CE29x Team-Project Challenge

Critical Path Analysis

Professor Anthony Vickers

Room: 1NW.3.17 e-mail: vicka

with acknowledgements to Keith Primrose, Iain Langdon, Michael Fairbank

Review of Previous Lecture

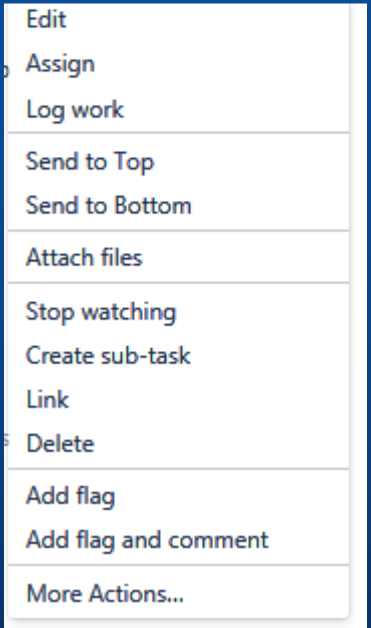
Looked at Estimating tasks and projects:

- Reviewed Agile vs Waterfall
  - And Scrum vs Kanban
- Work Breakdown Structure
- Methods of Estimating tasks and projects
  - Costing projects
- Story points and average velocity
- Effort, Duration and Resources

Jira Tip

It’s possible to subscribe/unsubscribe to issues to receive automatic emails when comments are added.

Just use the issue’s context menu and select “start watching” or “stop watching”



Critical path analysis

Today's lecture

- ✳ Dependencies between tasks
- ✳ Activity Network Diagram
- ✳ Scheduling: Critical Path Method (CPM)
- ✳ Total duration compression techniques
- ✳ Gantt charts in Microsoft Project

Critical path analysis

Some tasks can’t be done until others are done first

⇒Task Dependencies

Are some tasks more important than others to be done as soon as possible, for the project to be finished quickly?

Q: What tasks are required to make a cup of tea?

Q: Which task dependencies are there?

Q: Which of those tasks are critical to start as soon as possible?

Critical path analysis

This leads to the idea of certain activities being "critical ones"

- To start as soon as possible
- If those tasks are late then the whole project will be late.

And certain tasks not being critical:

- they have some "slack" in when they can start

Critical path analysis

- Critical path analysis is part of "Program evaluation and review technique", (PERT)
  - developed by the US Navy in the 1950s

Ordering tasks

- Calculating precedence table
- Generating an activity network

Dependency Table / Precedence Table

- “Tasks” are also known as “activities”
- A “precedence table” (Dependency table) below shows the tasks (activities) involved in a project with their durations, and immediate predecessors

Task (Activity)	Duration	Immediate Predecessors
A	2	-
B	4	A
C	4	-
D	5	B, C
E	6	C
F	3	E

Example Task List: moving home

- Imagine the main tasks required
- Then detail them

Task	Duration	Immediate Predecessors
A. Pack boxes		
B. Purchase boxes		
C. Hire removal company		
D. Load the van		
E. Wrap glasses and other delicates in newspaper		
F. Cancel utility companies		
G. Drive van to new house		
H. Unload van		
I. Redirect mail		

Example Task List: moving home

- Imagine the main tasks required
- Then detail them

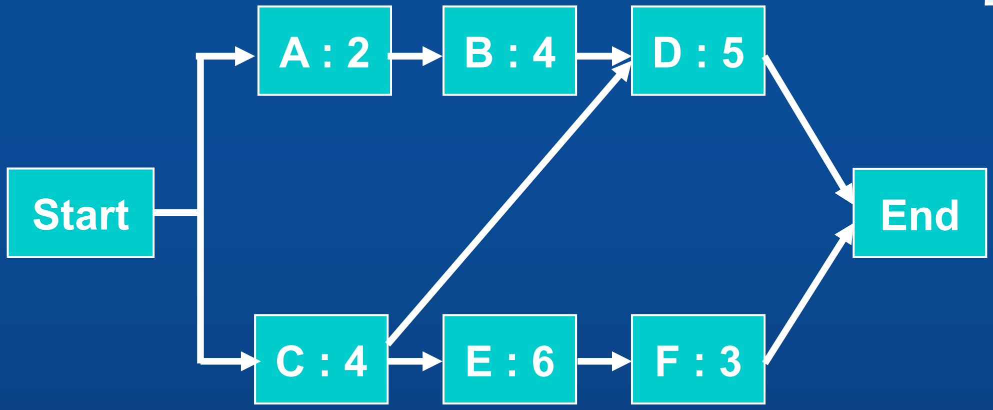
Task	Duration	Immediate Predecessors
A. Pack boxes		B,E
B. Purchase boxes		-
C. Hire removal company		-
D. Load the van		A, C
E. Wrap glasses and other delicates in newspaper		-
F. Cancel utility companies		-
G. Drive van to new house		D
H. Unload van		G
I. Redirect mail		-

Activity network

Task	Duration	Immediate Predecessors
A	2	-
B	4	A
C	4	-
D	5	B, C
E	6	C
F	3	E

We can display the precedence table as an "activity network":

- Nodes represent the activities (tasks)
- Connections show the dependencies



- This shows activity B cannot start until A has finished
- It also shows D cannot start until all arrows coming into it indicate completion (i.e. until B and C have finished).
- Note, no need to say D depends on A! This occurs automatically through dependency chaining.

Activity network

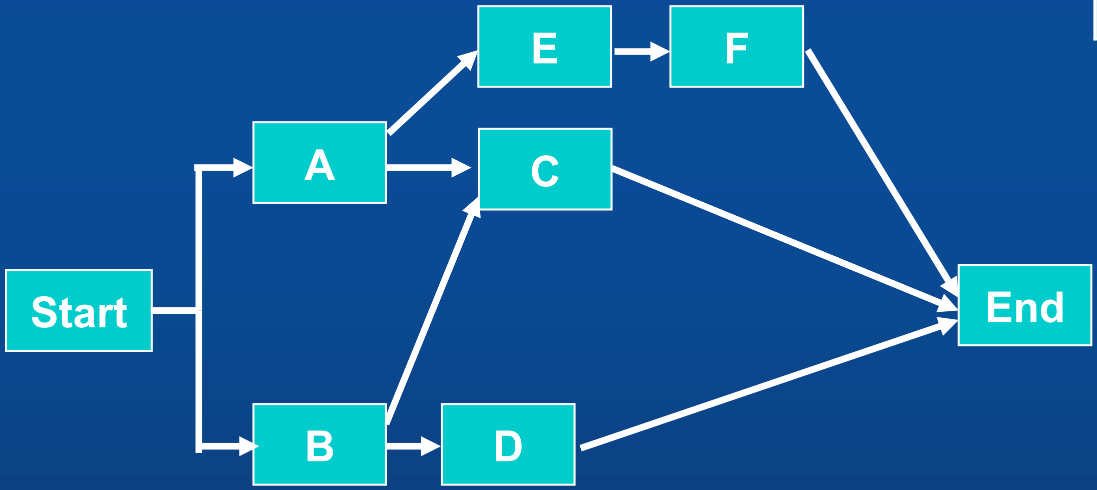
Task	Duration	Immediate Predecessors
A	2	-
B	4	-
C	4	A, B
D	5	B
E	6	A
F	3	E

1. Try and represent the precedence table above as an activity network:

Activity network

Task	Duration	Immediate Predecessors
A	2	-
B	4	-
C	4	A, B
D	5	B
E	6	A
F	3	E

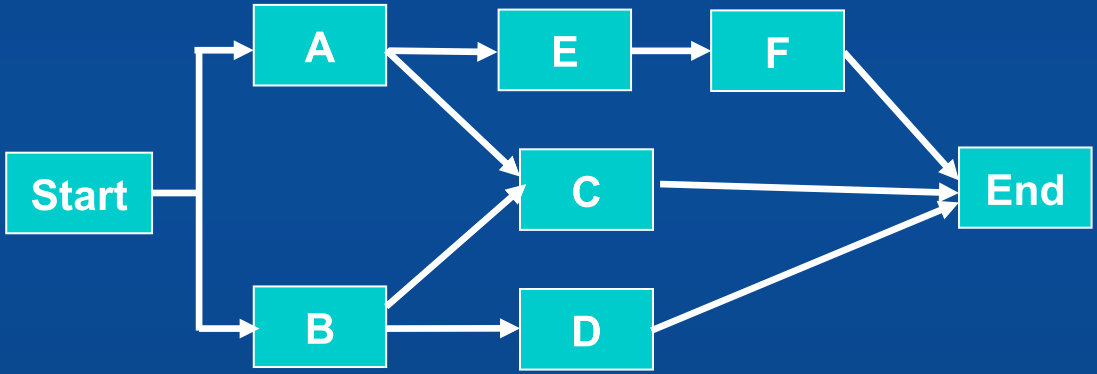
1. Try and represent the precedence table above as an activity network:



Activity network

Ta sk	Dura tion	Immediate Predecessors
A	2	-
B	4	-
C	4	A, B
D	5	B
E	6	A
F	3	E

1. Try and represent the precedence table above as an activity network:



Just tidying it up a bit.

- Try to avoid crossing arrows
- Try to make arrows all point towards the right (so time flows right)

Activity network

2. Try and represent the precedence table above as an activity network:

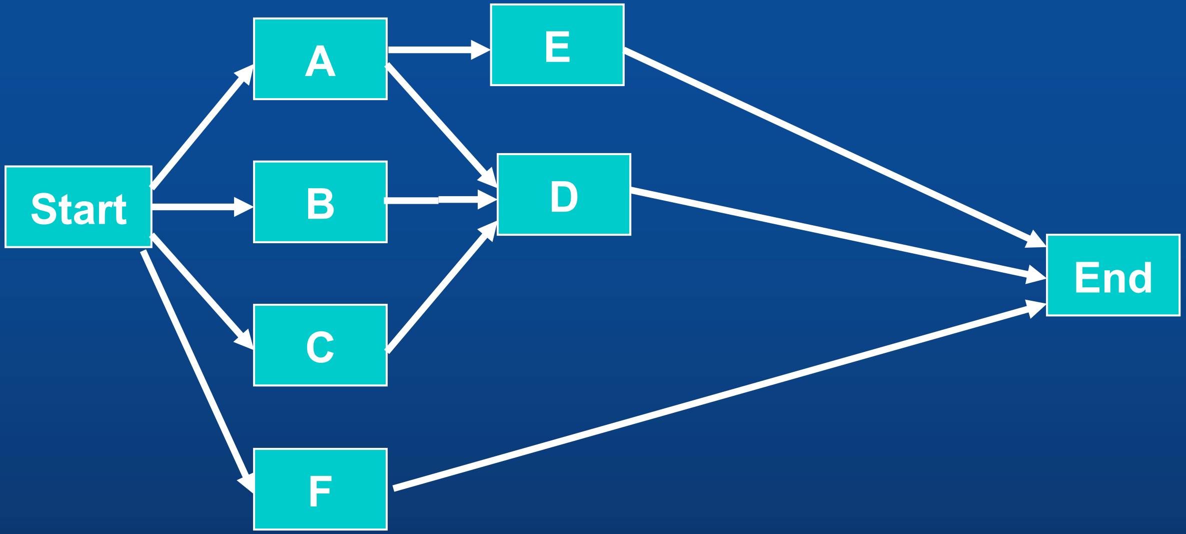
Ta sk	Dura tion	Immediate Predecessors
A	2	-
B	4	-
C	4	-
D	5	A,B,C
E	6	A
F	3	-

Start

Activity network

Task	Duration	Immediate Predecessors
A	2	-
B	4	-
C	4	-
D	5	A,B,C
E	6	A
F	3	-

2. Try and represent the precedence table above as an activity network:

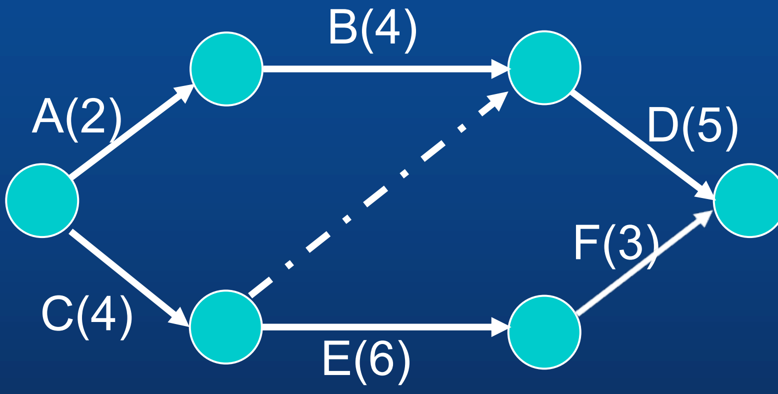


Activity network – alternative display format

Or can use Arrow diagramming method:

- Now arrows represent the activities
- Connections between nodes show dependencies
- Nodes represent “milestones”. These are events which indicate when we are ready to move on.
- Milestone nodes “fire” (i.e. Milestones are completed) when all arrows coming into that node indicate completion of all preceding activities.

Task	Duration	Immediate Predecessors
A	2	-
B	4	A
C	4	-
D	5	B, C
E	6	C
F	3	E



The Activity Network Diagram

- The Activity Network Diagram is the end-product of the task decomposition process
  - It should be accompanied by narrative to explain dependencies

What you have done up to now

- Listed, from lesser to greater detail, the tasks that need to be achieved to cover the complete scope of the project
- Defined dependencies between tasks
- Built a network of tasks

To finish the plan you need to:

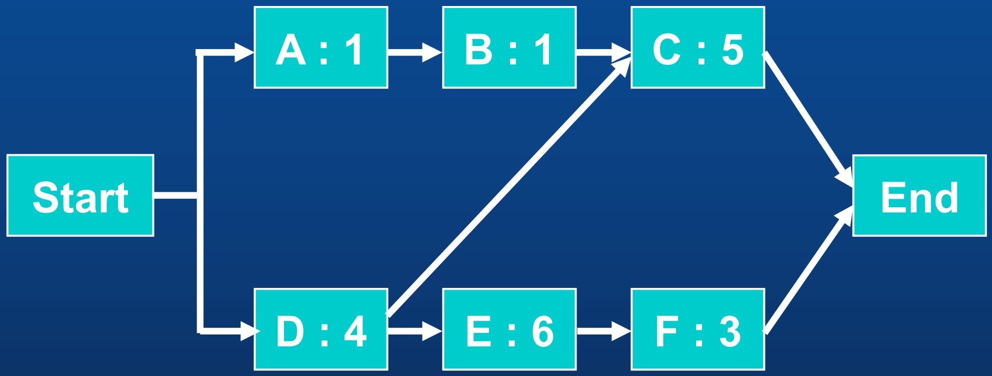
- Estimate the duration of the tasks (previous lecture)
- Plan your project
    - Critical path analysis technique
- Review your plan
    - Does it fit it into a specific global elapsed time (if one exists)?

Analysing the Activity Network

1. Forward path calculation…
2. Backward path calculation….
3. Slack calculation….
4. Critical activities….
5. Critical path.

Critical Path Method

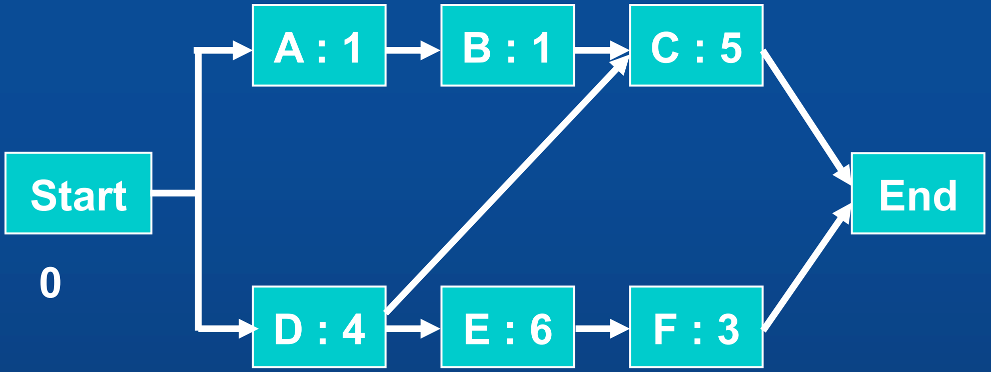
- A path is a sequence of activities that starts with a beginning activity and travels through a series of immediate successors until it terminates with an ending activity
- The length of a path is the total time it takes to go through the complete path
- For any network, there is a maximum of the lengths of all the paths through the network
- Any path whose length is equal to this maximum is called a critical path



The Forward Path

- Calculate the Earliest Start Time (EST) and the Earliest Finish Time (EFT)
- These are "earliest" times in the sense that it is not possible to start/end these tasks any early than this.

Activity	Duration
A	1
B	1
C	5
D	4
E	6
F	3

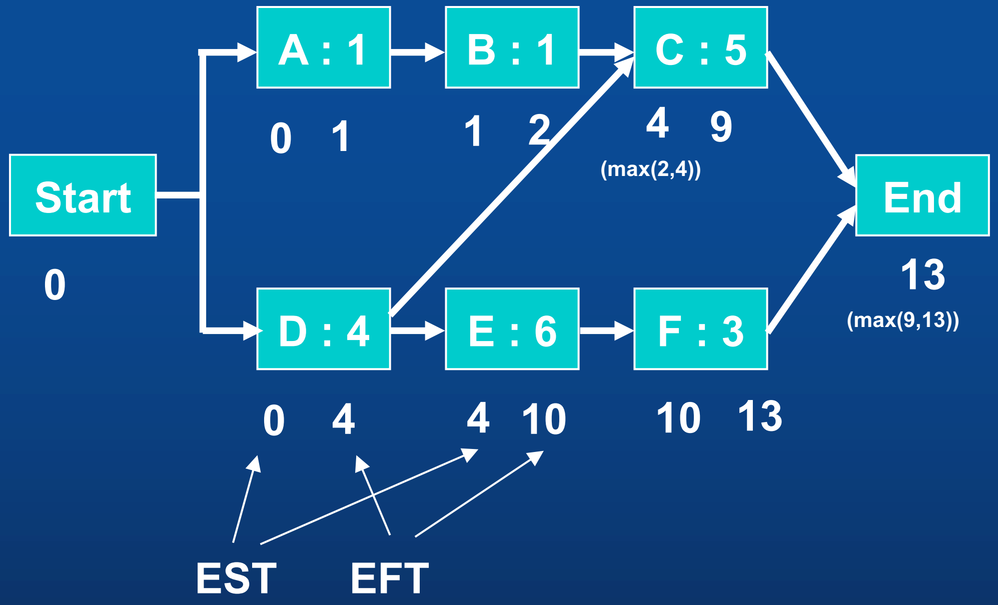


Complete the forward-path calculation

The Forward Path

- Calculate the Earliest Start Time (EST) and the Earliest Finish Time (EFT)
- These are "earliest" times in the sense that it is not possible to start/end these tasks any early than this.

Activity	Duration
A	1
B	1
C	5
D	4
E	6
F	3



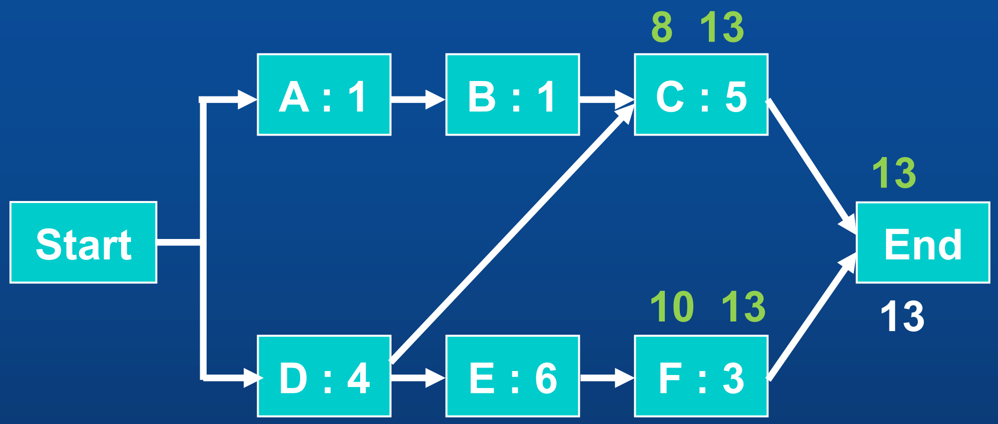
The Forward Path Algorithm

1. The EST for the beginning activity is 0
2. For any task, EFT=EST+duration
3. For any task, the EST is the maximum of the EFT for all its immediate predecessors

- This requires a “forward pass” through all of the network’s nodes

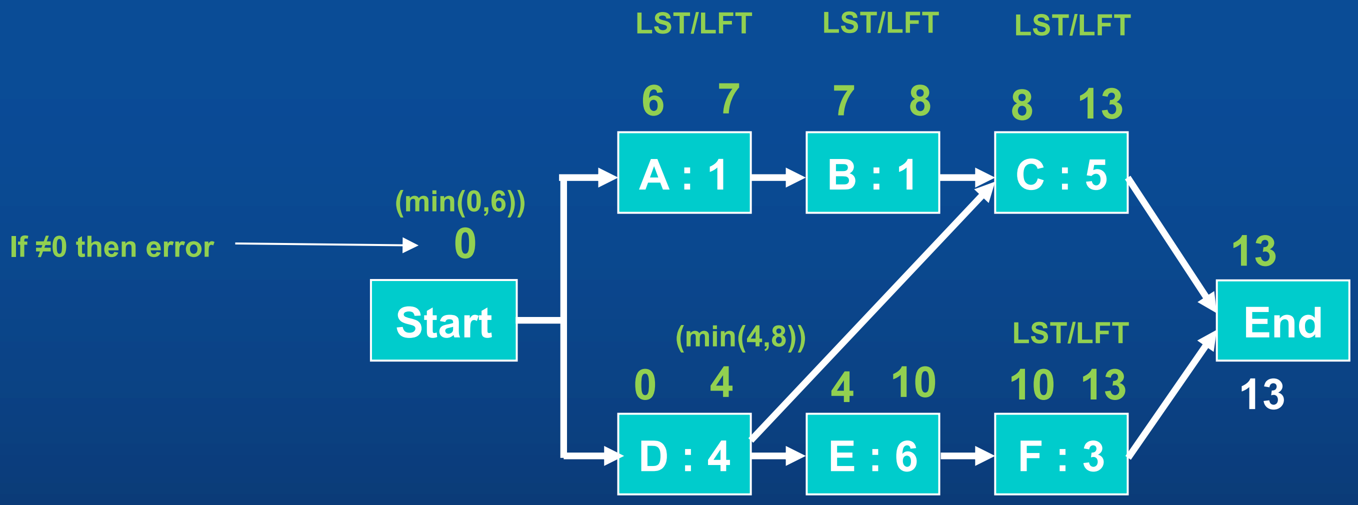
The Backward Path

- Calculate Latest Start Time (LST) and Latest Finish Time (LFT)
- "Latest" here means latest possible *assuming we want the project completed in optimal time,* which we know from the previous slide is 13 in this example.
- Complete the diagram:



The Backward Path

- Calculate Latest Start Time (LST) and Latest Finish Time (LFT)
- "Latest" here means latest possible *assuming we want the project completed in optimal time,* which we know from the previous slide is 13 in this example.
- Complete the diagram:



The Backward Path Algorithm

1. For the “End” node, the LFT = EFT (calculated from the forward pass)
2. For each task, LST = LFT – duration
3. For each task, the LFT is the minimum of the LST for all of its immediate successors

- This requires a backward pass through the network

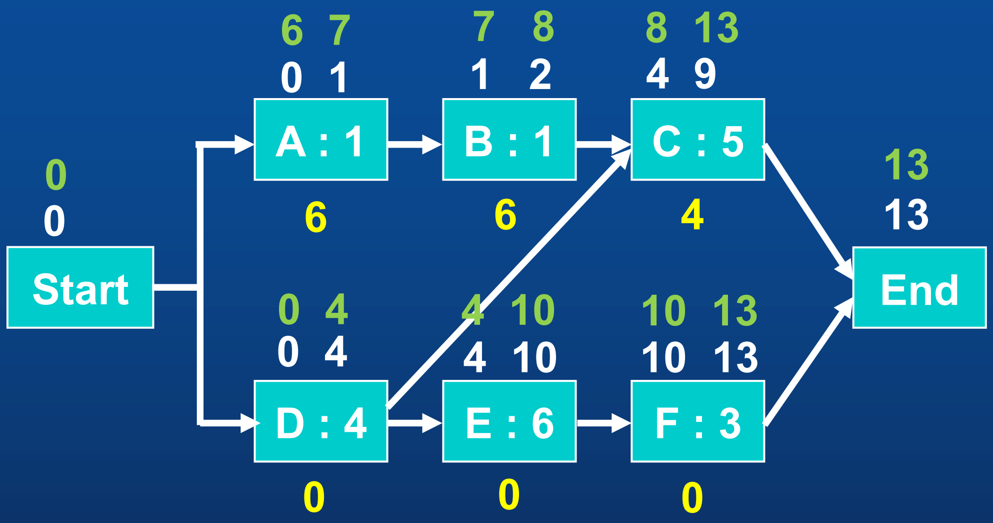
Calculating Slack

- Slack is the amount of time that an activity can be delayed past its EST or EFT without delaying the project.
- For each activity, the Slack is:

Latest Finish Time – Earliest Finish Time (LFT-EFT) or

Latest Start Time – Earliest Start Time (LST-EST)

- E.g. below, C has a slack of 13-9=4

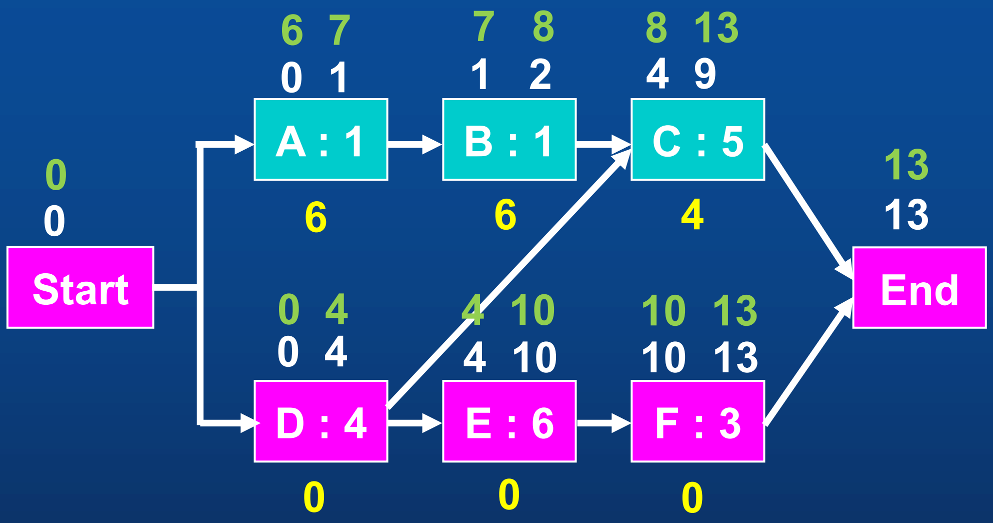


Key:

LST	LFT
EST	EFT
Slack	

Critical Activities and Critical Path

- Activities with zero slack are called critical activities
- If these are late then the whole project will be late
- The critical path here is Start-D-E-F-End
- The project manager must pay most attention to those activities on the critical path



0
0
Start

6 7
0 1
A : 1
6

7 8
1 2
B : 1
6

8 13
4 9
C : 5
4

0 4
0 4
D : 4
0

4 10
4 10
E : 6
0

10 13
10 13
F : 3
0

13
13
End

Key:

LST	LFT
EST	EFT
Slack	

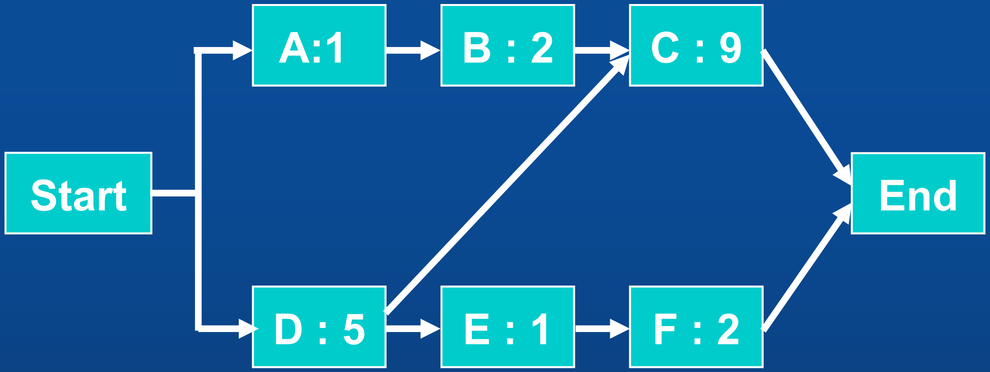
Why Critical Path is important

- Slippage on a critical path results in slippage on the project completion date
- Improvement of the total project duration is mainly made by improvements along a critical path
- Critical activities need special attention, and their responsibilities and resources need to be carefully managed
- Activities on the critical path may not previously have been thought of as the most "important" (relevant, difficult…) activities of a project

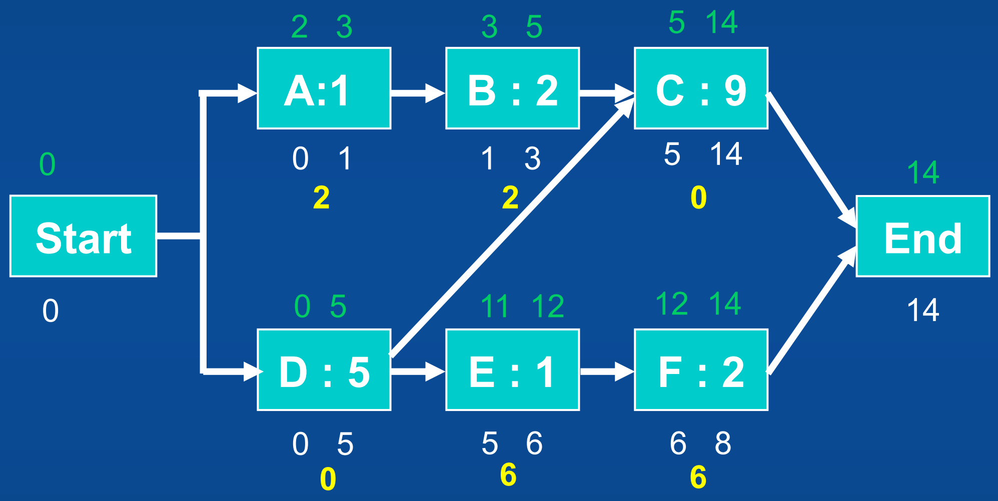
Q: What is the minimum number of critical paths?

Critical Path Calculation Exercise

1. Do a forward pass and calculate the EST and EFTs
2. Do a backward pass and calculate the LST and LFTs
3. Calculate the slack of each activity
4. Identify the critical activities and critical path



Critical Path Calculation Exercise



Critical path is "Start-D-C-End"

Summary of Lecture

- Now you should be able to:
    - Generate a precedence table for tasks
    - Draw an activity network
    - Do a forward path and backward path calculation to find the LST/LFT/EST/EFT times
    - Calculate slack and critical activities

- Further practice on today’s lecture

Try the first quiz on Critical Path Analysis