Software Project Management

María Gómez

Slides by: María Gómez, Sudipta Chattopadhyay, Rahul Premraj, Andreas Zeller

The UK Electronic Health Records Project

- Unified medical records system for British citizens (launched 2002)
- Cost: $16 Billion
- Over budget and over schedule!

Project abandoned in 2011!!

Why Sw Projects Fail?

Caper Jones

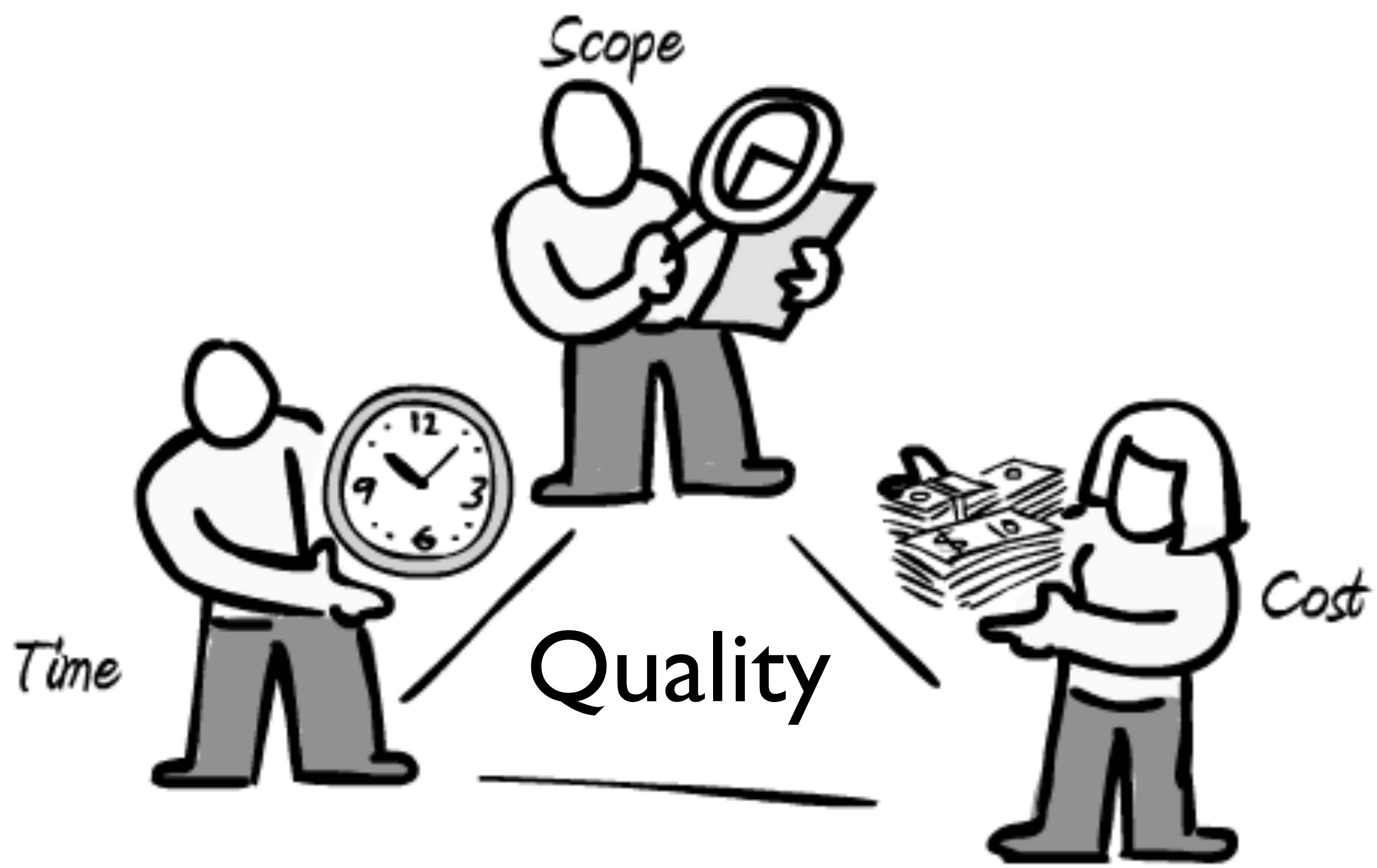
- Investigated 250 large projects.
- Unsuccessful projects showed weaknesses in:
  - Project Planning
  - Cost Estimation
  - Measurements
  - Milestone Tracking
  - Change Control
  - Quality Control

Why Sw Projects Fail?

Caper Jones

...the most interesting aspect of these six problem areas is that all are associated with **project management** rather than with technical personnel.

How do we plan a project?



Project Management

The goal of **Project Management** is
to help projects to *finish on time, within budget* and
without *cutting scope*

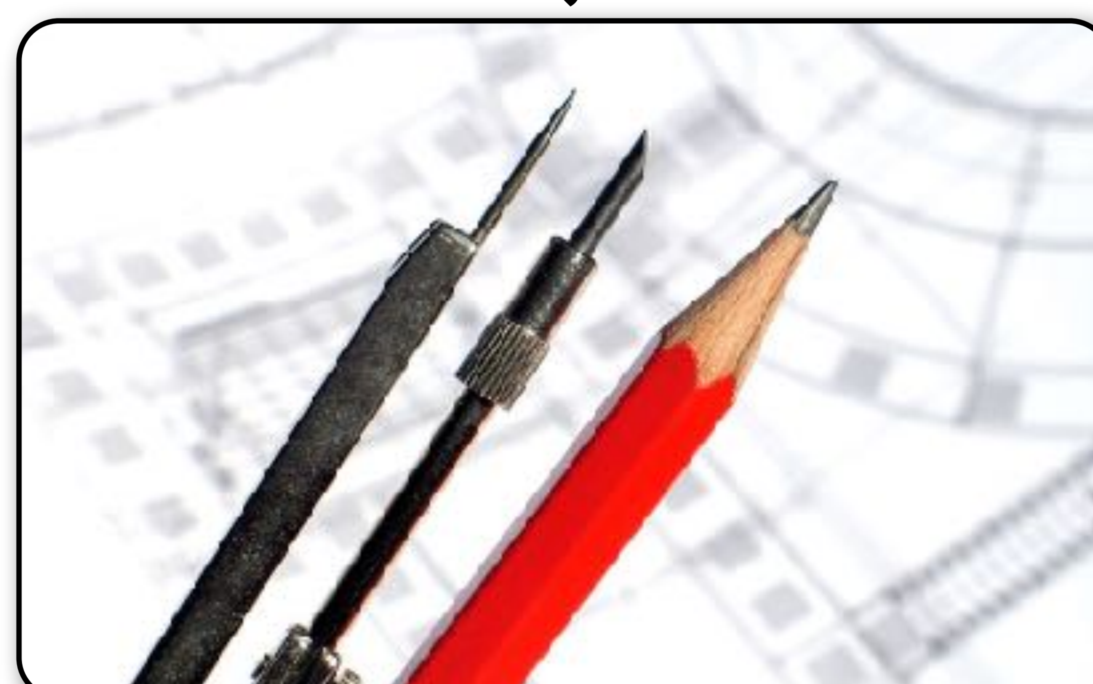
Four P's of Project Management

People

Product

Process

Project



People

"The most important ingredient that was successful on this project was having smart people... very little else matters in my opinion." *[1]*

[1] Curtis, B., et al., "A field study of the Software Design Process for Large Systems". IEEE Trans. Sw Eng

People

Communication & Coordination

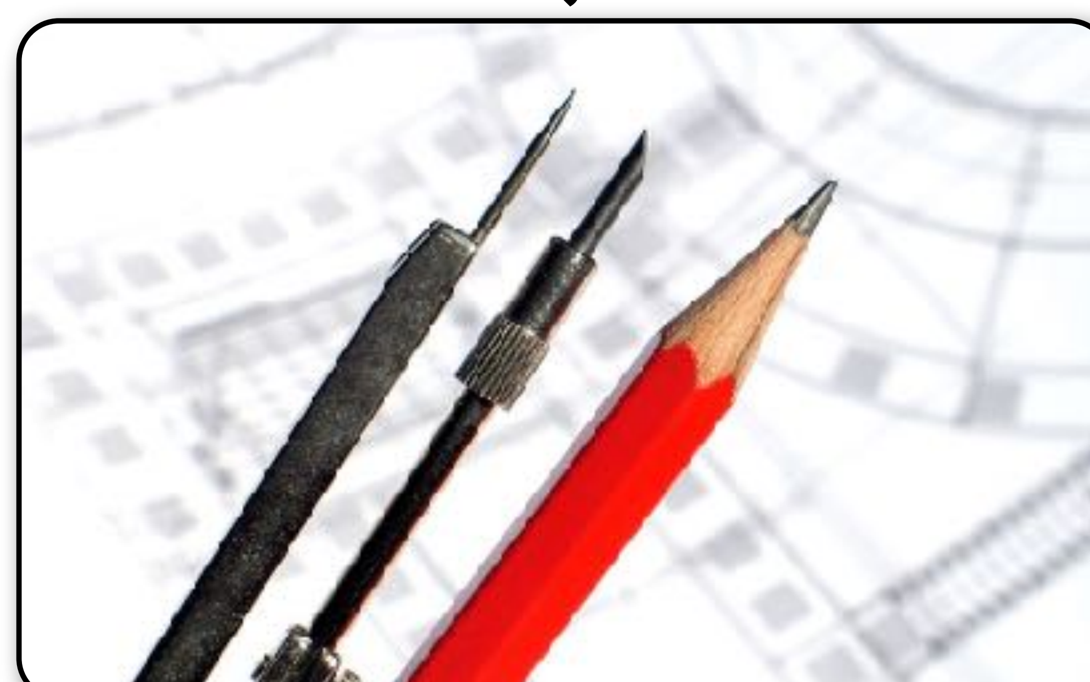
Four Ps of Project Management

People

Product

Process

Project



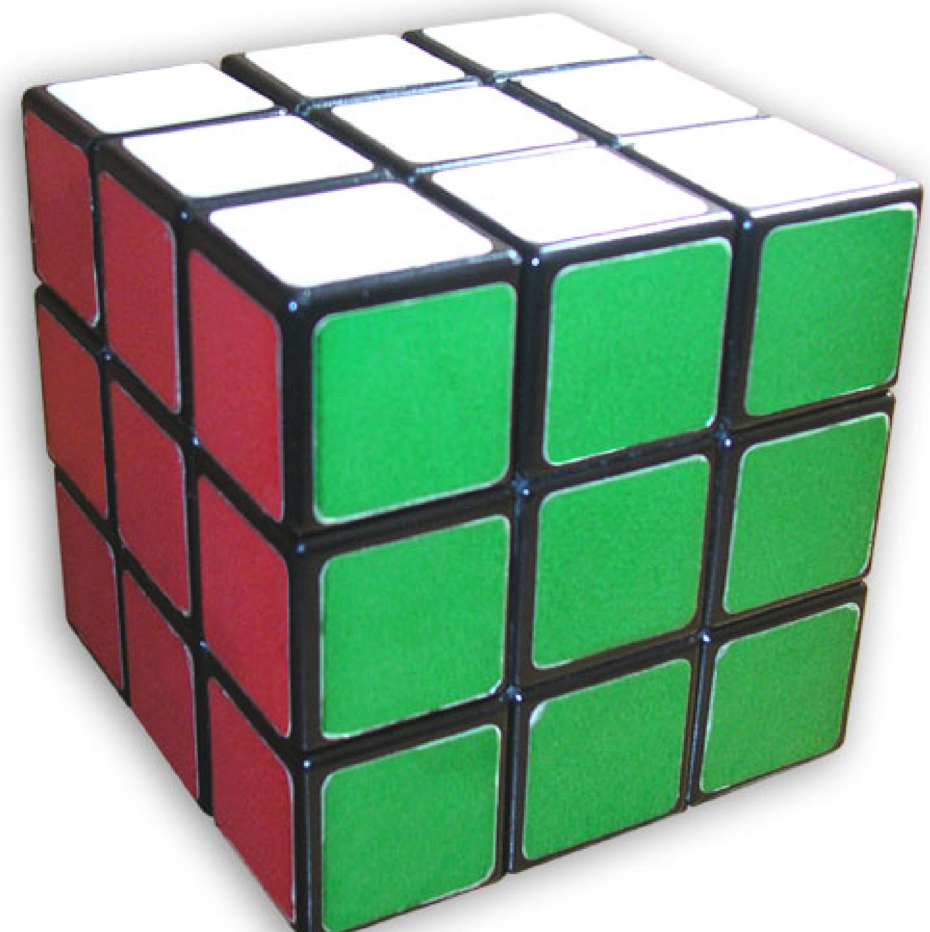
Product

Define the Scope of the Project

- Context: How does the software fit into a larger system, product, or business context, and what constraints are posed?
- Information objectives: What are the inputs and outputs of the system?
- Function and performance: What functions are to be performed to transform the inputs to outputs?

Product

Divide & Conquer



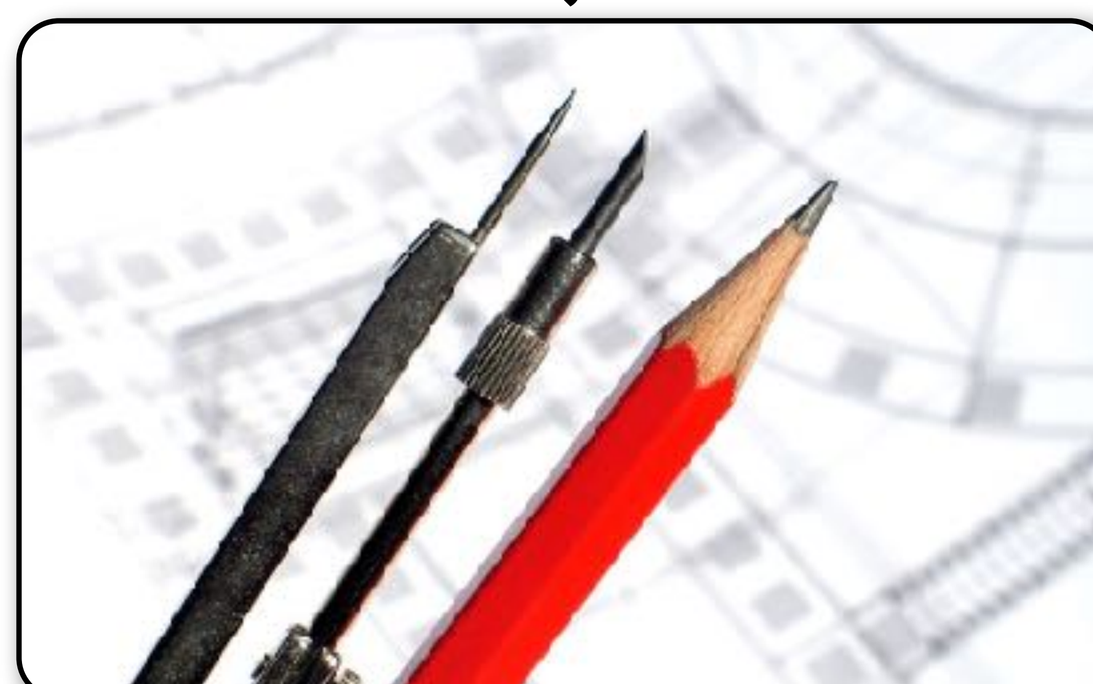
Four Ps of Project Management

People

Product

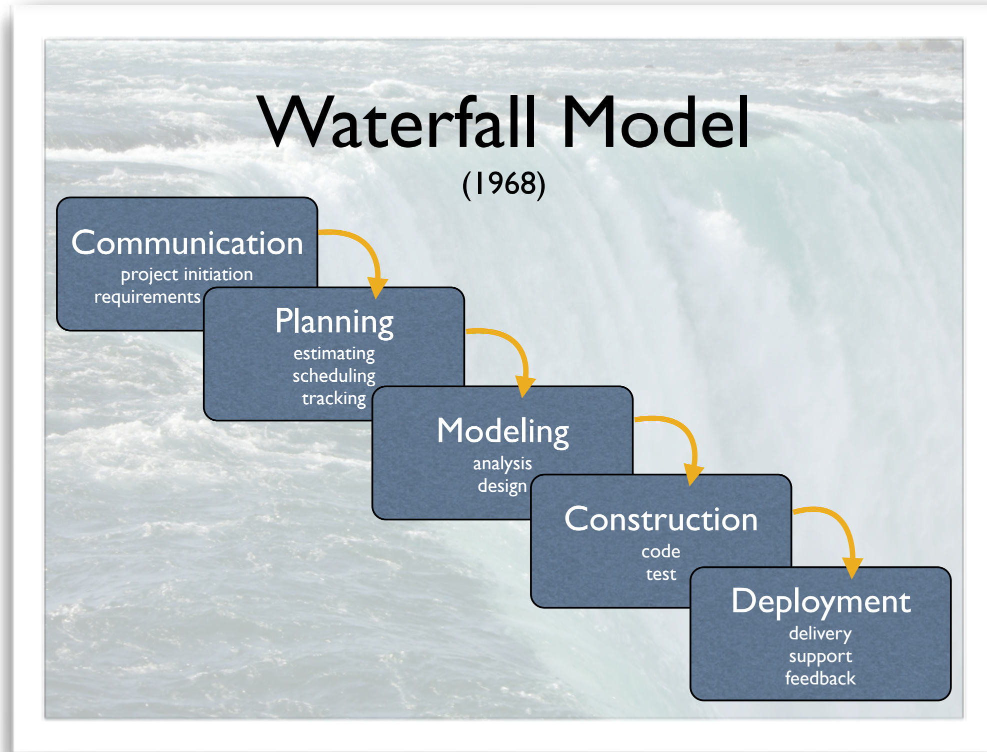
Process

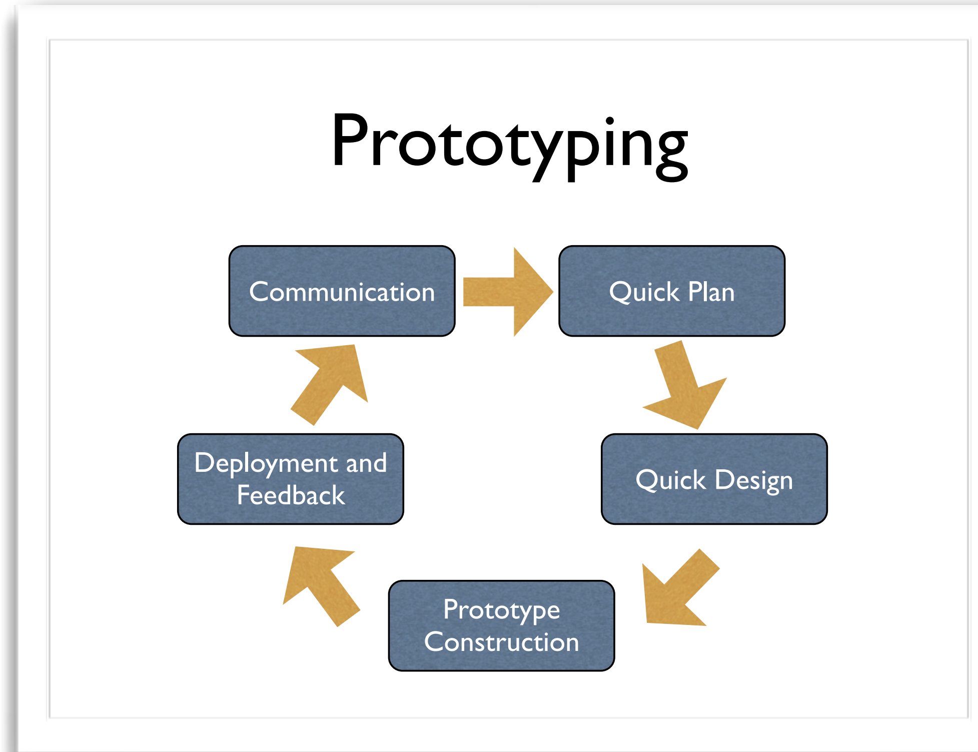
Project

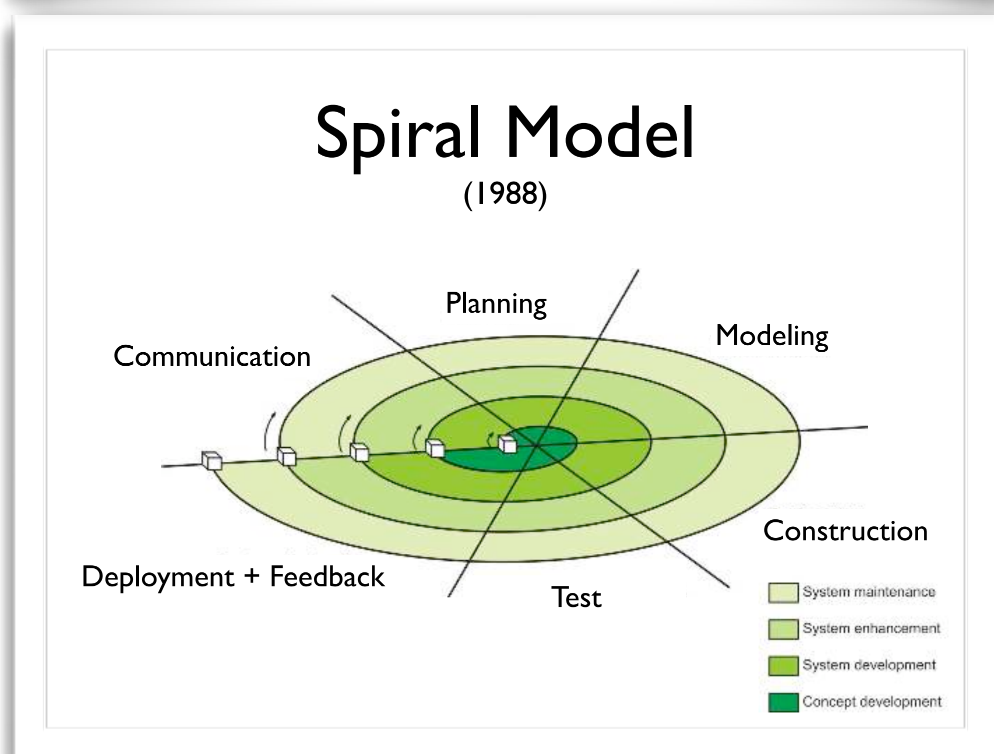


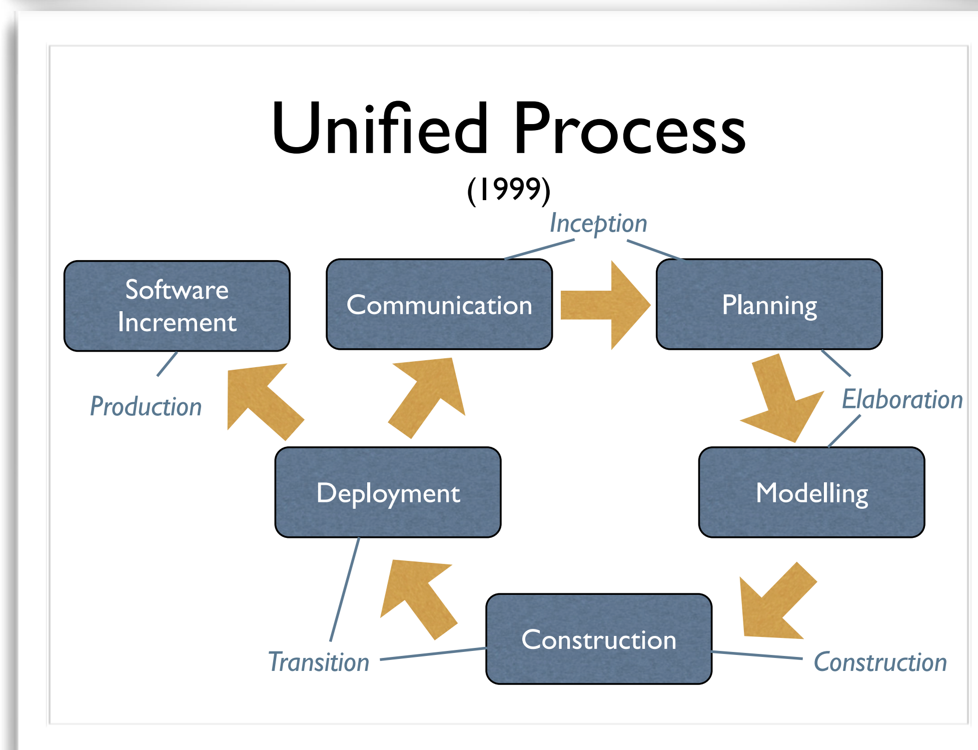
Process

Many processes to choose from!









Process

What to keep in mind while choosing the process?

- customers who requested the product and the end-users.
- the product's characteristics.
- the project environment in which the software is developed.

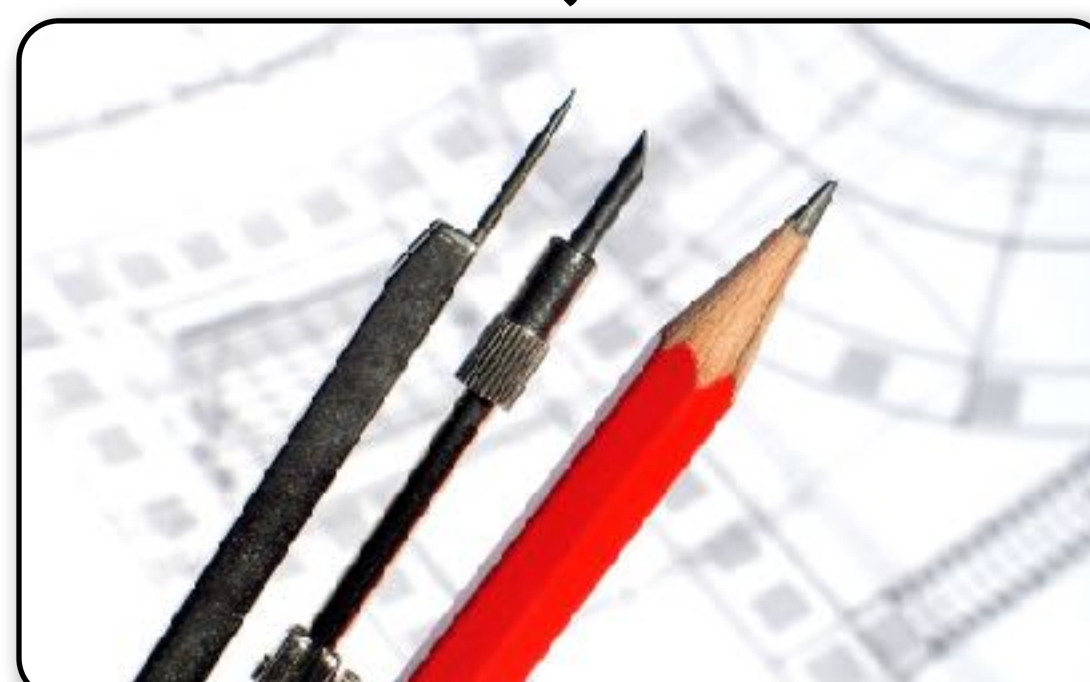
Four Ps of Project Management

People

Product

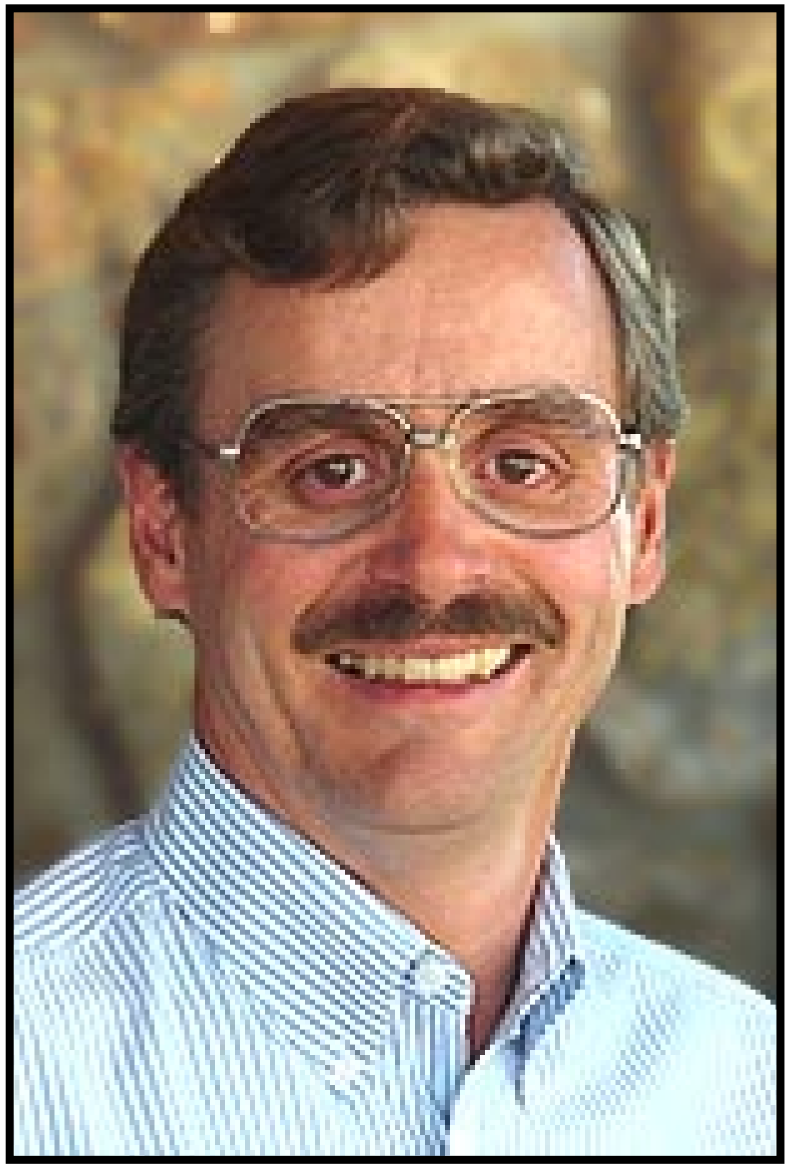
Process

Project



Project

Tom Cargill



The first 90% of the code accounts for the first 90% of the development time.

The remaining 10% of the code takes another 90% of the development time.

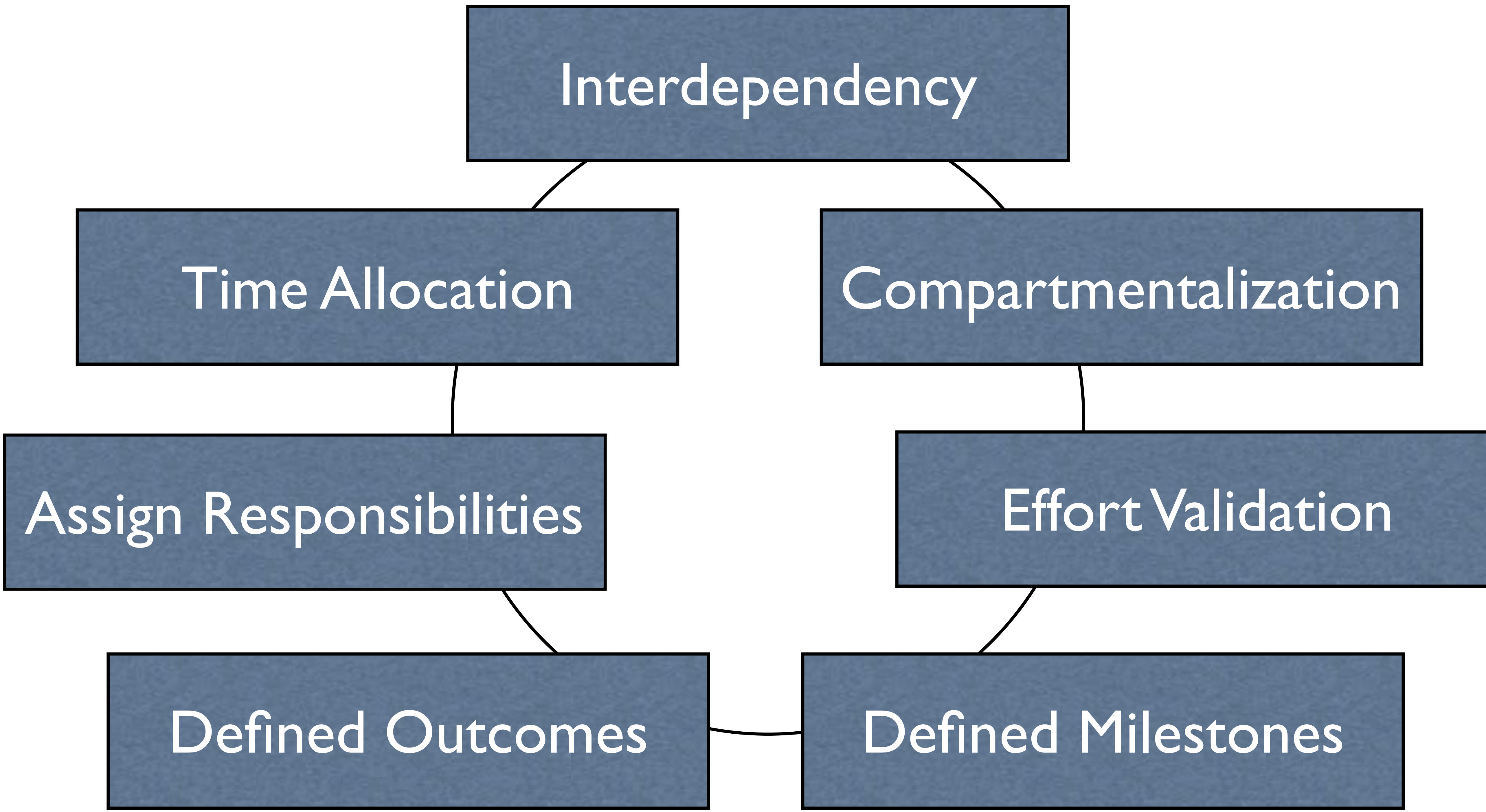
Signs of Project in Risk

- Development team doesn't understand customer's needs
- Product scope is poorly defined
- Poorly managed changes
- Chosen technology changes
- Unrealistic deadlines
- Inexperienced team
- Poor management

Project Scheduling

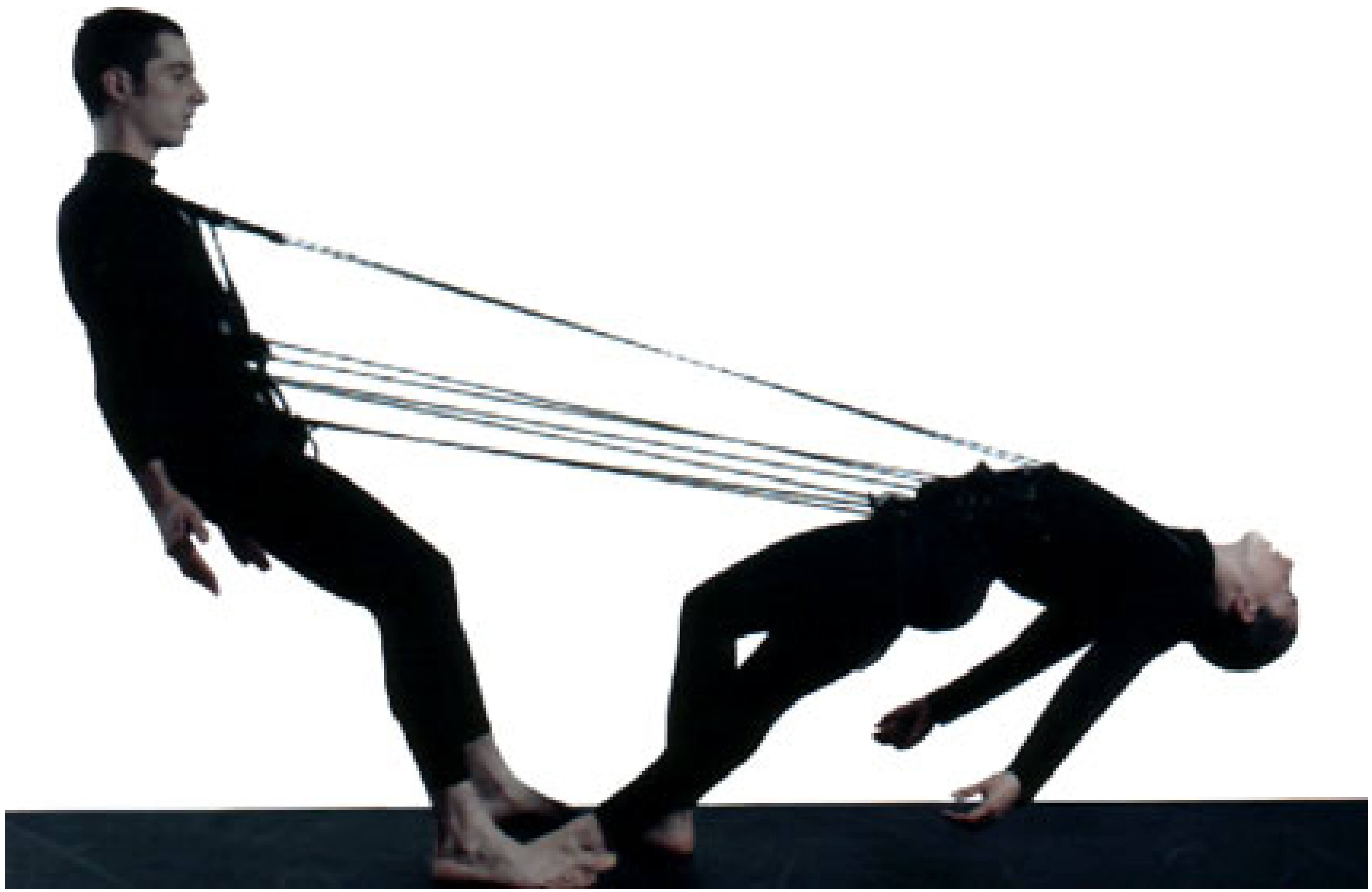
People commonly assume as will go as planned –
Each task will take as long as it ought to take.

Principles of Project Scheduling



Compartmentalization

Interdependency



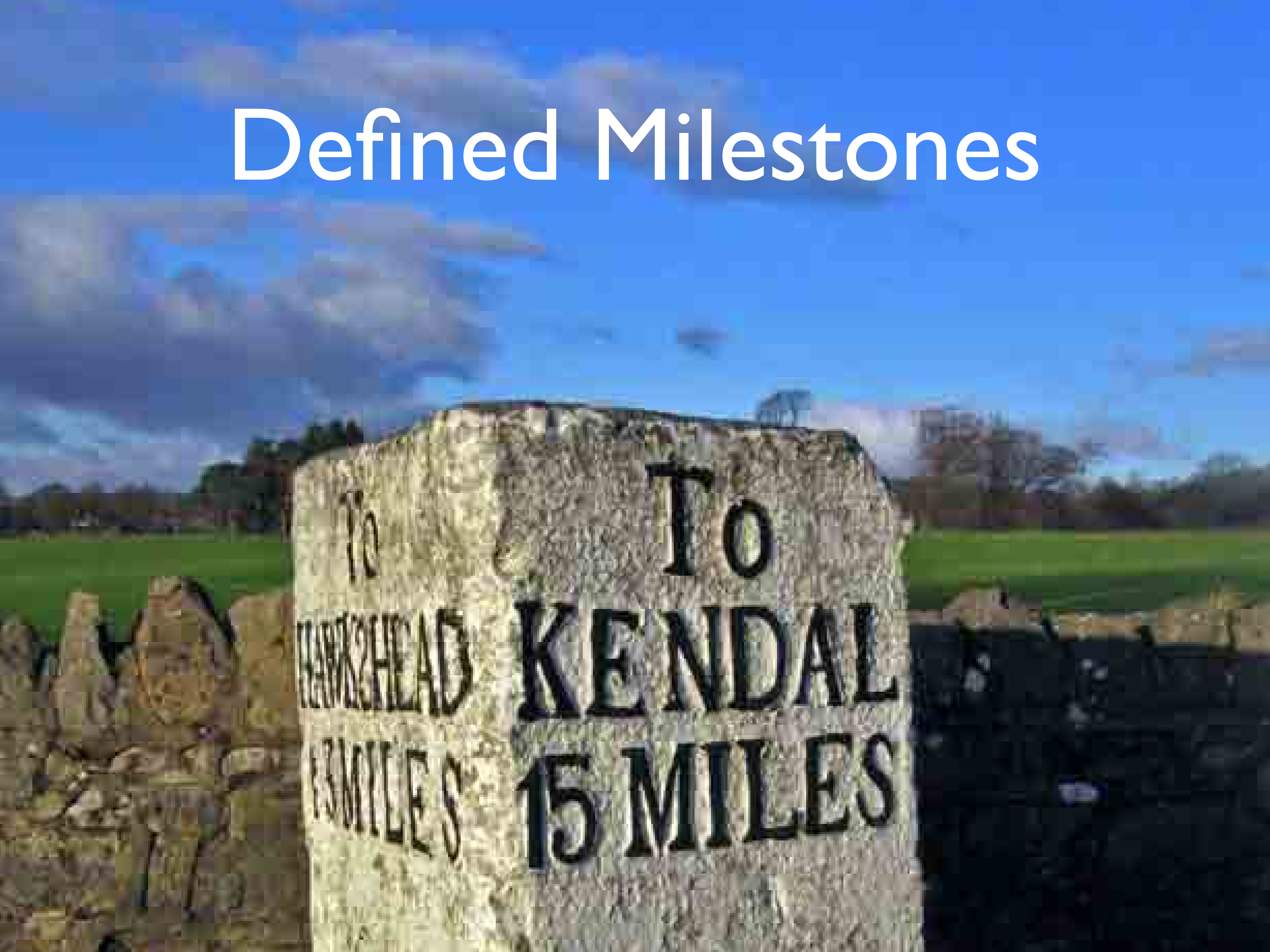
Time Allocation

Effort Validation

Assign Responsibilities

Defined Outcomes

Defined Milestones



Scheduling Tools

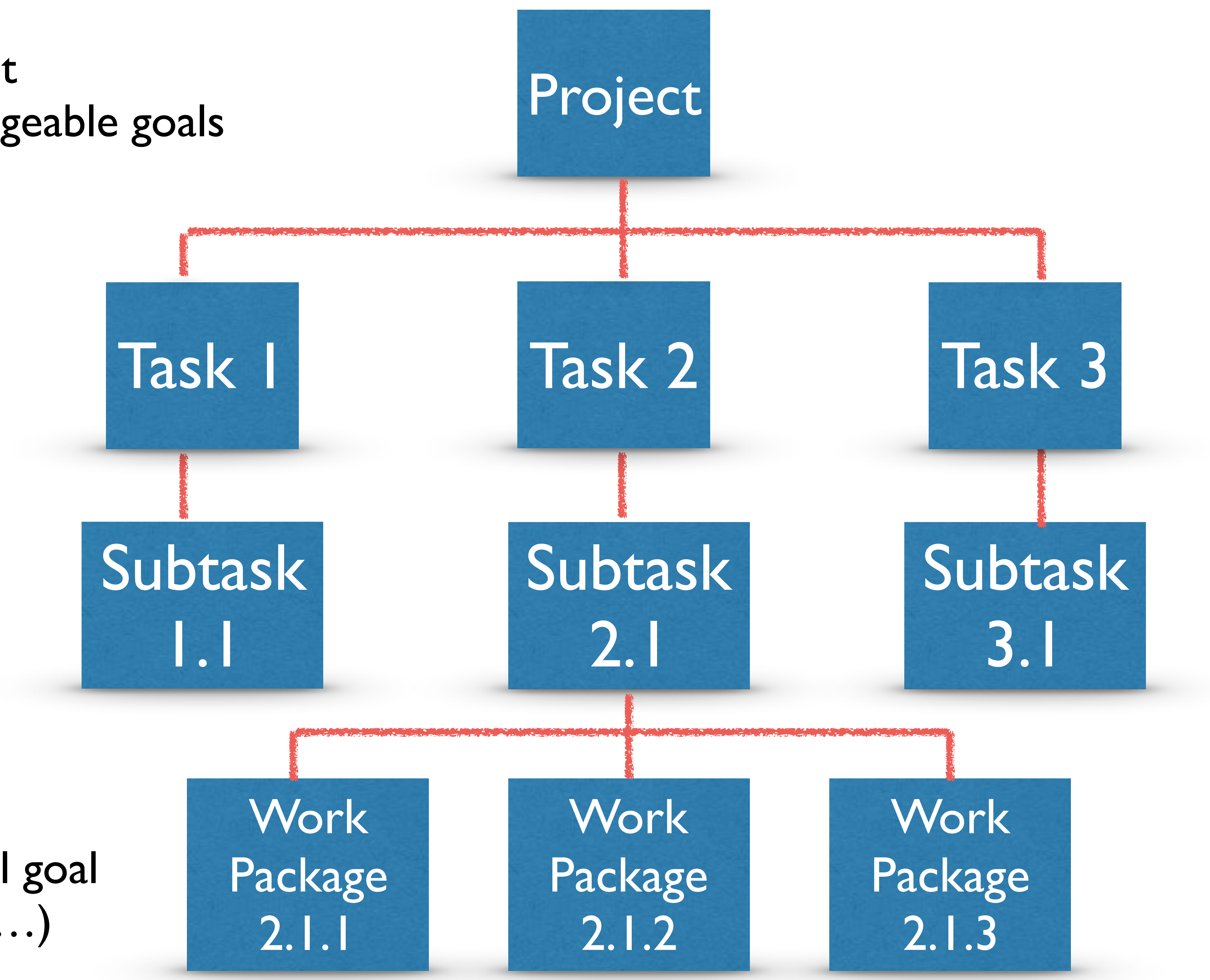
Compartmentalization

Work Breakdown Structure (WBS)

Breakdown the project
into simpler and manageable goals

Repeat process

Plan for each individual goal
(scheduling, resources…)



Work Breakdown

How to build one?

Top-down Approach

Brainstorming

Work Breakdown: Example

WBS LEVEL 1:

1. Bicycle_ 100

WBS LEVEL 2:

1. Bicycle

1.1 Frame Set_	15
1.2 Crank Set_	5
1.3 Wheels_	30
1.4 Braking System_	5
1.5 Shifting System_	5
1.6 Integration_	35
1.7 Project Mgt_	5
	100

WBS LEVEL 3:

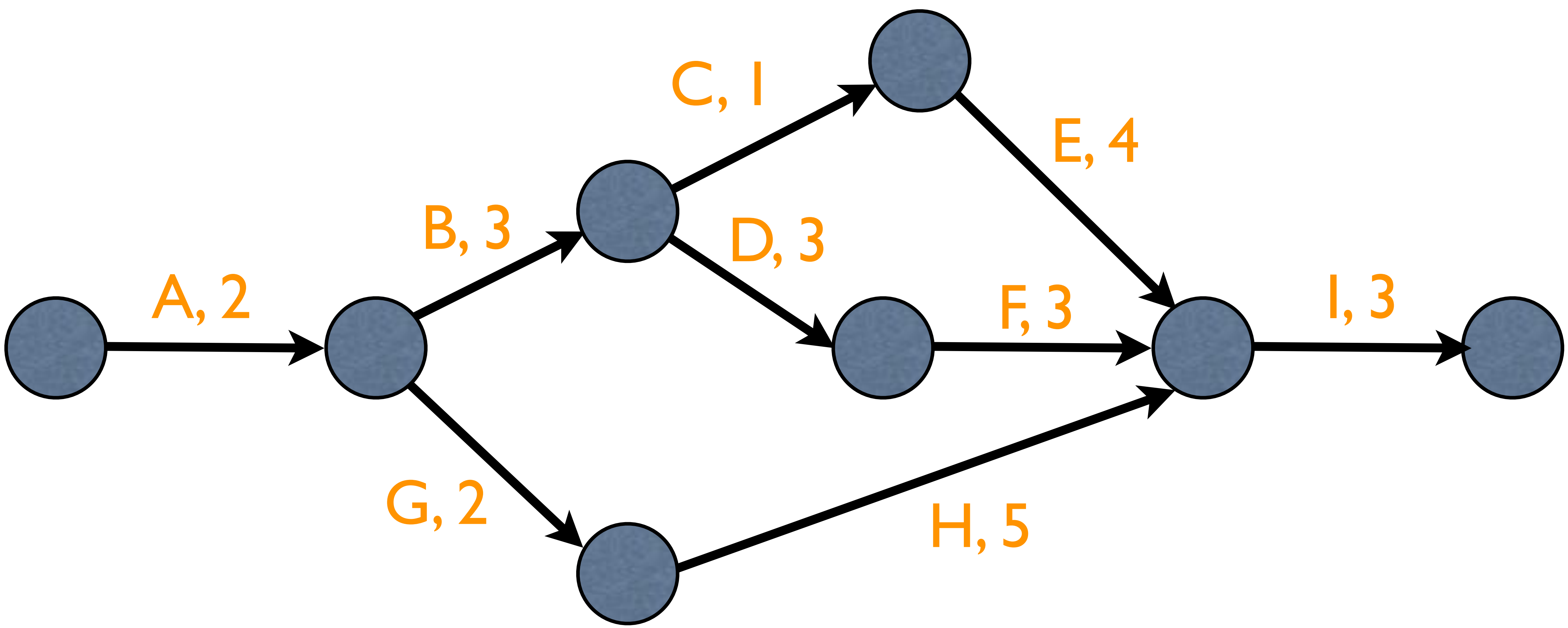
1. Bicycle

1.1 Frame Set	
1.1.1 Frame_	7
1.1.2 Handlebar_	2
1.1.3 Fork_	3
1.1.4 Seat_	3
1.2 Crank Set_	5
1.3 Wheels	
1.3.1 Front Wheel_	13
1.3.2 Rear Wheel_	17
1.4 Braking System_	5
1.5 Shifting System_	5
1.6 Integration	
1.6.1 Concept_	3
1.6.2 Design_	5
1.6.3 Assembly_	10
1.6.4 Testing_	17
1.7 Project Mgt_	5
	100

Critical Path Method (CPM)

- Method for scheduling interdependent tasks in a project

  E.g., Unit testing cannot start before development



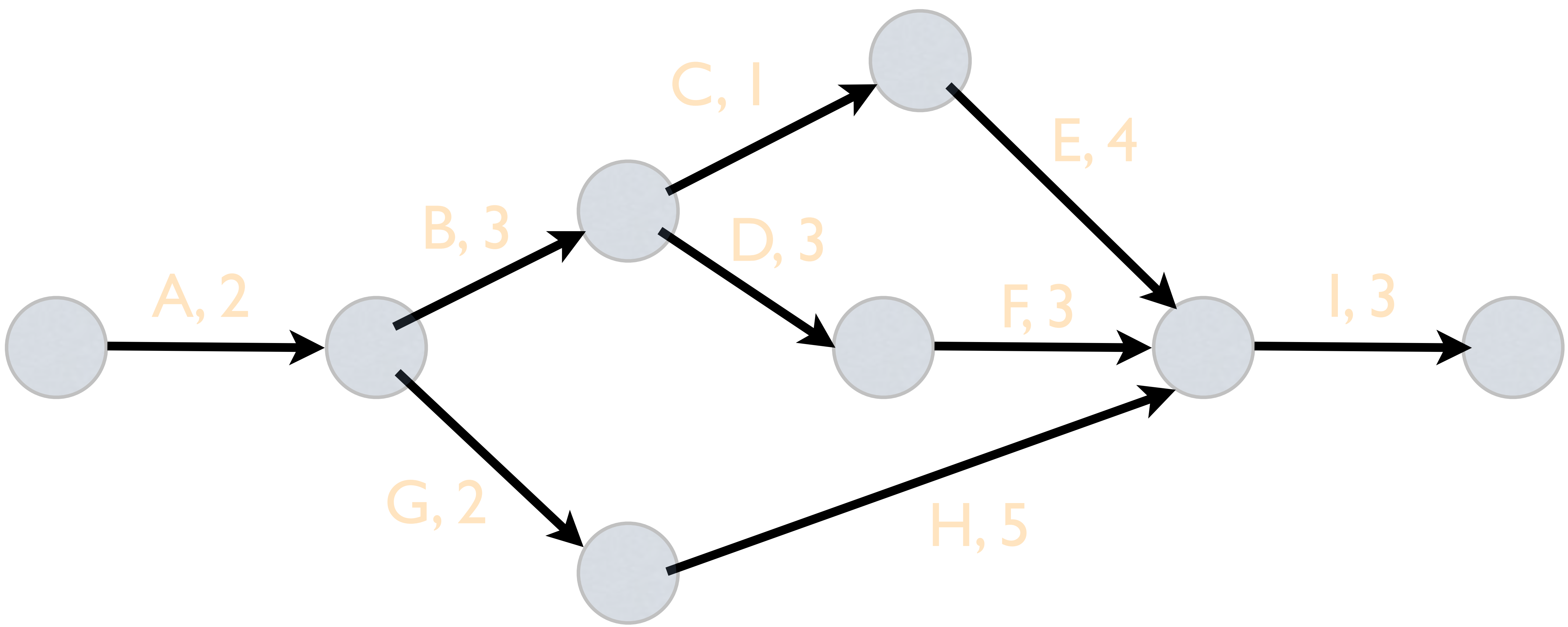
Critical Path Method (CPM)

- **CPM** identifies:
  - Required time to complete the project
  - Activities that must be completed on time to complete project on time
  - Earliest and latest dates each activity must start to keep schedule

Critical path is the sequence of activities that takes the longest time to complete

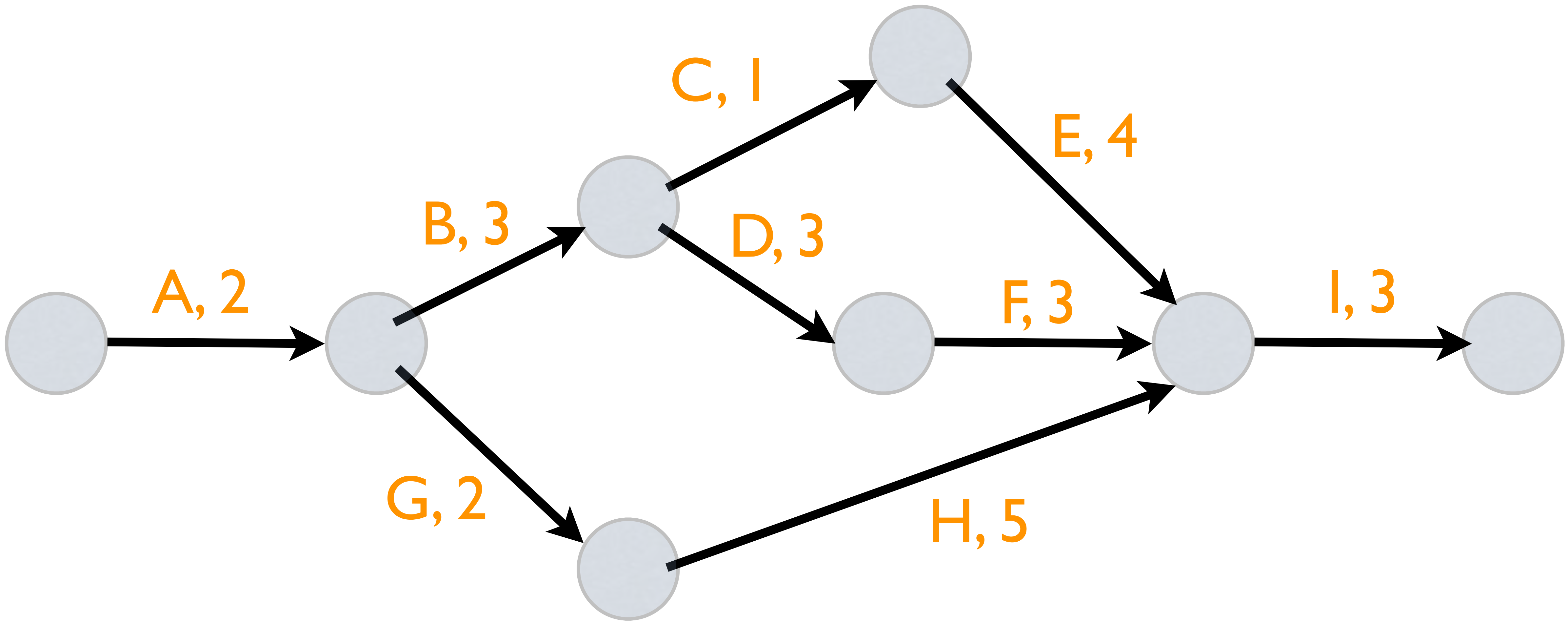
Critical Path Method

Arrows indicate tasks



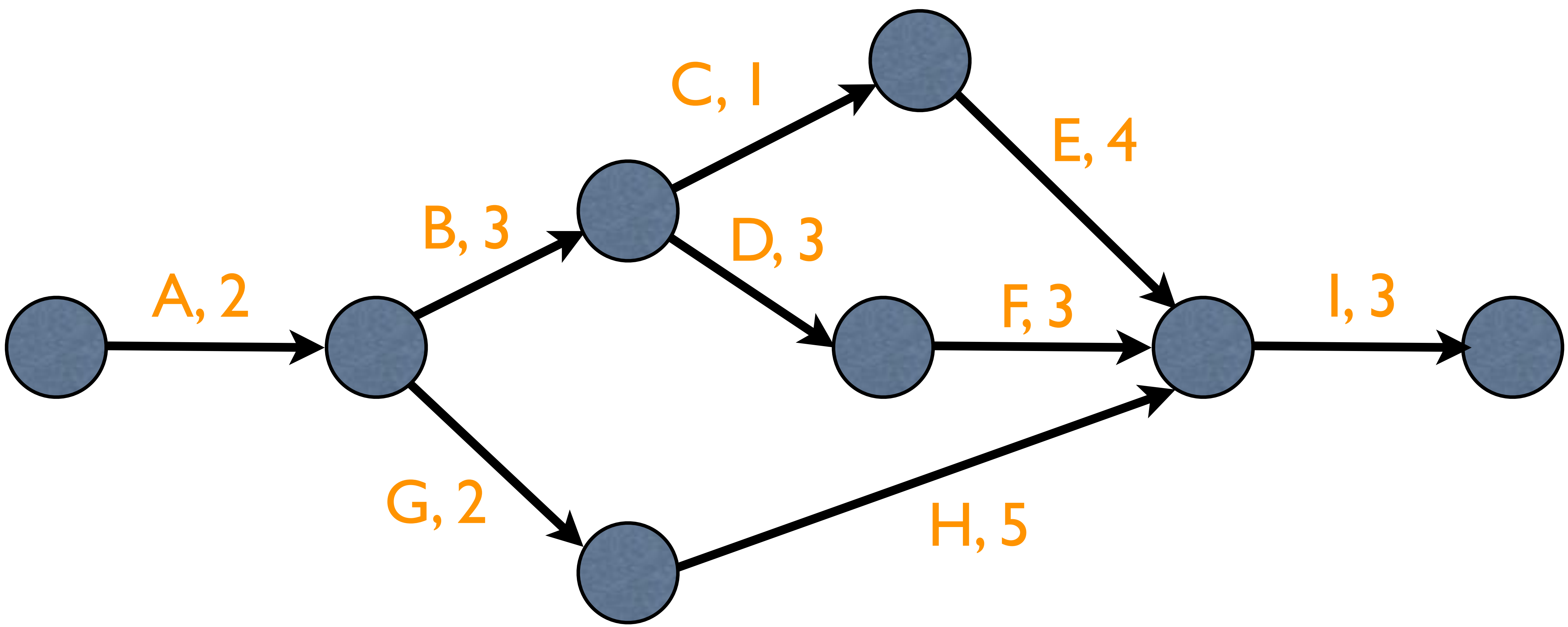
Critical Path Method

Labels indicate task name and duration



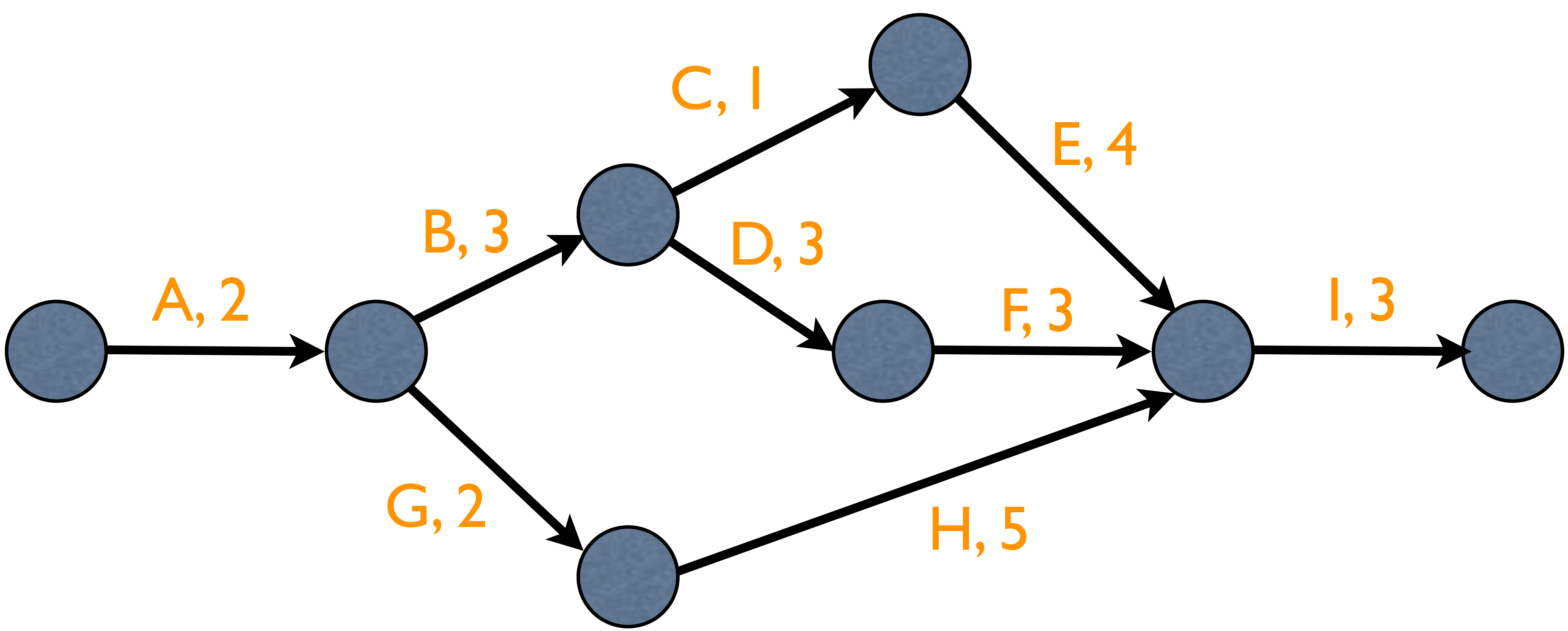
Critical Path Method

Nodes indicate the start and end points of tasks.



Critical Path Method

Partial order between edges capture project dependency



Critical Path Method

Determine Earliest Start & Finish Time

- **Earliest Start Time (ES)**

  Earliest time an activity can start

  ES = max EF of immediate predecessor activities

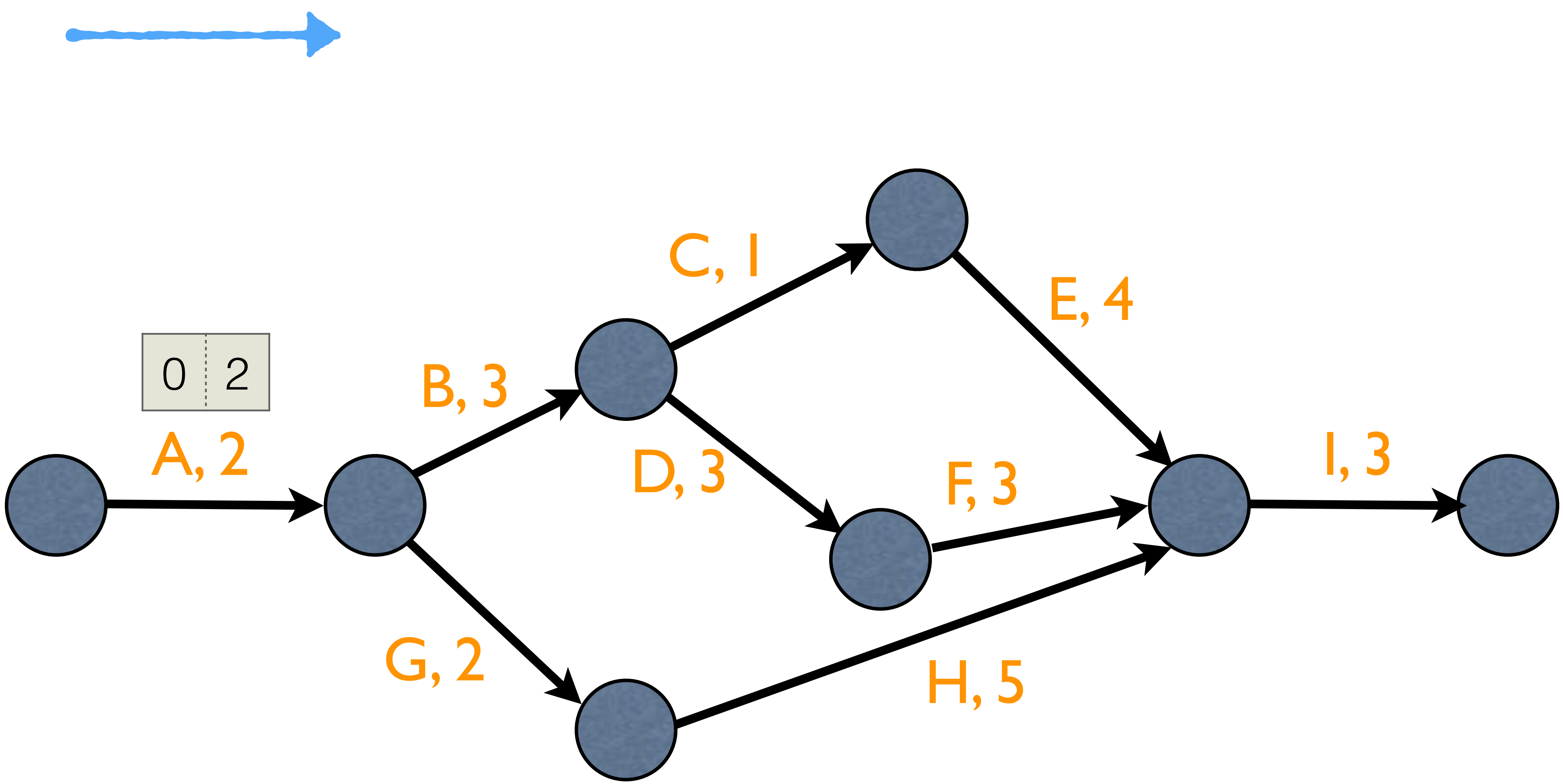
- **Earliest Finish Time (EF)**

  Earliest time an activity can finish

  EF = ES + activity_duration

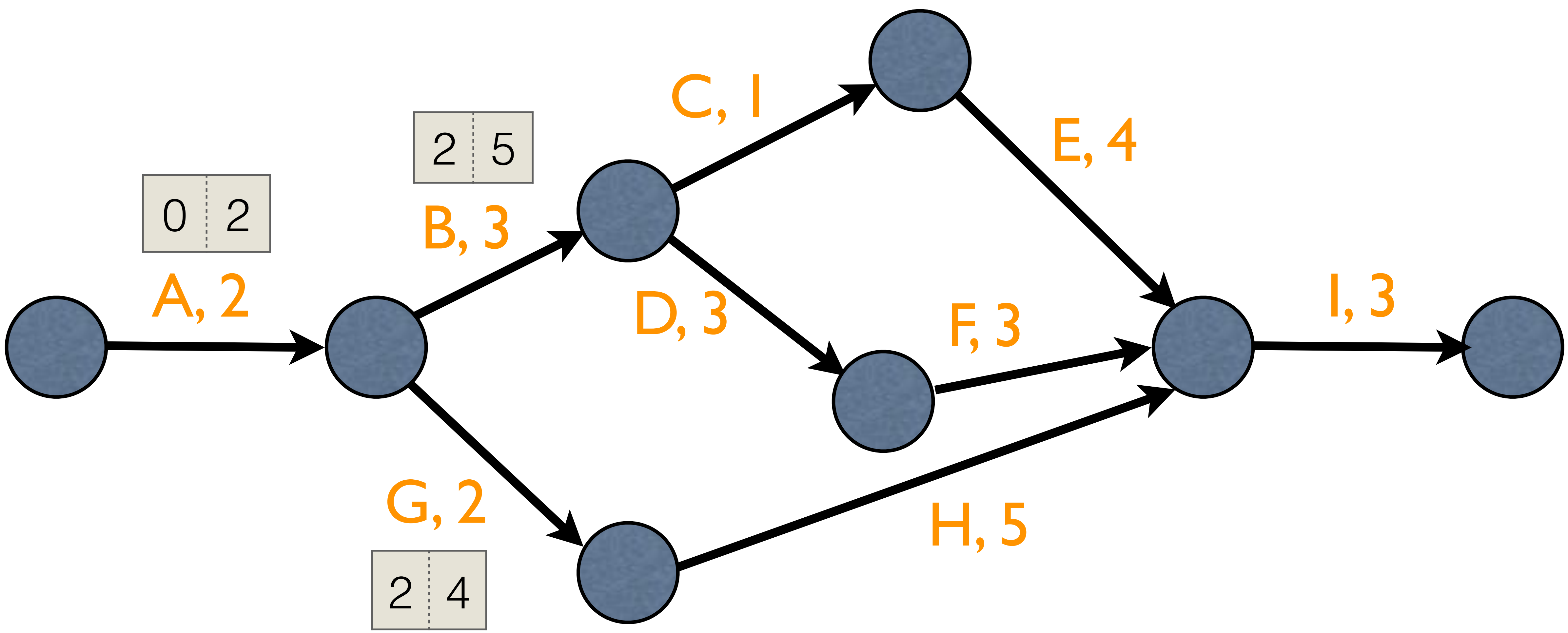
Critical Path Method

Determine Earliest Start & Finish Time



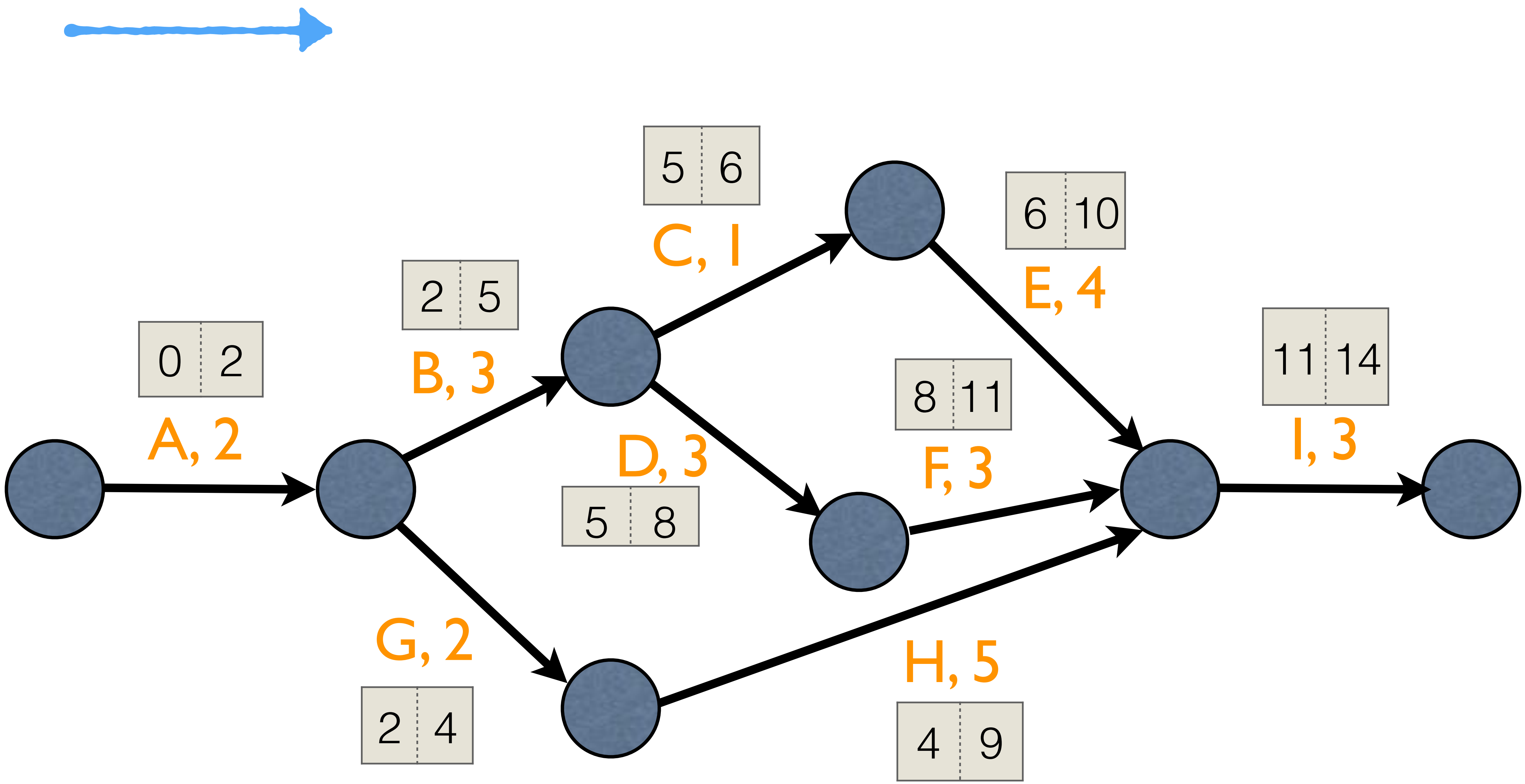
Critical Path Method

Determine Earliest Start & Finish Time



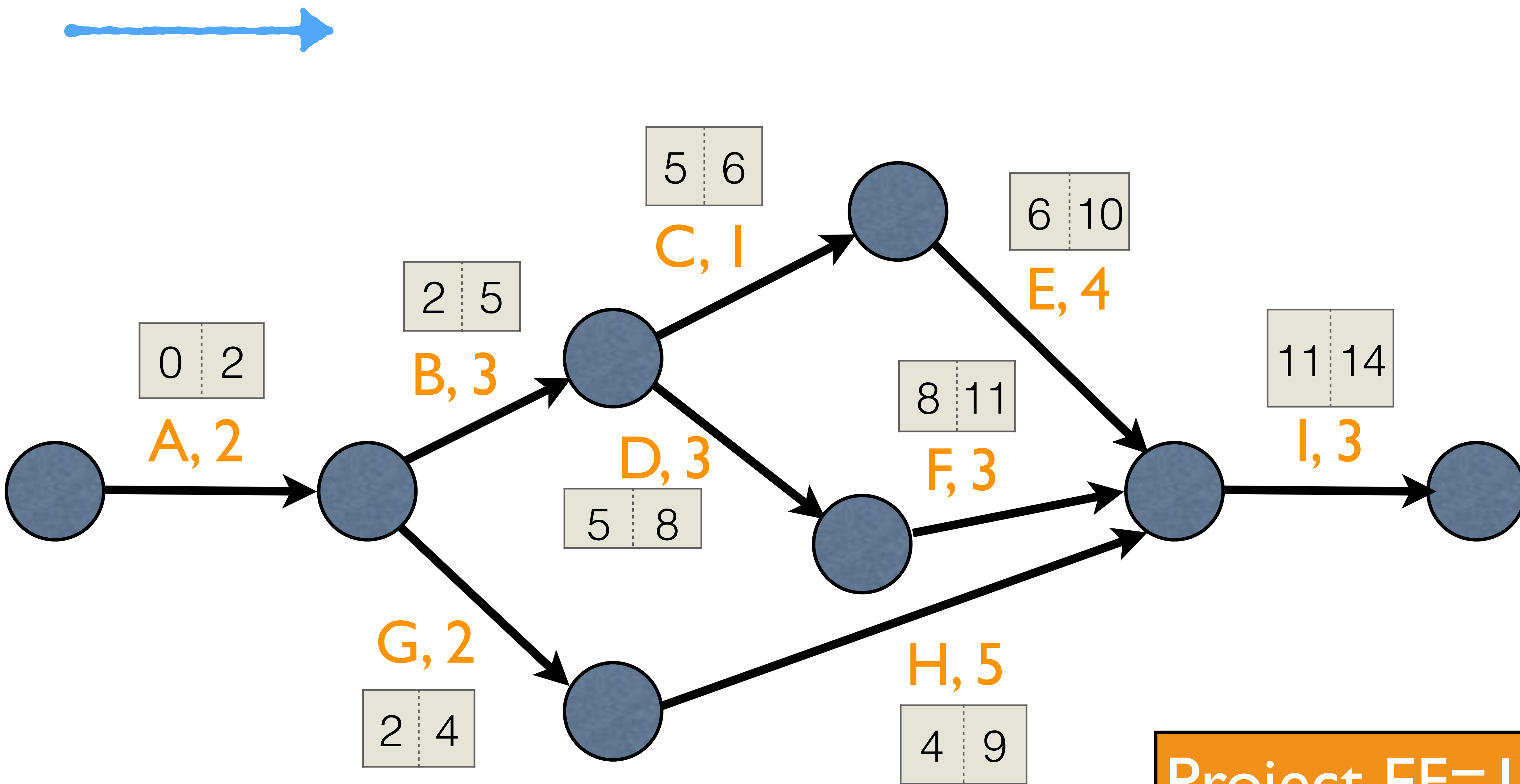
Critical Path Method

Determine Earliest Start & Finish Time



Critical Path Method

Determine Earliest Start & Finish Time



Project EF=14

Critical Path Method

Determine Latest Start & Finish Time

- **Latest Start Time (LS)**

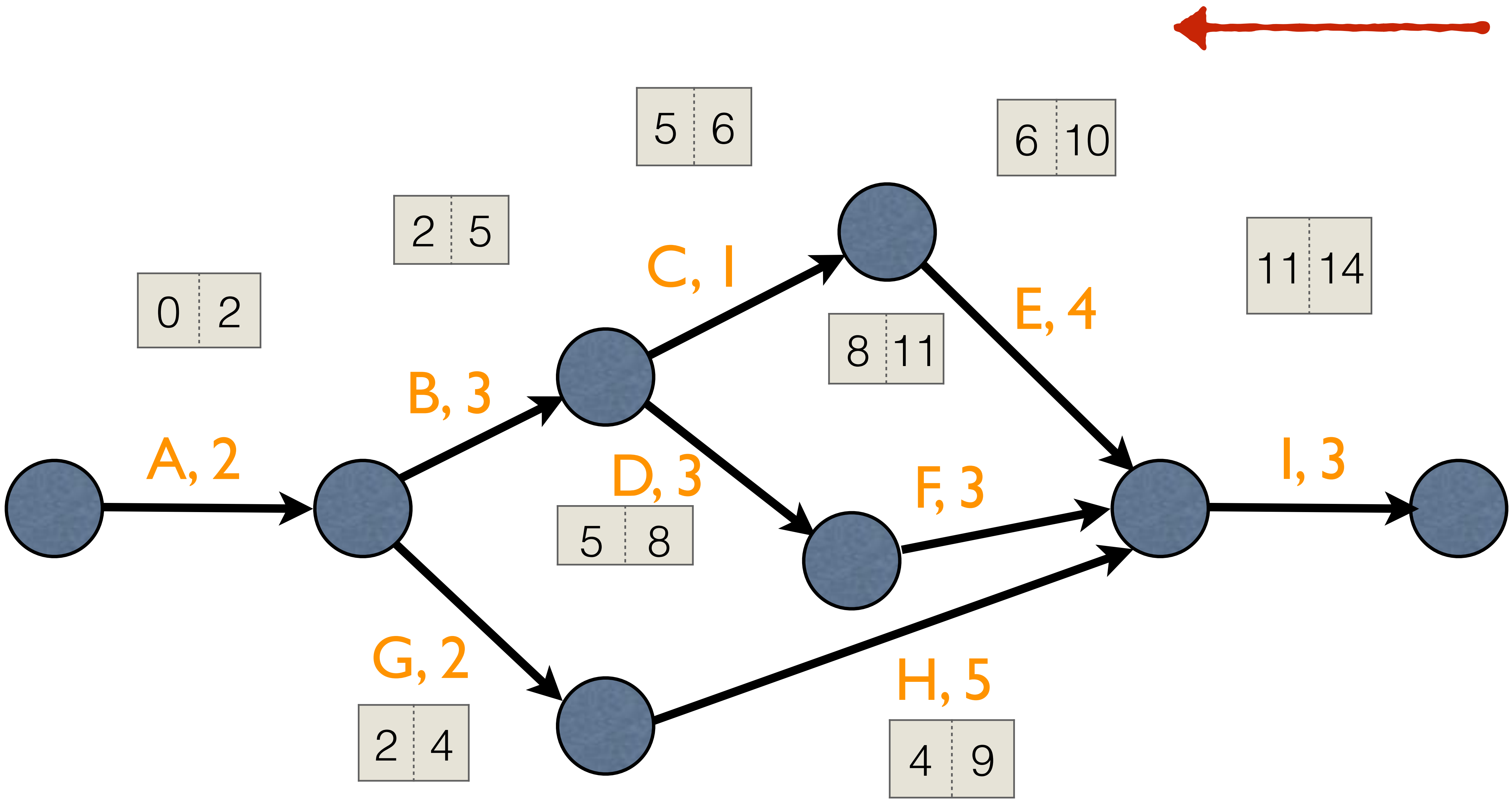
  Latest time an activity can start without delaying critical path time
  LS = LF - activity_duration

- **Latest Finish Time (LF)**

  Latest time an activity can finish without delaying critical path time
  LF = min LS of immediate predecessors

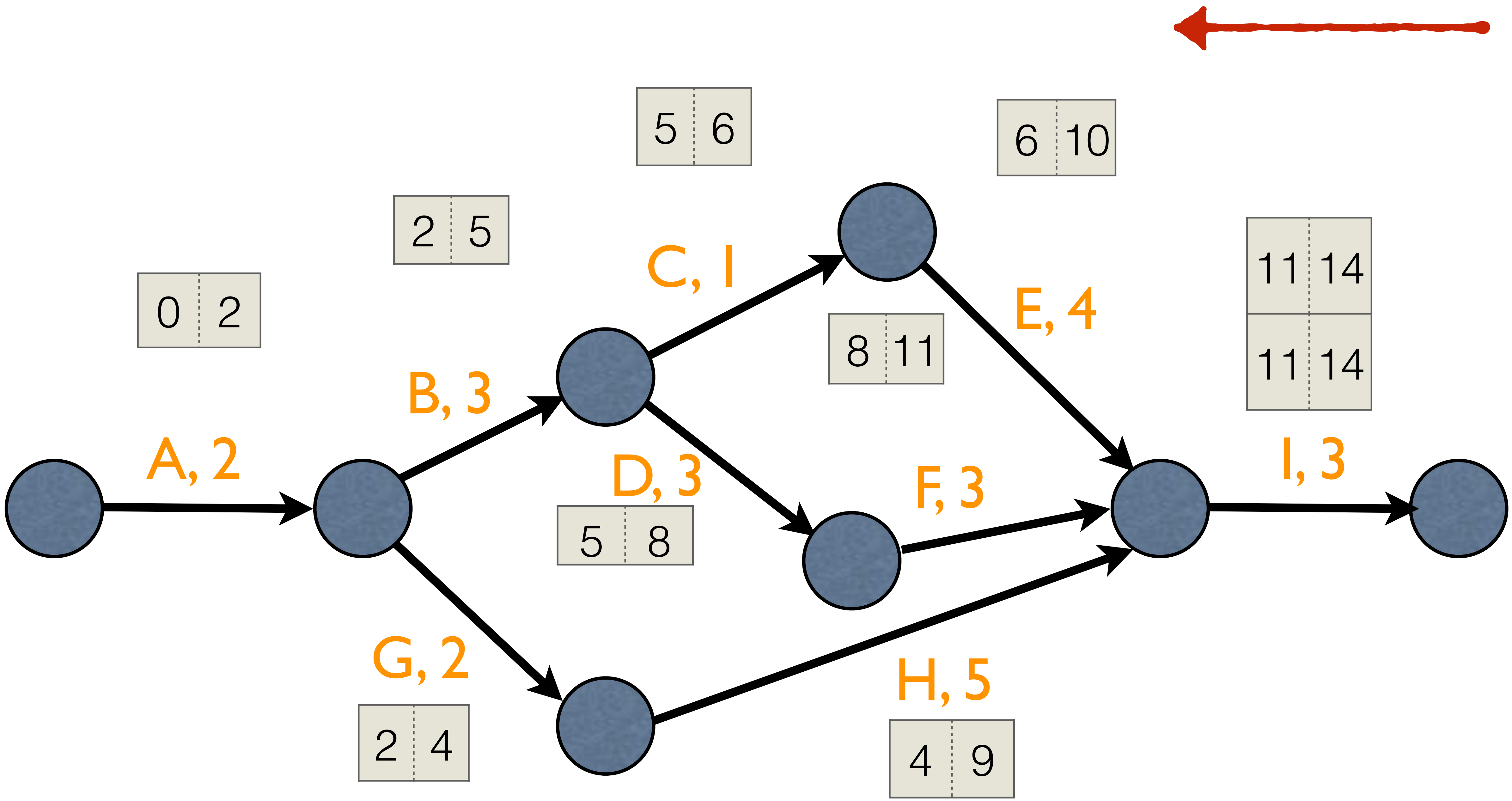
Critical Path Method

Determine Latest Start & Finish Time



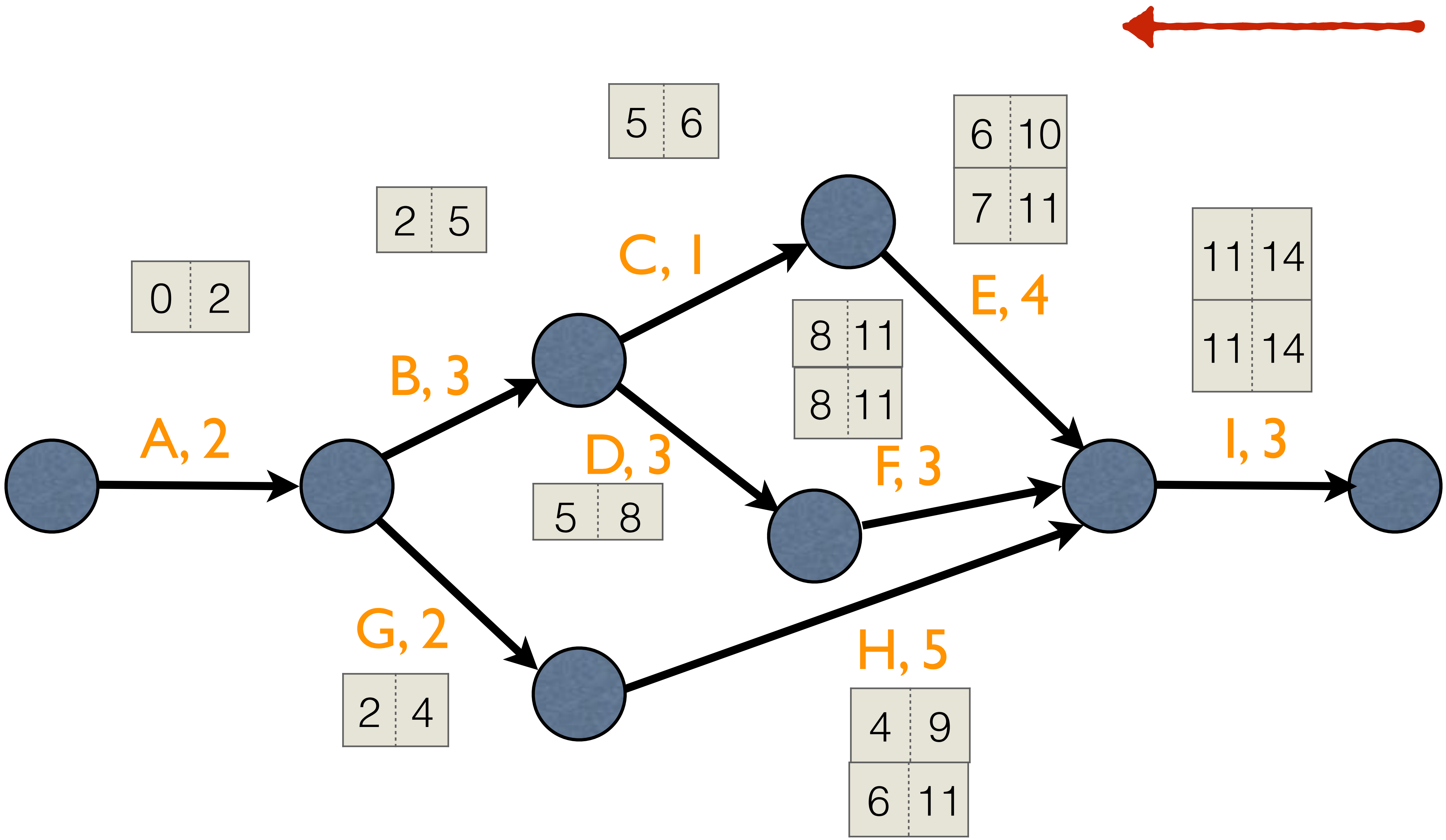
Critical Path Method

Determine Latest Start & Finish Time



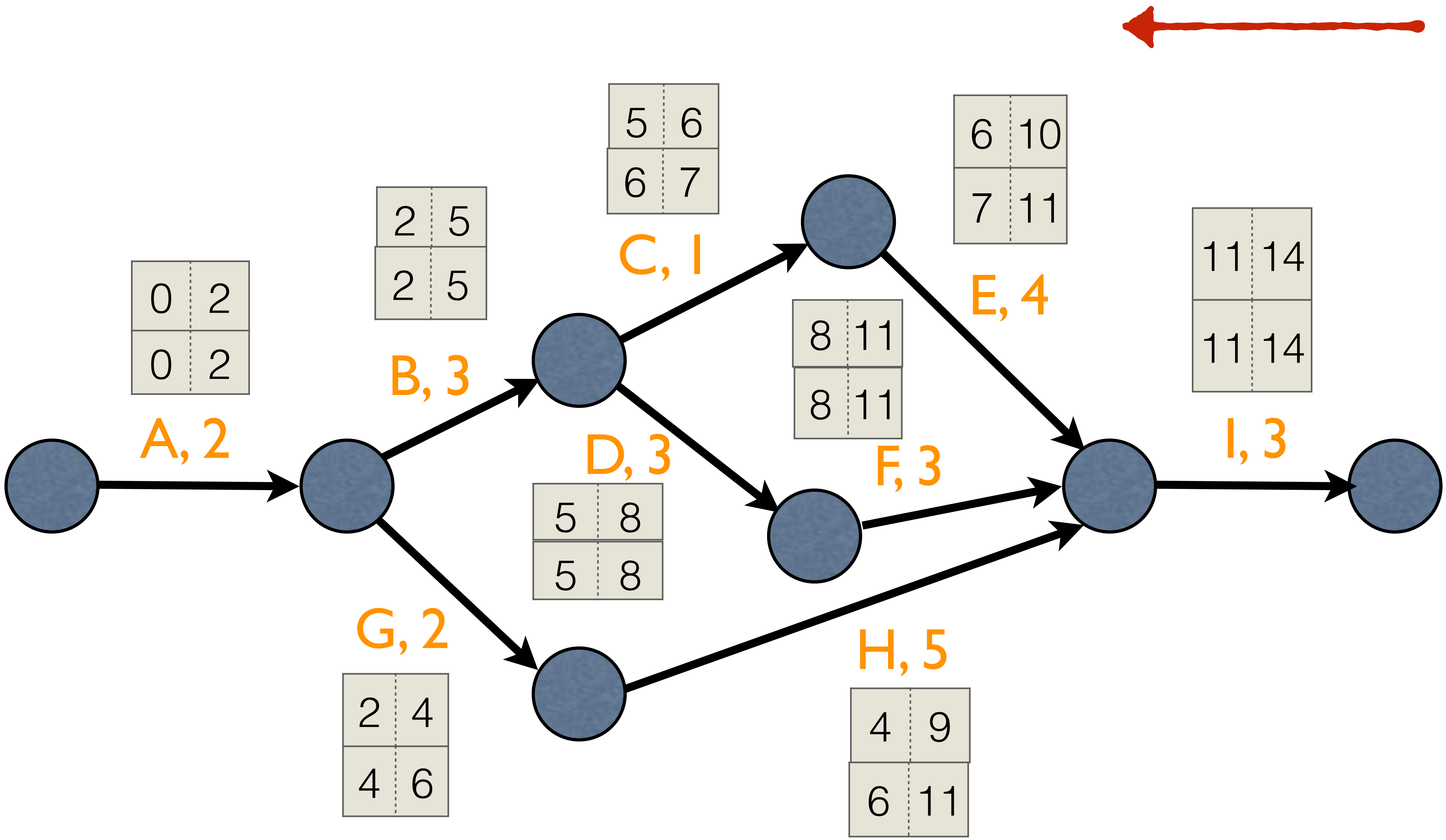
Critical Path Method

Determine Latest Start & Finish Time



Critical Path Method

Determine Latest Start & Finish Time

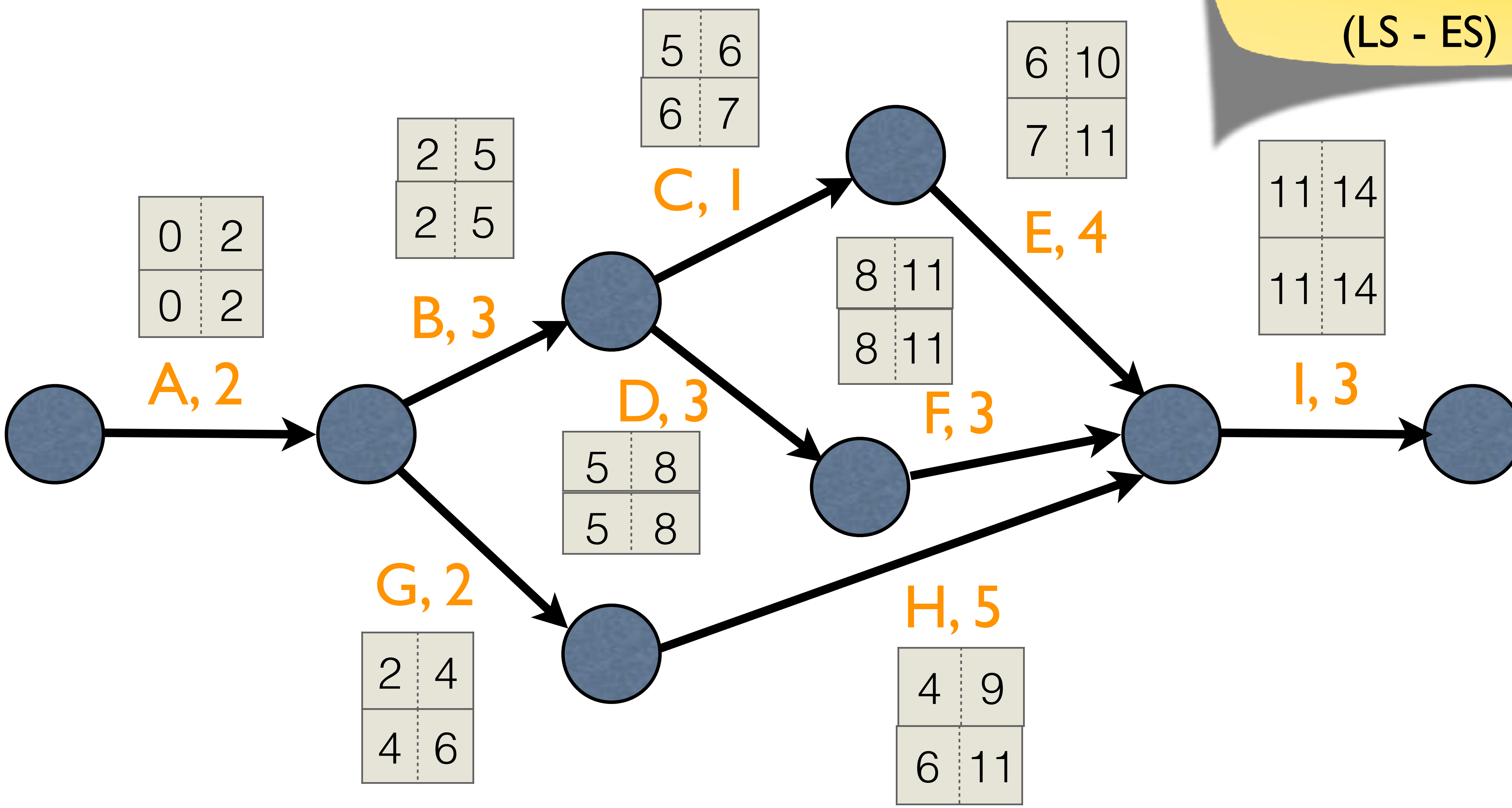


Critical Path Method

Critical path has zero slack

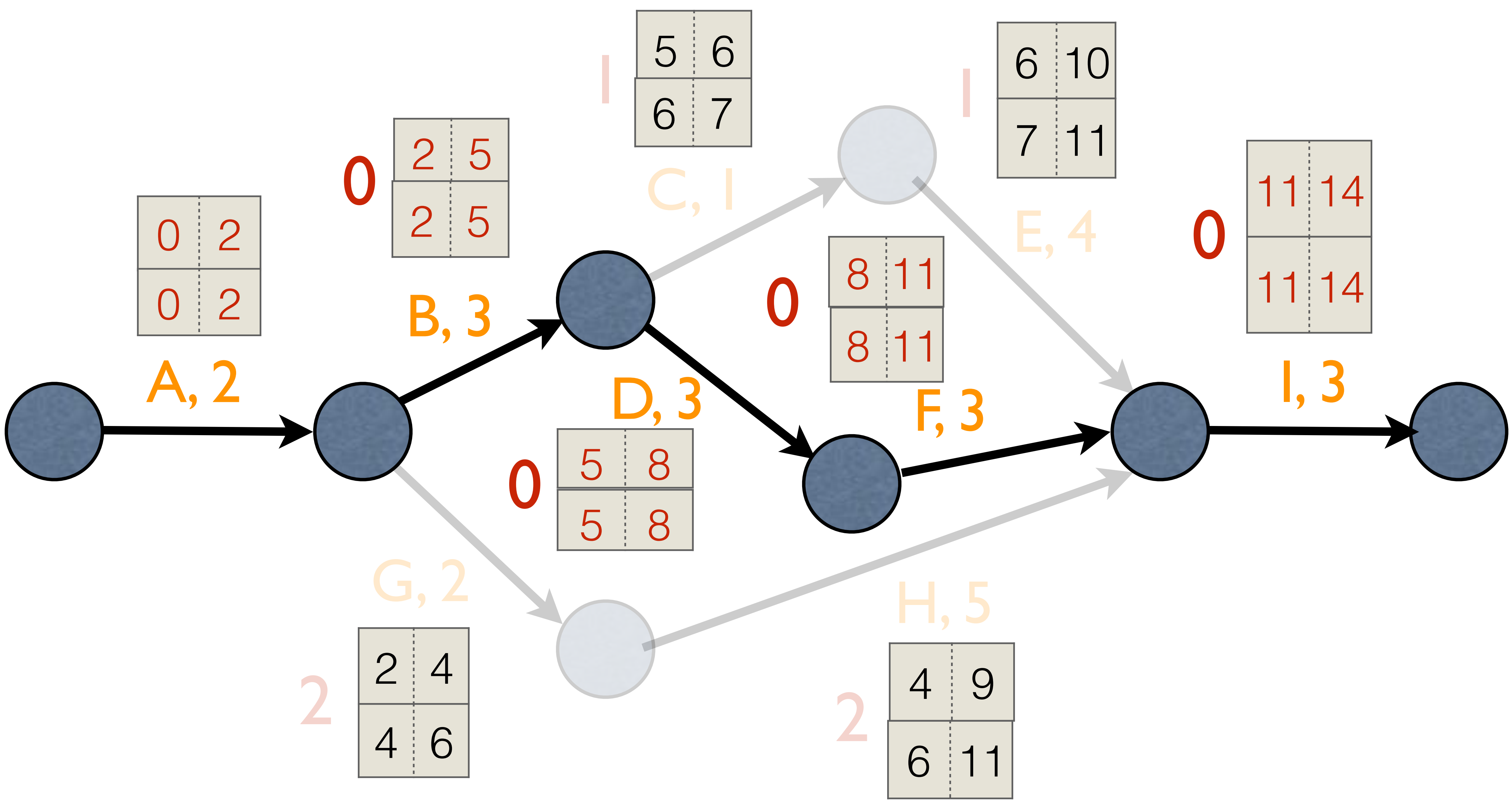
Slack: Max time activity can be delayed without delaying the project

(LS - ES)



Critical Path Method

Critical path has zero slack



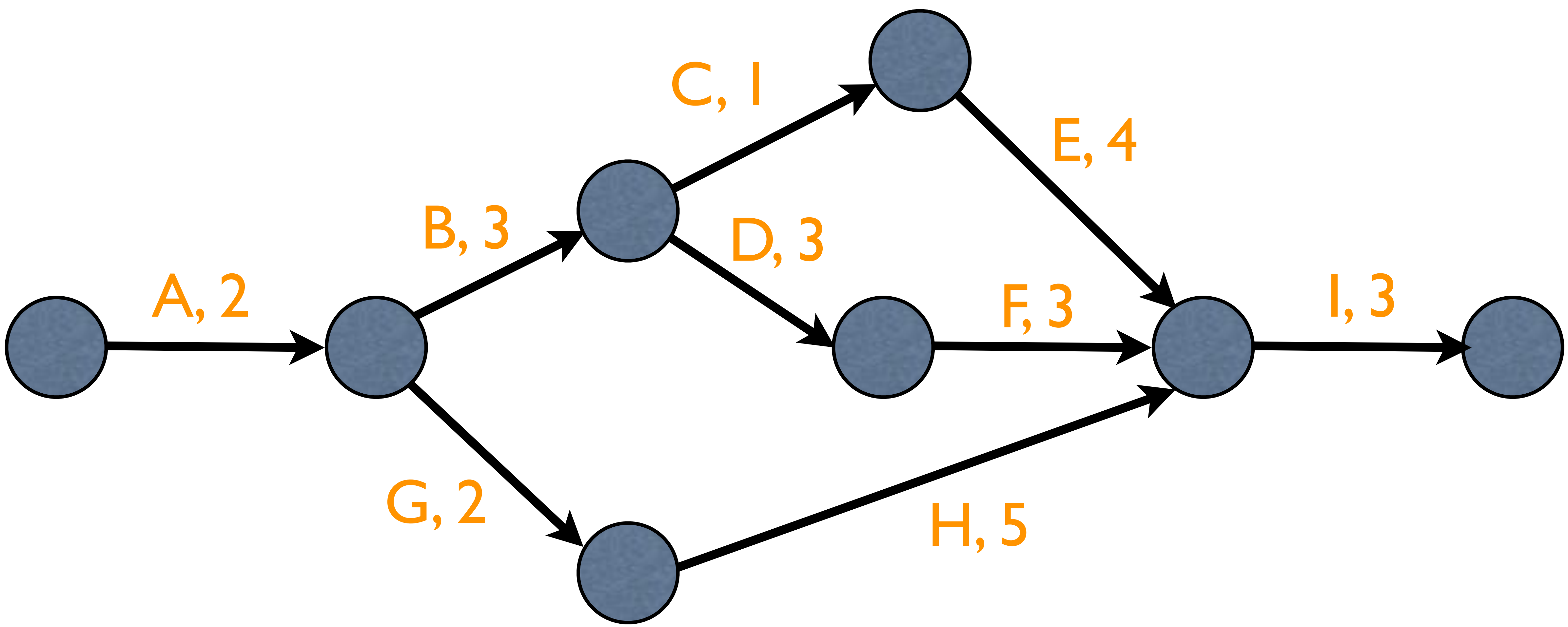
Critical Path Method

- Any delay to an activity in the critical path will cause delays to the overall project
- Delays to activities not on the critical path may relax, but keep a watch on *slack*

PERT Charts

PERT: Program Evaluation and Review Techniques

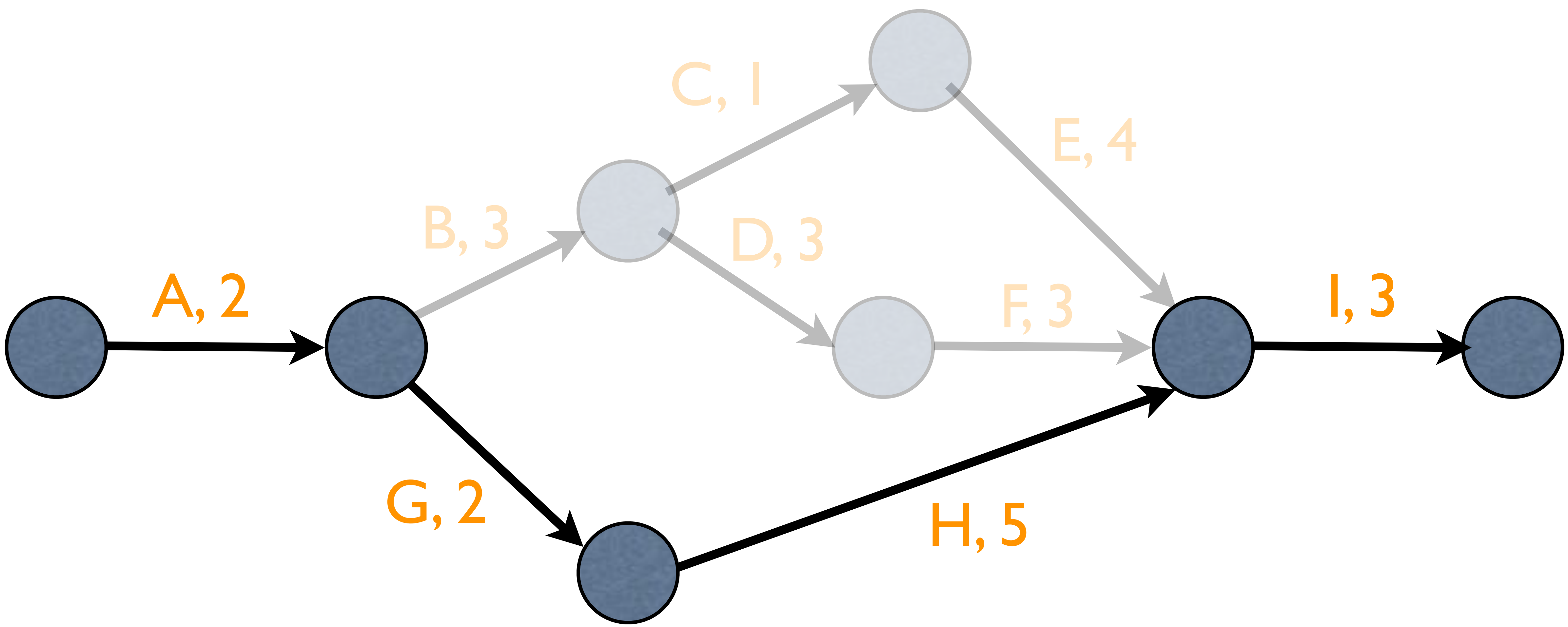
Analysing best-case and worst-case scenarios



PERT Charts

There are several routes to reach from start to finish.

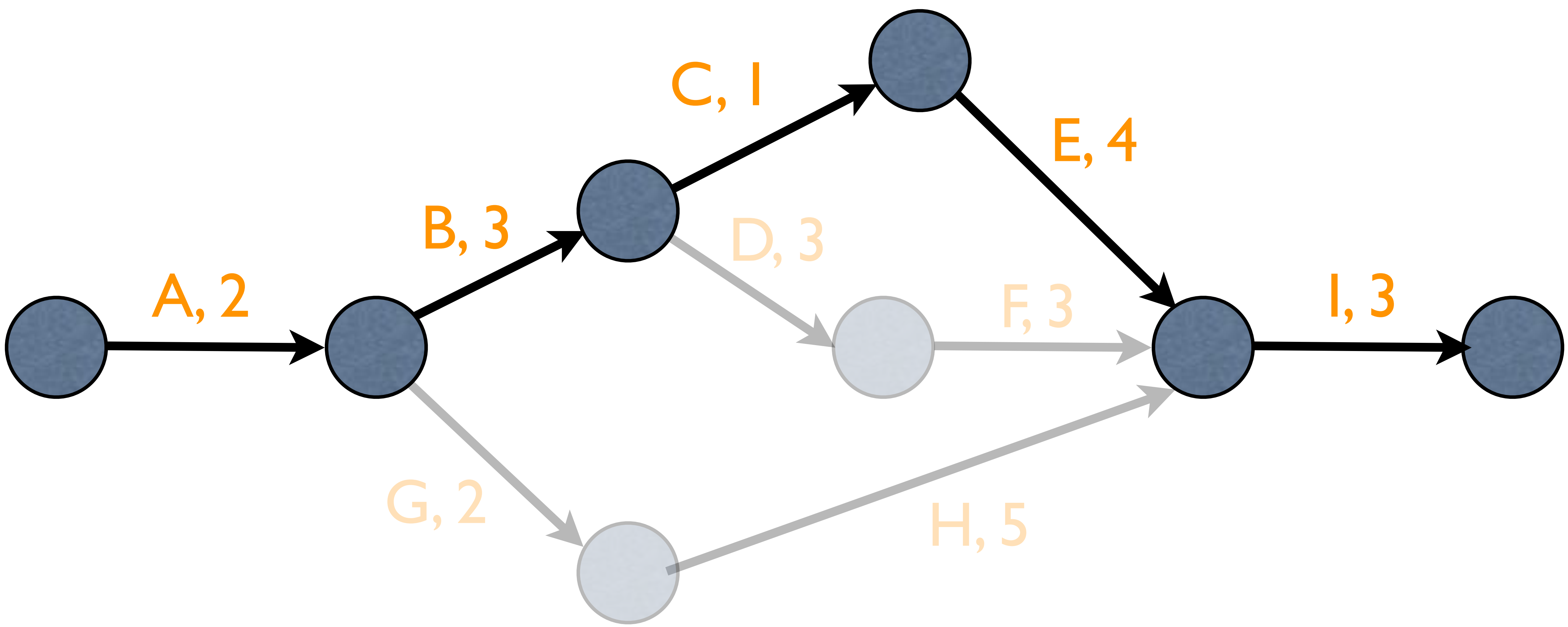
Time to complete: 12 days!



PERT Charts

There are several routes to reach from start to finish.

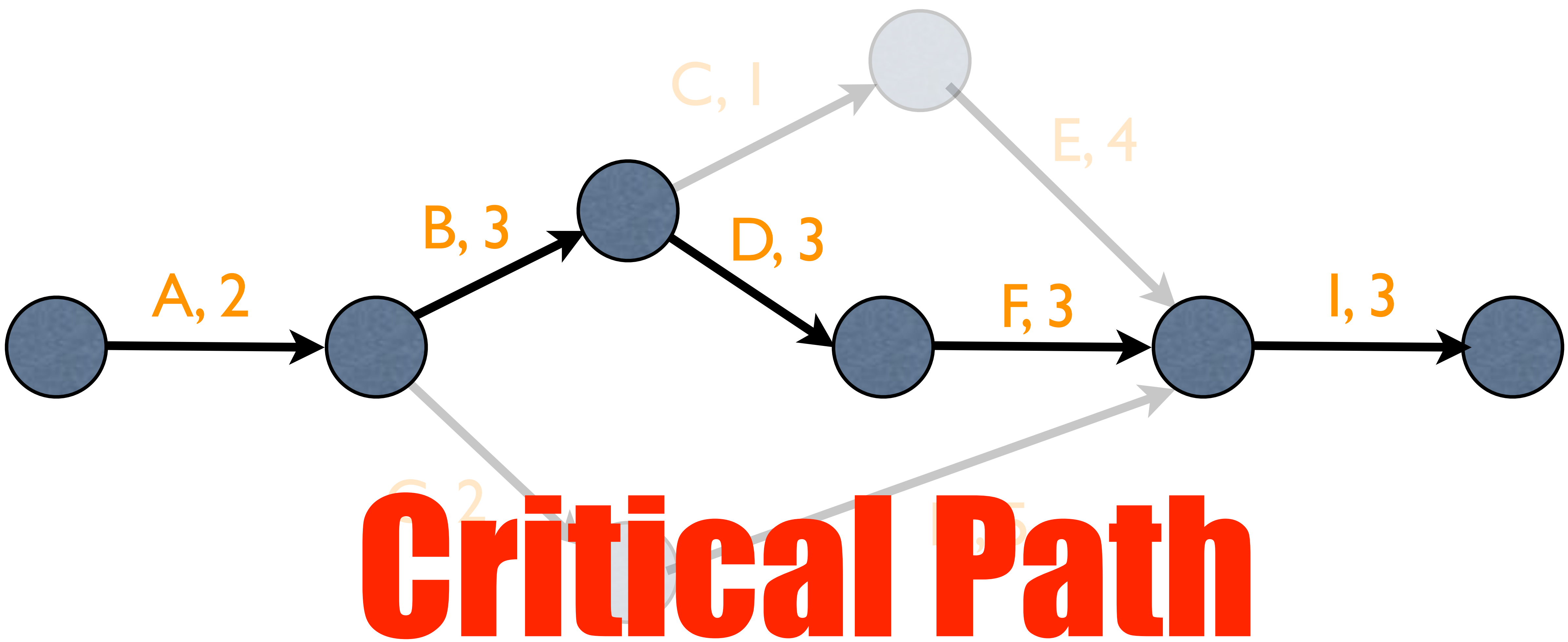
Time to complete: 13 days!



PERT Charts

There are several routes to reach from start to finish.

Time to complete: 14 days!



PERT charts

Dealing with uncertainty in activity completion times

- For each activity, 3 time estimates:

  Optimistic time (O): *minimum possible time required to complete a task*

  Pessimistic time (P): *maximum possible time required to complete a task*

  Most likely time (M): *most likely time required to complete a task*

PERT Charts

- PERT assumes a beta probability distribution for the time estimates

Expected time (T_E): *the best estimate of the time required to accomplish a task*

$$T_E = (O + 4M + P) / 6$$

Example:

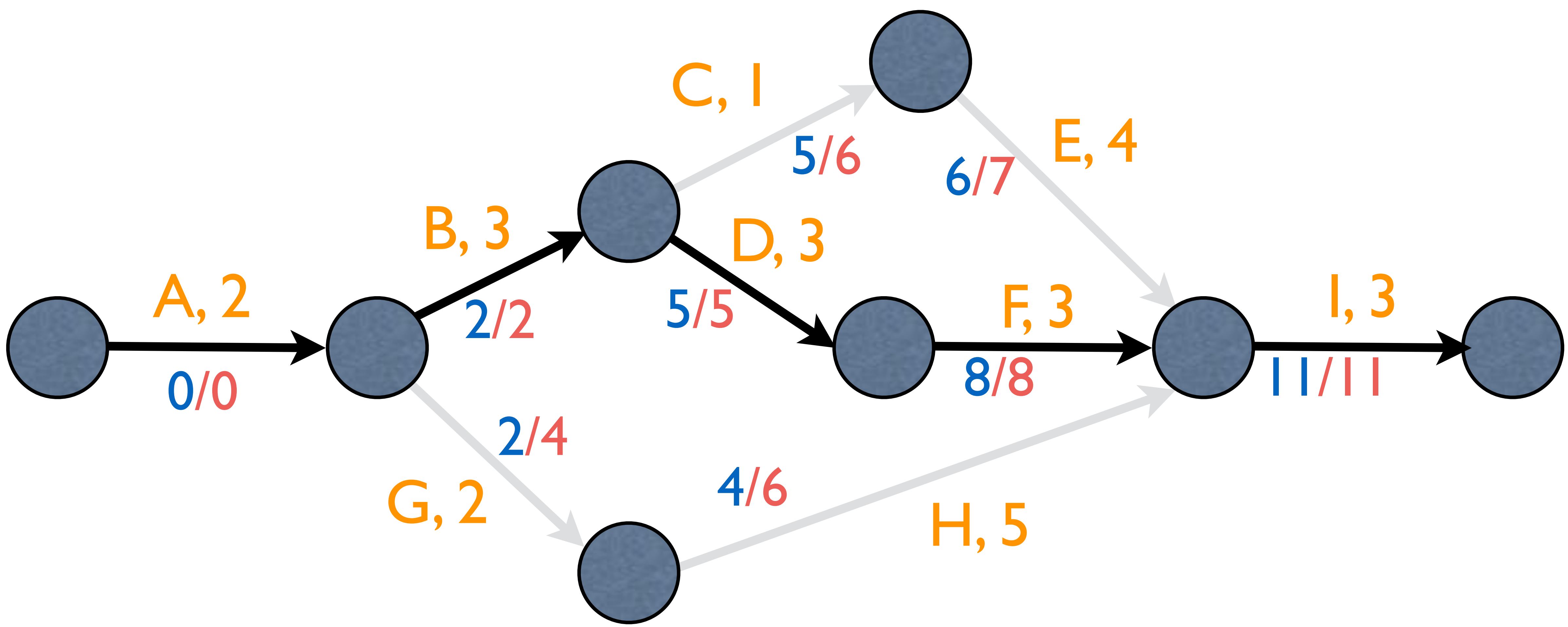
Optimistic	Most Likely	Pessimistic
3 months	4 months	6 months

T_E = (3 months + 4*4months + 6 months) / 6 = **4.17 months**

PERT Charts

Critical path

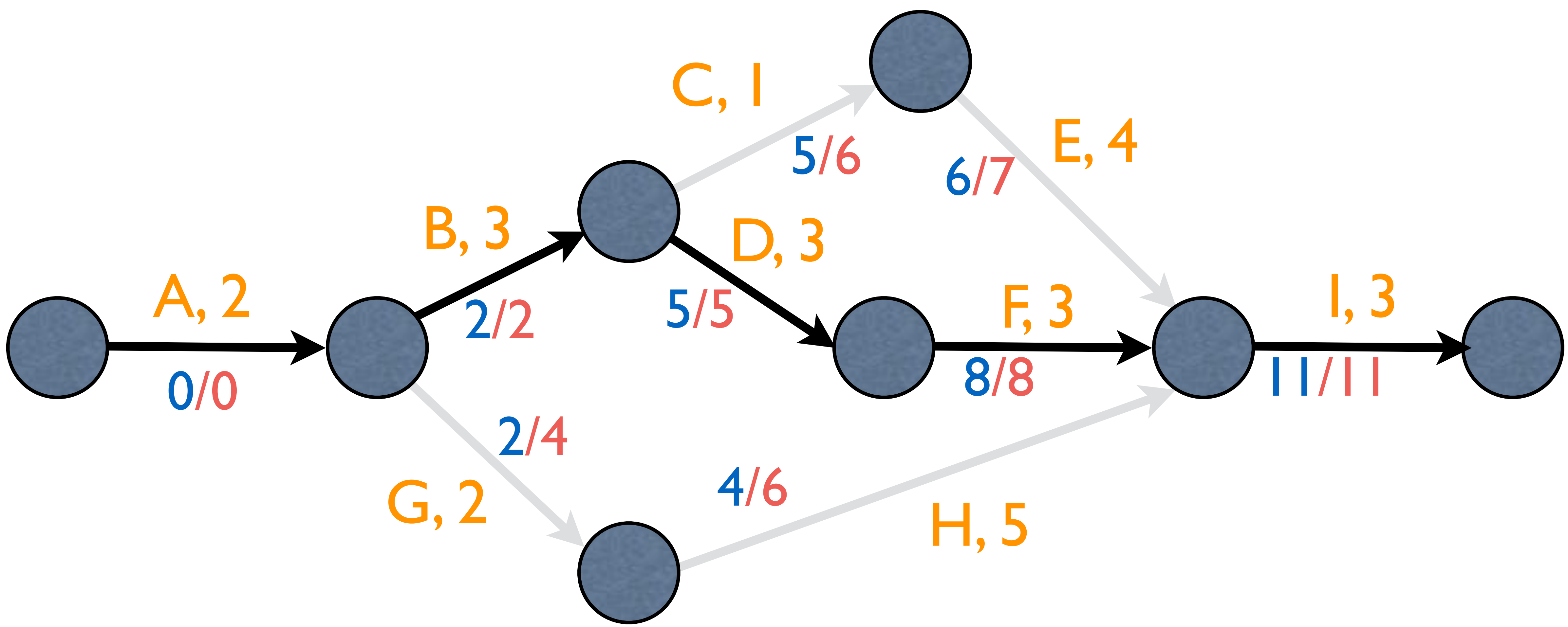
- Why is it called *critical?*

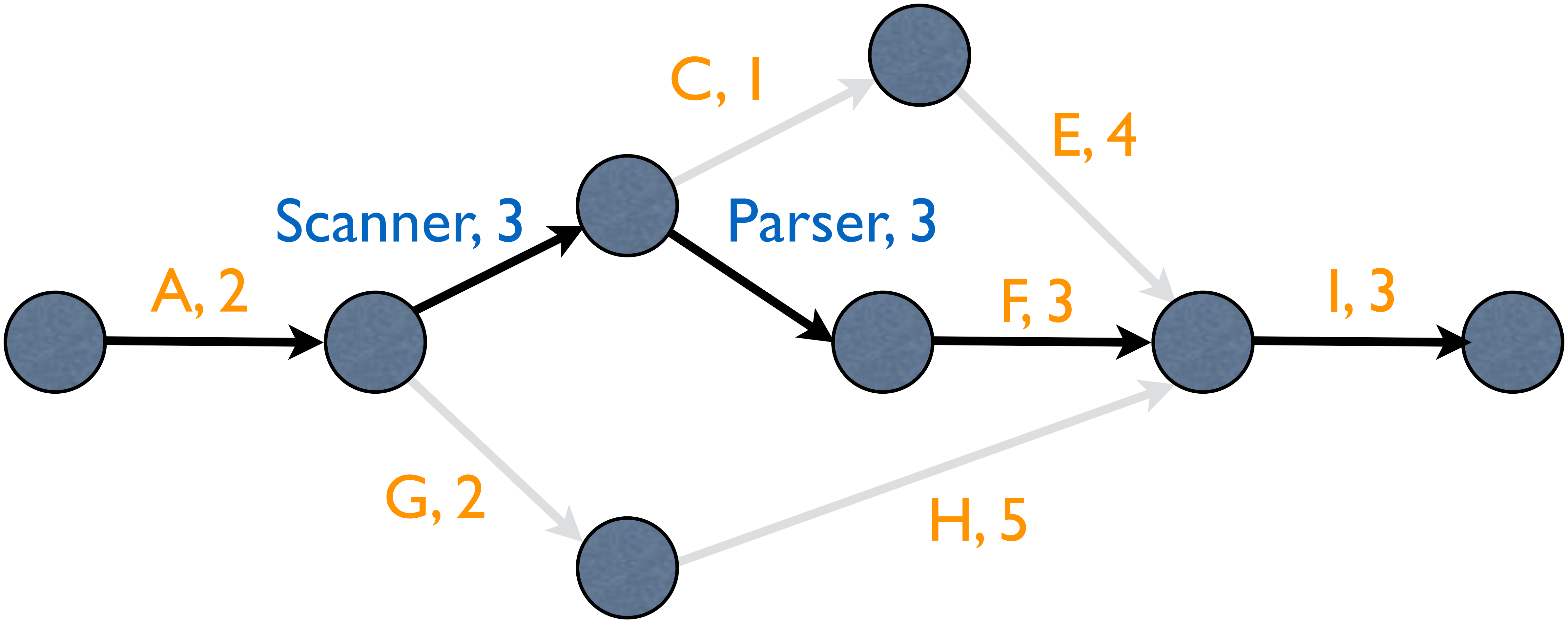


PERT Charts

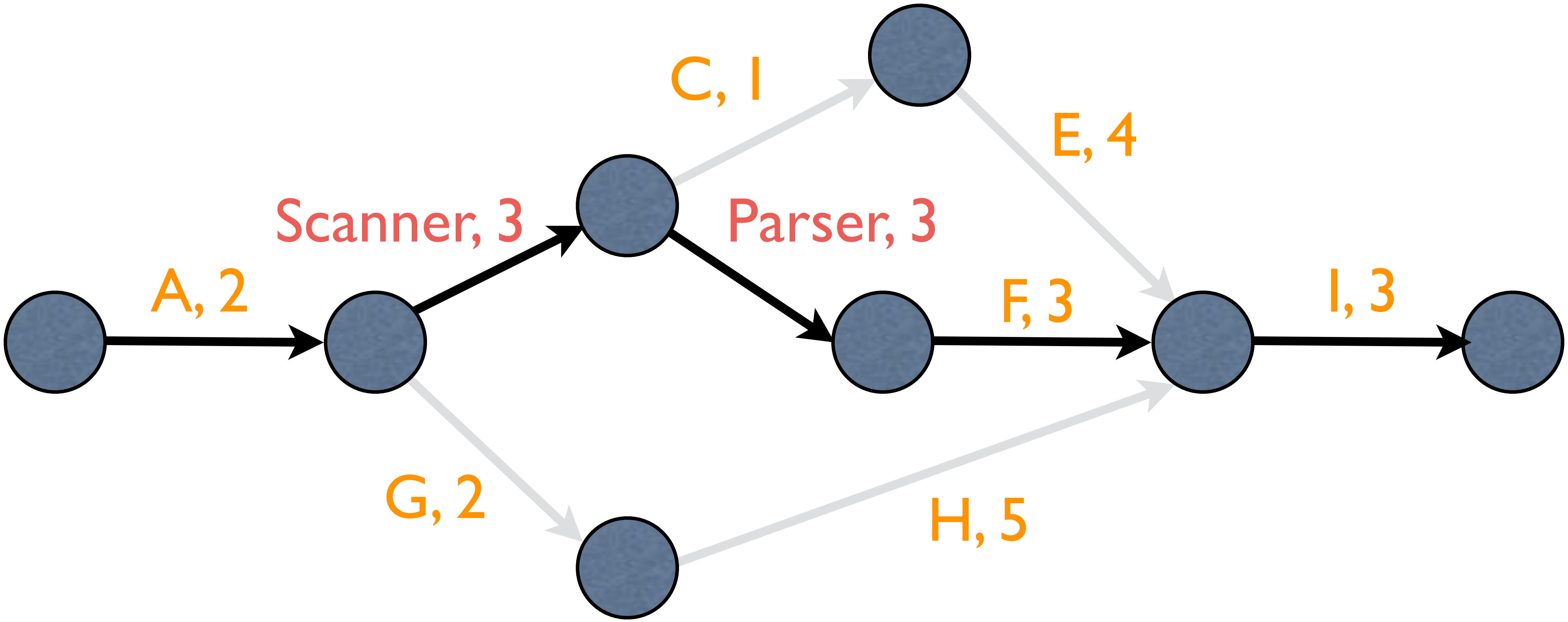
Critical path

- Why is it called *critical?*
- *How should we optimise critical path?*

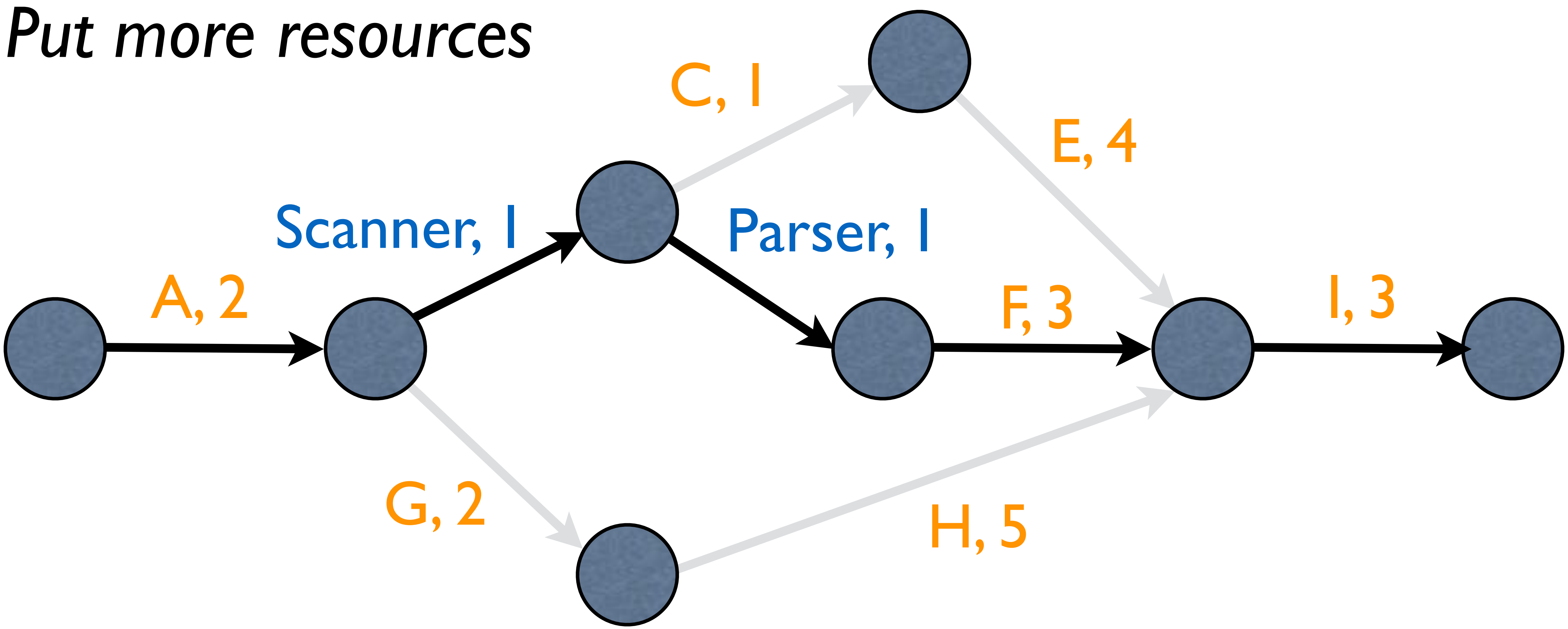


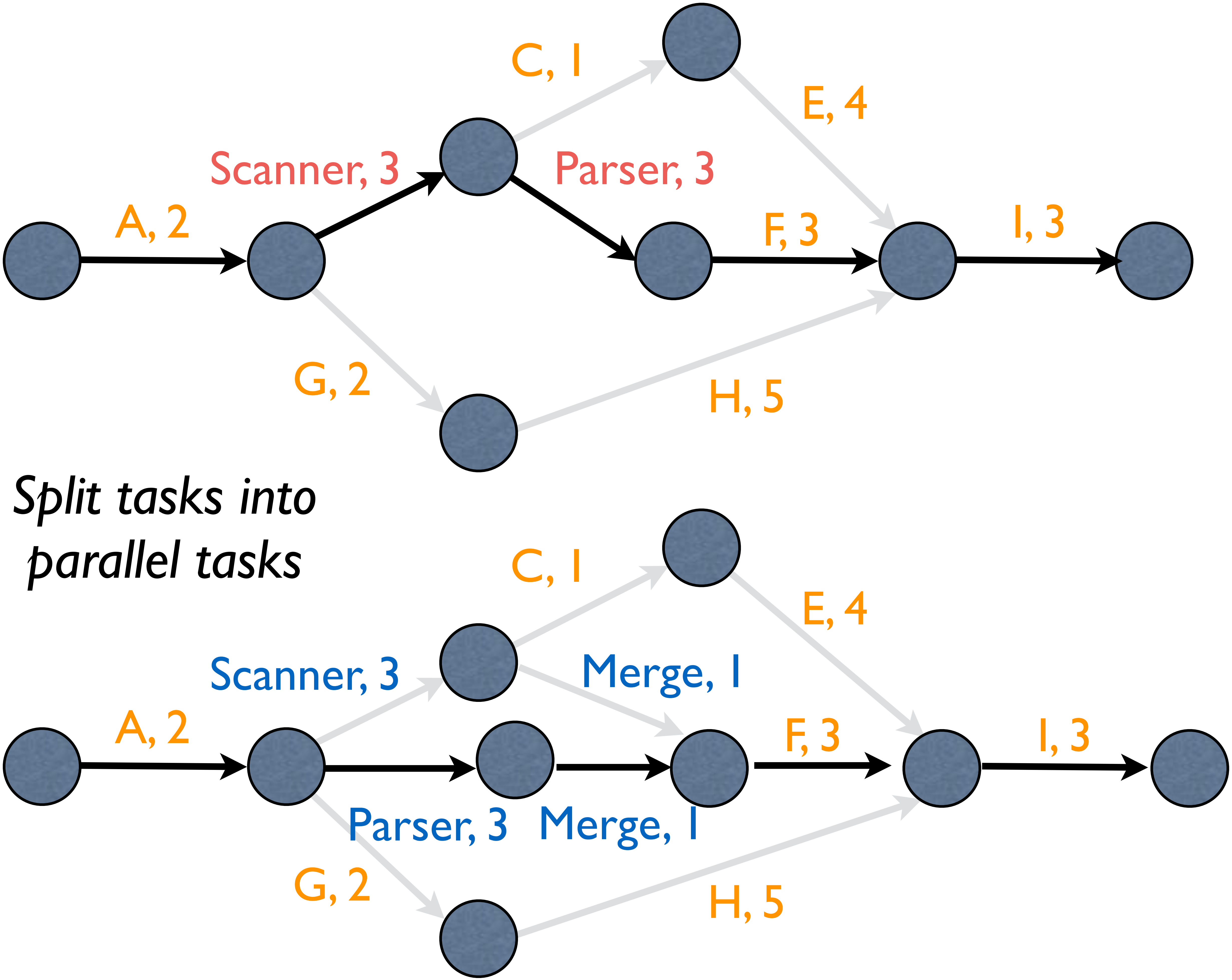


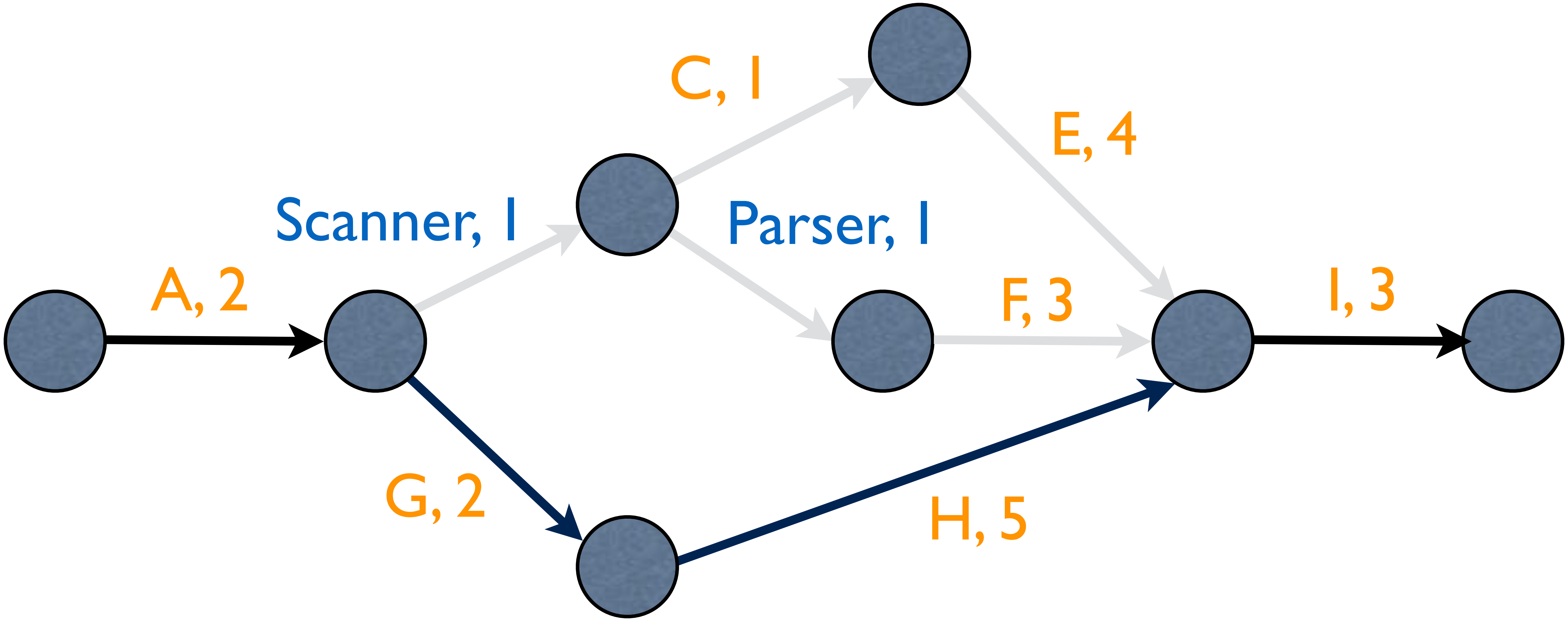
C, 1
E, 4
Scanner, 3
Parser, 3
A, 2
F, 3
I, 3
G, 2
H, 5



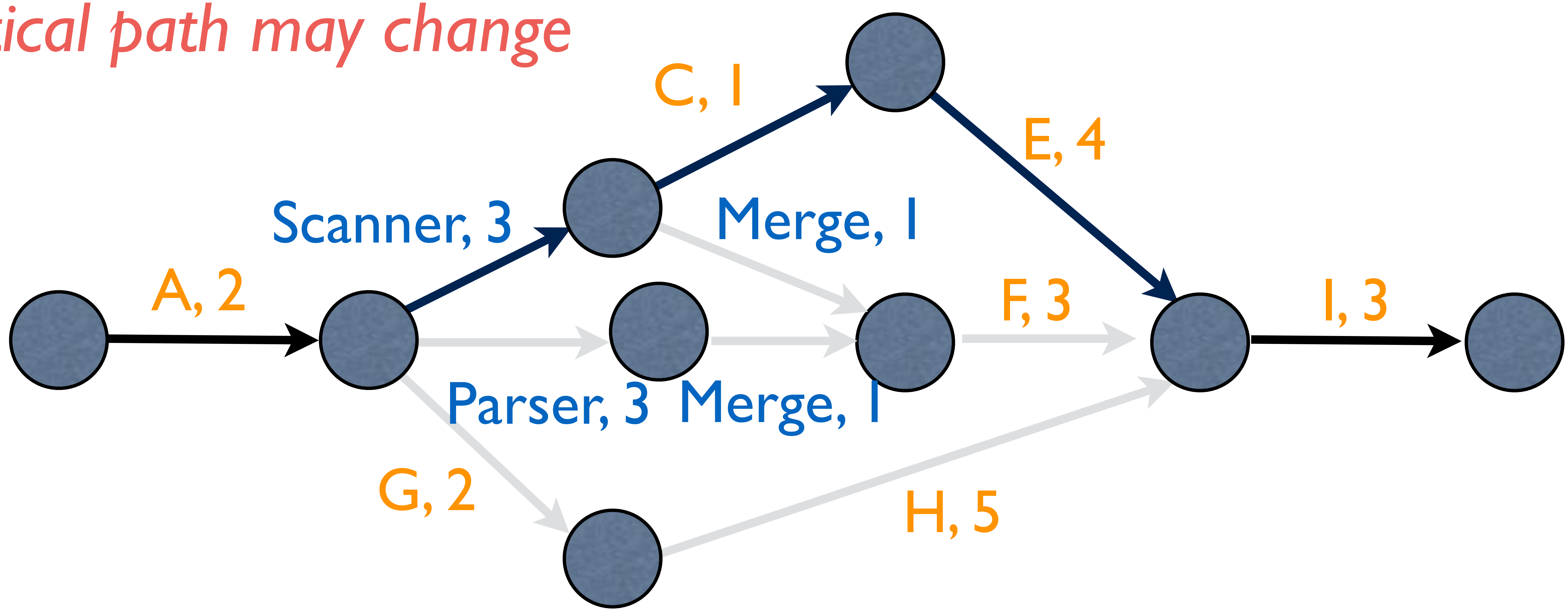
Put more resources



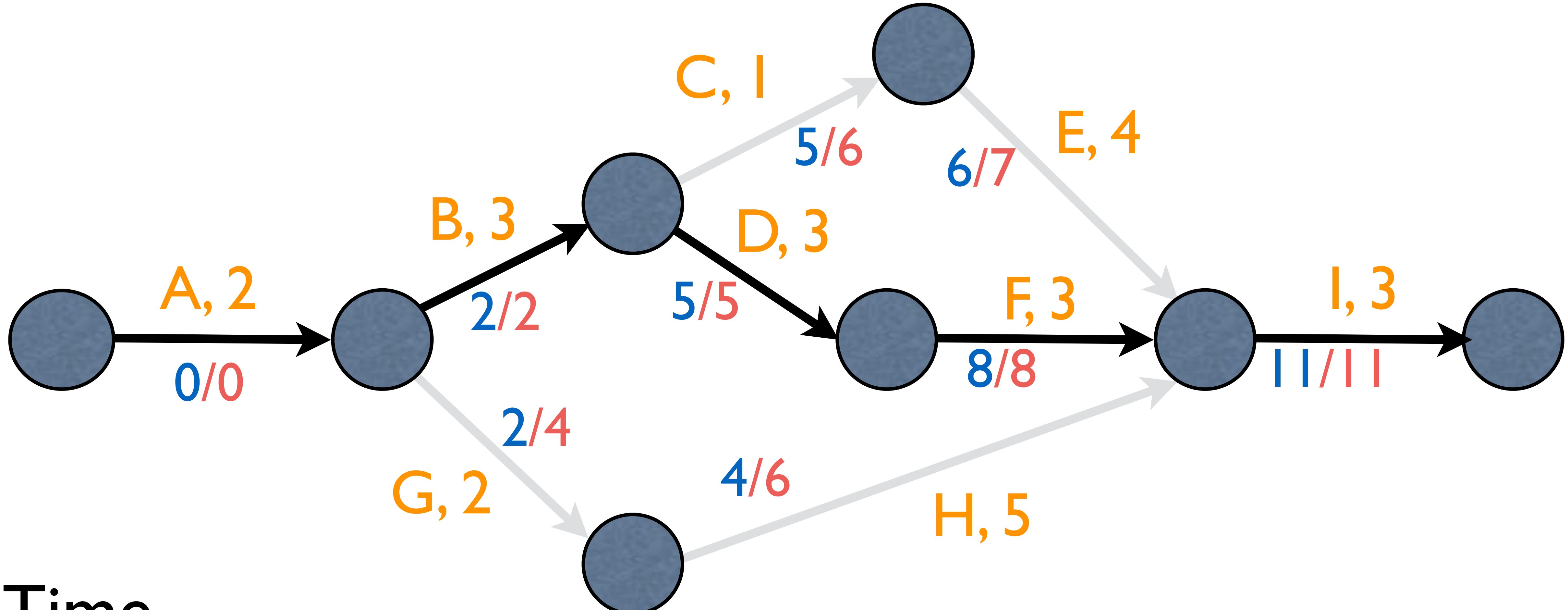

C, 1
E, 4
Scanner, 3
Parser, 3
A, 2
F, 3
I, 3
G, 2
H, 5
Split tasks into parallel tasks
C, 1
E, 4
Scanner, 3
Merge, 1
A, 2
F, 3
I, 3
Parser, 3
Merge, 1
G, 2
H, 5



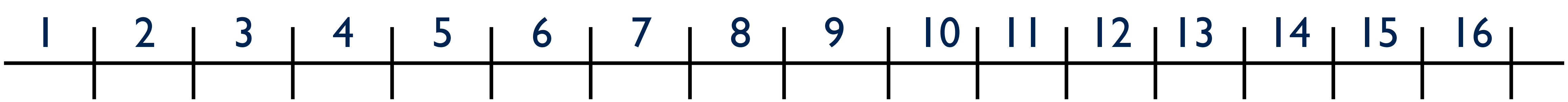
Critical path may change



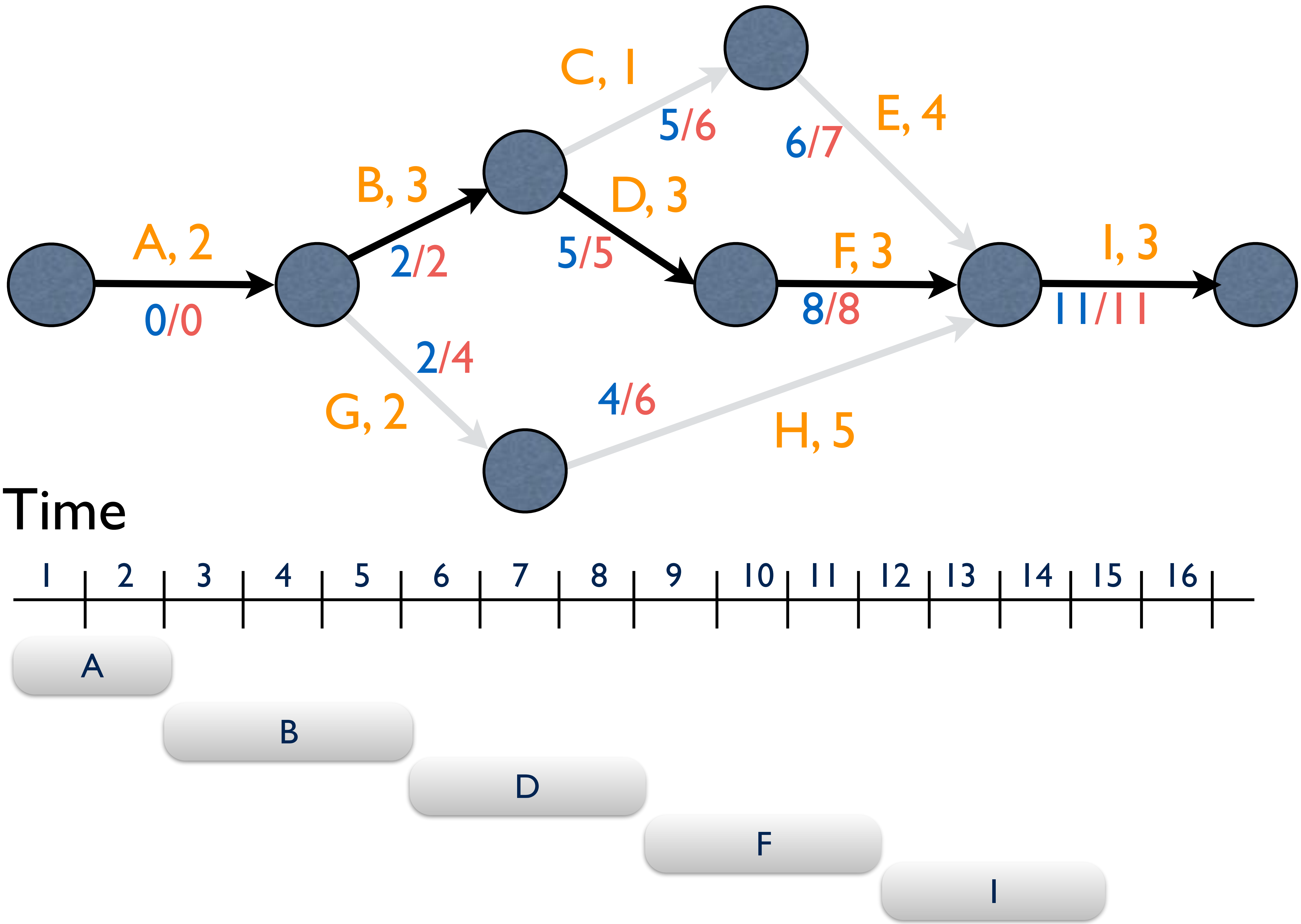
Gantt Chart



Time

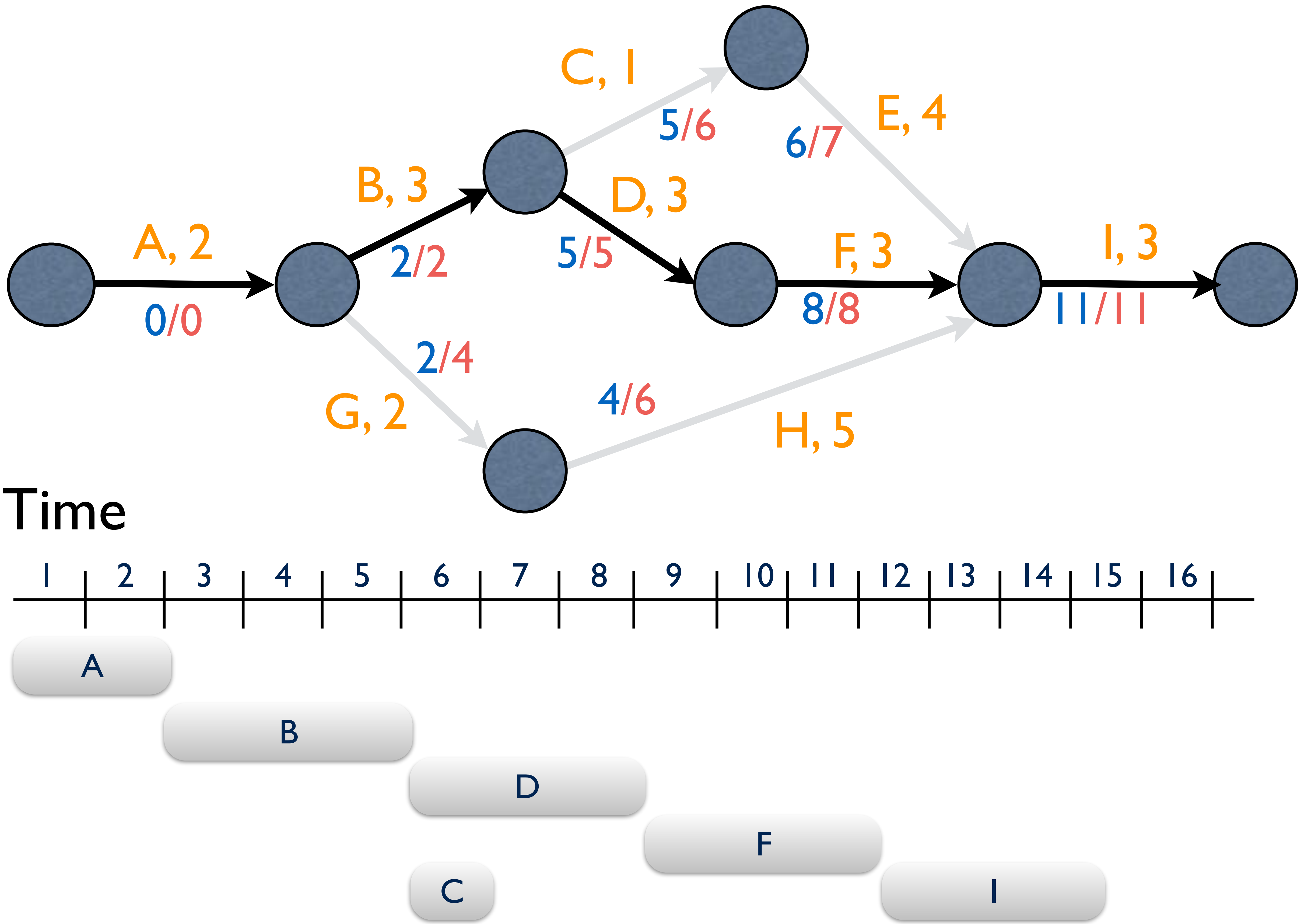


Gantt Chart



A, 2
0/0
B, 3
2/2
C, 1
5/6
D, 3
5/5
E, 4
6/7
F, 3
8/8
G, 2
2/4
H, 5
4/6
I, 3
11/11
Time
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
A
B
D
F
I

Gantt Chart

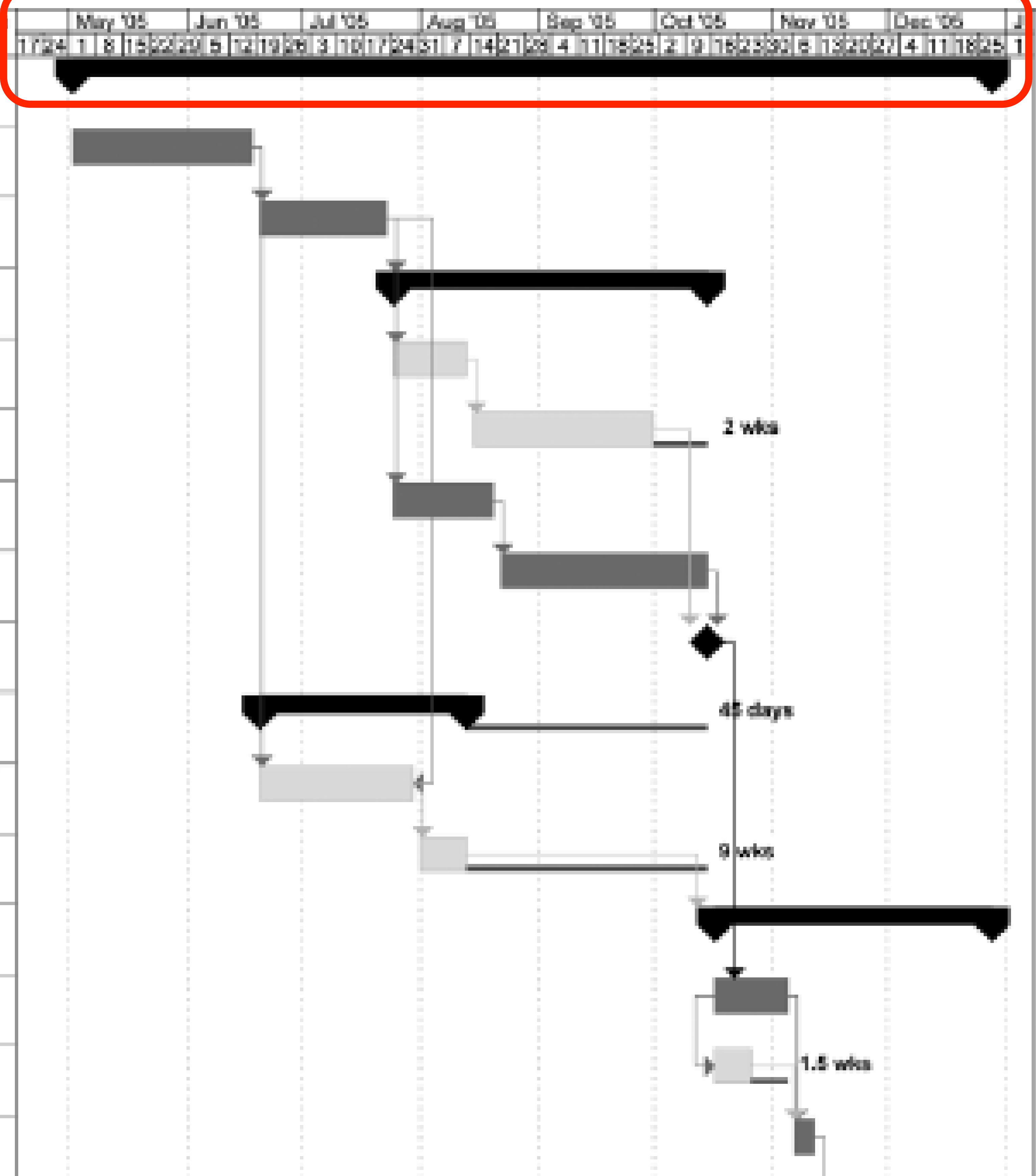

A, 2
0/0
B, 3
2/2
C, 1
5/6
D, 3
5/5
E, 4
6/7
F, 3
8/8
G, 2
2/4
H, 5
4/6
I, 3
11/11
Time
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
A
B
D
F
C
I

Gantt Chart

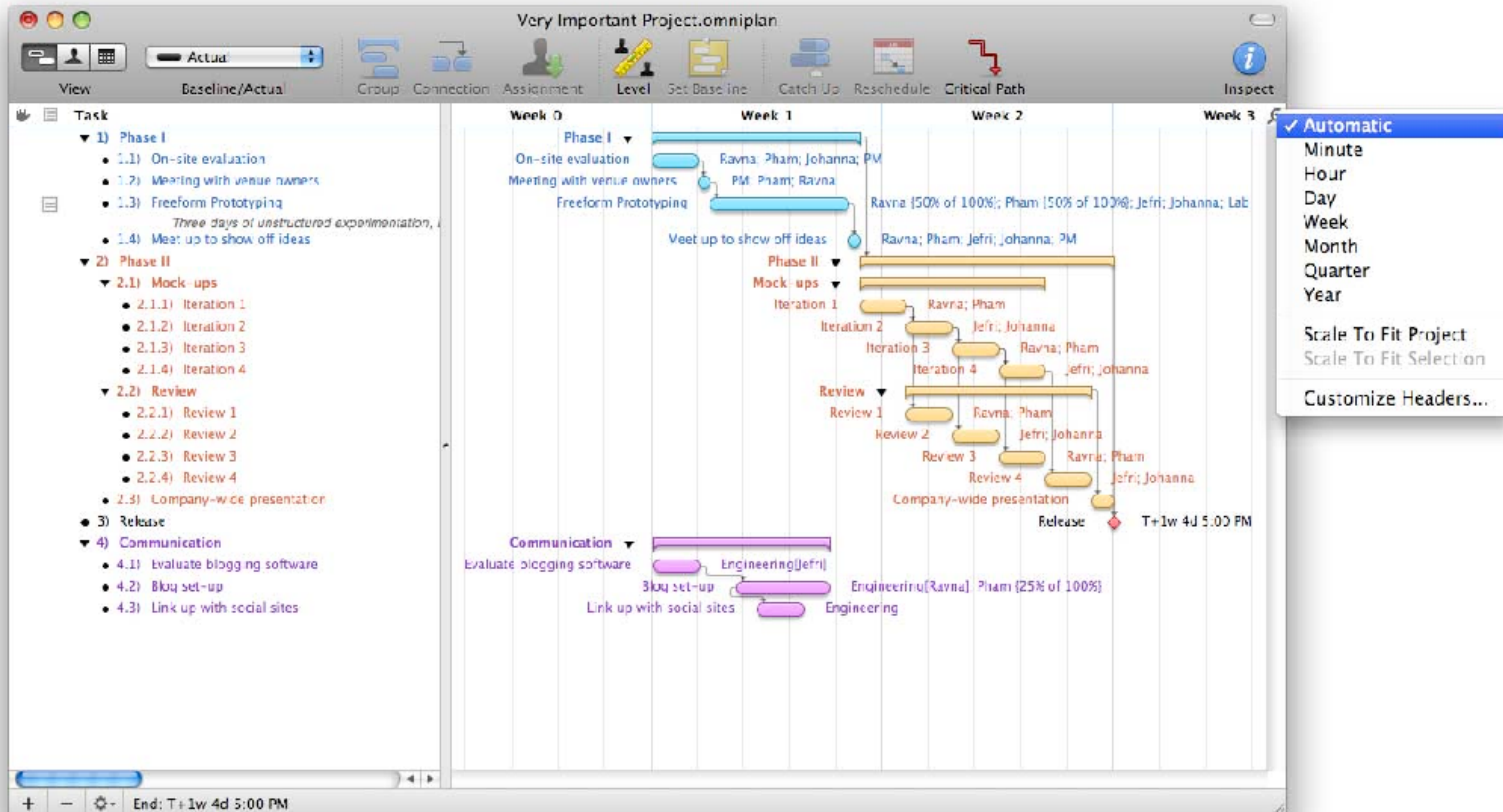
Task List

ID		Duration	Predece
1	**Software Project**	**172.5 days**	
2	Requirements	7 wks	
3	Design	5 wks	2
4	**Programming**	**60 days**	**3**
5	Unit Tests for Feature A	3 wks	3
6	Program Feature A	7 wks	5
7	Unit Tests for Feature B	4 wks	3
8	Program Feature B	8 wks	7
9	Feature-Complete Build	0 days	6,8
10	**Test Preparation**	**40 days**	
11	Build Test Plans	6 wks	2,3FF
12	Review, Correct Test Plan	2 wks	11
13	**Test Execution**	**52.5 days**	**12**
14	Execute Test Plan A	3 wks	9
15	Execute Test Plan B	1.5 wks	14SS
16	Fix Defects	1 wk	14,15

Duration

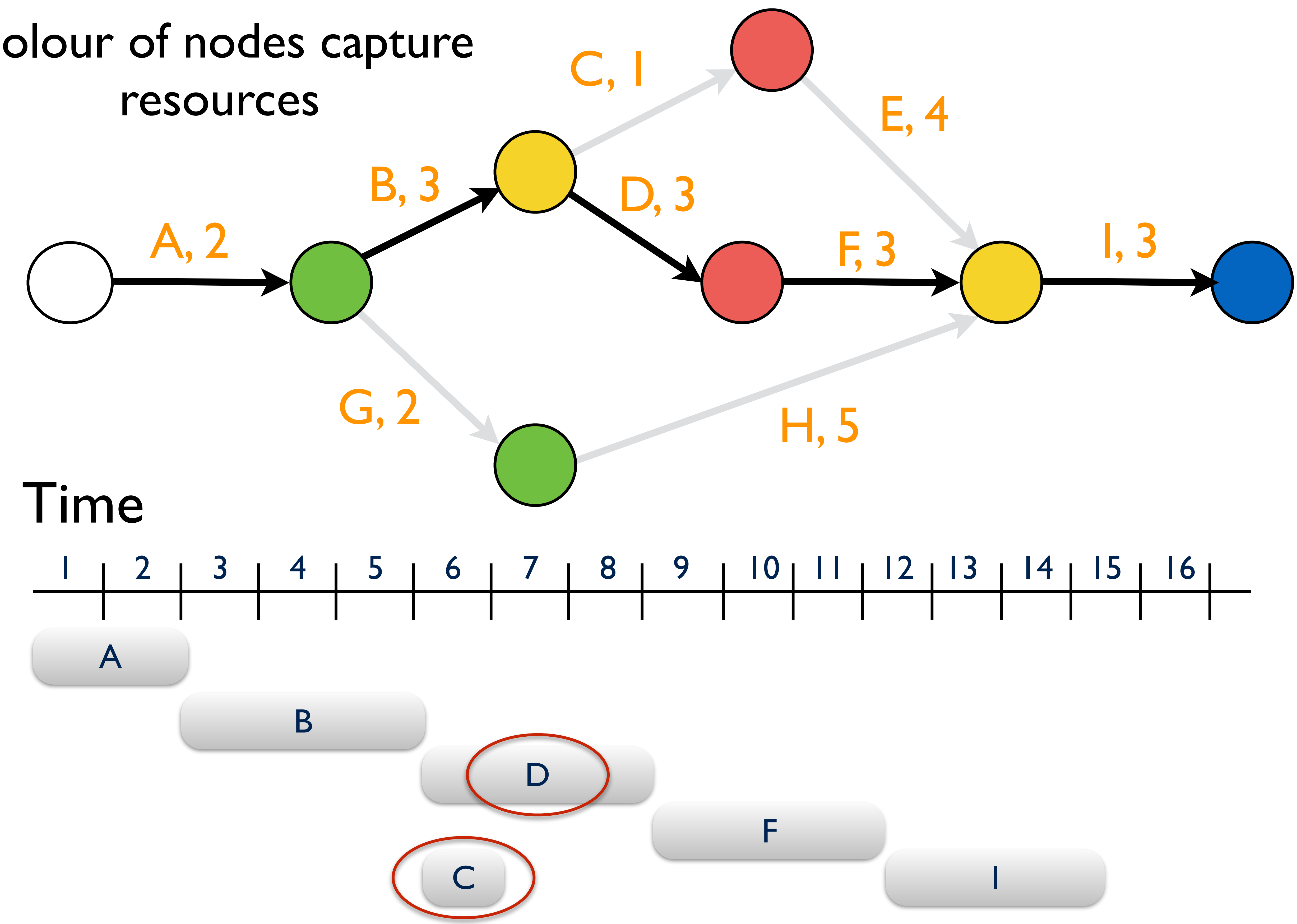


Project Planning Tools

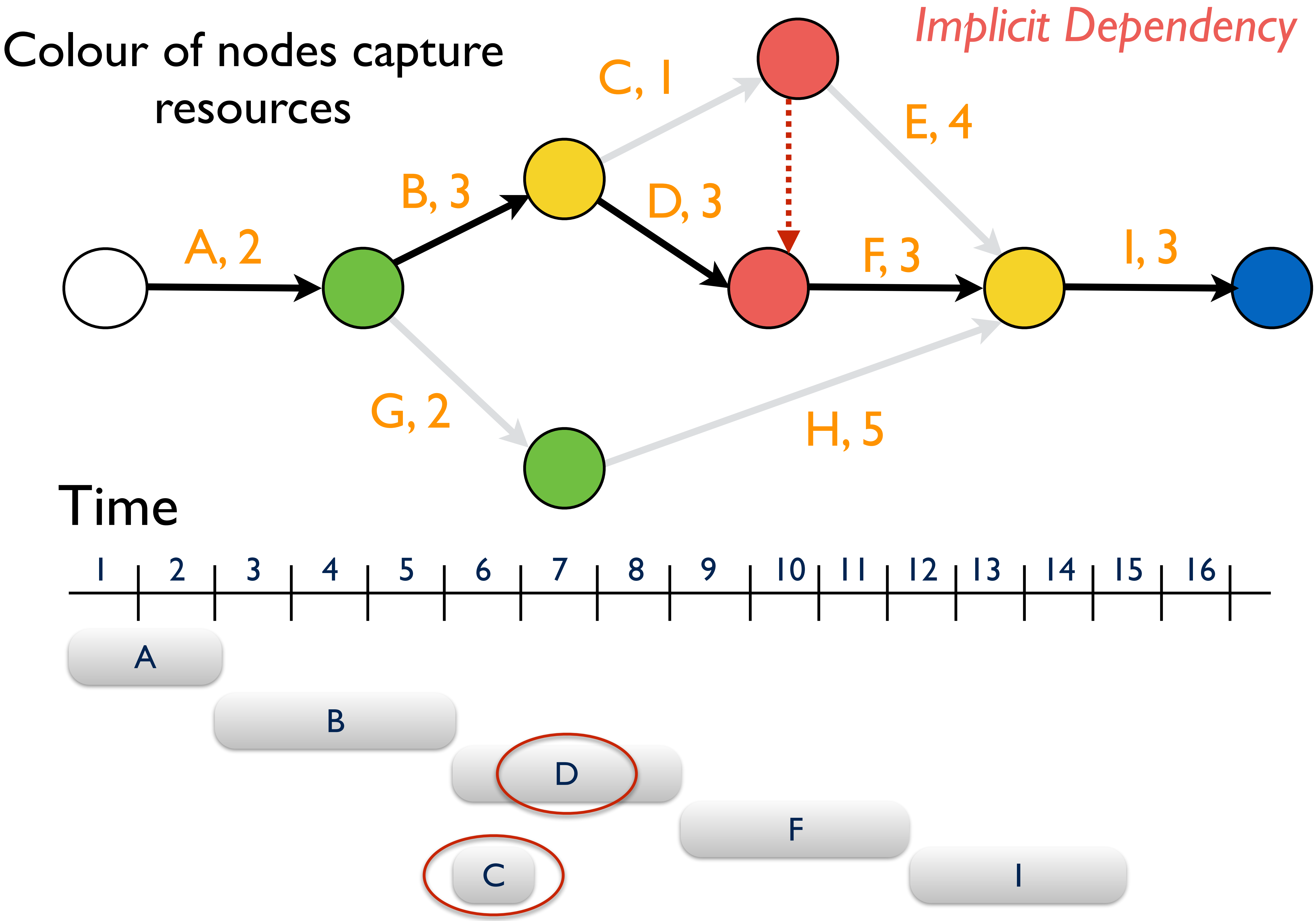


Wait!!!

Colour of nodes capture resources

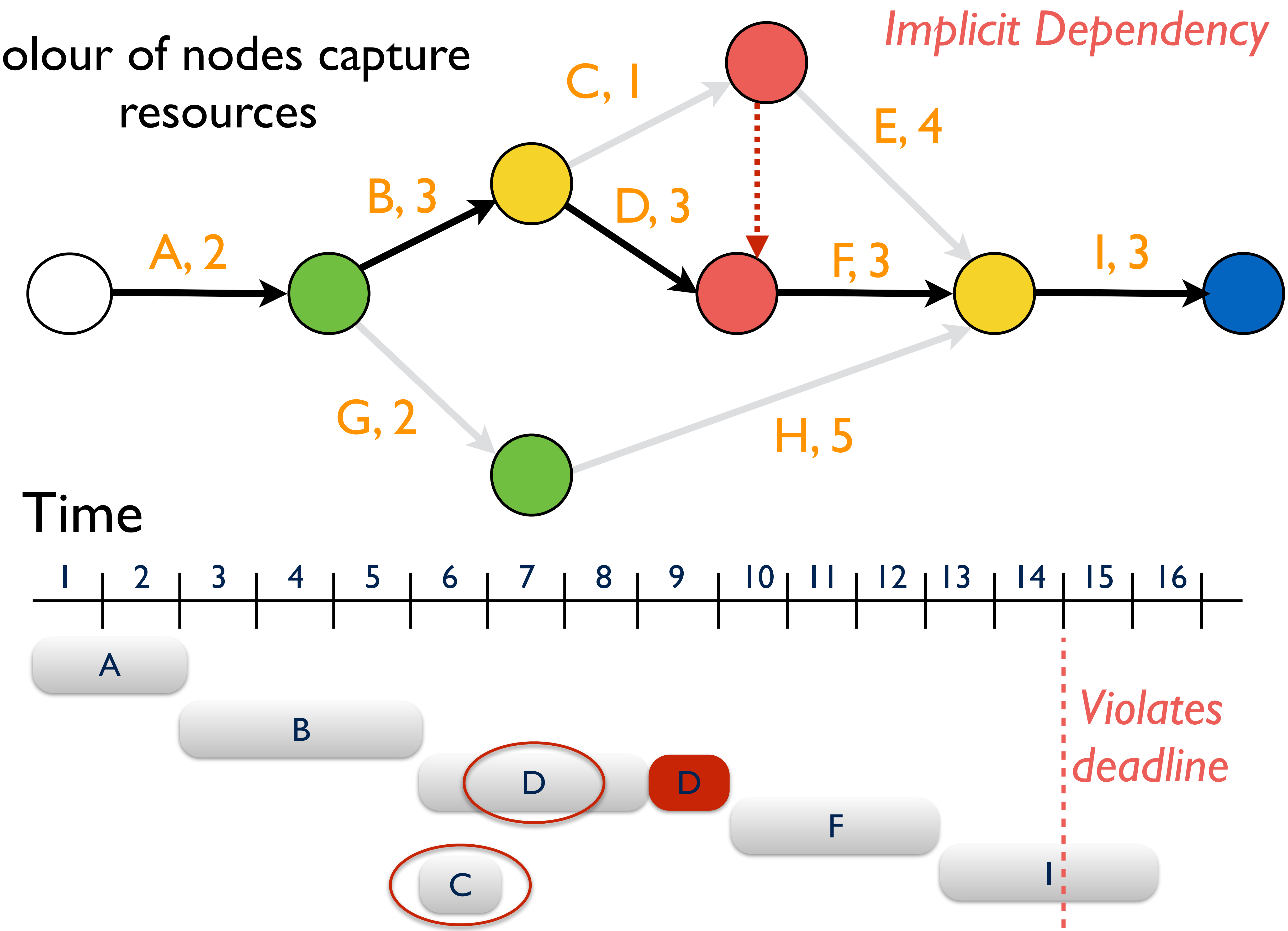


Wait!!!

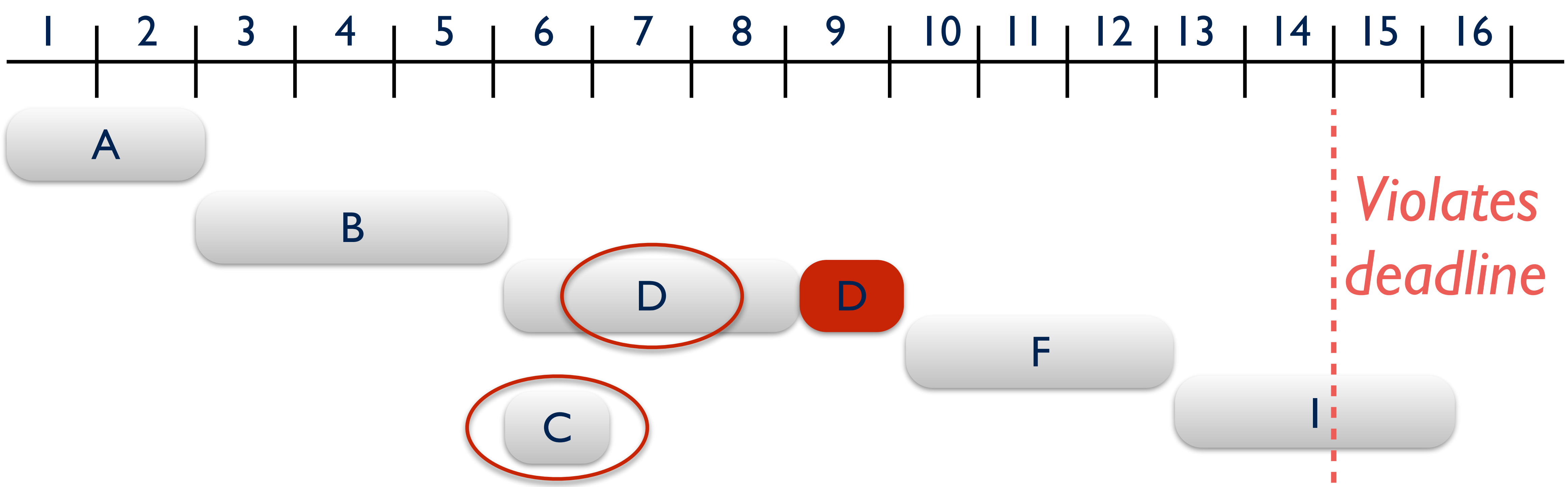

Colour of nodes capture resources
Implicit Dependency
A, 2
B, 3
C, 1
D, 3
E, 4
F, 3
G, 2
H, 5
I, 3
Time
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
A
B
D
F
C
I

Wait!!!

Colour of nodes capture resources



Critical Path



In real world

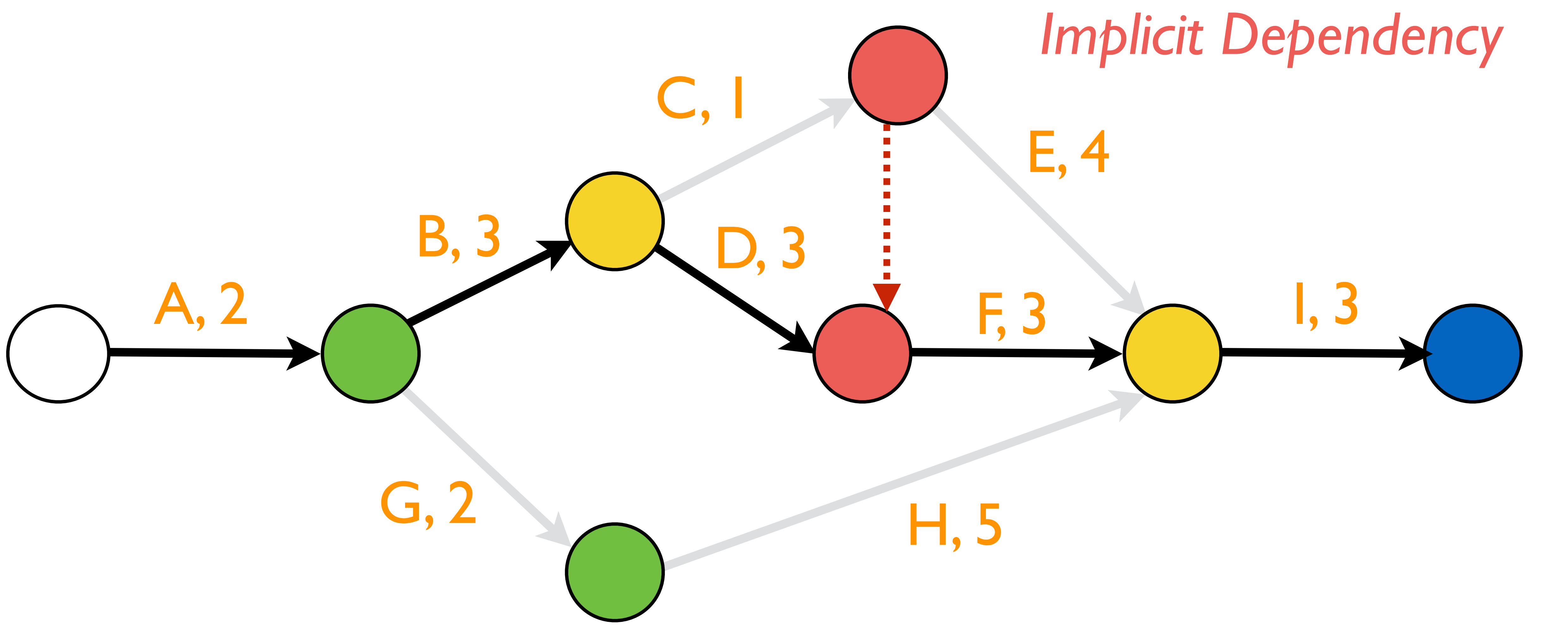
- we have finite resources
- project inevitably gets delayed
  - student syndrome (procastination)
  - murphy's law (whatever can go wrong, will)
  - parkinson's law (delaying completion of task)

Critical Chain

- Critical Chain method focuses on resources (people, equipment, physical space, etc.) required to execute project tasks.
- Insertion of *buffers* to guarantee that the project fulfils the schedule
    - Project buffer
    - Feed buffers
    - Resource buffers

Critical chain longest path considering both task and resource dependencies

Critical Path vs Chain

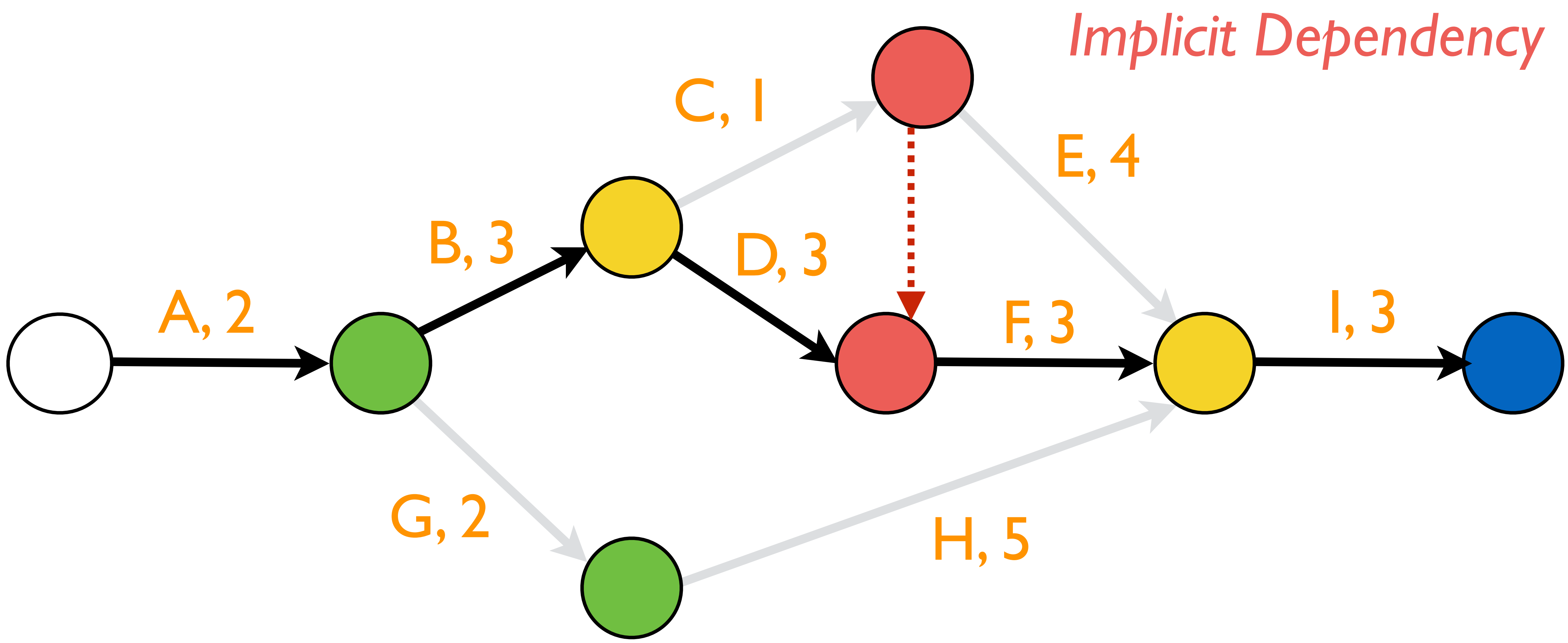


Critical path: A-B-D-F-I

Critical chain: A-B-C-D-F-I

When is critical chain the same as critical path?

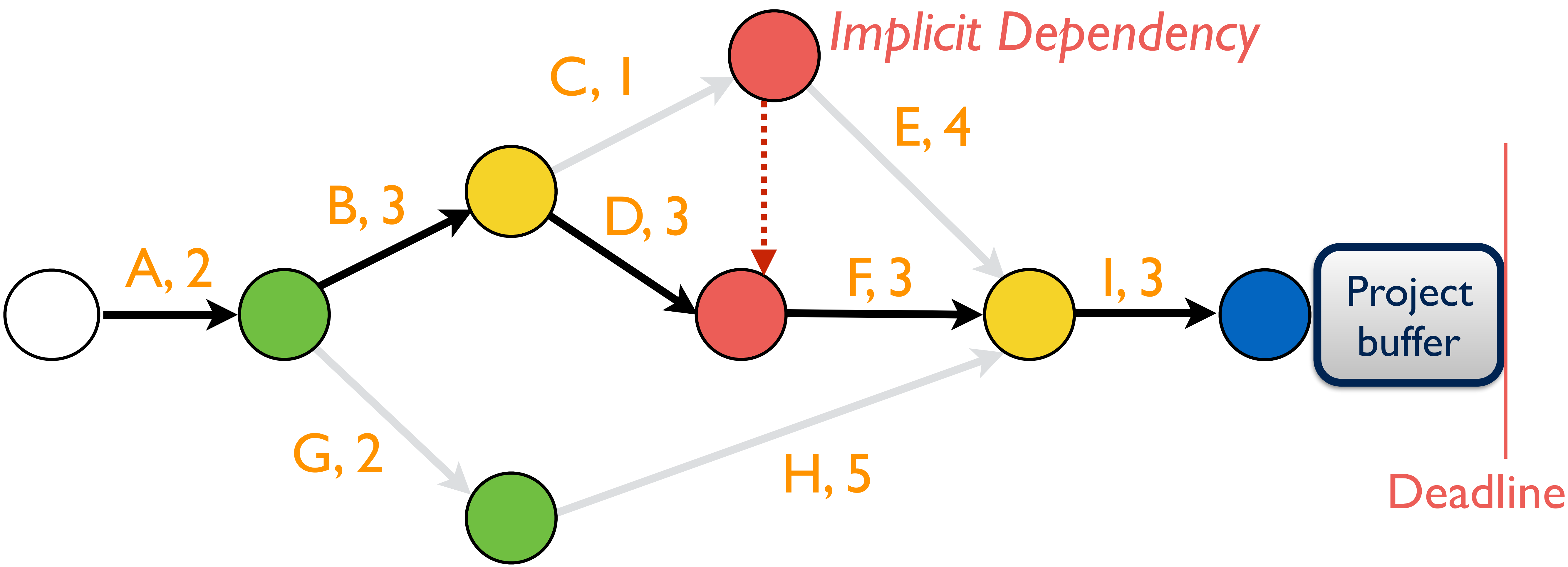
Critical Path vs Chain



Critical path: track progress of individual task

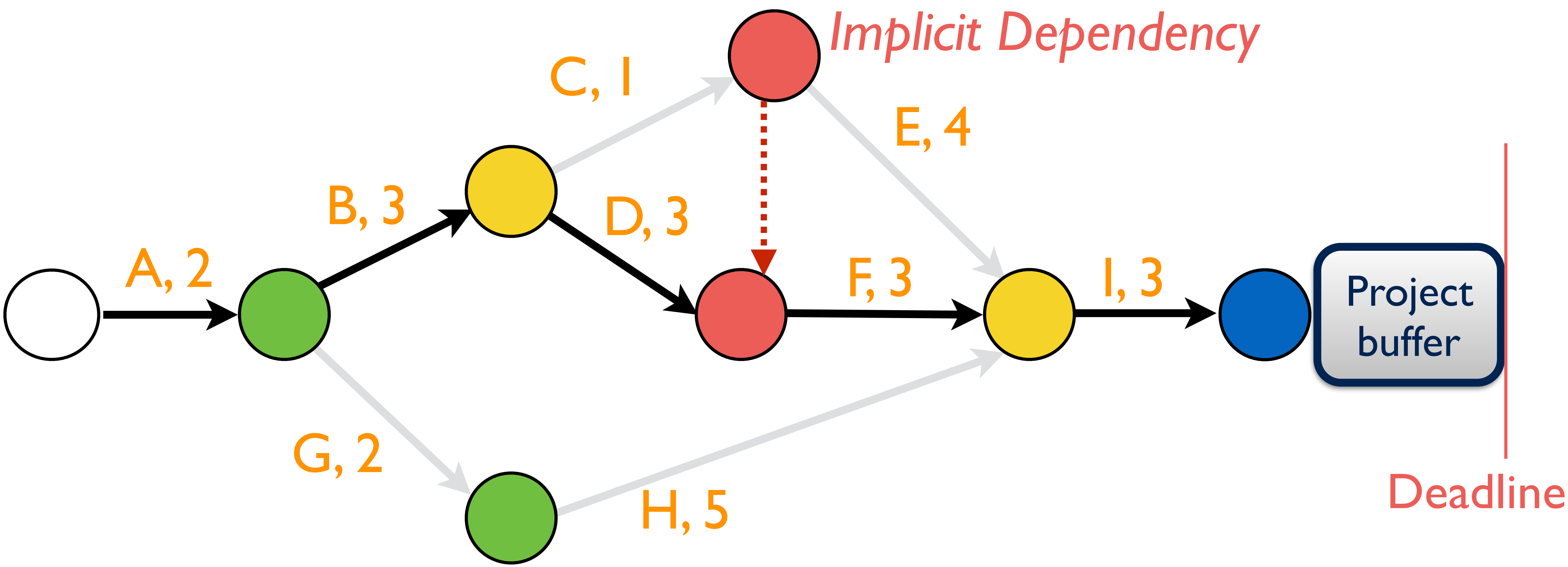
Critical chain: track progress of buffers

Critical Chain Buffers



Project buffer between the final task and deadline

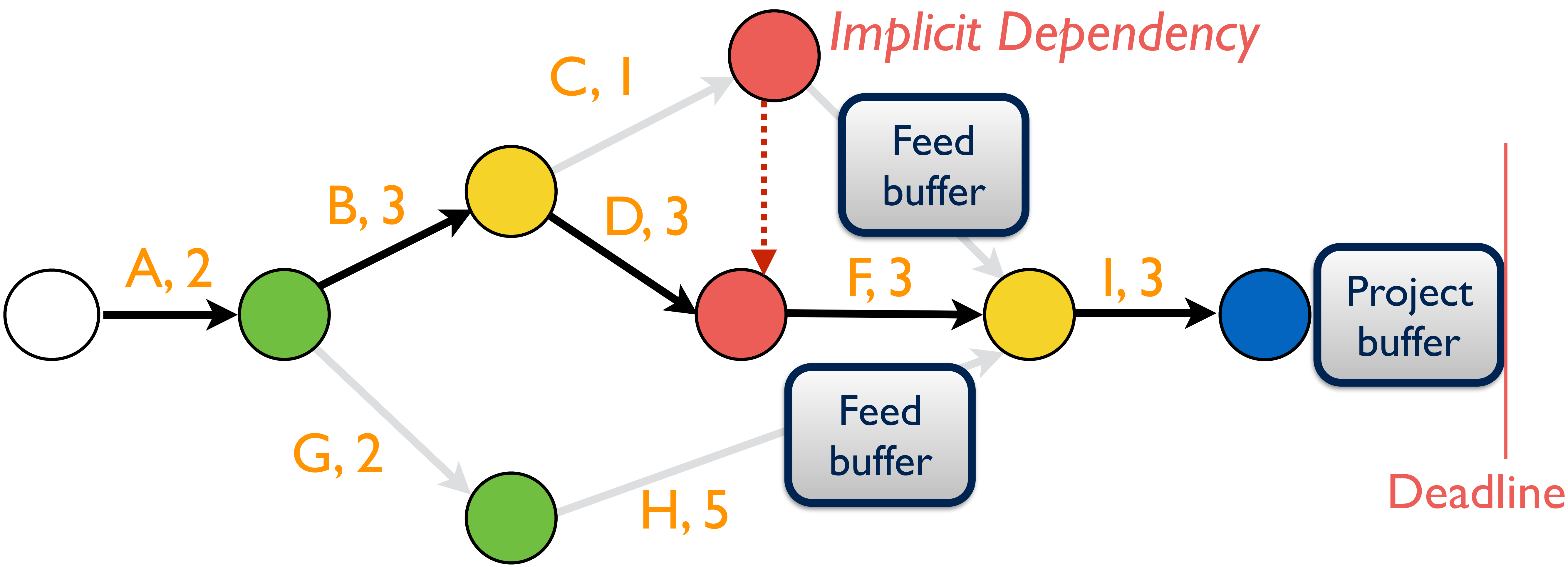
Critical Chain Buffers



Feeding chain: path of activities merging into critical chain

Feeding buffer: placed at the merge point

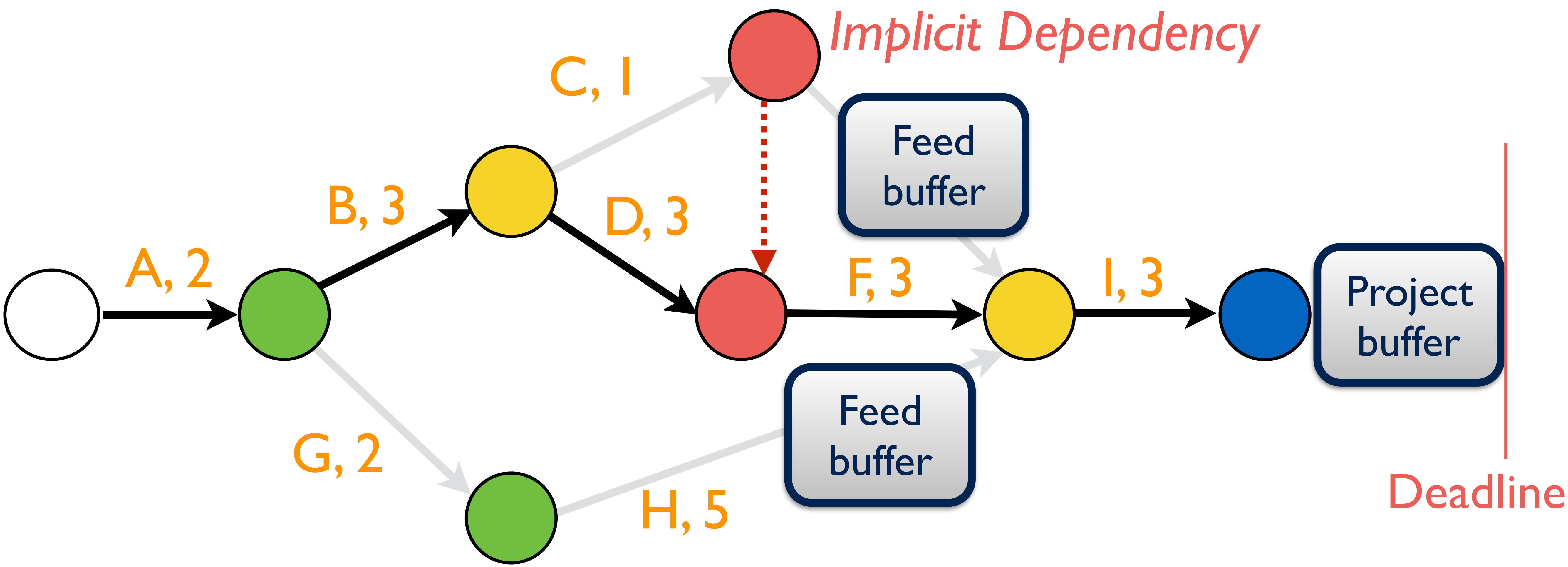
Critical Chain Buffers



Feeding chain: path of activities merging into critical chain

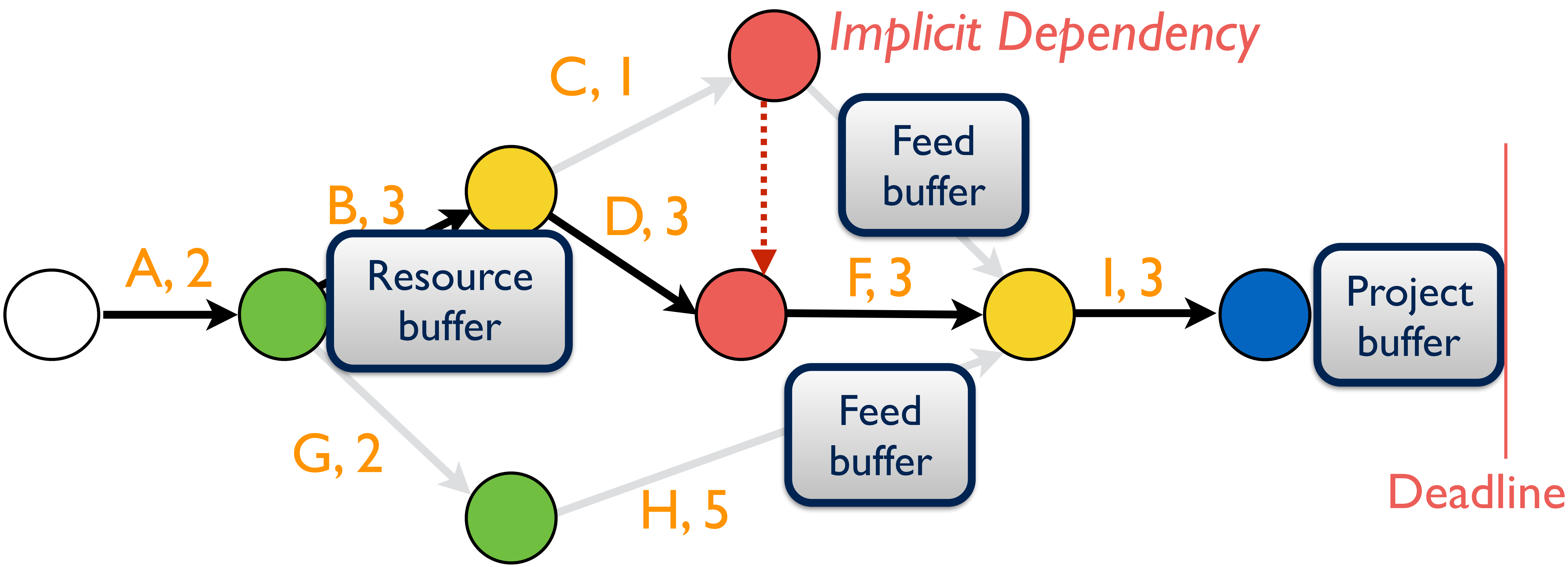
Feeding buffer: placed at the merge point

Critical Chain Buffers



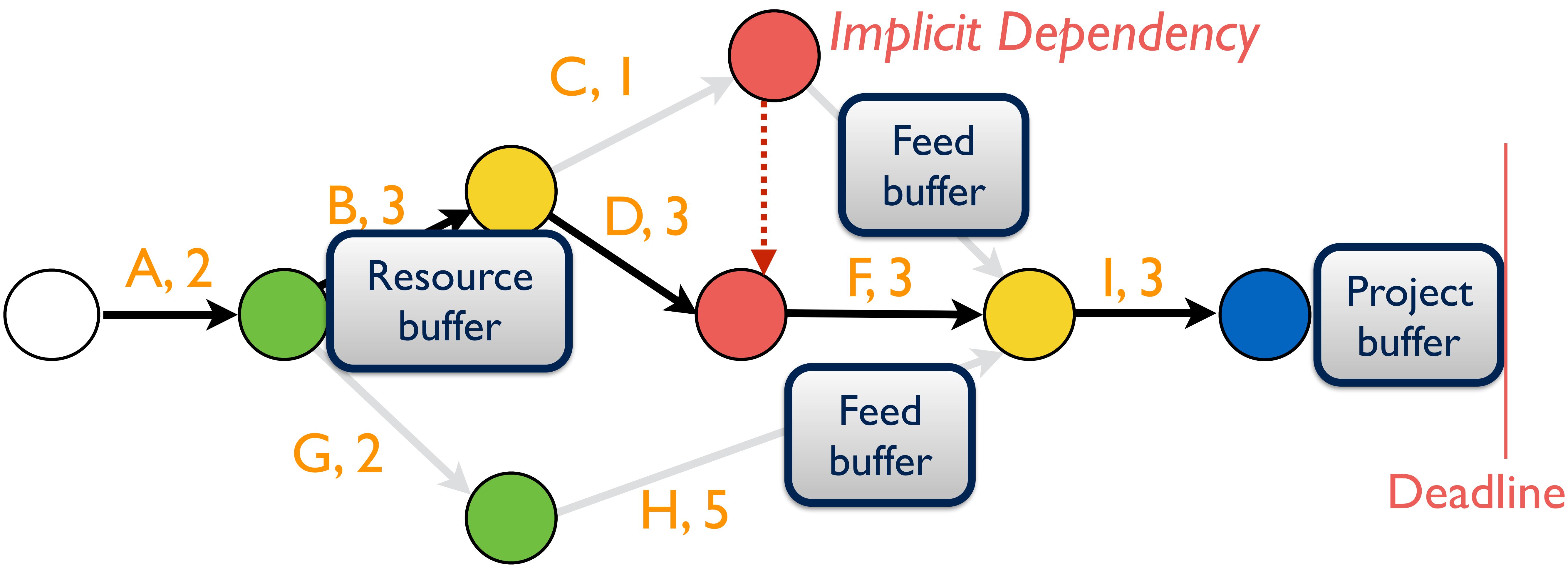
Resource buffer:
timely availability of resource in the critical chain

Critical Chain Buffers



Resource buffer:
timely availability of resource in the critical chain

Critical Chain Buffers



Critical chain: why track progress of buffers?

Risk Management

He who will not risk cannot win (John Paul Jones, 1791).

Types of Risks

- Project risks threaten the *project plan*.
  Causes project to slip and increase cost.
- Technical risks threaten the *quality* and *timeliness* of the project.
  Causes implementation to become difficult or impossible.
- Business risks threaten the *viability* of the project to be built.
  Causes project to be irrelevant or redundant.

Types of Risks

- Known risks are those that can be uncovered during careful evaluation of the project, and the business and technical environment
  (e.g. unrealistic delivery data, lack of documented requirements).
- Predictable risks can be extrapolated by past experience/projects
  (e.g. poor productivity or communication).
- Unpredictable risks are those that are difficult to identify
  (e.g. manager falls of a horse).

Risk Management

Types of Risks

- Generic risks
- Product-specific risks

Risk Management

Similar story with software development!

Risk Table

Risk	Category	Probability	Impact
Size estimate low	PS	20%	2
Change in req.	PS	45%	3
Lack of training	DE	15%	2
Staff inexperienced	ST	40%	4
Delivery deadline tightened	BU	60%	5

Impact values:

1 - catastrophic
2 - critical
3 - marginal
4 - negligible

Assessing Risk Impact

Risk Exposure (RE) = **P** x **C**

P = probability of risk
C = cost if the risk occurs

RMMM

Risk Mitigation, Monitoring & Management

- Risk avoidance (prevention better than cure)
- Risk monitoring
  - monitor and collect information for future risk analysis
- Risk management and contingency plans.
  - Risk has become a live problem

Summary

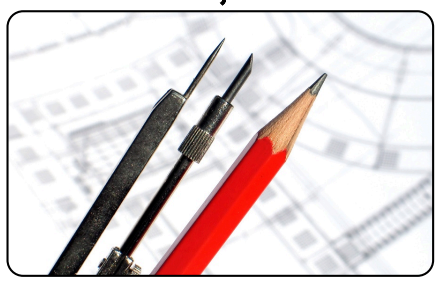
Four Ps of Project Management

People

Product

Process

Project



Software Project Management

Caper Jones

...the most interesting aspect of these six problem areas is that all are associated with project management rather than with technical personnel.

Project Scheduling

Risk Management

He who will not risk cannot win (John Paul Jones, 1791).

Further Reading