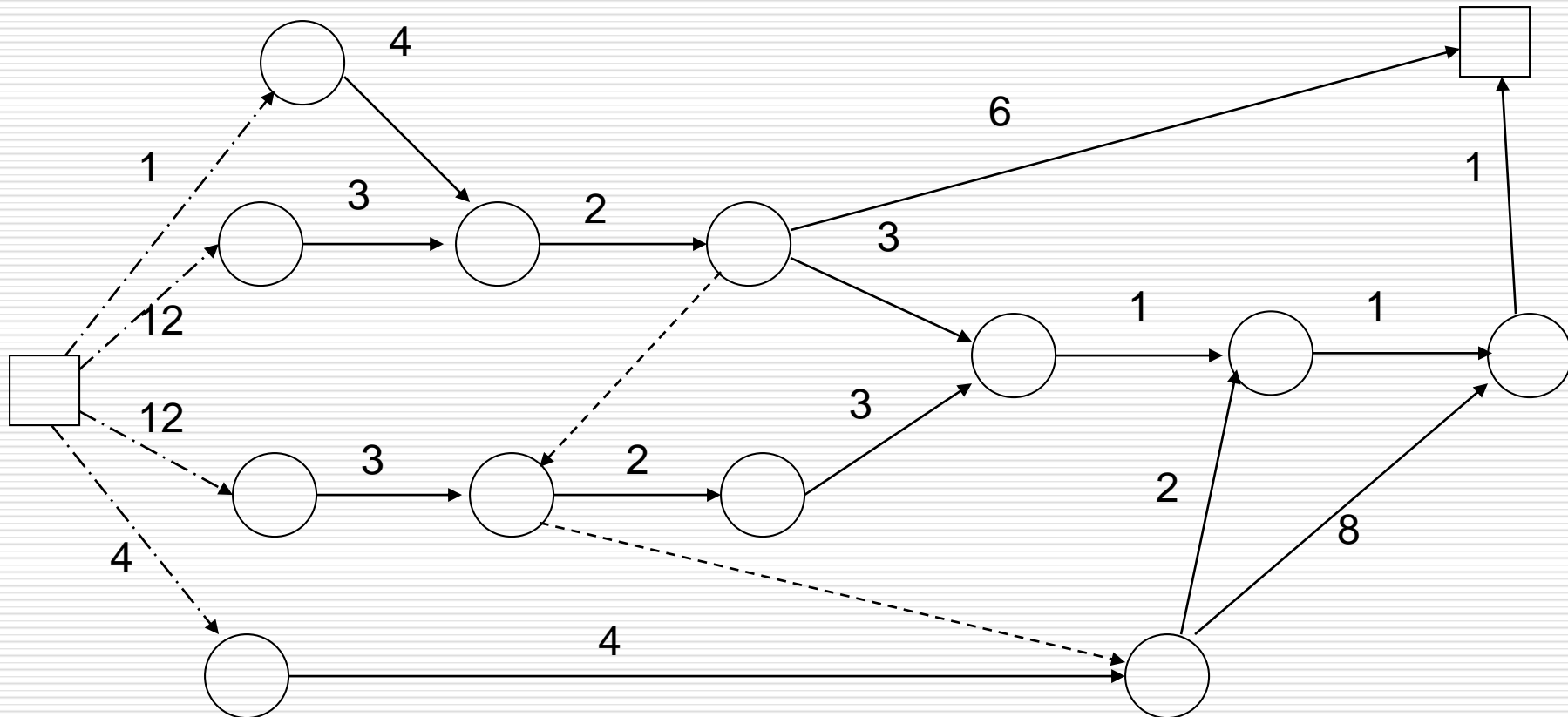
Critical Path Method



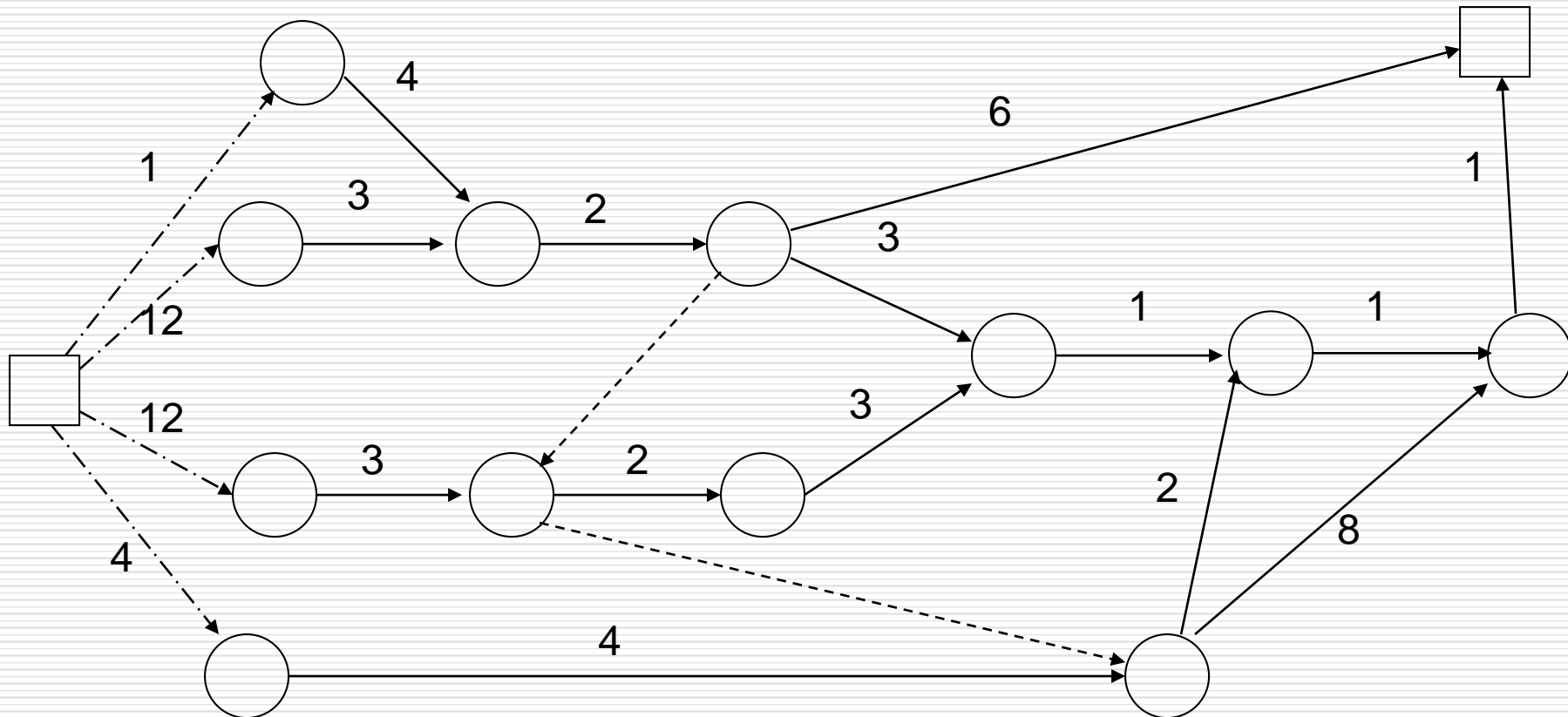
Critical Path Method



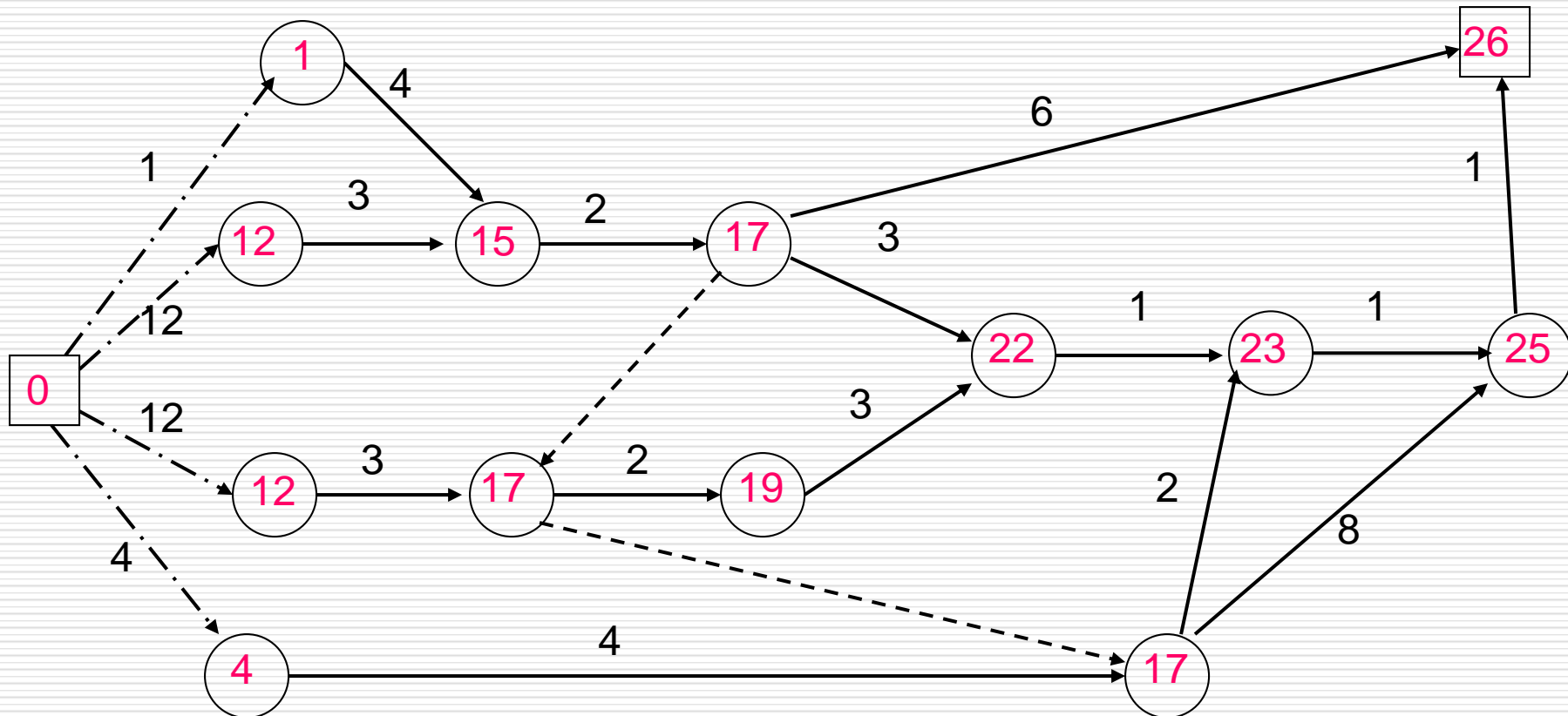
Time Estimates for Activities



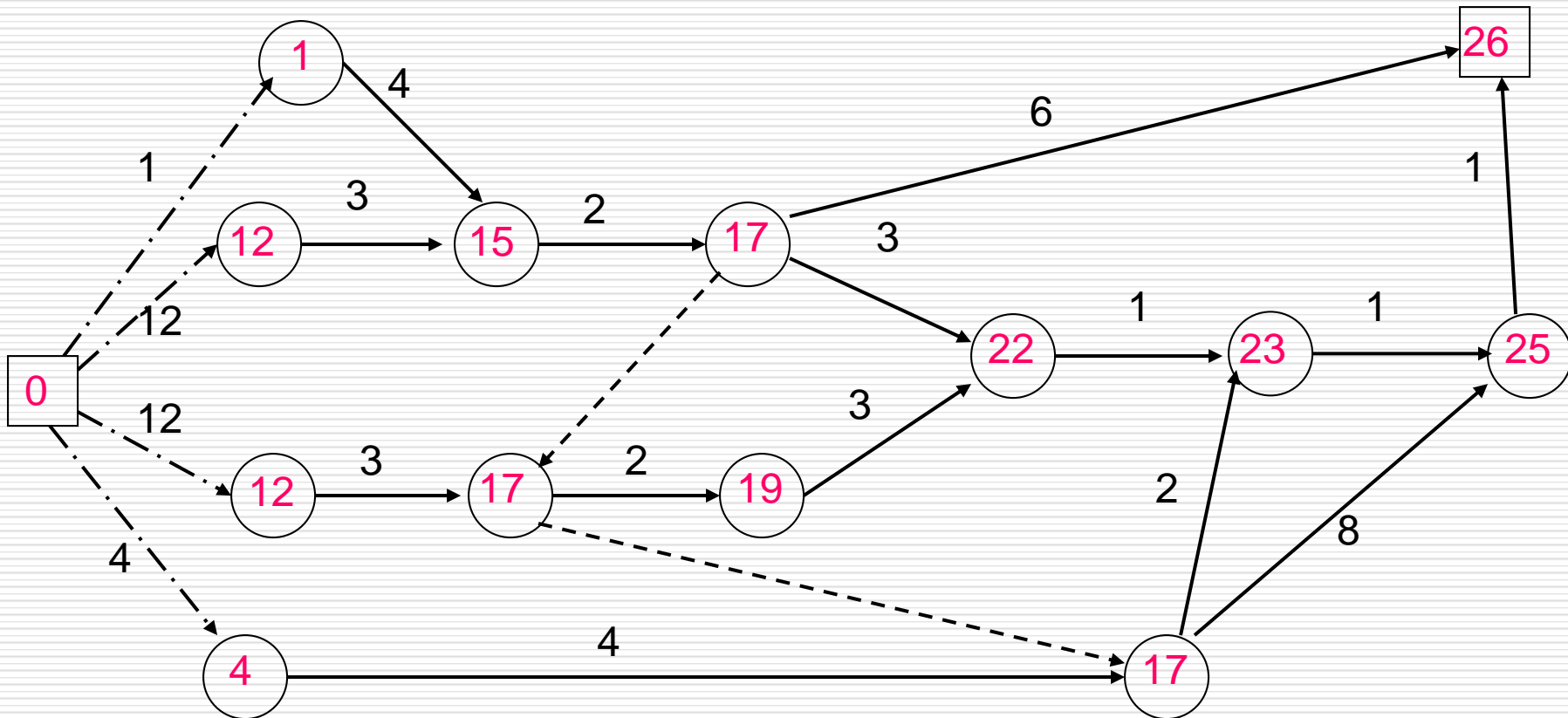
Time Estimates for Activities



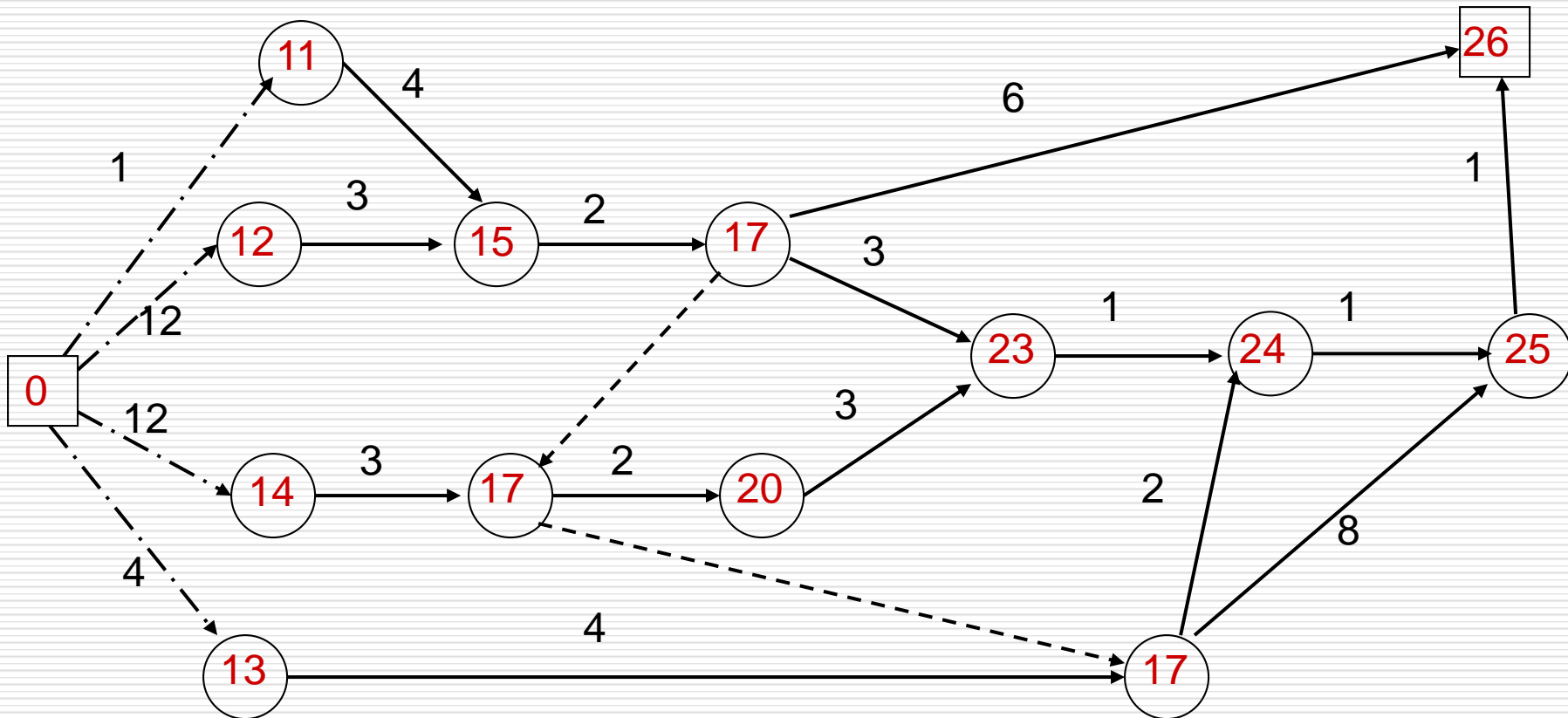
Earliest Start Dates



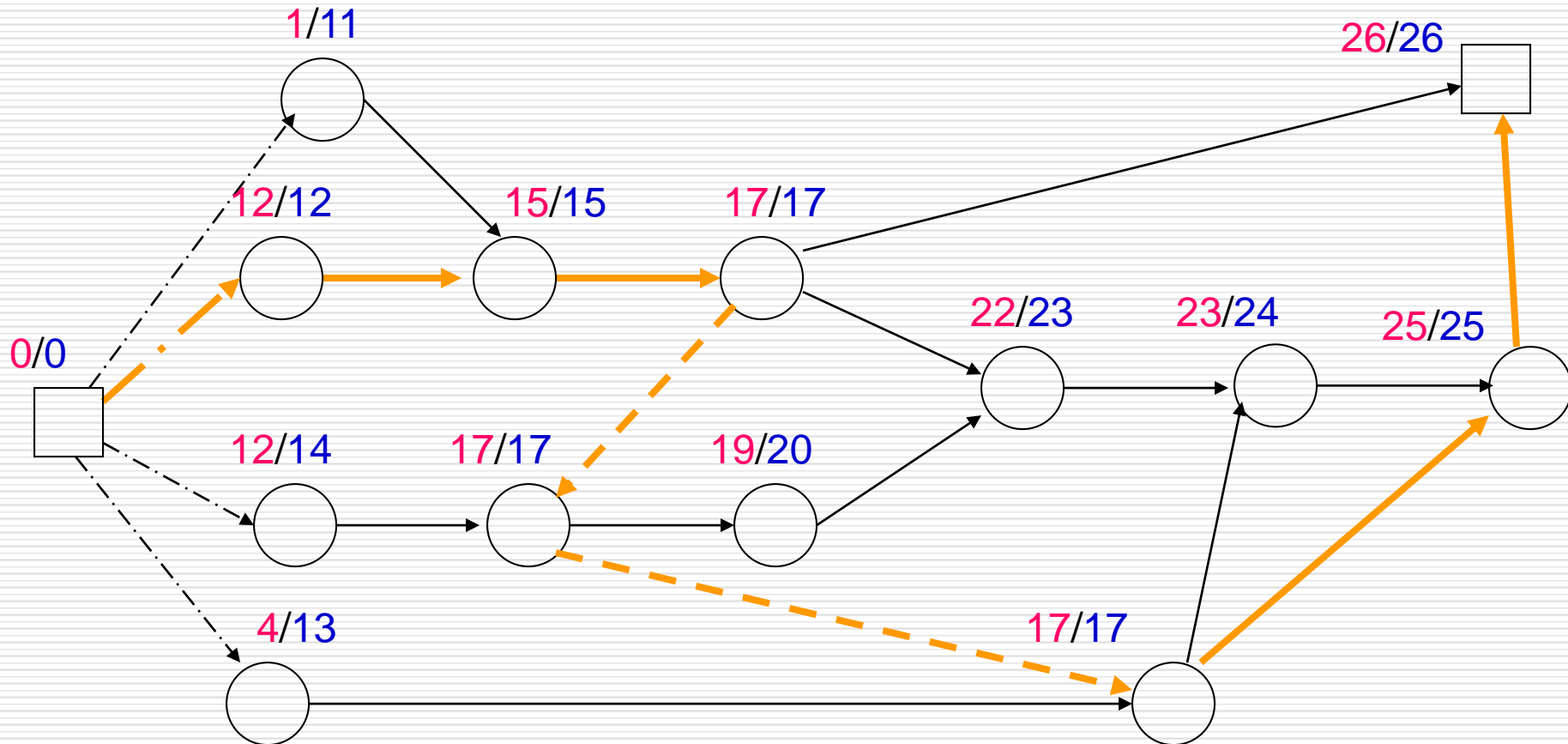
Earliest Start Dates



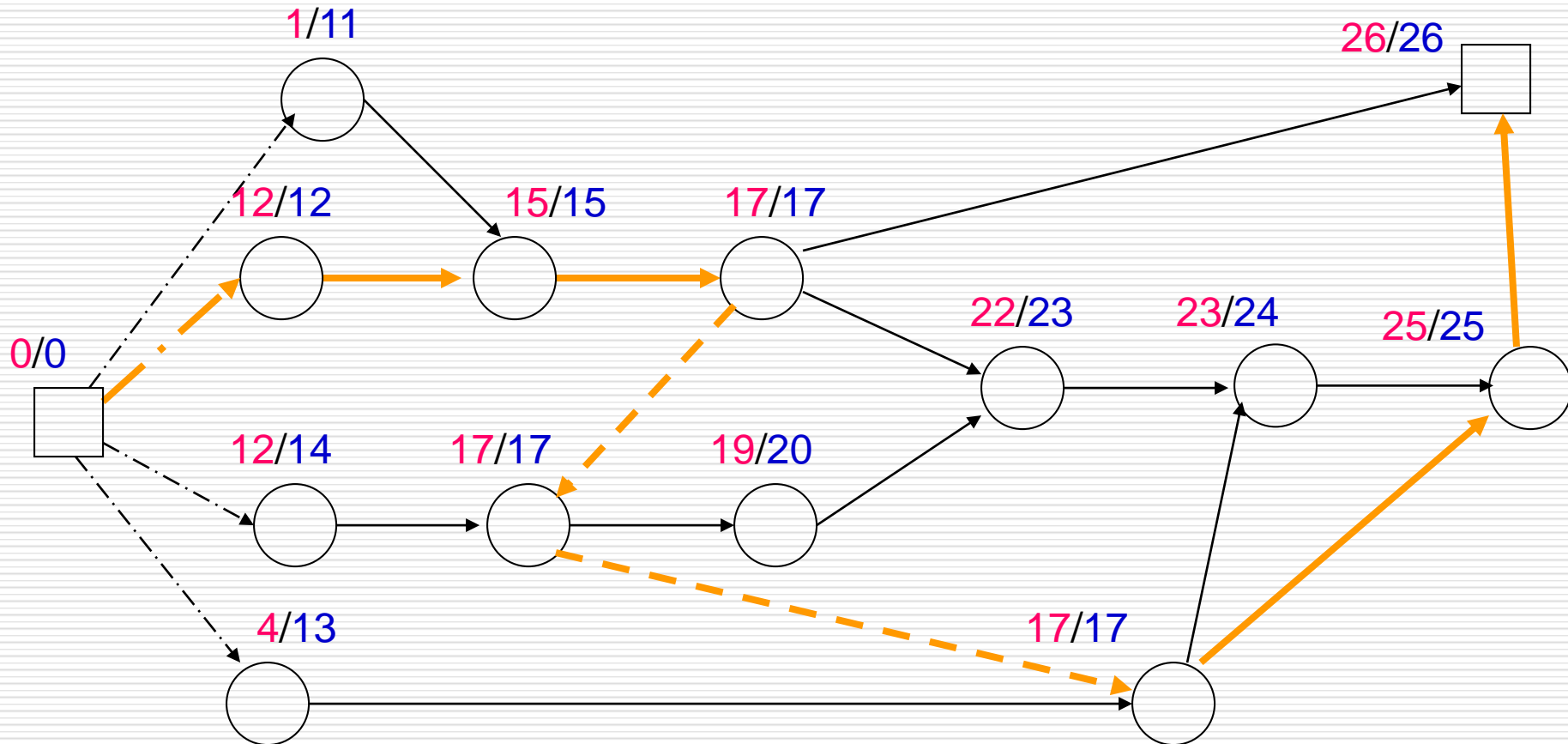
Latest Start Dates



Critical Path



Critical Path



Slack

