
HUST
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
ONE LOVE. ONE FUTURE.

1


SOICT
School of Information and Communication Technology
ONE LOVE. ONE FUTURE.

2

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

IT3180 – Introduction to Software Engineering

7 – Project Management

ONE LOVE. ONE FUTURE.

3

3

The Aim of Project Management

To complete a project:

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

To provide visibility about the progress of a project

To give early warning of problems so that corrections can be made

4

4

The challenge of Project Management (1)

What do clients want to know?

- Will the system do what was promised? (**Function**)
- When will it be delivered? If late, how late? (**Time**)
- How does the cost compare with the budget? (**Cost**)

The challenge of Project Management (2)

Often, the software is a part of larger activity:

- If the system is a product, marketing and development must be combined (e.g., Microsoft Office)
- If the system has to work with other systems, developments must be coordinated (e.g., embedded systems in an automobile)

The challenge of Project Management (3)

BUT:

- Every software system is different.
- Most systems are not well specified, or the requirements change during development.
- Estimate time and effort is full of errors, even when the system is well understood.

7

7

Aspects of Project Management (1)

- **Planning**
  - Outline schedule during feasibility study
  - Full schedule for each part of a project (e.g., each process step, iteration, or sprint)
- **Contingency planning**
  - Anticipate possible problems (risk management)

8

8

Aspects of Project Management (2)

- **Progress tracking**
  - Regular comparison of progress against plan
  - Regular modification of the plan
  - Changes of scope, etc. made jointly by client and developers
- **Final analysis**
  - Analysis of project for improvements during next project

Terminology (1)

- **Deliverable**
  - Work product that is provided to the client (mock-up, demonstration, prototype, report, presentation, documentation, code, etc.)
  - Release of a system or subsystem to customers and users
- **Milestone**
  - Completion of a specified set of activities (e.g., delivery of a deliverable, completion of a process step, end of a sprint)

Terminology (2)

- **Activity**
  - Part of a project that takes place over time (also known as a **task**)
  - Release of a system or subsystem to customers and users
- **Event**
  - The end of a group of activities, e.g., agreement by all parties on the budget and plan
- **Dependency**
  - An activity that cannot begin until some event is reached
- **Resource**
  - Staff time, equipment, or other limited resource required by an activity

11

11

Standard approach to Project Management

- The **scope** of the project is defined early in the process.
- The development is divided into **tasks** and **milestones**.
- **Estimates** are made of the **time** and **resources** needed for each task.
- The estimates are combined to create a **schedule** and a **plan**.
- **Progress** is continually **reviewed** against the plan, perhaps weekly.
- The **plan** is **modified** by changes to scope, time, resources, etc.

Typically the plan is managed by a **separate project management team**, not by the software developers.

Used with the Modified Waterfall Model and Iterative Refinement.

12

12

Agile Approach to Project Management

- Planning is divided into high level **release forecasting** and low level **detailed planning**.
- Release planning is a best guess, high level view of what can be achieved in a sequence of **time-boxes**.
- Release plans are continually **modified**, perhaps daily.
- **Clients** and **developers** take joint control of the release plans and choice of sprints.
- For each **time-box**, the **team** plans what it can achieve.

The team may use **Gantt charts** or other conventional planning tools.

13

13

Project Planning Tools

Critical Path Method, Gantt Charts

- Build a work-plan from activity data
- Display work-plan in graphical or tabular form.

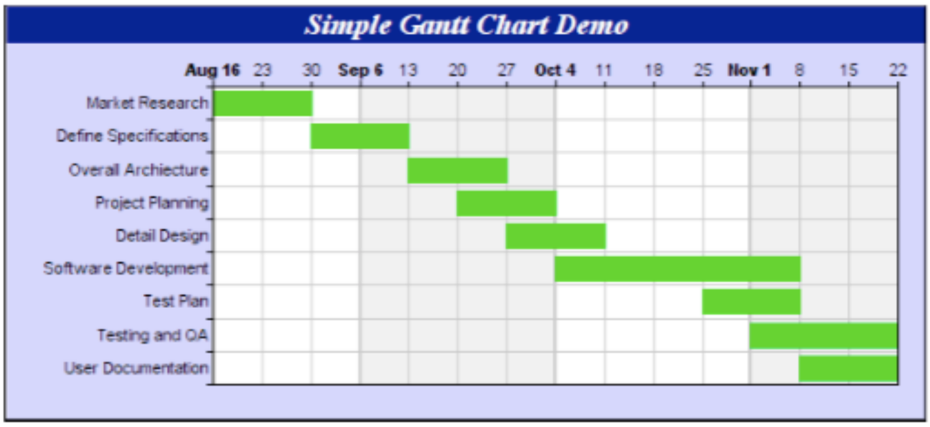
Project planning software (e.g., Microsoft Project)

- Maintain a database of activities and related data
- Calculate and display schedules
- Manage progress reports

14

14

A Simple Gantt Chart



Source: Advanced Software Engineering Limited

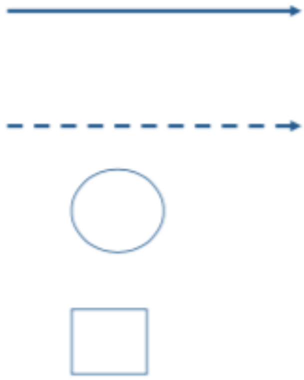
Gantt Chart (1)

Used for small projects, single 1me-boxes, and sprints

- Dates run along the top (days, weeks, or months).
- Each row represents an activity.
- Activities may be sequential, in parallel or overlapping.
- The schedule for an activity is a horizontal bar.
- The left end marks the planned beginning of the task.
- The right end marks the expected end date.
- The chart is updated by filling in each activity to a length proportional to the work accomplished. This is often difficult.
- Progress to date can be compared with the plan by drawing a vertical line through the chart at the current date.

Activity Graph

A group of scheduling techniques that emphasizes dependencies:



An activity (task)

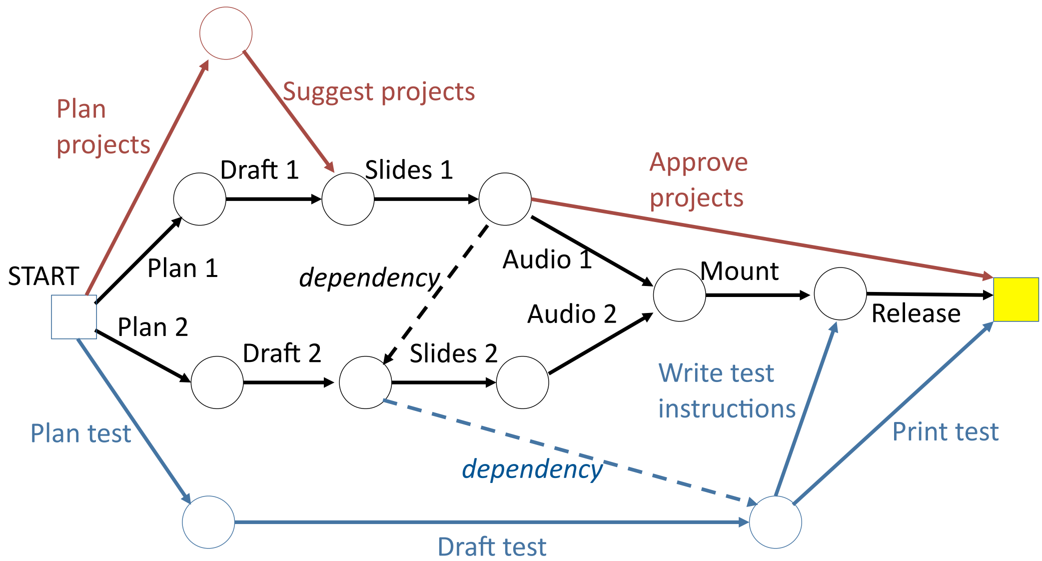
A dummy activity (dependency)

An event

A milestone

17

Example: Activity Graph



18

Scheduling using Activity Graphs: History

PERT

- Program Evaluation and Review Technique introduced by the U.S. Navy in 1957 to support the development of its Polaris submarine missile program.

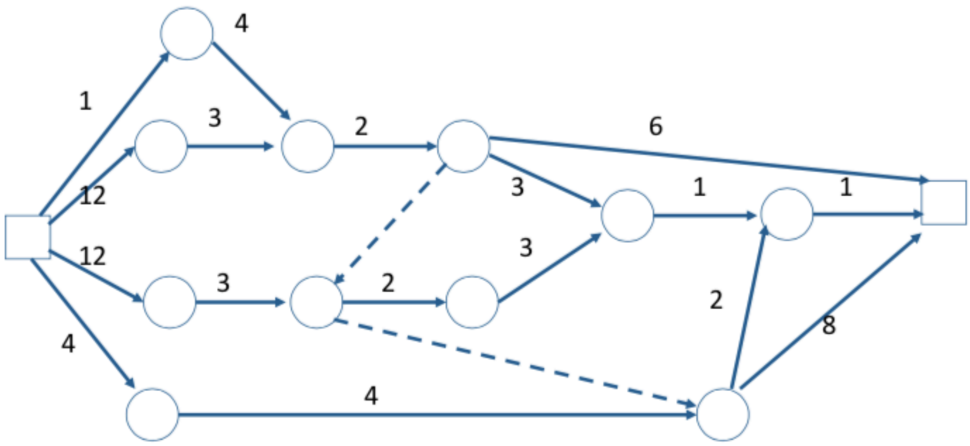
PERT/Time

- Activity graph with three time estimates (shortest, most probable, longest) on each activity to compute schedules.
- Because of the difficulty of obtaining good time estimates, usually only one esDmate is made. This is called the **Critical Path Method**.

PERT/Cost

- Added scheduling of resources (e.g., facilities, skilled people, etc.)

Time Estimates for Activities (weeks)



Example: Building a house

Project activities:

- install landscaping
- pour foundations
- frame walls
- install plumbing systems
- get permits
- install electrical systems
- move in

Example: Building a house

Activities in order, Durations, Labels, Dependencies

Project tasks	Durations	Labels	Preds.	Post
Get permits				
Pour foundations				
Frame walls				
Install plumbing systems				
Install electrical systems				
Install landscaping				
Move in				

Example: Building a house

Activities in order, **Durations**, Labels, Dependencies

Project tasks	Durations	Labels	Preds.	Post
Get permits	2			
Pour foundations	6			
Frame walls	5			
Install plumbing systems	4			
Install electrical systems	6			
Install landscaping	9			
Move in	3			

Example: Building a house

Activities in order, Durations, **Labels**, Dependencies

Project tasks	Durations	Labels	Preds.	Post
Get permits	2	A		
Pour foundations	6	B		
Frame walls	5	C		
Install plumbing systems	4	D		
Install electrical systems	6	E		
Install landscaping	9	F		
Move in	3	G		

Example: Building a house

Activities in order, Durations, Labels, Dependencies

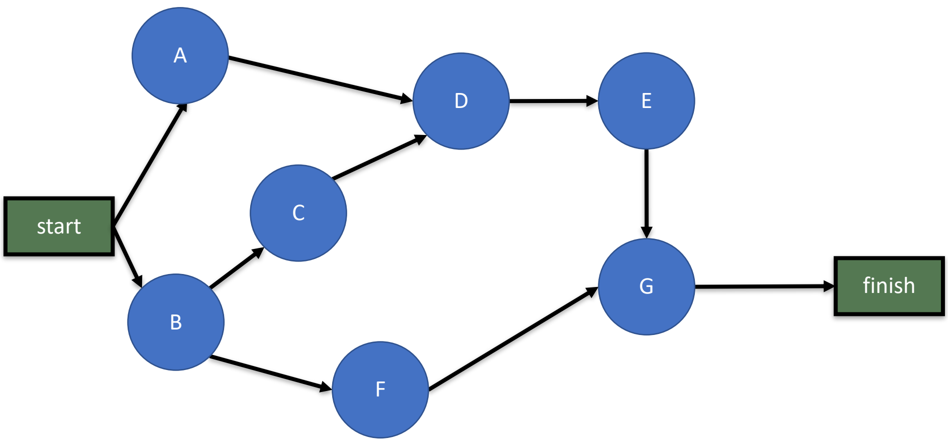
Project tasks	Durations	Labels	Preds.	Post
Get permits	2	A	--	B
Pour foundations	6	B	--	C, F
Frame walls	5	C	B	D
Install plumbing systems	4	D	A, C	E
Install electrical systems	6	E	D	G
Install landscaping	9	F	B	G
Move in	3	G	E, F	--

25

25

Example: Building a house

Create a precedency diagram

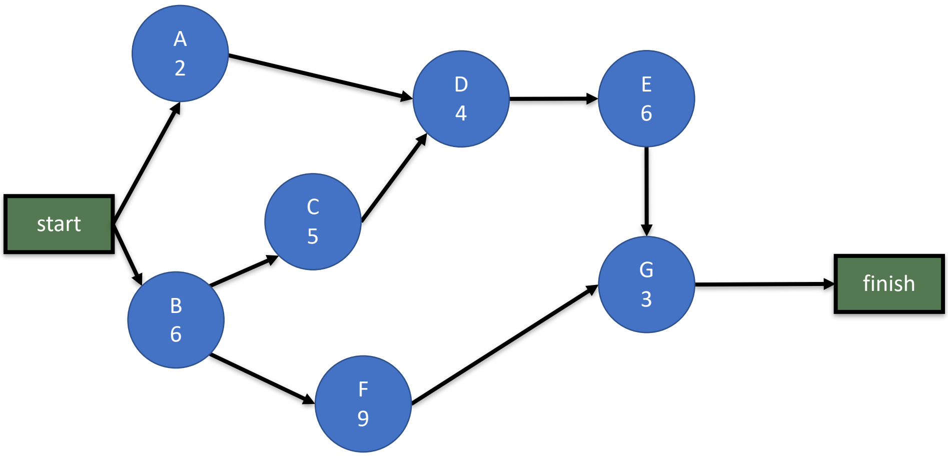


26

26

Example: Building a house

Add duration to each node in the diagram



Critical Path Method

- Uses an **Activity Graph** with single time estimate for each activity
- A standard method for managing large construction projects
- On big projects, activity graphs with more than 10,000 activities are common
- Based on the estimated duration, calculate the theoretical Early Start , Early Finish, Late Start and Late Finish for each activity

ES, EF, LF, LS

- **Earliest start date (ES)**: the earliest date that it is possible to start an activity, given that its precedent activities must be completed first
- **Earliest finish date (EF)**: the date that all the activities ending at that node will be completed, assuming that every activity begins at its earliest start date
  - Equal to the earliest start time for the activity plus the time required to complete the activity
- **Latest finish time (LF):** the latest time at which the activity can be completed without delaying the project
- **Latest start time (LS):** equal to the latest finish time minus the time required to complete the activity

29

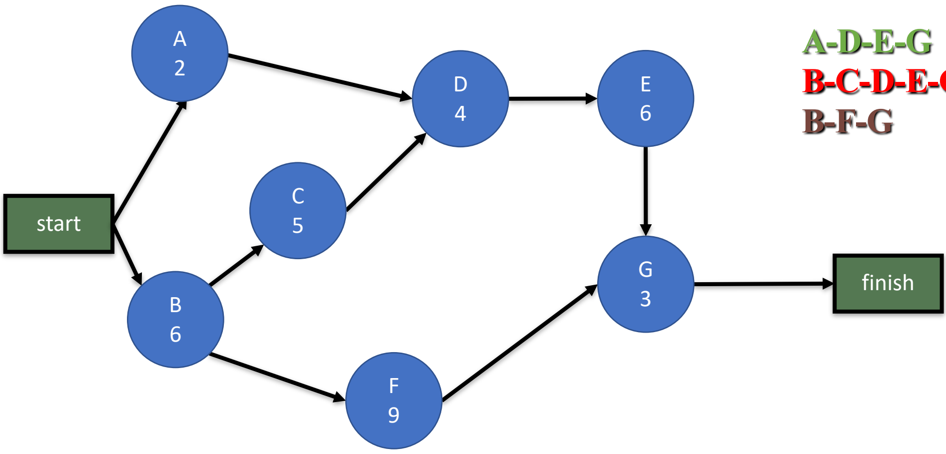
Identify Critical Path

- The critical path is the longest-duration path through the network
- Determining the following four parameters for each activity
- **Slack time** (float time): how much extra time you have available for a particular activity?

30

Example: Building a house

How many paths are there in this network?



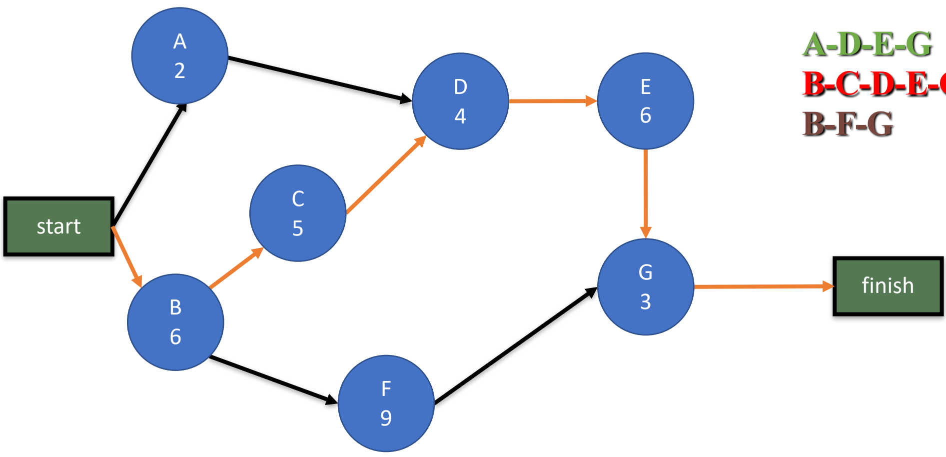
A-D-E-G = 15
B-C-D-E-G = 24
B-F-G = 18

31

31

Example: Building a house

Which is the longest path?



A-D-E-G = 15
B-C-D-E-G = 24
B-F-G = 18

32

32

Calculate Slack time

For each activity, calculate ES, EF, LS, LF and slack time

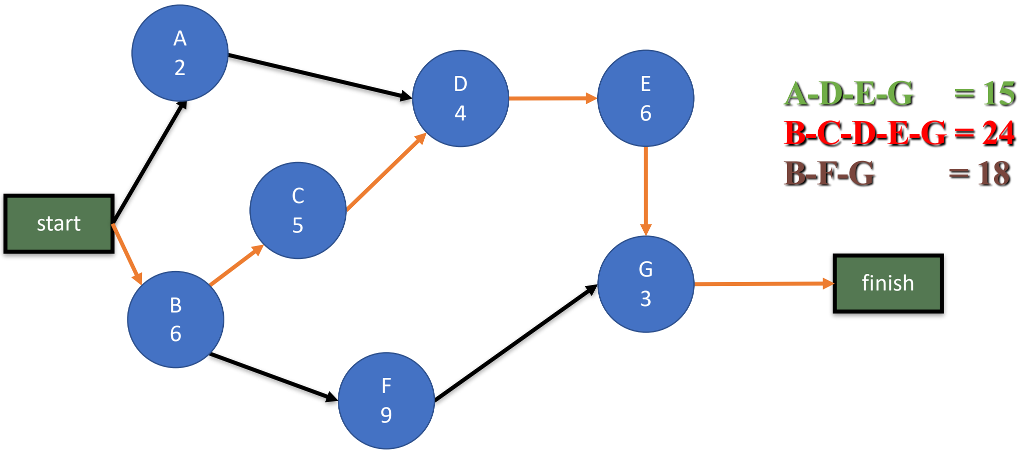
ES	Duration	EF
Activity		
LS	Slack	LF

Calculate Slack time (2)

- The float is how long an activity's duration can extend before it lengthens the project duration
- The float for any activity on the critical path is **zero**
- The float for **non-critical activities** is the critical path duration ***minus*** the duration of the activity's path
- If **an activity** is on **multiple paths**, its float is the one that is **least**

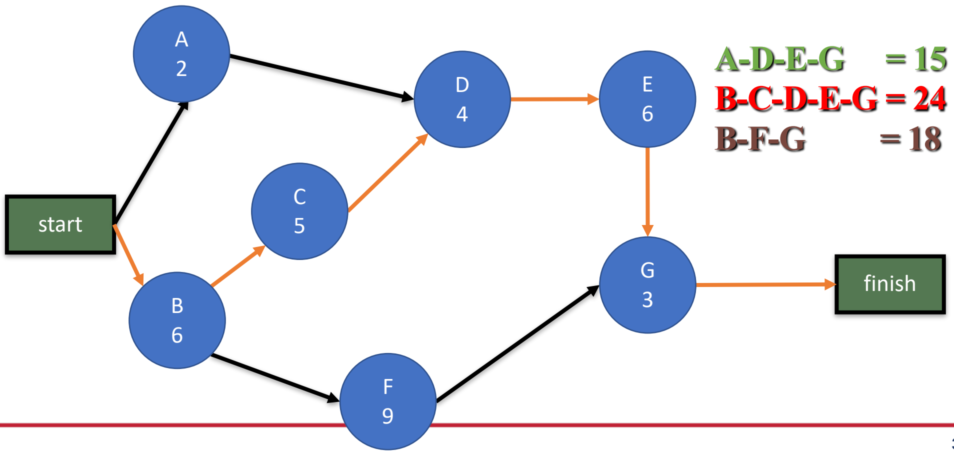
Calculate Slack time (3)

- The critical path has a duration of 24
- The Slack time of activities B, C, D, E, G are all 0.

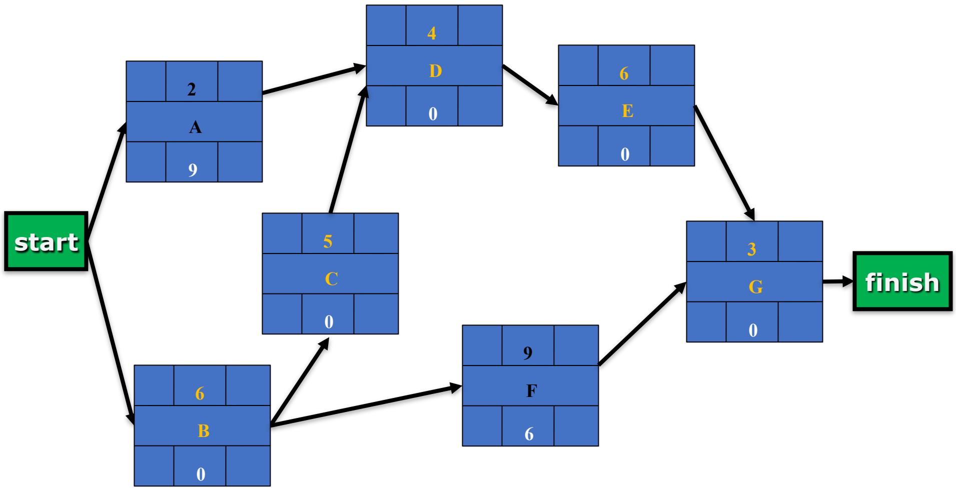


Calculate Slack time (4)

- With path B-F-G has a duration of 18, the Slack time of F (non-critical path activities) is 24–18 = 6
- What's about the activity A?



Calculate Slack time (5)



37

37

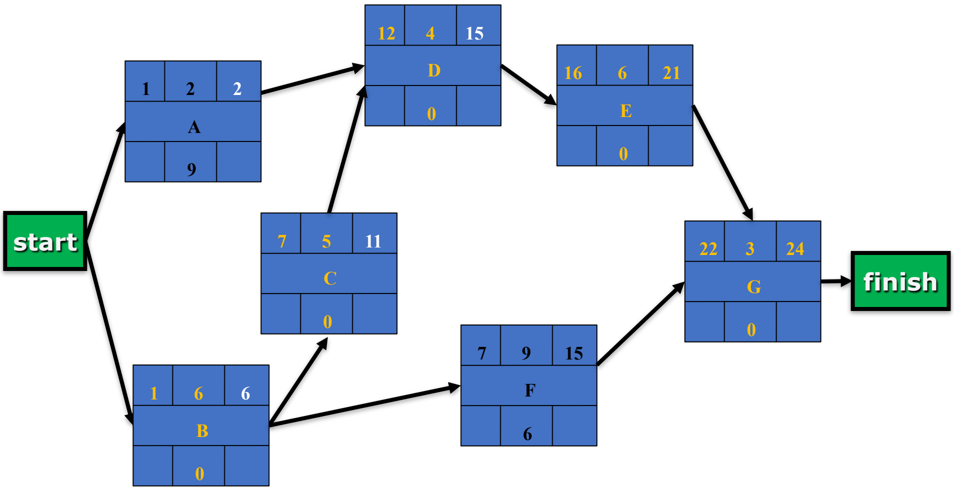
Calculate ES and EF (1)

- ES and EF are calculated by doing a **forward** pass through the diagram
- The **ES** of *activities after the start node* is **1**
- The **EF** of an activity is its **ES** *plus* **its duration** *minus* **1**
- The **ES** is the **EF** of the predecessor activity *plus* **1**
- If there are multiple predecessor activities, use the *greatest EF*

38

38

Calculate ES and EF (2)



Calculating LS and LF (1)

- Late start is the latest time that an activity can start
  - If an activity is on a path that's much **shorter** than the critical path, then it can start very **late without delaying** the project
- Late finish is the latest time that an activity can finish
  - If an activity is on a **shorter** path than the critical path and all of the other activities on that path start and finish early, then it can finish very **late without delaying** the project

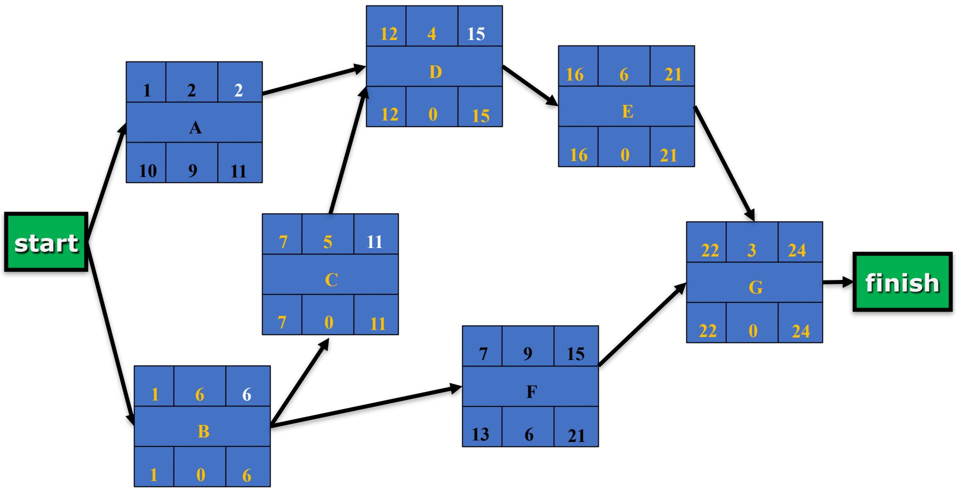
Calculating LS and LF (2)

- LS and LF are calculated by doing a backward pass through the diagram
- Start with the longest path and work your way from the end node to the start node
  - Do the same thing for the next longest path, and so on
  - Don't recalculate the LS or LF for an activity that's already been calculated on a prior backward pass.

Calculating LS and LF (3)

- The **LS** and **LF** of the **last activity** in the **critical path** will be *the same as its ES and EF*
- The **LF** of **non-critical activities** with the end node as their successor will be the **LF** of the last critical path activity
- The **LF** of an activity is the **LS** of its successor *minus* **1**
  - If there are multiple successor activities, use the *least LS*
- The **LS** is the **LF** of the activity *minus* its **duration** *plus* **1**

Calculating LS and LF (4)



Discussion

- What are the critical activities?
- How long will it take to complete this project?
- Can activity D be delayed without delaying the entire project? If so, how many weeks?
- Can activity F be delayed without delaying the entire project? If so, how many weeks
- What is the schedule for activity C?

7. Project Management

(end of lecture)

ONE LOVE. ONE FUTURE.

45