Logistics

- Forum available on Piazza (piazza.com/shanghaitech.edu.cn/fall2021/cs132)

- QQ group

- Office Hour:
  - Time: 7:00 – 8:00 PM, Wed (W5-W16)
  - Loc: 1-106A @SIST

Lecture 5: Software Project Management (2)

Outline

- Project Size Estimation
- Project Time Management

Software Project Size Estimation

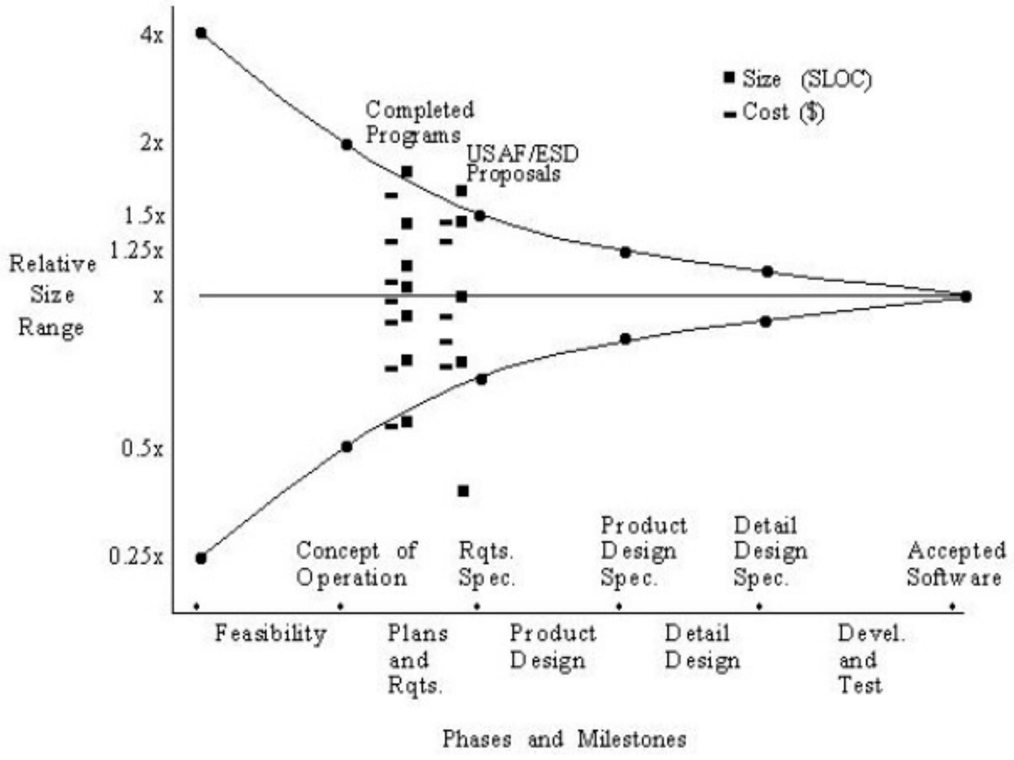
Software Project Size Estimation

- Software Project Size Estimation is a process of calculating and predicting the costs necessary for its successful realization.

- Challenges
  - Software complexity;
  - Requirement can be vague;
  - No historical data (benchmark) to reference;

Software Project Size Estimation (2)

- There are so many uncertainties that it is impossible to estimate system development costs accurately during the early stages of a project.



Software Project Size Estimation (3)

- Two types of techniques can be used for making estimates:
  - Experience-based techniques
    - The estimate of future effort requirements is based on the manager’s experience of past projects and application domain.
    - Essentially, the manager makes an informed judgment of what the effort requirements are likely to be.

Software Project Size Estimation (4)

- Algorithm cost modelling
  - A formulaic approach is used to compute the project effort based on estimates of product attributes, such as size, process characteristics, and experience of staff involved.

Function Point Analysis

COCOMO

Use-Case Points

Machine Learning based Approach

Lines of Code (LOC) Analysis

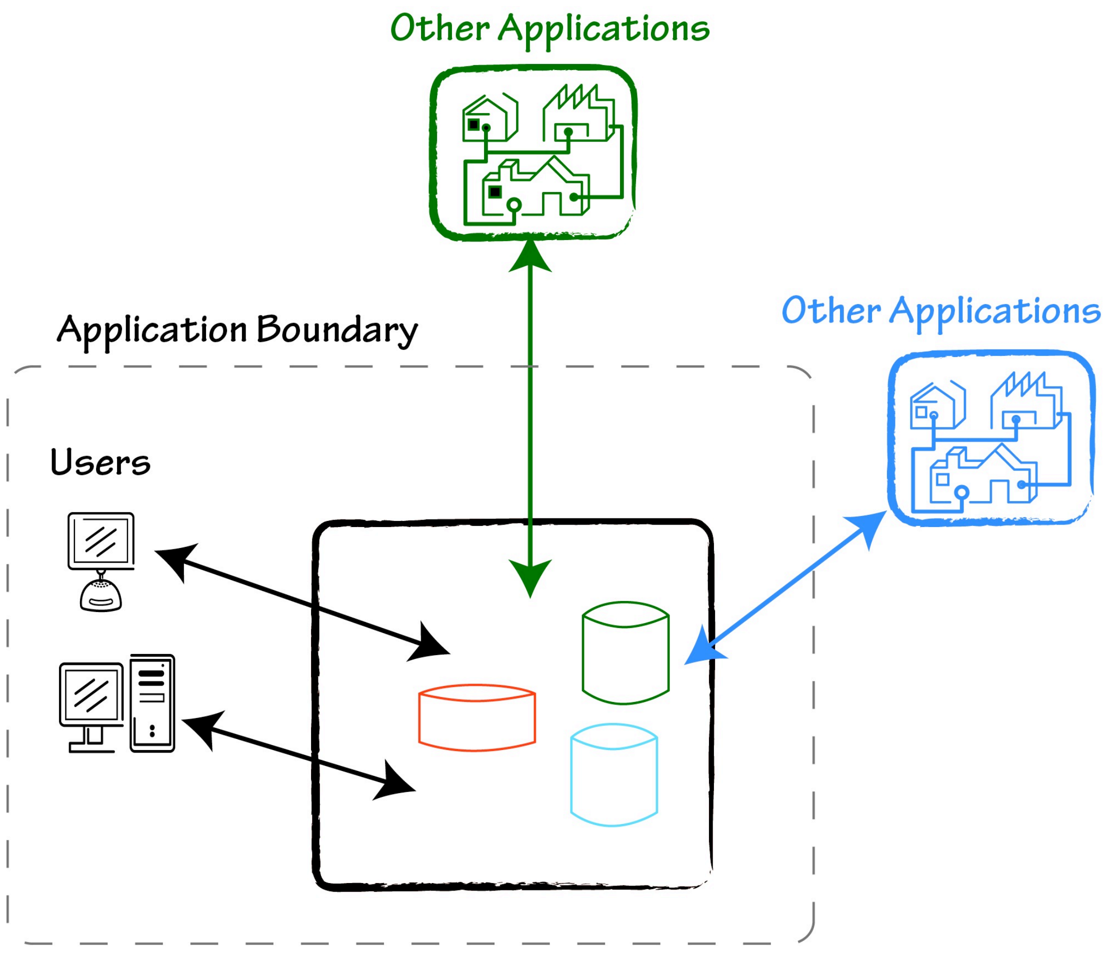
- What's a line of code?
  - The measure was first proposed when programs were typed on cards with one line per card;
  - How does this correspond to statements as in Java which can span several lines or where there can be several statements on one line.
- A key thing to understand about early estimates is that the uncertainty is more important than the initial line – don't see one estimate, seek justifiable bounds
- KLOC (Kilo Line of Code)

Function Point Analysis (FPA)

- The FPA is a reliable method for measuring the size of computer software. It also measures the software development and maintenance cost and size independently of the technology used for implementation.
- Based on a combination of program characteristics
  - External inputs
  - External outputs;
  - External inquiries;
  - Internal logical files; and
  - External logical files.

Function Point Analysis (2)

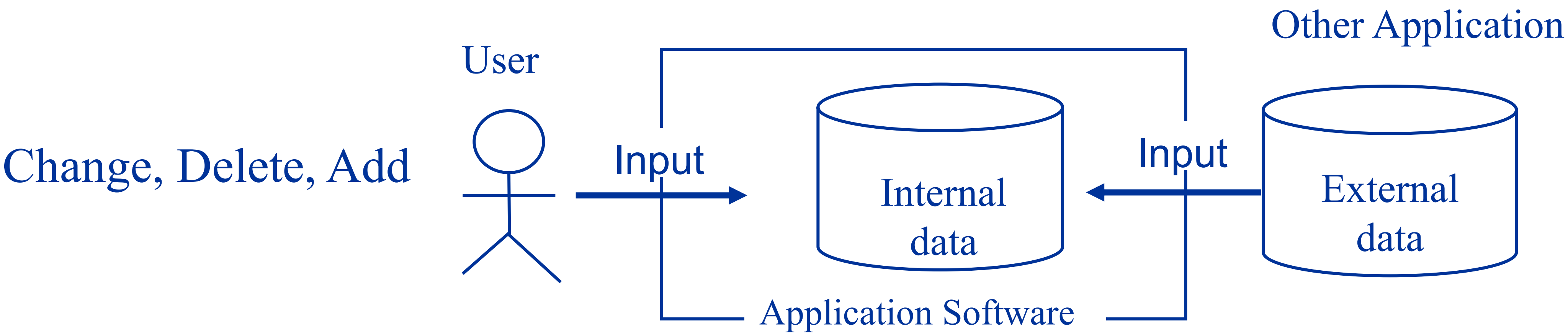
- It is common for computer systems to interact with other computer systems and/or human beings.
- A boundary must be drawn around each system to be measured prior to classifying components.
- This boundary must be drawn according to the sophisticated user's point of view.



Application being measured

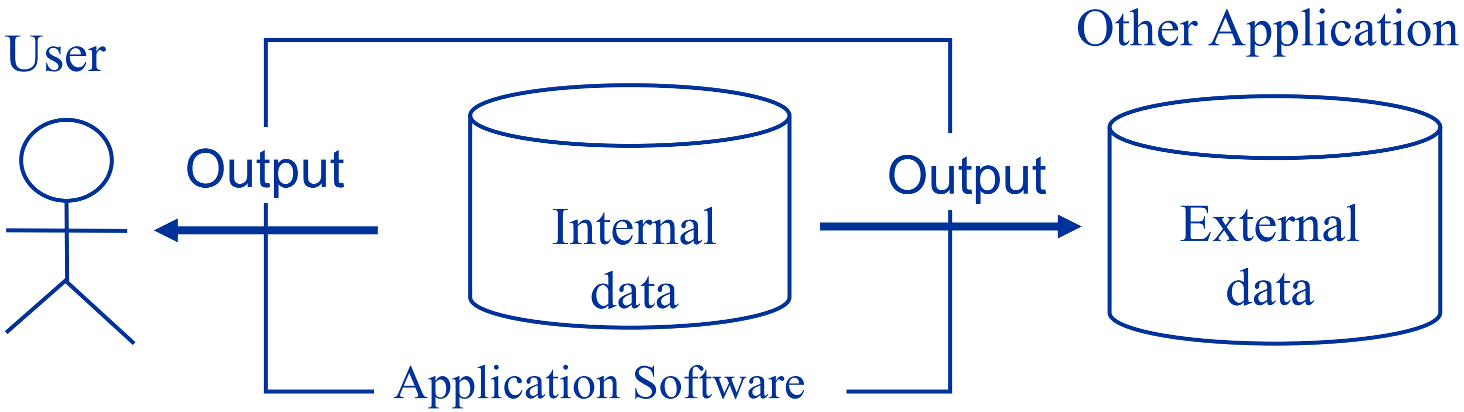
Function Point Analysis (2)

- External inputs (input): user or control data element entering an application;
  - E.g, Data entry by users.
  - E.g, Data or file feeds by external applications.



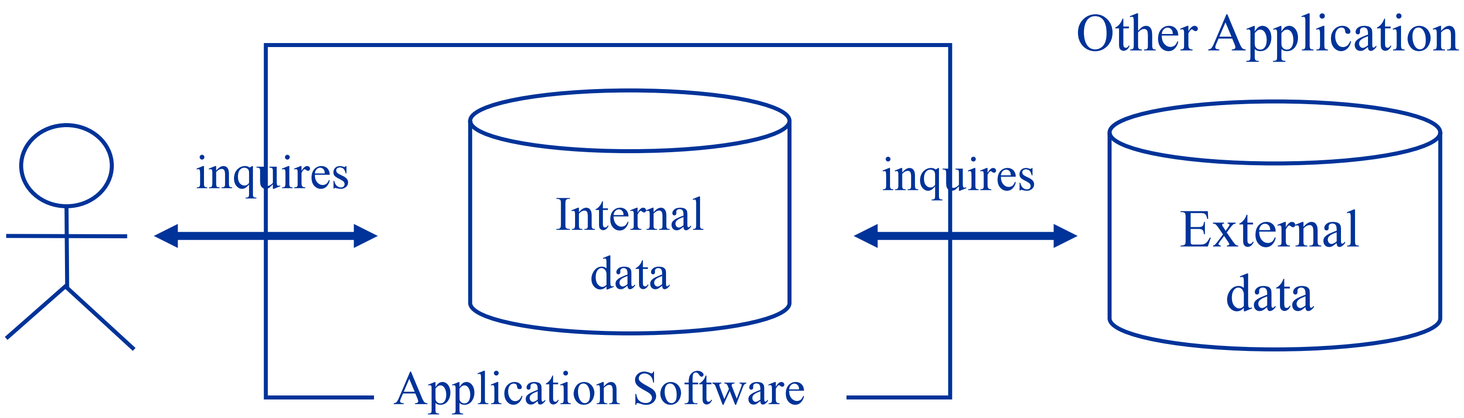
Function Point Analysis (3)

- External outputs (output): user or control data element leaving an application;



Function Point Analysis (4)

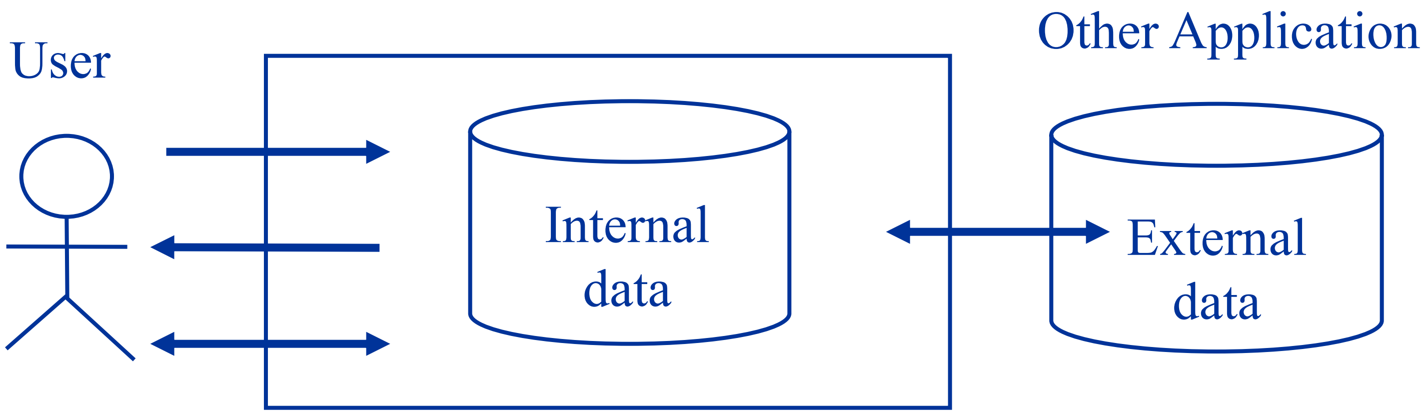
- External inquiries (inquires): Count each unique input-output combination, where an input causes and generates an immediate output, as an external inquiry type.



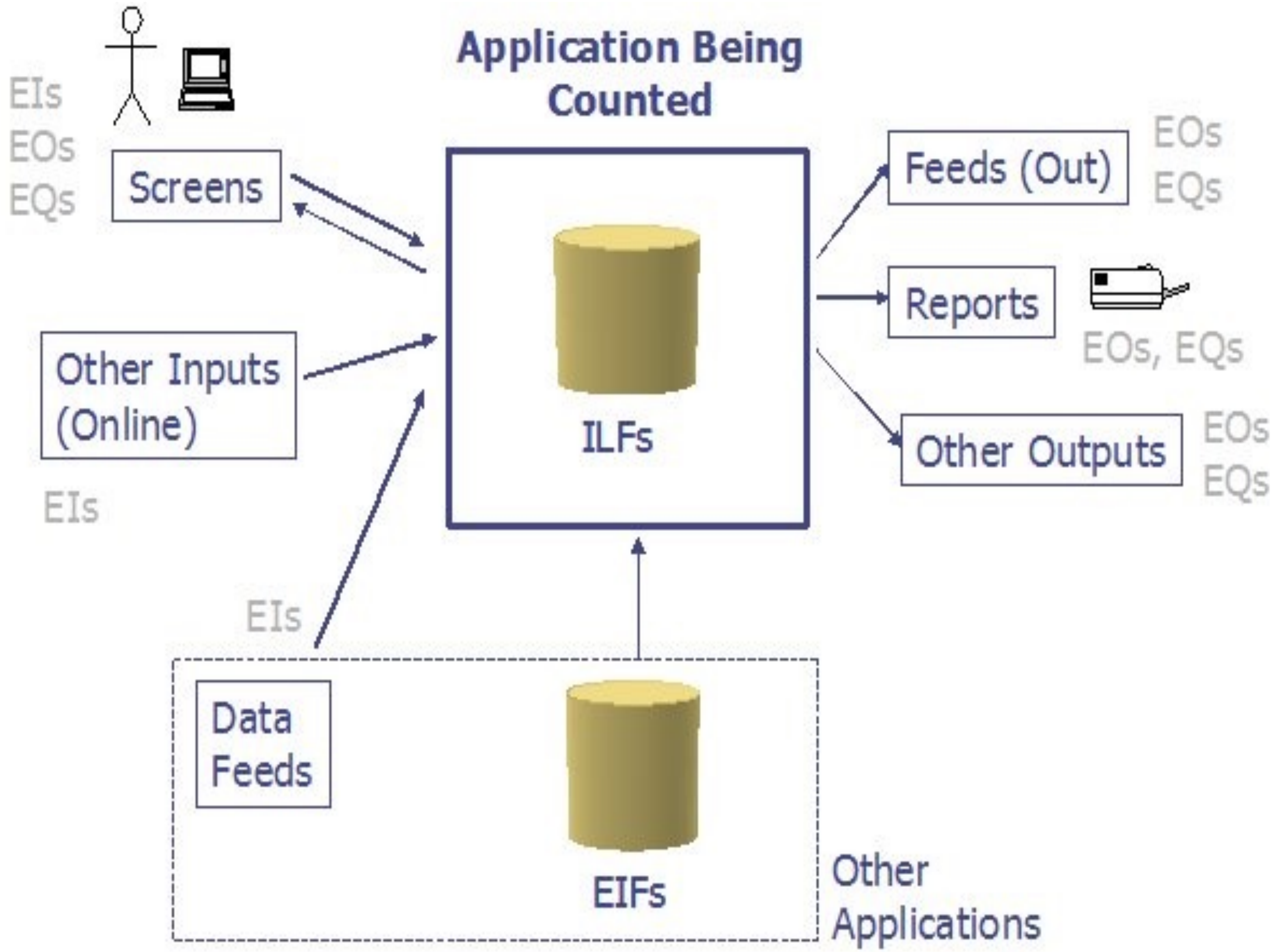
✗ Change, Delete, Add ✓ Inquires

Function Point Analysis (5)

- Internal logical files (files): an internal file or a data store acted on by the user;
- External logical files (interfaces): Files or data passed or shared between software systems should be counted as external interface file types within each system.



Function Point Analysis (6)



Function Point Analysis (7)

Number of FPs	Complexity			Total
	Low	Average	High	
External input type	3	4	6	Σ
External output type	4	5	7	Σ
External inquiry type	3	4	6	Σ
Internal logical files	7	10	15	Σ
External logical files	5	7	10	Σ
Unadjusted Function Points (UFP)				Σ

$3*a+4*b+6*c$

$4*d+5*e+7*f$

…

		Weighting Factor			Count
		Simple	Average	Complex	
Inputs	Member Login	3			
	Member Registration		4		
	Select research question for regression analysis		4		
	Select research question for correlation analysis		4		
	Select research question for hypothesis test analysis		4		
	Select research question for Chi Square test analysis		4		27
Outputs	Member login confirmation	3			
	Member Registration confirmation	3			
	Graph/Table of regression analysis				
	Graph/Table of correlation analysis	3			
	Graph/Table of hypothesis test analysis	3			
	Graph/Table of Chi square test analysis	3			15
Inquiries	Validate member information		4		
	View alumni list		4		8
Files	Linear regression		10		
	correlation		10		
	Hypothesis test		10		
	Chi square test		10		40
Interfaces	Application server to database			10	
	User to application server			10	20
Total UFP					110

Function Point Analysis (8)

General System Characteristic		Brief Description
F1.	Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?
F2.	Distributed data processing	How are distributed data and processing functions handled?
F3.	Performance	Did the user require response time or throughput?
F4.	Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?
F5.	Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?
F6.	On-Line data entry	What percentage of the information is entered On-Line?
F7.	End-user efficiency	Was the application designed for end-user efficiency?

Fi (Complexity Adjustment Factor) [0-5];
0 for not important or not applicable; 5 for absolutely essential

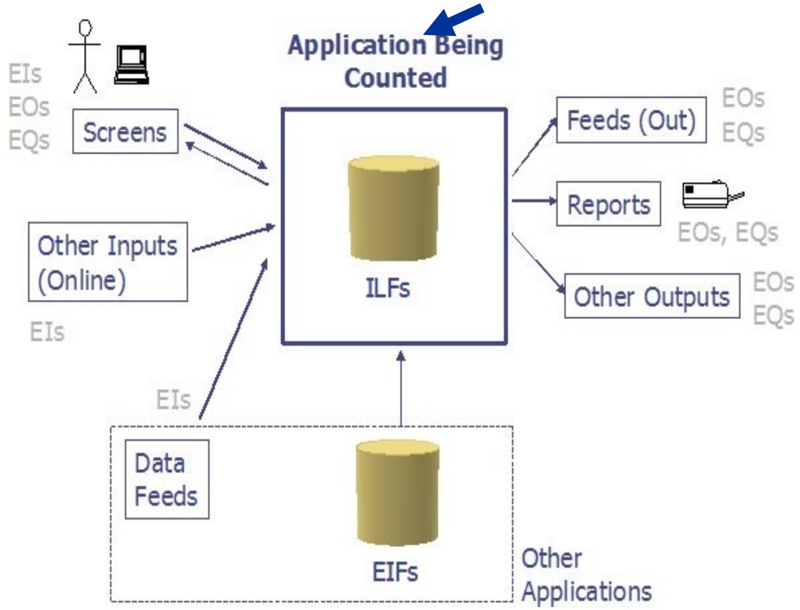
Function Point Analysis (8) cont’

General System Characteristic		Brief Description
F8.	On-Line update	How many Internal Logical File’s are updated by On-Line transaction?
F9.	Complex processing	Does the application have extensive logical or mathematical processing?
F10.	Reusability	Was the application developed to meet one or many user’s needs?
F11.	Installation ease	How difficult is conversion and installation?
F12.	Operational ease	How effective and/or automated are start-up, back up, and recovery procedures?
F13.	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?
F14.	Facilitate change	Was the application specifically designed, developed, and supported to facilitate change?

Fi factor: [0-5];

Function Point Analysis (9)

$$Function\ Point\ (FP) = UFP * \left[0.65 + 0.01 * \sum_{i=1}^{14} F(i)\right]$$



Fi factor: [0-5];

F1.	Data communications
F2.	Distributed data processing
F3.	Performance
…	….

Lines of Code per Function Point

Language	Approximate LOC/Function Point
C	130
Java	55
C++	50
Visual Basic	30
Power Builder	15
HTML	15
Package, e.g. Excel, Access, …	10-40

COCOMO Model

- COCOMO (CONSTRUCTIVE COST MODEL)
  - First published by Dr. Barry Boehm, 1981

- Interactive cost estimation software package that models the cost, effort and schedule for a new software development activity.
  - Can be used on new systems or upgrades

COCOMO Model (2)

- COCOMO defines three types of systems:
- Organic（组织性）
  - if the team size required is adequately small, the problem is well understood and has been solved in the past.
- Embedded （嵌入性）
  - A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category.
- Semi-detached （半独立性）
  - if the vital characteristics such as team-size, experience, knowledge lie in between that of organic and Embedded.

COCOMO Model (3)

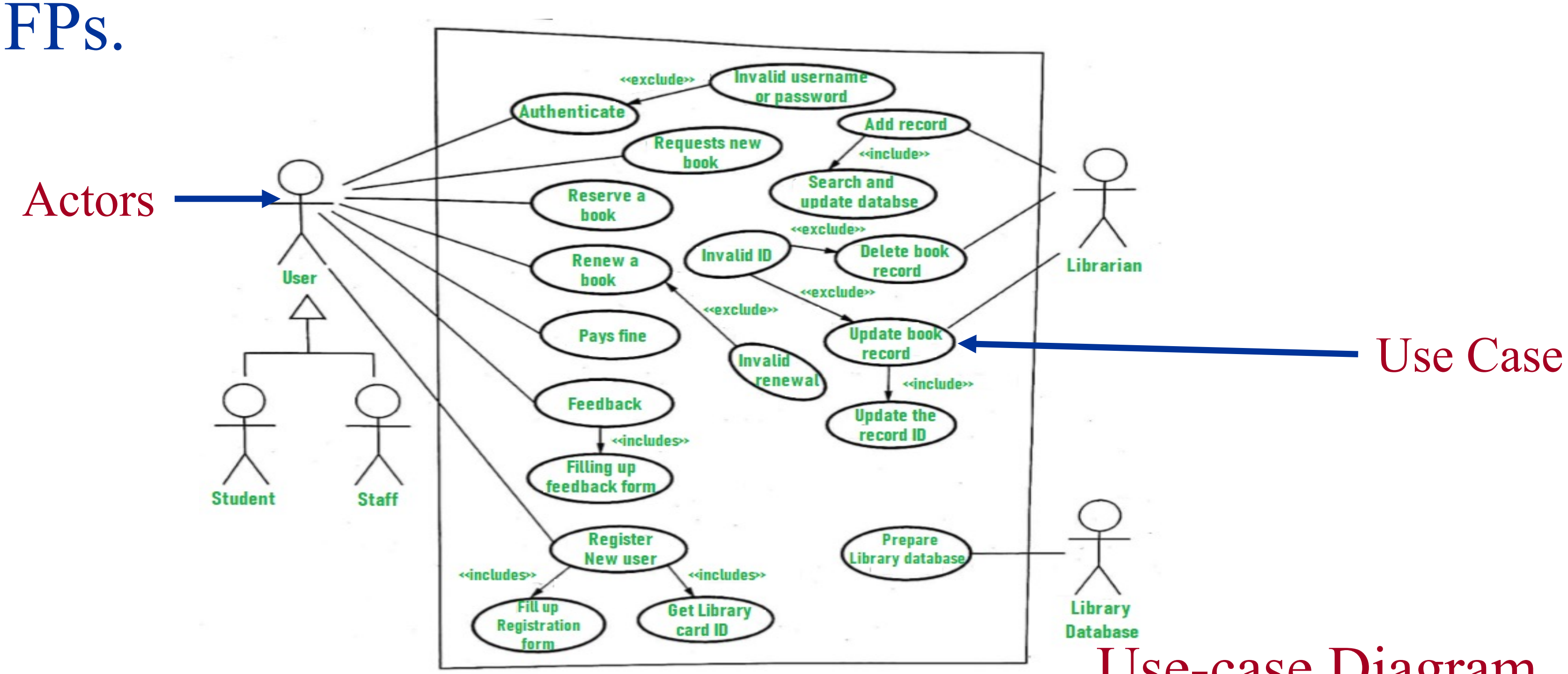
$$\mathrm{E} = A \ * \ (Size)^B; TDEV = 2.5 * E^C$$

- E： Effort applied in person-months;
- TDEV: (Time to Development) applied in months;
- A, B, C: coefficients;
- Size: KLOC or function point;

Type	A	B	C
Organic	2.4	1.05	0.38
Embedded	3.6	1.2	0.32
Semi-detached	3.0	1.12	0.35

Use Case Point

- Use-Case Points (UCP) is a software estimation technique used to measure the software size with use cases. The concept of UCP is similar to FPs.



Use-case Diagram

Use Case Point (2) – STEP 1

- Determine Unadjusted Actor Weight (UAW)
- STEP 1.1 Determine the type of Actors;

Actor Complexity	Example	Actor Weight
Simple	A System with defined API	1
Average	A System interacting through a Protocol (e.g., TCP/IP) A User interacting through text-based interface	2
Complex	A User interacting through GUI	3

Use Case Point (3) – STEP 1

- Determine Unadjusted Actor Weight (UAW)
- STEP 1.2 Determine the type of Actors;

Actor Complexity	Actor Weight	Number of Actors	Product
Simple	1	NSA	$1 \times NSA$
Average	2	NAA	$2 \times NAA$
Complex	3	NCA	$3 \times NCA$
Unadjusted Actor Weight (UAW)			$1 \times NSA + 2 \times NAA + 3 \times NCA$

Use Case Point (4) – STEP 2

- Determine Unadjusted Use-Case Weight (UCCW)
- STEP 2.1 Classify each Use-Case as Simple, Average or Complex

Use-Case Complexity	Number of Transactions	Use-Case Weight
Simple	≤3	5
Average	4 to 7	10
Complex	>7	15

Use Case Point (5) – STEP 2

- Determine Unadjusted Use-Case Weight (UCCW)
- STEP 2.2 Classify each Use-Case as Simple, Average or Complex

Use-Case Complexity	Use-Case Weight	Number of Use-Cases	Product
Simple	5	NSUC	5 × NSUC
Average	10	NAUC	10 × NAUC
Complex	15	NCUC	15 × NCUC
Unadjusted Use-Case Weight (UUCW)			5 × NSUC + 10 × NAUC + 15 × NCUC

Use Case Point (6) – STEP 3

- Technical Complexity Factor (TCF)

Factor	Description	Weight (W)	Rated Value (0-5) (RV)	Impact (I= W *RV)
T1	Distributed System	2.0		
T2	Response time or throughput performance objectives	1.0		
T3	End-user efficiency	1.0		
T4	Complex processing	1.0		
T5	Reusable Code	1.0		
T6	Easy to Install	0.5		
T7	Easy to Use	0.5		

Use Case Point (7) – STEP 3

- Technical Complexity Factor (TCF)

Factor	Description	Weight (W)	Rated Value (0-5) (RV)	Impact (I= W *RV)
T8	Portable	2.0		
T9	Easy to change	1.0		
T10	Concurrent	1.0		
T11	Security objectives	1.0		
T12	Direct access to third parties	1.0		
T13	User training facilities	1.0		

Use Case Point (8) – STEP 3

- Technical Complexity Factor (TCF)

$$TCF = 0.6 + (0.01 * \sum_{i=1}^{13} Weight(i) * RatedValue(i))$$

Use Case Point (9) – STEP 4

- Environment Factor (EF)

Factor	Description	Weight (W)	Rated Value (0-5)	Impact (I = W*RV)
F1	Familiarity with the project	1.5		
F2	Application experience	0.5		
F3	OO Programming experience	1.0		
F4	Lead analyst capability	0.5		

Use Case Point (10) – STEP 4

- Environment Factor (EF)

Factor	Description	Weight (W)	Rated Value (0-5)	Impact (I = W*RV)
F5	Motivation	1.0		
F6	Stable requirements	2.0		
F7	Part-time staff	-1.0		
F8	Difficult programming language	-1.0		

Use Case Point (11) – STEP 4

- Environment Factor (EF)

$$EF = 1.4 + (-0.03 * \sum_{i=1}^{8} Weight(i) * RatedValue(i))$$

Use Case Point (12) – STEP 5

- The adjusted use case points (AUCP) are calculated as:

$$AUCP = (UAW+UCCW)*TCF*EF$$

Unadjusted Actor Weight (UAW)

Unadjusted Use-Case Weight (UCCW)

Technical Complexity Factor (TCF)

Environment Factor (EF)

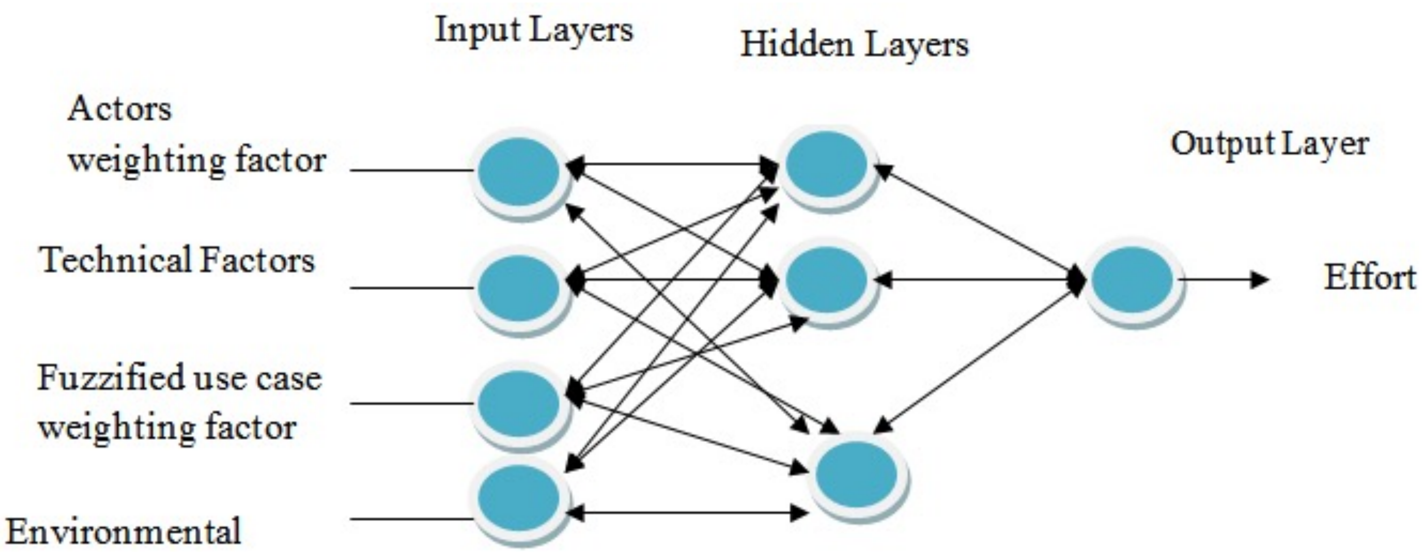
Use Case Point (13)

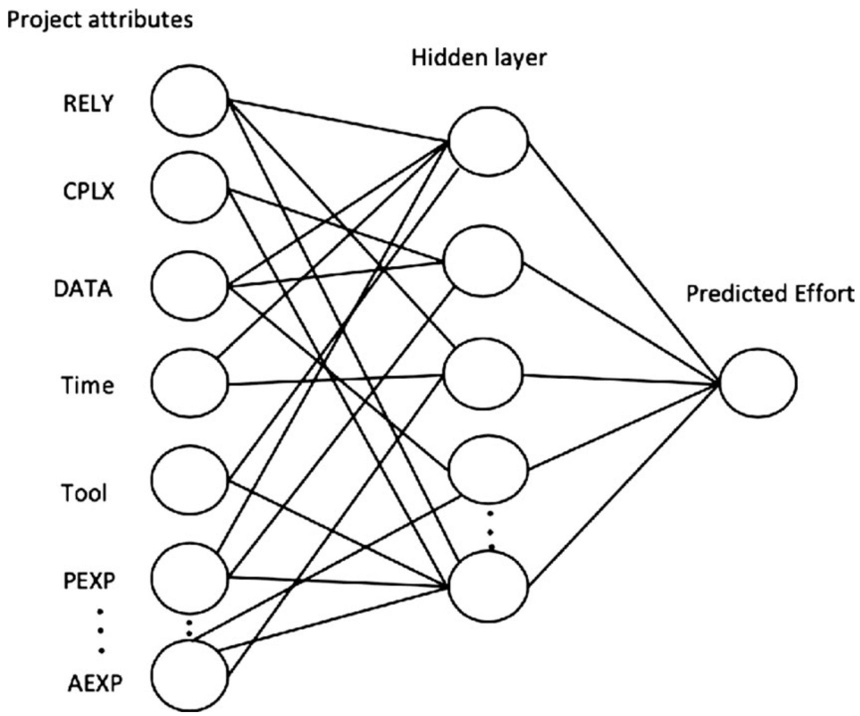
- Suggestion (Experience)
- For each UCP, 16-30 hours per use case point.
- On average, 20 hours per use case point.

- E.g UCP = 545
  - 545*20 = 10900 hour

Machine Learning based Approach

- Neural Network
- Historical Data





Project Time Management

Project Time Management

- Managers often cite the need to deliver projects on time as one of their biggest challenges and the main cause of conflict.
- People often compare planned and actual project completion times without taking into account the approved changes in the project.
- Time is the variable that has the least amount of flexibility. Time passes no matter what happens on a project.

Workflow of Project Time Management

Defining Activities

↓

Sequencing Activities

↓

Estimating Activity Resources

↓

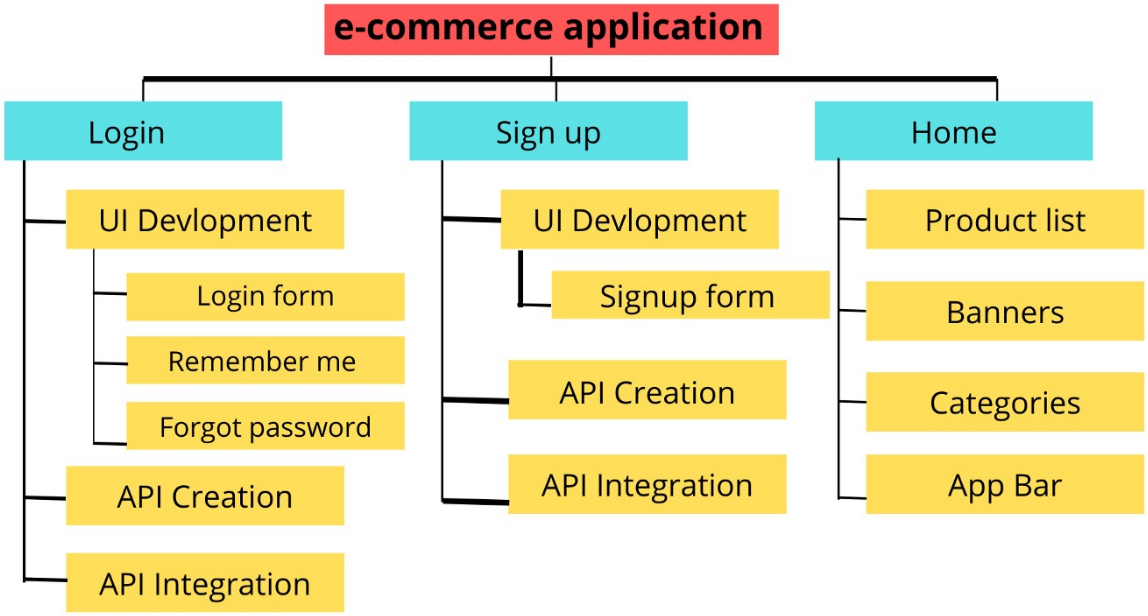
Estimating Activity Durations

↓

Developing the Schedule

Defining Activities

- Activity: An activity or task is an element of work normally found on the work breakdown structure (WBS) that has expected duration, cost, and resource requirements.

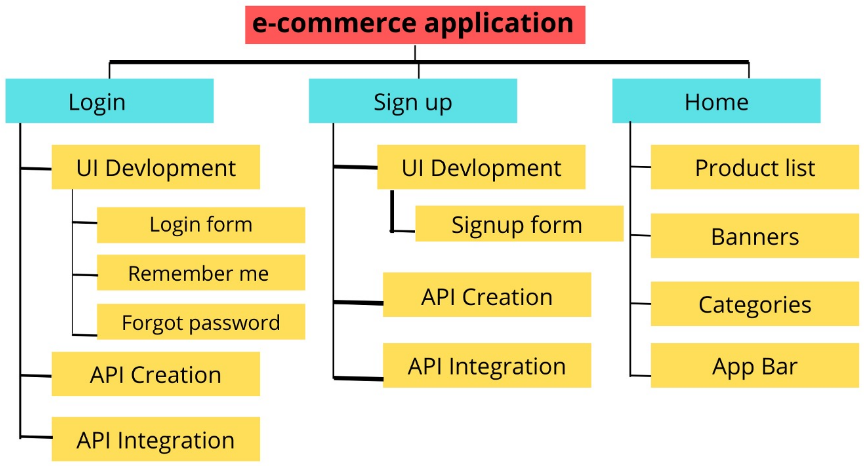


Defining Activities (2)

- Activity List: is a tabulation of activities to be included on a project schedule. It should include:
  - The activity name;
  - An activity identifier or number;
  - A brief description of the activity
- Activity Attribute: provide schedule-related information about each activity, such as
  - predecessors, successors;
  - logical relationships;
  - resource requirements, constraints, imposed dates, and assumptions

Defining Activities (3)

- Milestones: on a project is a significant event that normally has no duration. It often takes several activities and a lot of work to complete a milestone.
- Milestones are also useful tools for setting schedule goals and monitoring progress.



Defining Activities (4)

- Defining activities involves identifying the specific actions that will produce the project deliverables in enough detail to determine resource and schedule estimates.

Sequencing Activities

- A dependency or relationship pertains to the sequencing of project activities or tasks.
- For example,
  - Does a certain activity have to be finished before another can start?
  - Can the project team do several activities in parallel?
  - Can some overlap?
- Determining these relationships or dependencies among activities is crucial for developing and managing a project schedule.

Sequencing Activities -- Dependency

- There are three basic reasons for identifying dependencies among project activities:
- Mandatory dependencies: are inherent in the nature of the work being performed on a project.
  - You cannot test code before the code is written;
- Discretionary dependencies: are defined by the project team.
  - a team might not start the detailed design until the users sign off on all of the analysis work.

Dependency

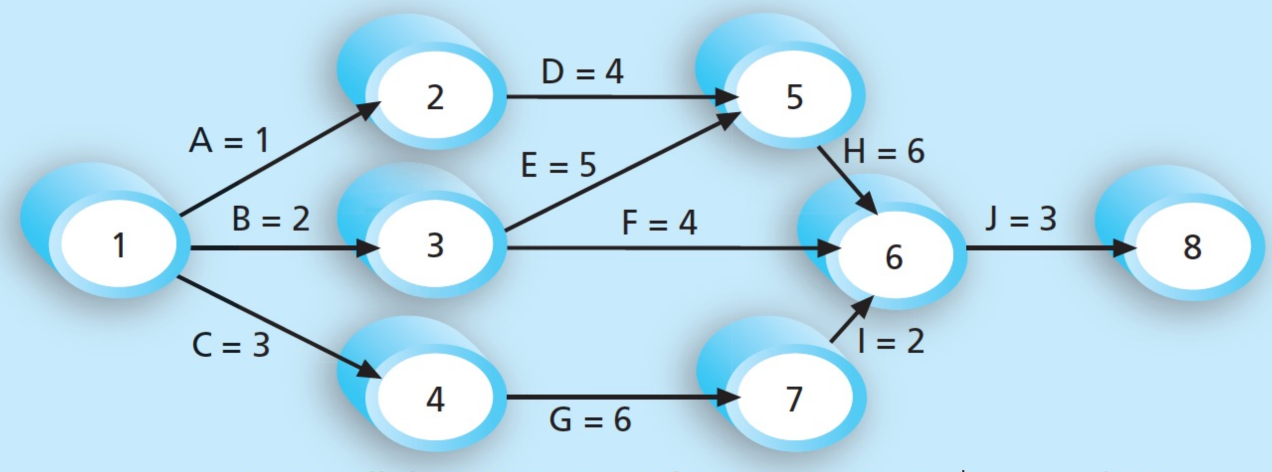
- External dependencies: involve relationships between project and non-project activities.
  - the installation of a new operating system and other software may depend on delivery of new hardware from an external supplier.

Sequencing Activities --- Network Diagram

- Network diagrams are the preferred technique for showing activity sequencing.
- A network diagram is a schematic display of the logical relationships among project activities and their sequencing.
- Types of network diagram:
  – AOE network diagram
  – AOV network diagram

AOE Network Diagram

- The activity-on-edge (AOE) approach
- Vertex: is simply the starting and ending point of an activity.
- Edge: the activity and its attribute.



Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

Draw AOE Network Diagram

- Follow these steps to create an AOE network diagram:
- STEP 1.
  - 1.1 Find all of the activities that start at Node 1.
  - 1.2 Draw their finish nodes, and draw arrows between Node 1 and each of the finish nodes.
  - 1.3 Put the activity letter or name on the associated arrow.

AOE Network Diagram – STEPs

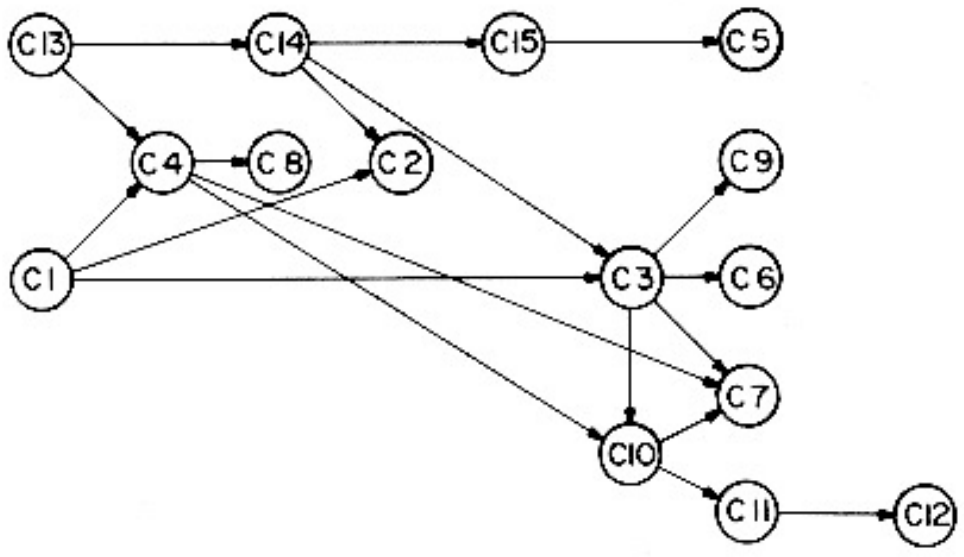
- STEP 2.
  - 2.1 Continue drawing the network diagram, working from left to right.
  - 2.2 Look for bursts and merges. Bursts occur when two or more activities follow a single node. A merge occurs when two or more nodes precede a single node.
- STEP 3. Continue drawing the AOE network diagram until all activities are included.
- STEP 4. All arrowheads should face toward the right, and no arrows should cross on an AOE network diagram

AOE Network Diagram

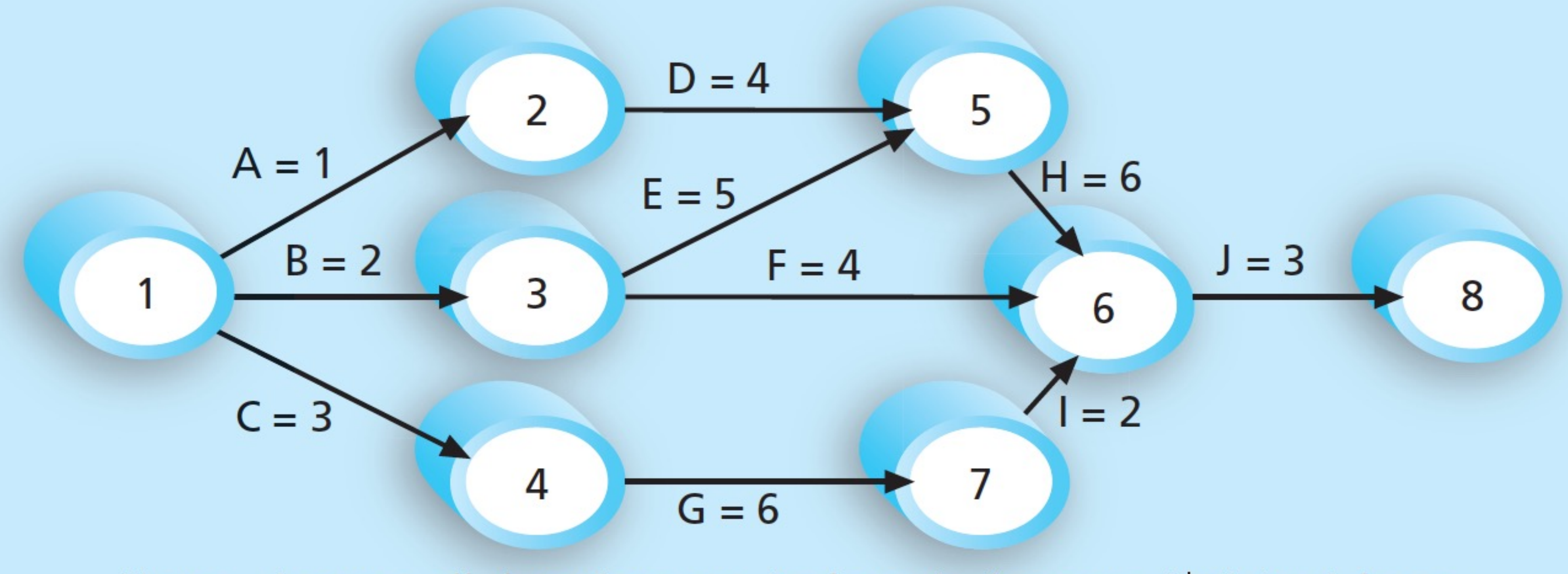
Activity	Initial Node	Final Node	Estimated Duration
A	1	2	2
B	2	3	2
C	2	4	3
D	2	5	4
E	3	6	2
F	4	6	3
G	5	7	6
H	6	8	2
I	6	7	5
J	7	8	1
K	8	9	2

AOV Network Diagram

- The activity-on-vertex (AOV) approach
- Vertex: the activity and its attribute.
- Edge: is simply the starting and ending point of an activity.



AOE to AOV



Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

Dummy Activity

- Dummy Activity
- Dummy activities have no duration and no resources, but are occasionally needed on AOA network diagrams to show logical relationships between activities.
- These activities are represented with dashed arrow lines and have zeros for their duration estimates.

Estimating Activity Resources

- Before estimate the duration for each activity, the quantity and type of resources (people, equipment, and materials) that will be assigned to each activity.
- The nature of the project and the organization will affect resource estimates.

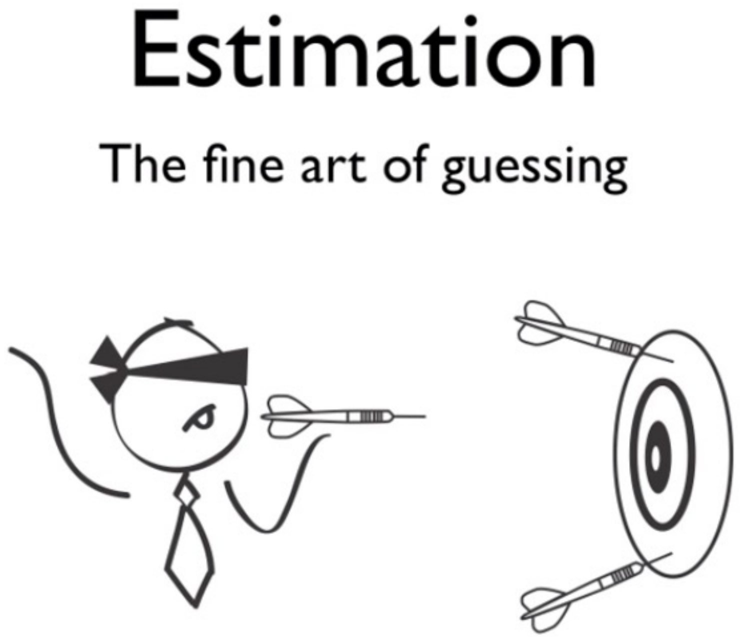
Estimating Activity Resources (2)

- Important questions to answer when estimating activity resources include:
  - How difficult will specific activities be on this project?
  - Is anything unique in the project's scope statement that will affect resources?
  - What is the organization's history in doing similar activities? Has the organization done similar tasks before? What level of personnel did the work?
  - Does the organization have people, equipment, and materials that are capable and available for performing the work? Could any organizational policies affect the availability of resources?
  - Does the organization need to acquire more resources to accomplish the work?

Estimating Activity Durations

- Duration includes the actual amount of time worked on an activity plus elapsed time.

Ref: Project Size Estimation



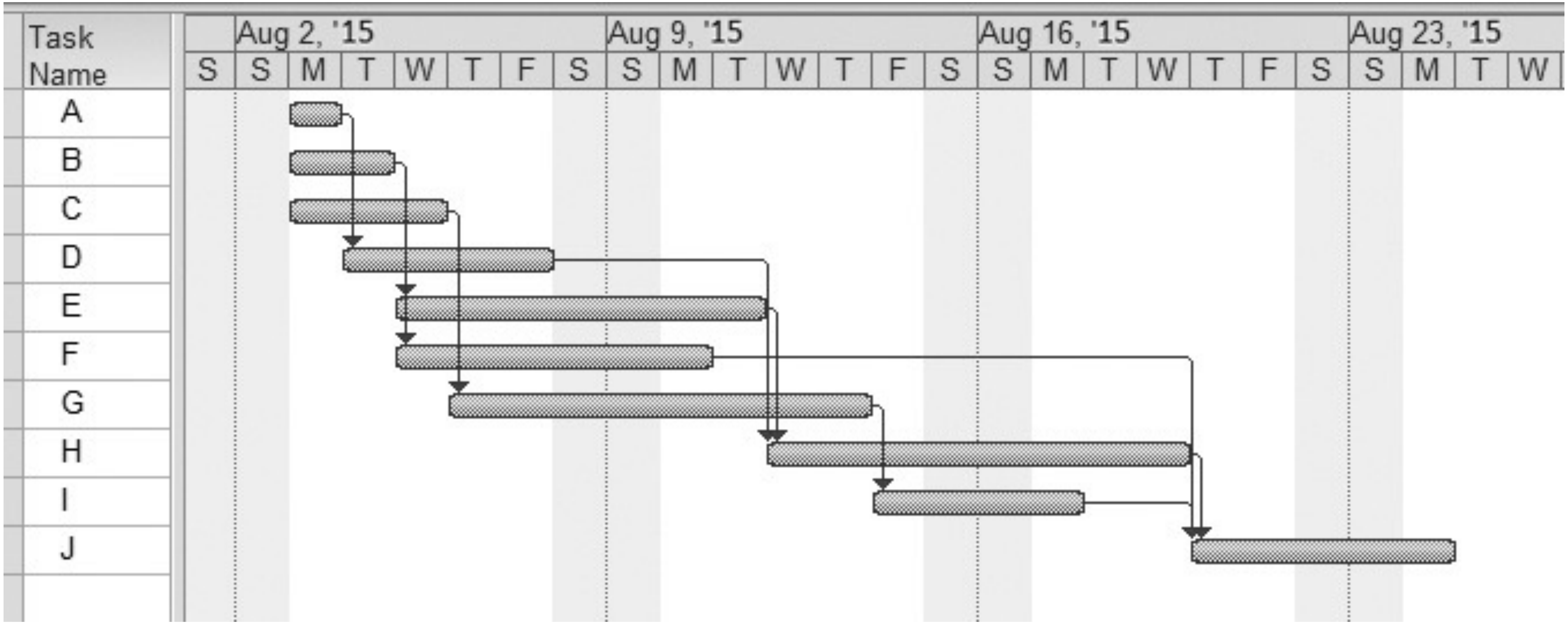
Developing the Schedule

- Schedule development uses the results of all the preceding project time management processes to determine the start and end dates of the project and its activities.
- The ultimate goal of developing a realistic project schedule is to provide a basis for monitoring project progress for the time dimension of the project.

Developing the Schedule (2)

- Several tools and techniques assist in schedule development:
- Gantt chart （甘特图）
  - A Gantt chart is a common tool for displaying project schedule
- Critical path analysis
  - Critical path analysis is a very import tool for developing and controlling project schedules.
- PERT analysis
  - PERT analysis is a means for considering schedule risk on project.

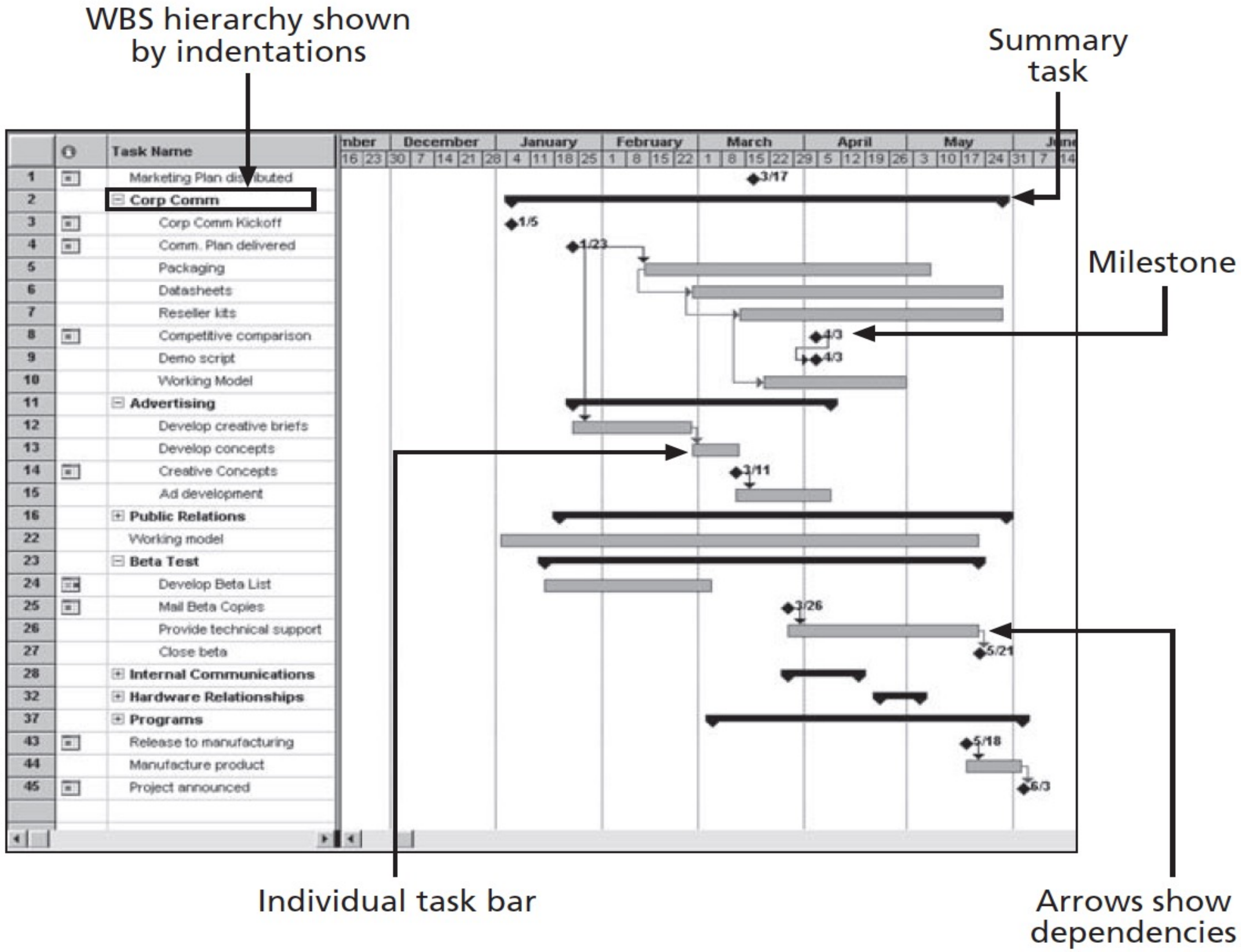
Gantt Chart


Task Name
Aug 2, '15
Aug 9, '15
Aug 16, '15
Aug 23, '15
S S M T W T F S S M T W T F S S M T W T F S S M T W
A
B
C
D
E
F
G
H
I
J

Gantt Chart (2)

- The activities on the Gantt chart
  - driven by the deliverables on the WBS;
  - coincide in turn with the activity list and milestone list.
- Gantt Chart contains
  - milestones
  - tasks
  - resources
  - individual task durations,
  - arrows showing task dependencies.

Gantt Chart (3)



Adding Milestones to Gantt Chart

- Milestones can be a particularly important part of schedules, especially for large projects.
- Milestones are used to
  - Emphasize important events;
  - Accomplishments on projects;
- Milestones are tasks with zero duration.

Adding Milestones to Gantt Chart (2)

- To make milestones meaningful, people use SMART criteria to help define them.
- Specific: milestones should be clear and specific;
- Measurable: milestones should be measurable, so that you can track your progress;
- Achievable: milestones needs to be realistic and attainable to be successful;
- Relevant: milestones needs to be aligned with relevant tasks;
- Time-bound: a milestone needs a target date;

Tools for Gantt Chart

- Microsoft Project
- GanttProject
- OpenProject
- TeamGantt
- WorkZone
- …

Gantt Chart with MS Project 2016

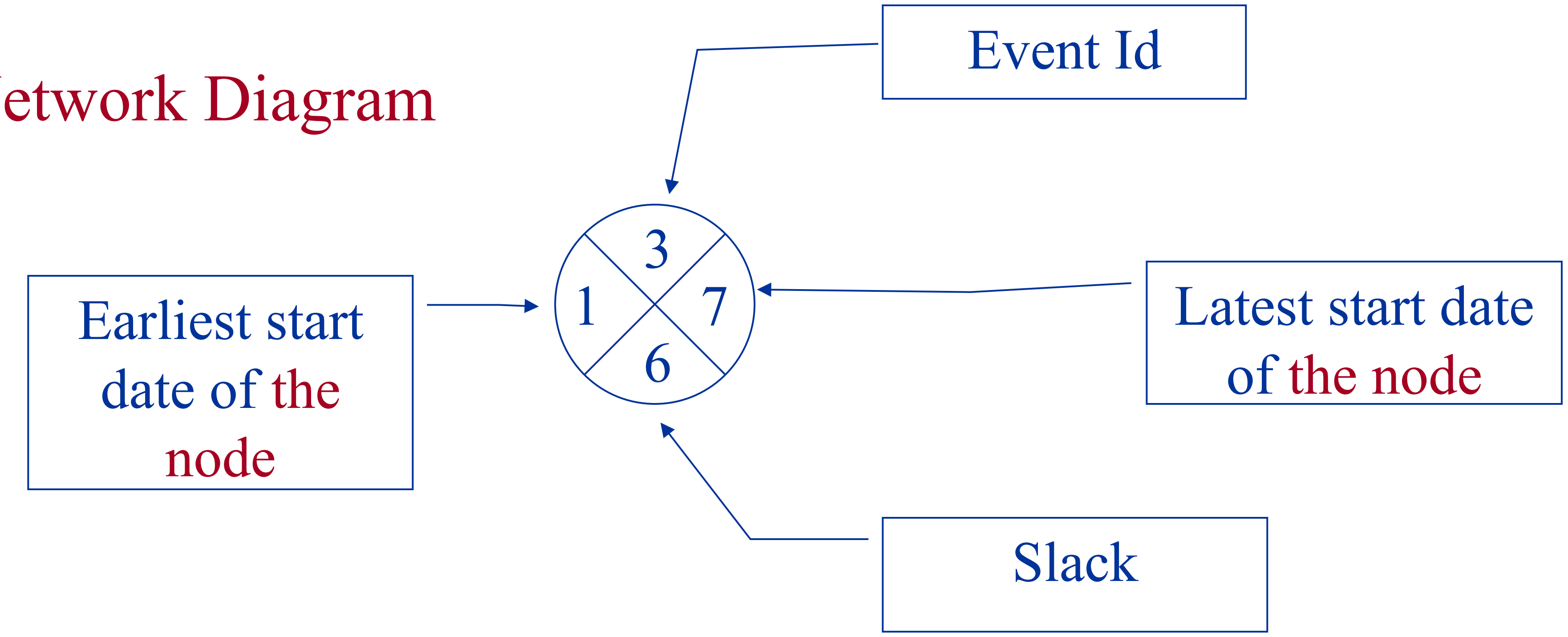
- STEP 1. Download MS Project 2016 from:
  - http://software.lib.shanghaitech.edu.cn/Microsoft/Project_Pro_2016_64Bit_ChnSimp.ISO (for Windows System Only)
- STEP 2. Install MS Project 2016
- STEP 3. Build Gantt Chart with MS Project

Critical Path Method (CPM)

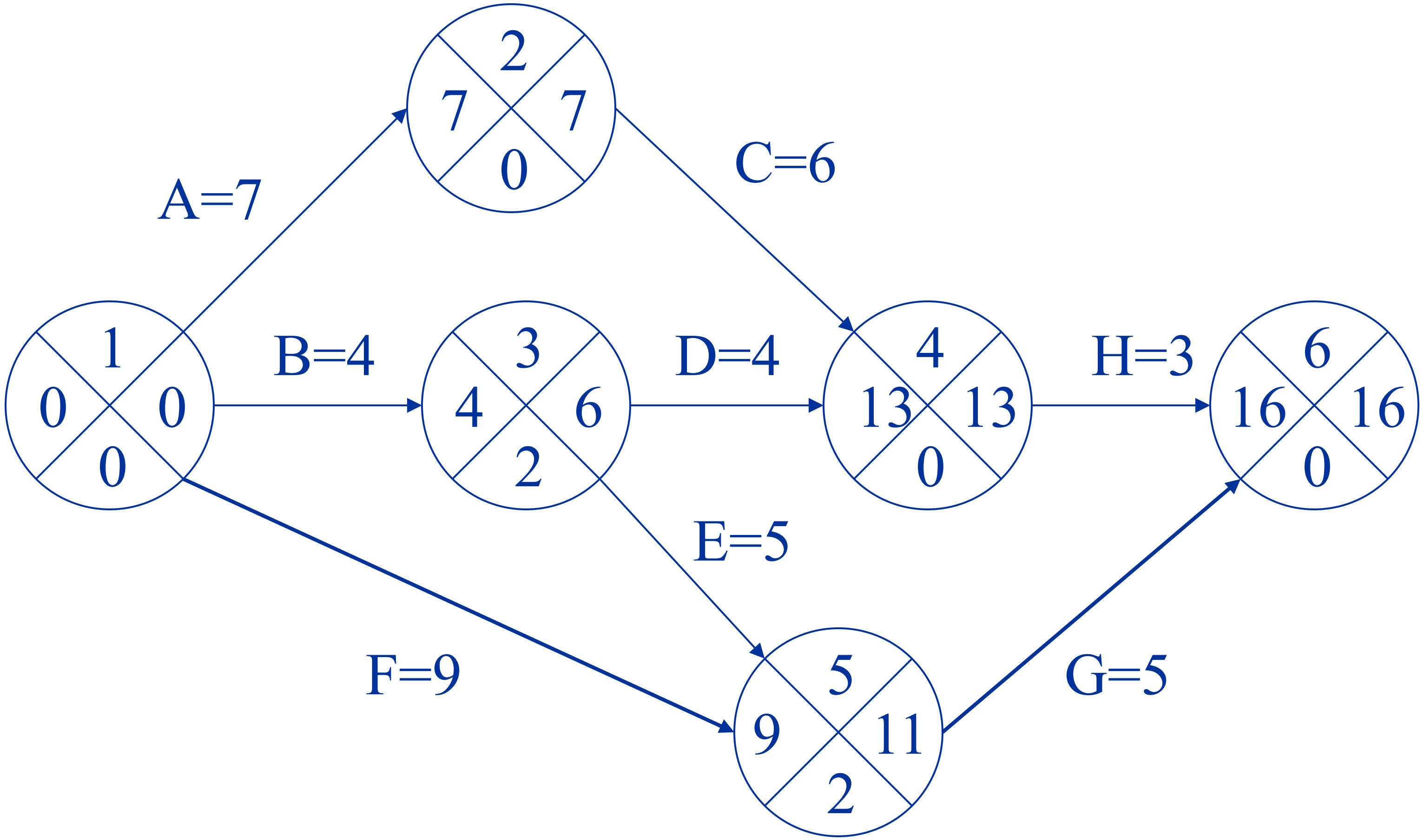
- **Critical Path Method (CPM):** is also critical path analysis—is a network diagramming technique used to predict total project duration.
- **Critical Path**: A critical path for a project is the series of activities that determine the earliest time by which the project can be completed.
- **Slack or float** is the amount of time an activity may be delayed without delaying a succeeding activity or the project finish date.

Critical Path Method (2)

AOE Network Diagram

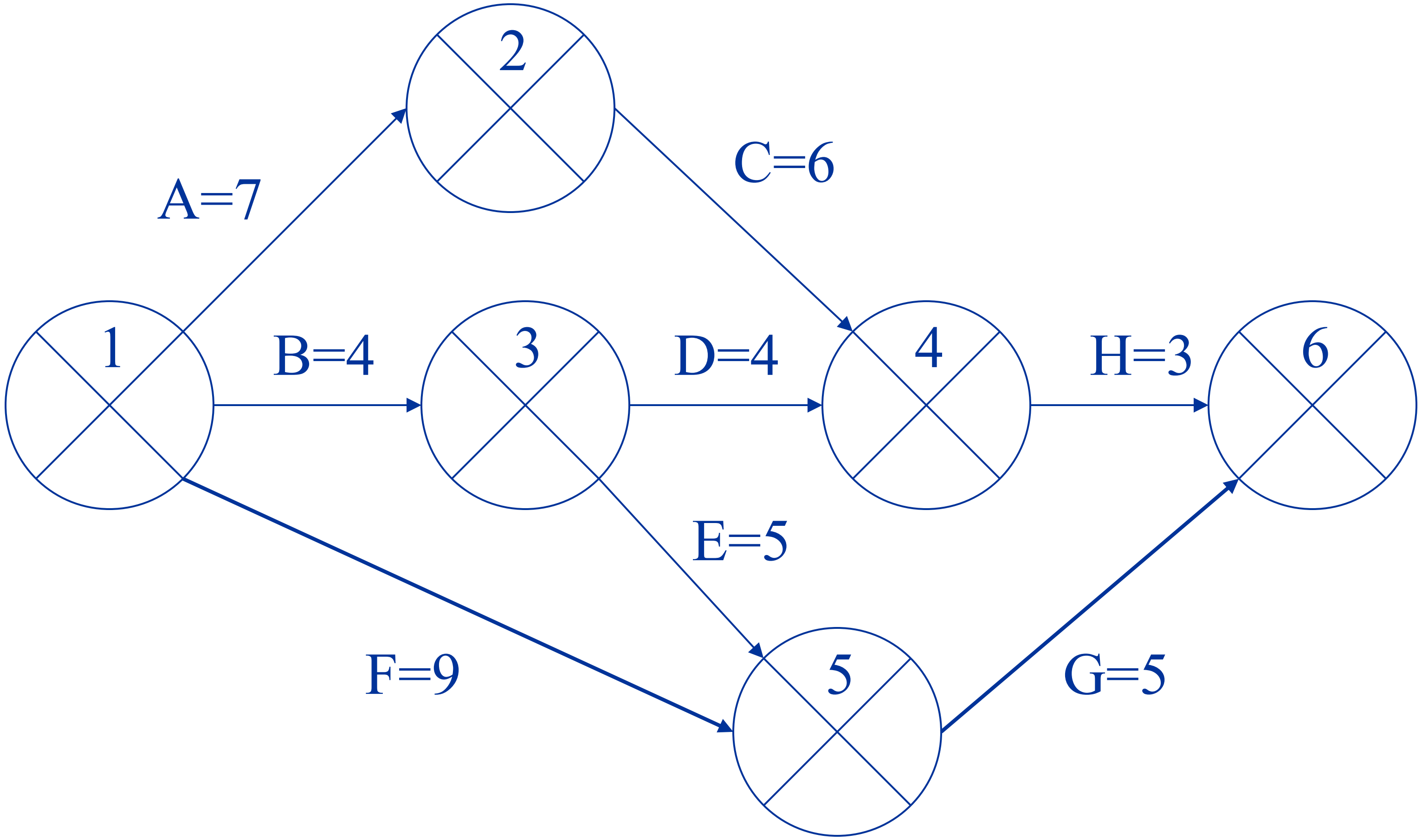


Critical Path Method (3)



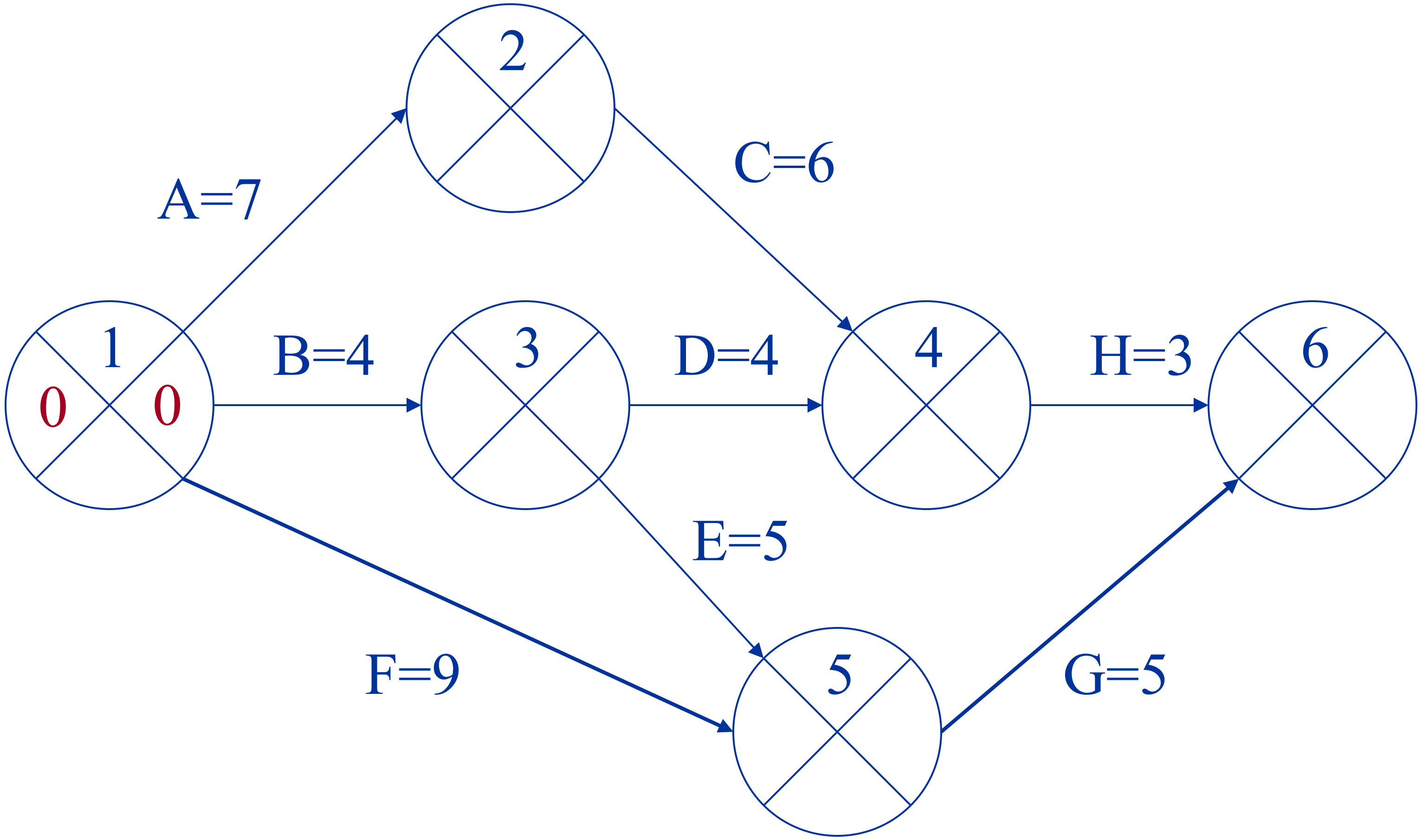
Critical Path Method (4) STEPs

- STEP 1. Build AOE Network Diagram.



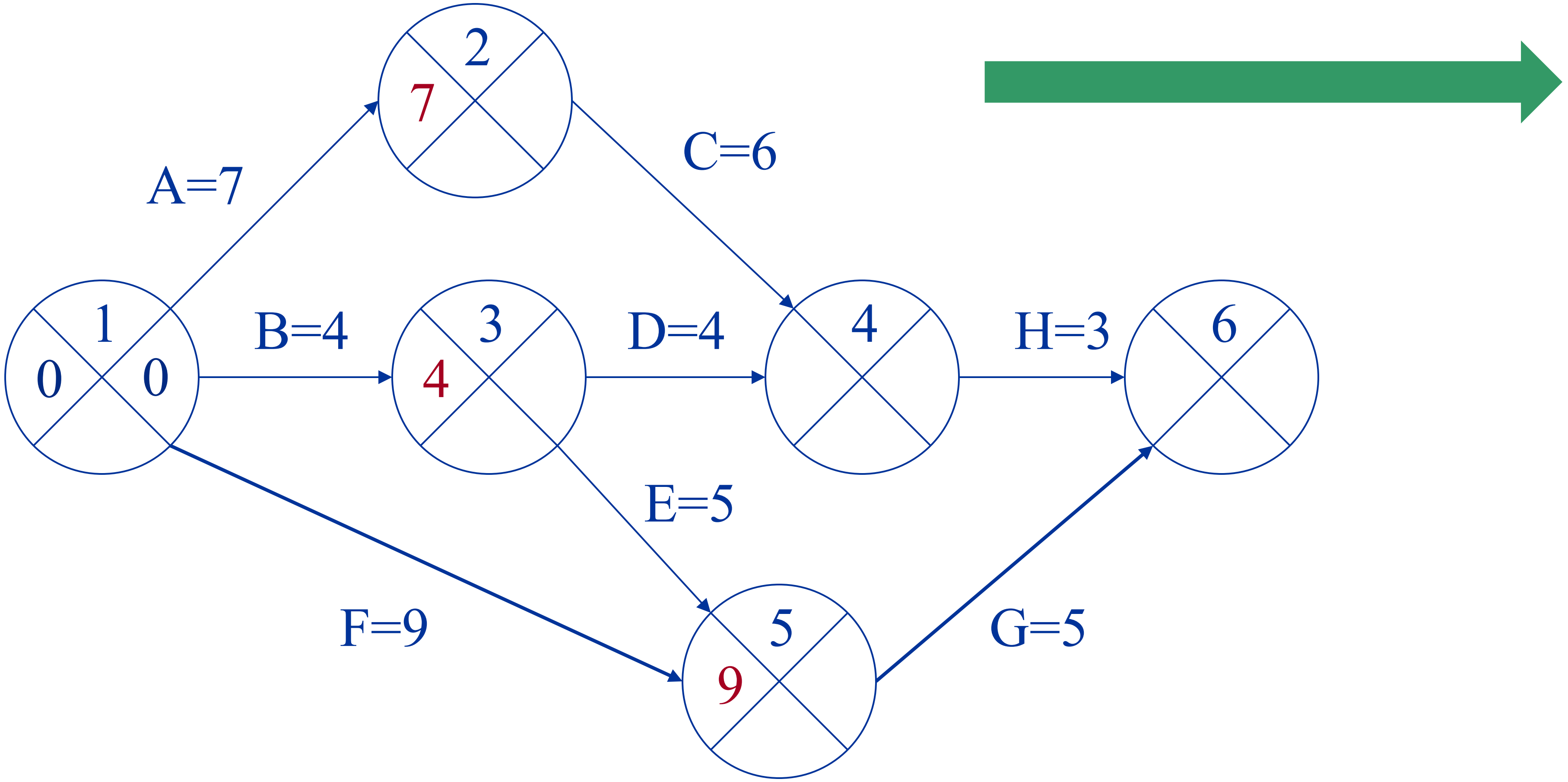
Critical Path Method (5) STEPs

- STEP 2. Set Node 1's Earliest start date and Latest start date to 0.



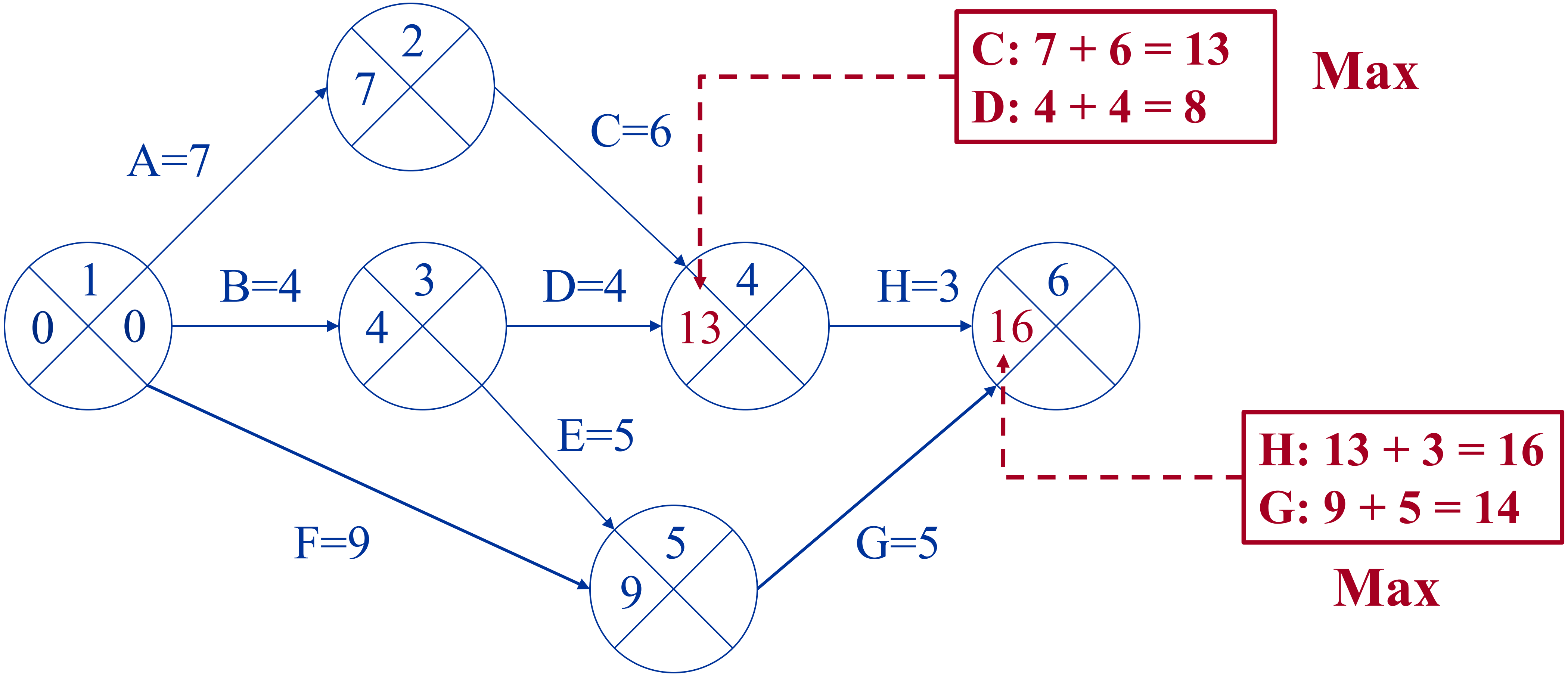
Critical Path Method (6) STEPs

- STEP 3. Compute the Earliest start date fields in diagram.



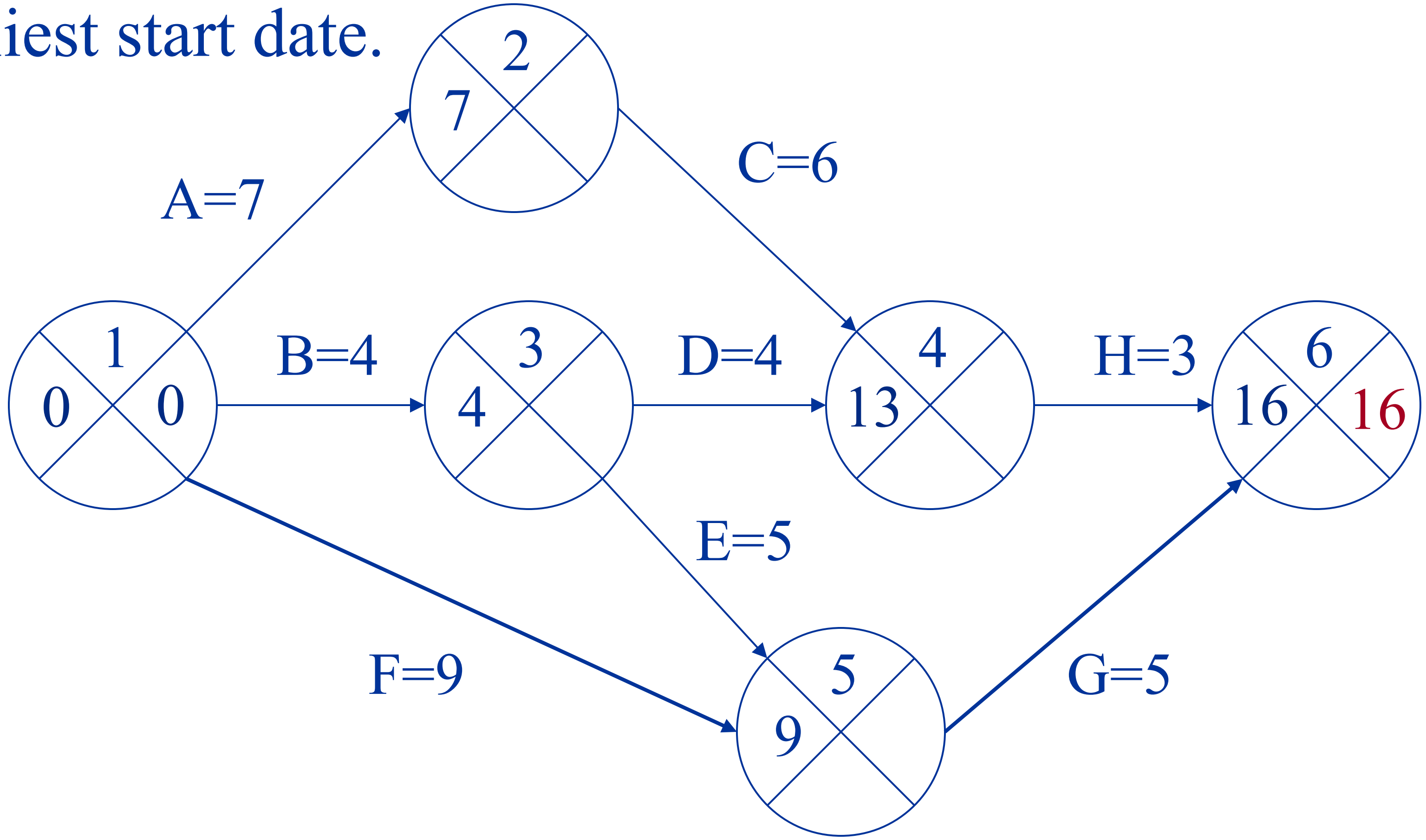
Critical Path Method (7) STEPs

- STEP 3. Compute the Earliest start date fields in diagram.



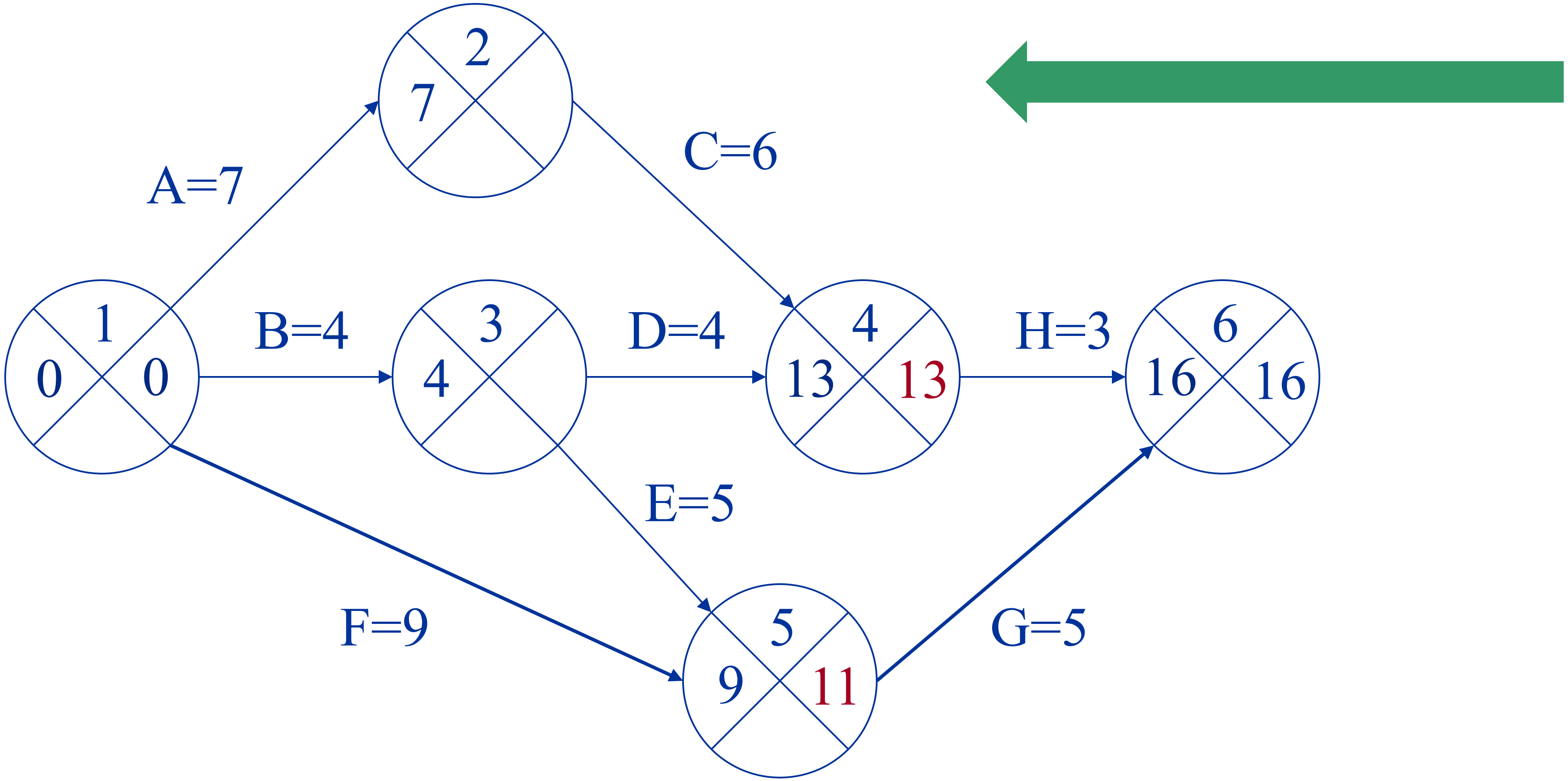
Critical Path Method (8) STEPs

- STEP 4. Set the Latest start date of the last node to the same as its Earliest start date.



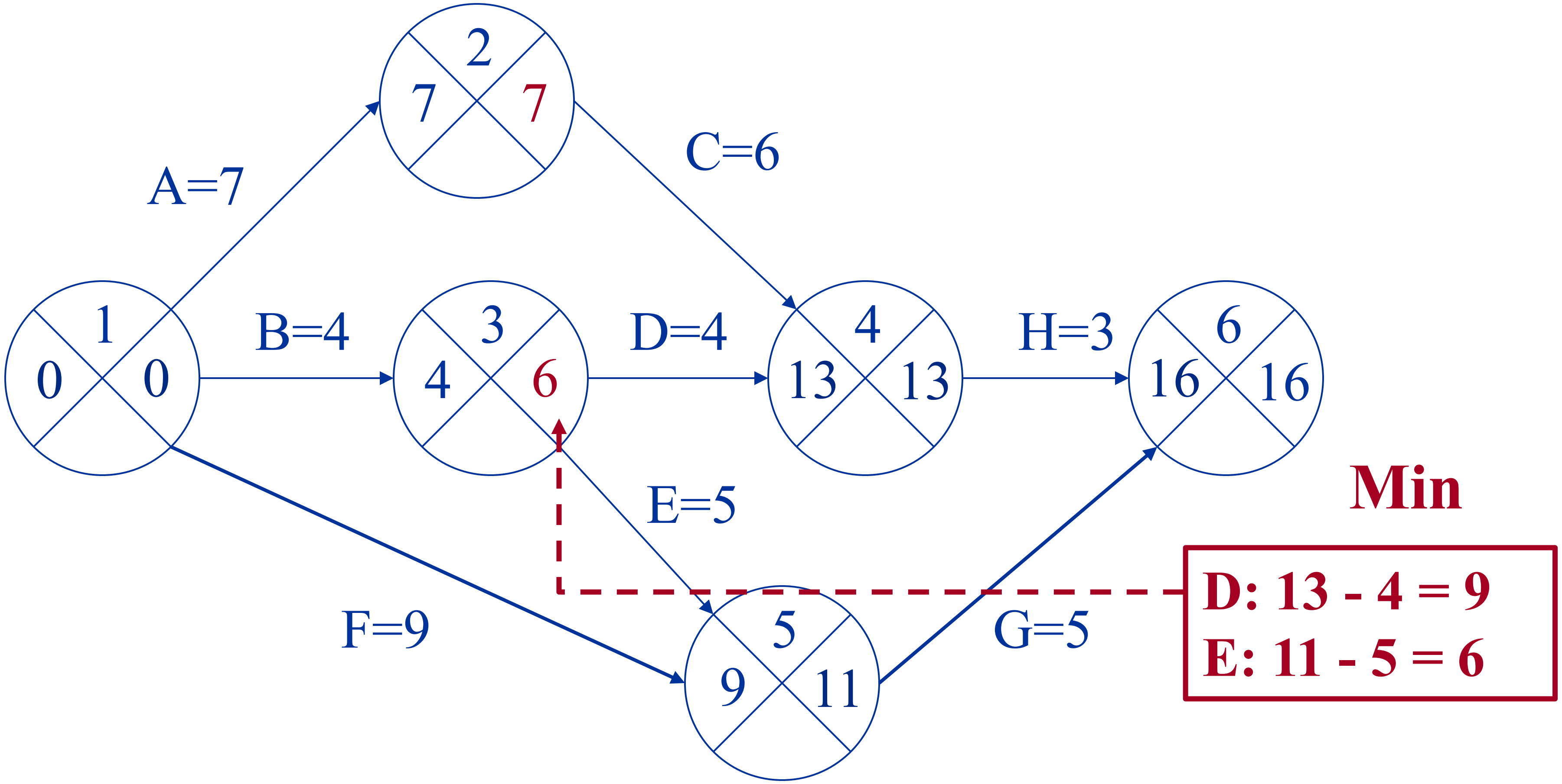
Critical Path Method (8) STEPs

- STEP 5. Compute the Latest start date fields in diagram..



Critical Path Method (9) STEPs

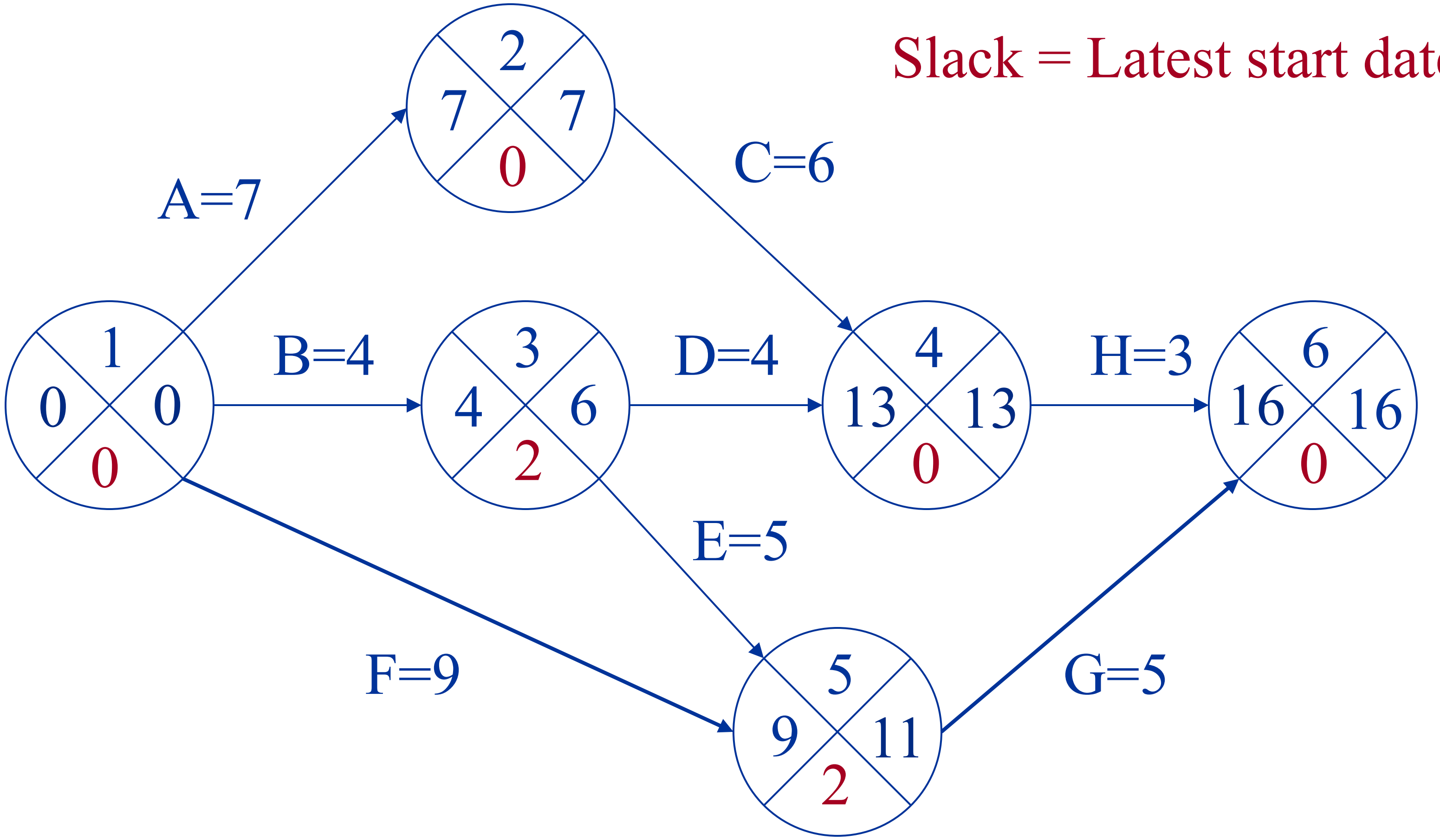
- STEP 5. Compute the Latest start date fields in diagram.



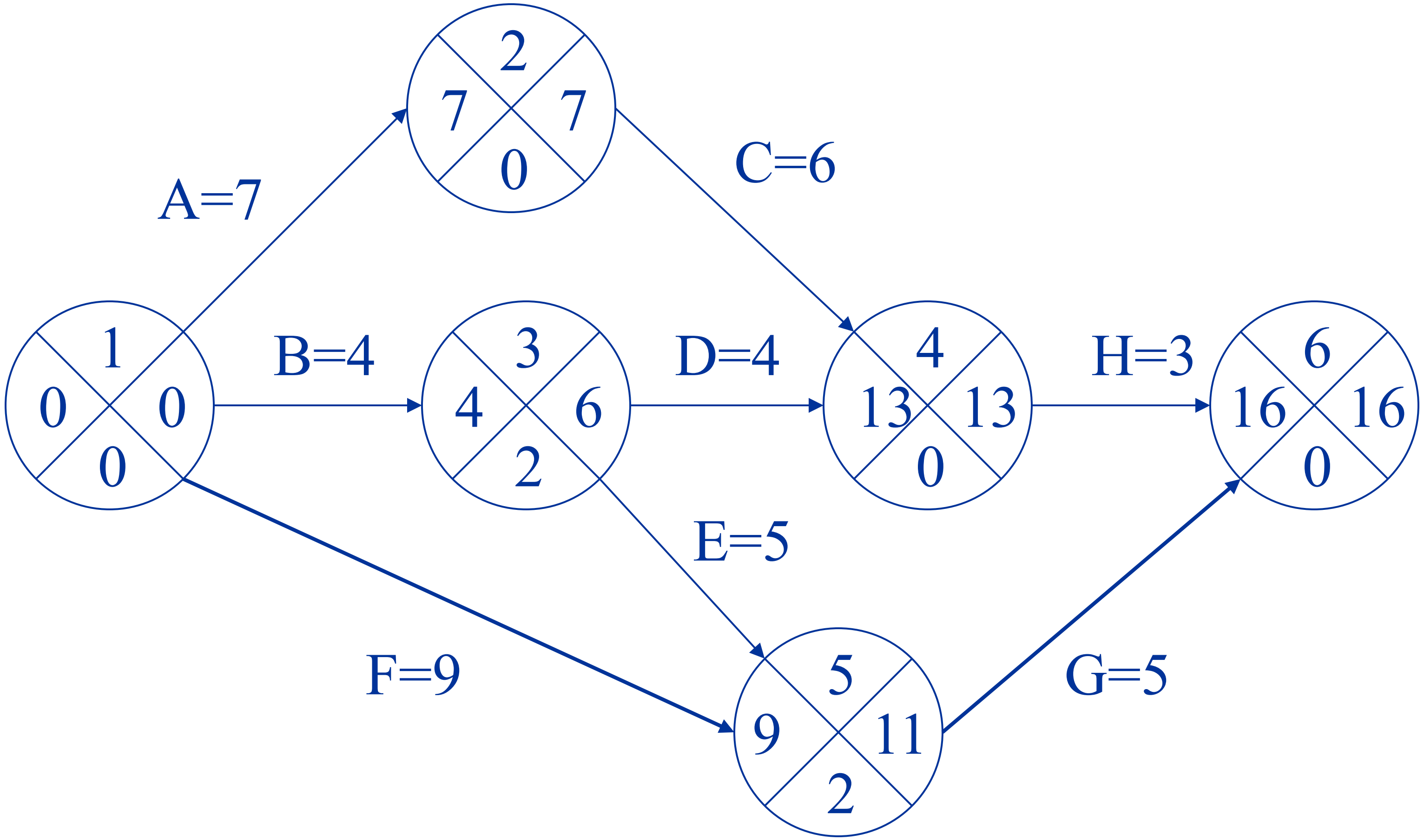
Critical Path Method (9) STEPs

- STEP 6. Compute all Stacks in diagram;

Slack = Latest start date- Earliest start date



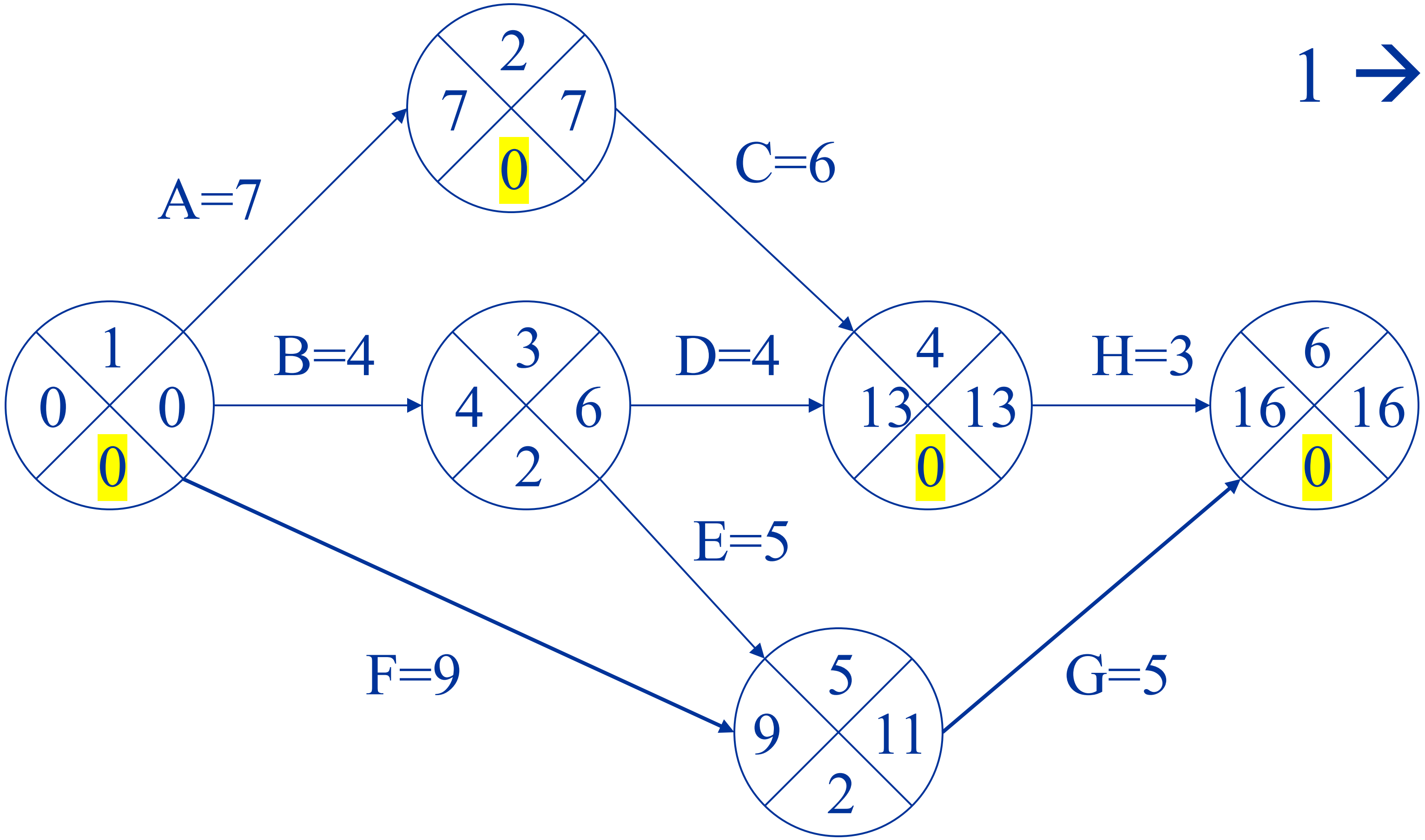
Critical Path Method (10)



Critical Activities and Critical Path

- Activities with zero slack cannot be delayed without affecting the entire project duration and are called critical activities.
- The critical path consists of a path of critical activities from the start to the end.

Critical Activities and Critical Path (2)



1 → 2 → 4 → 6

PERT Analysis

- Program Evaluation and Review Technique (PERT) can be used to estimate project duration.
- PERT uses probabilistic time estimates—duration estimates based on
  - using optimistic,
  - most likely, and
  - pessimistic estimates of activity durations

$$\text{PERT weight average} = \frac{\text{optimistic time} + 4 * \text{most likely time} + \text{pessimistic time}}{6}$$

Recap

- Defining Activities
- Sequencing Activities
- Estimating Activity Resources
- Estimating Activity Durations
- Developing the Schedule