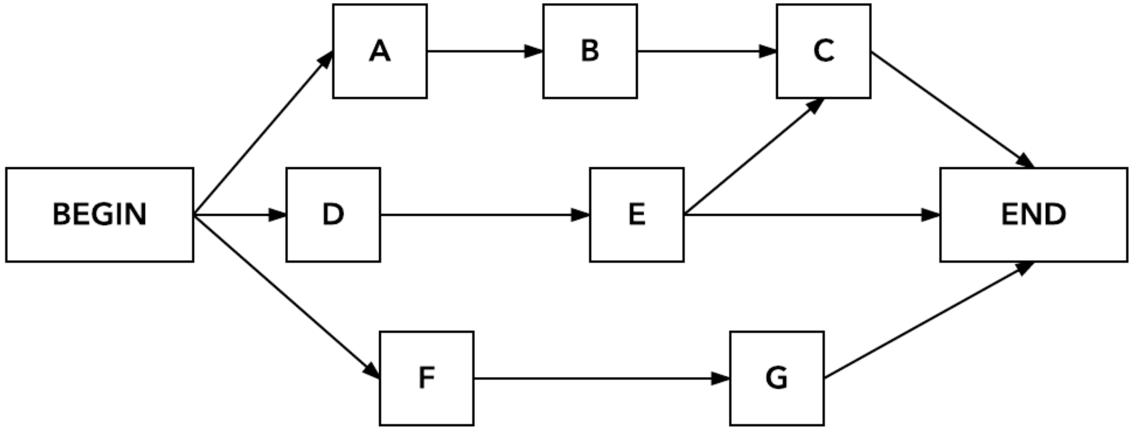
Critical Path Analysis

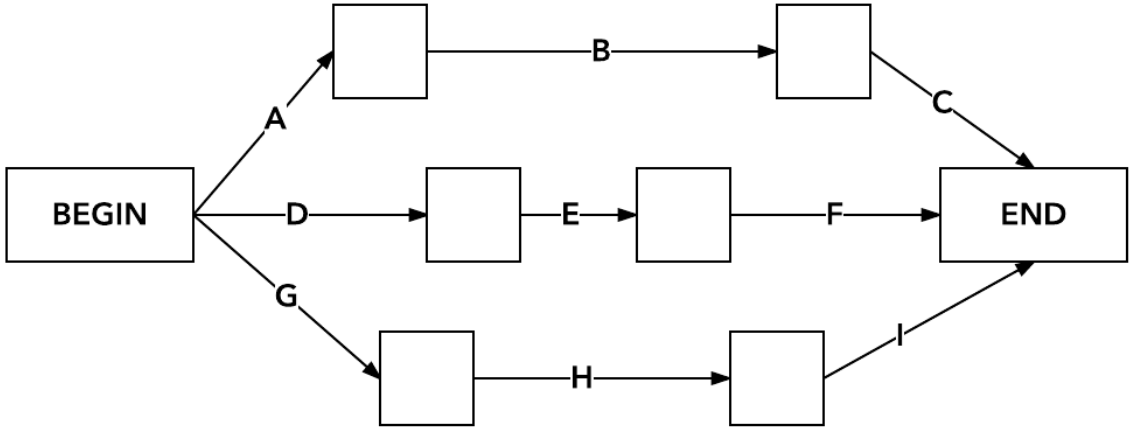
CRITICAL PATH ANALYSIS

- Critical Path Analysis (CPM) is used to schedule project tasks
  - Construction
  - Software development
- Project tasks have:
  - order
  - time and completion relationships with other tasks
  - duration
- Task schedules can be modeled using a **directed acyclic graph (DAG)**
- Project Managers usually call these activity network diagrams or schedules rather than DAGs
- An important part of the Project Management discipline (e.g. to obtain a PMP certification) and is (or should be) covered in IT343

TWO WAYS TO MODEL PROJECT TASKS WITH A GRAPH: AON AND AOA



Activity on Node

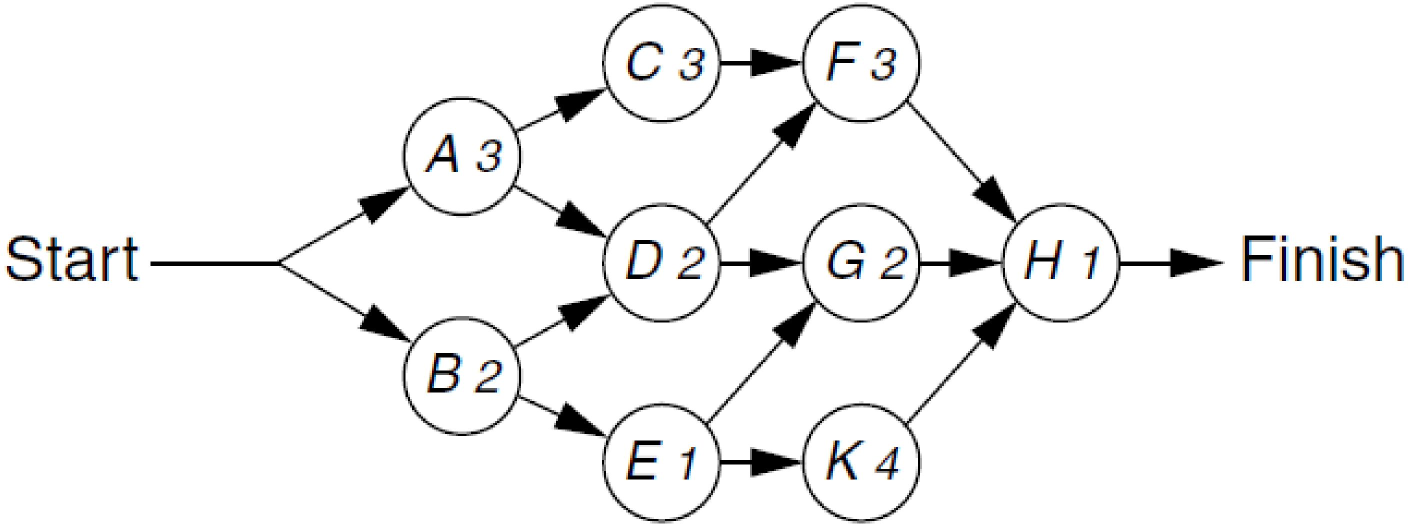


Activity on Arrow

ACTVITY-NODE GRAPH, ALSO CALLED “ACTIVITY ON NODE” (AON)

(WEISS FIG. 14.33)

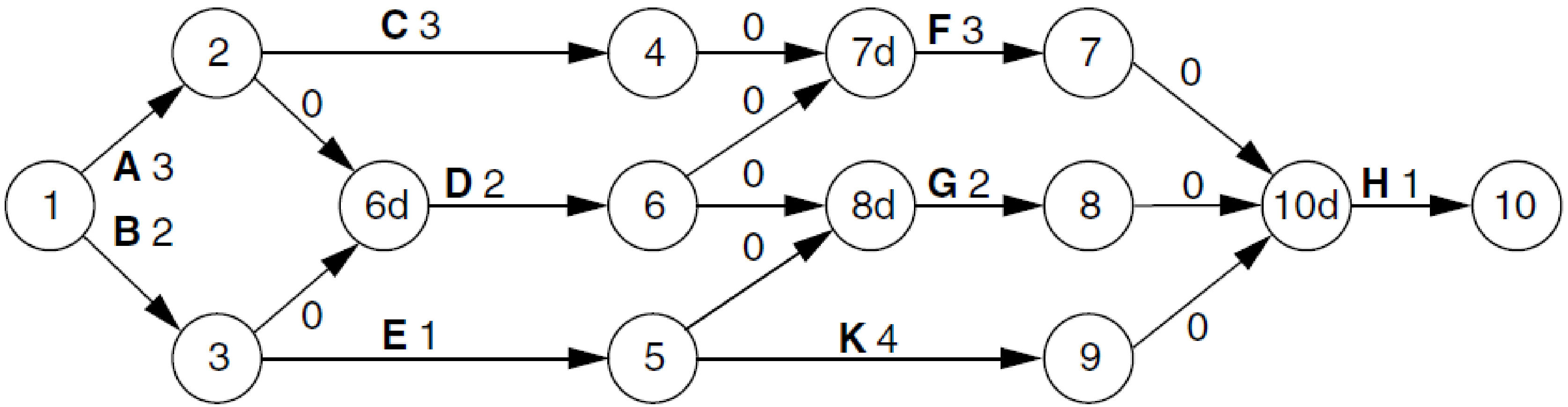
- Activities are vertices or nodes, edges are dependencies
- Letters = task identifiers, numbers = time needed to complete the activity
- Activity ‘A’ must be completed before ‘C’ can begin, ‘C’ before ‘F’, etc.
- Earliest completion time = 10 units (A-C-F-H)
- B, E, or K could be delayed by up to 2 units and not affect the earliest completion time



EVENT-NODE GRAPH, ALSO CALLED "ARROW ON ACTIVITY" (AOA)

(WEISS FIG. 14.34)

- Activities are edges, vertices start/completion milestones
- A milestone is a significant event in a project – typically completion of some activity or delivery of some component
- Project starts at MS 1, ends at MS10
- Earliest completion time = longest path from 1 to 10

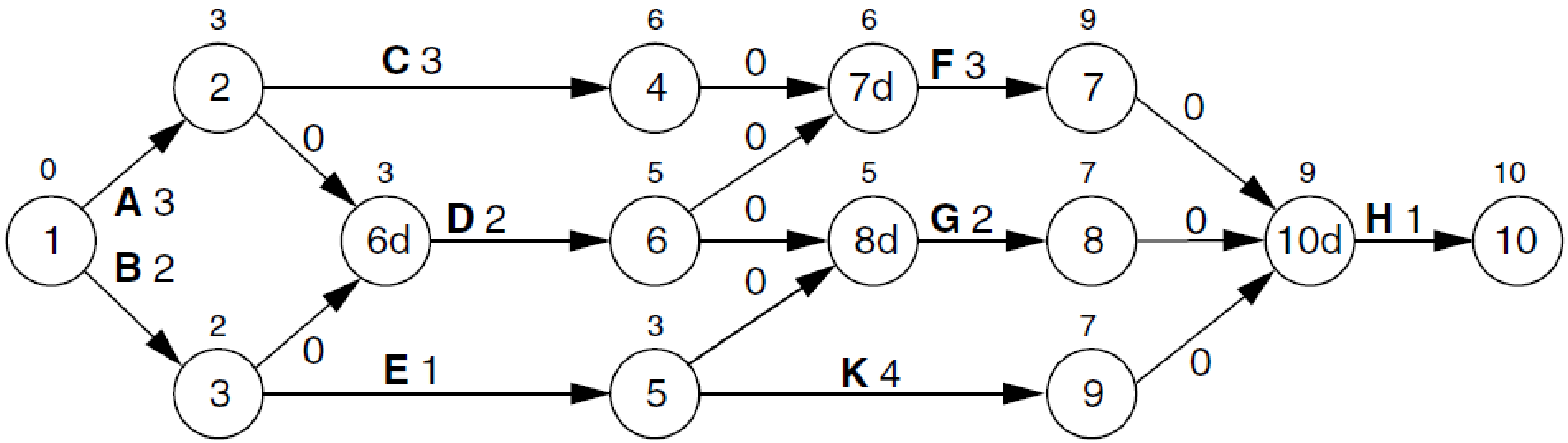


EARLIEST COMPLETION TIMES – COMPUTES THE MAXIMUM PATH LENGTH

(WEISS FIG. 14.35)

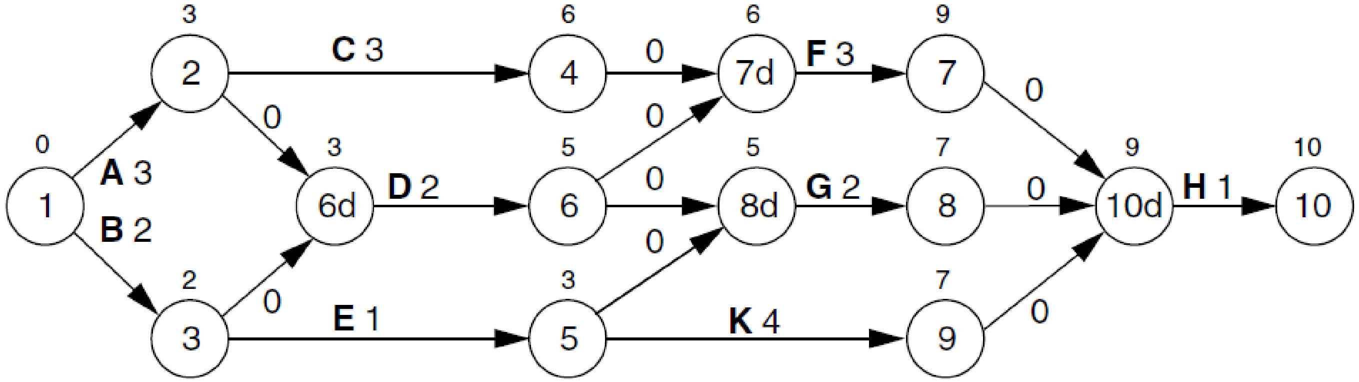
- EC_i = earliest completion time for node i based on the cost to get to node i
- EC_{10} = earliest completion time for the last node – the end of the project
- EC_{10} tells you the minimum amount of time that is required to complete the entire project
- Record each EC_i above the nodes circle in the diagram

$$EC_1 = 0 \quad \text{and} \quad EC_w = \text{Max}_{(v,\, w) \in E}(EC_v + c_{v,\, w})$$



PATHS THROUGH THE NODES

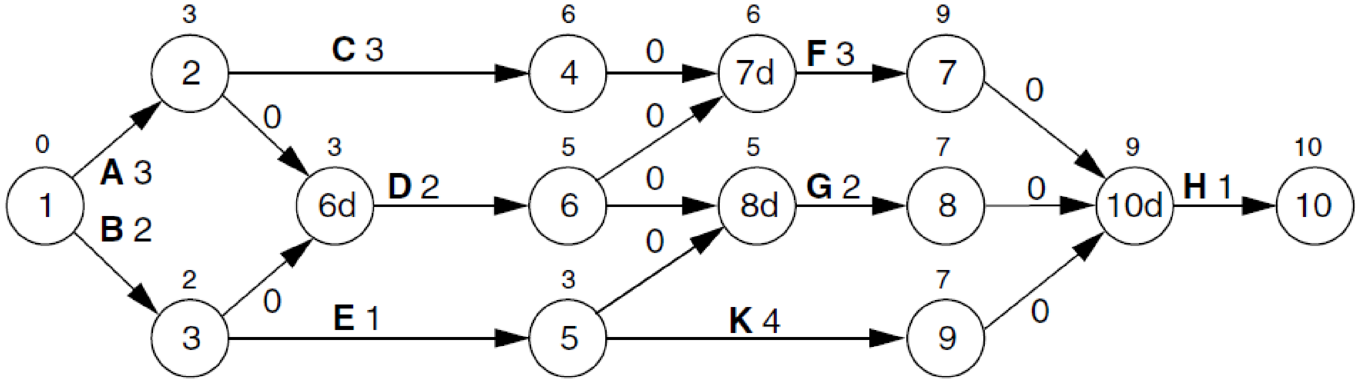
(WEISS FIG. 14.35)



#	Milestone Nodes on the Path								
1	1	2	4	7d	7	10d	10		
2	1	2	6d	6	7d	7	10d	10	
3	1	2	6d	6	8d	8	10d	10	
4	1	3	6d	6	7d	7	10d	10	
5	1	3	6d	6	8d	8	10d	10	
6	1	3	5	8d	8	10d	10		
7	1	3	5	9	10d	10			

PATHS THROUGH THE NODES AND THEIR LENGTHS – SUM THE EDGES

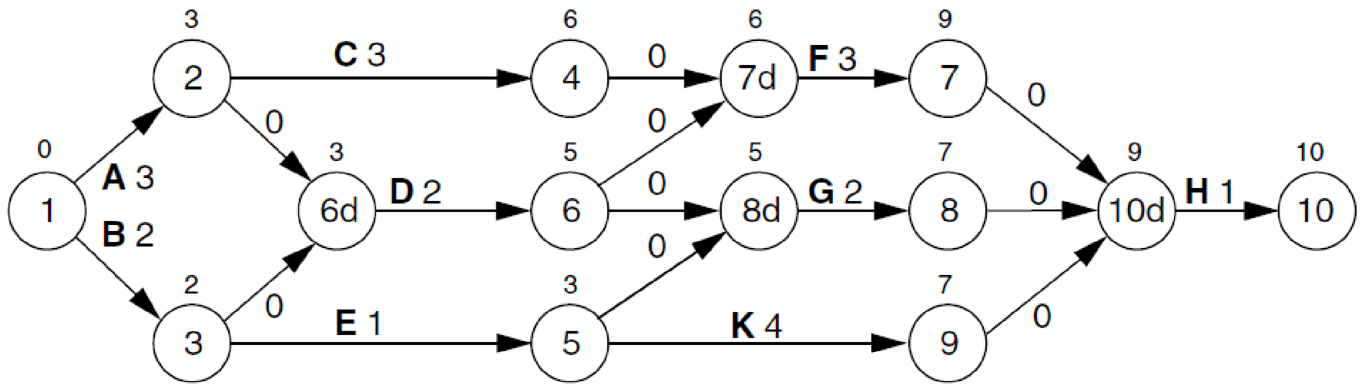
(WEISS FIG. 14.35)



#	Milestone Nodes on the Path								
1	1	2	4	7d	7	10d	10		**10**
2	1	2	6d	6	7d	7	10d	10	**9**
3	1	2	6d	6	8d	8	10d	10	**8**
4	1	3	6d	6	7d	7	10d	10	**8**
5	1	3	6d	6	8d	8	10d	10	**7**
6	1	3	5	8d	8	10d	10		**6**
7	1	3	5	9	10d	10			**8**

CRITICAL PATH – THE LONGEST PATH THROUGH THE DIAGRAM

(WEISS FIG. 14.35)



#	Milestone Nodes on the Path								
1	1	2	4	7d	7	10d	10		**10**
2	1	2	6d	6	7d	7	10d	10	**9**
3	1	2	6d	6	8d	8	10d	10	**8**
4	1	3	6d	6	7d	7	10d	10	**8**
5	1	3	6d	6	8d	8	10d	10	**7**
6	1	3	5	8d	8	10d	10		**6**
7	1	3	5	9	10d	10			**8**

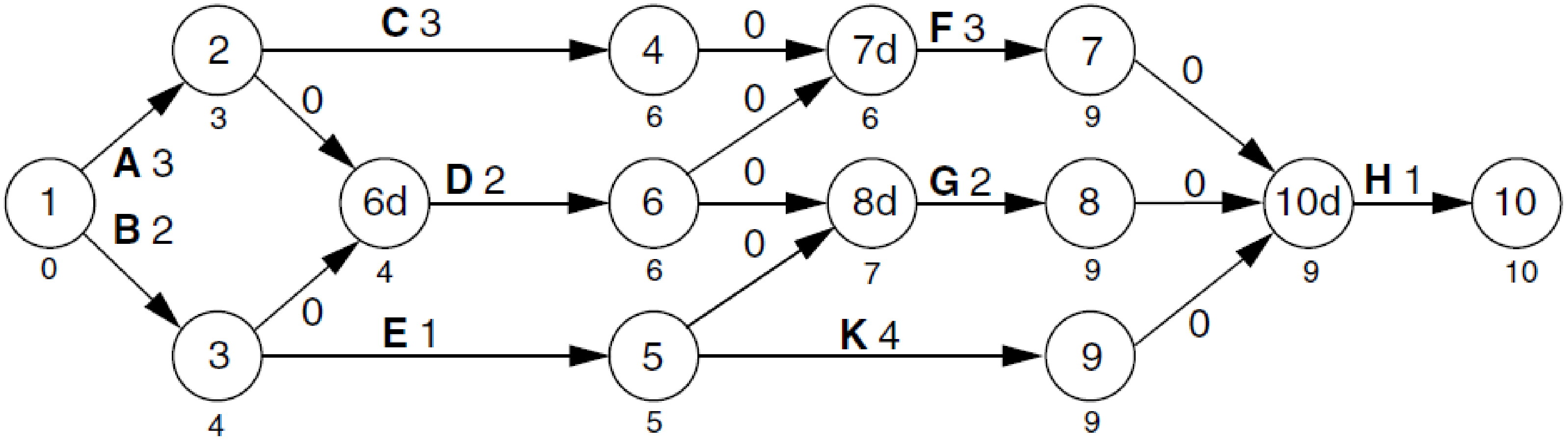
Earliest completion time is 10 – "the long pole in the tent"

This sequence of activities is called the **Critical Path**

LATEST COMPLETION TIMES

(WEISS FIG. 14.36)

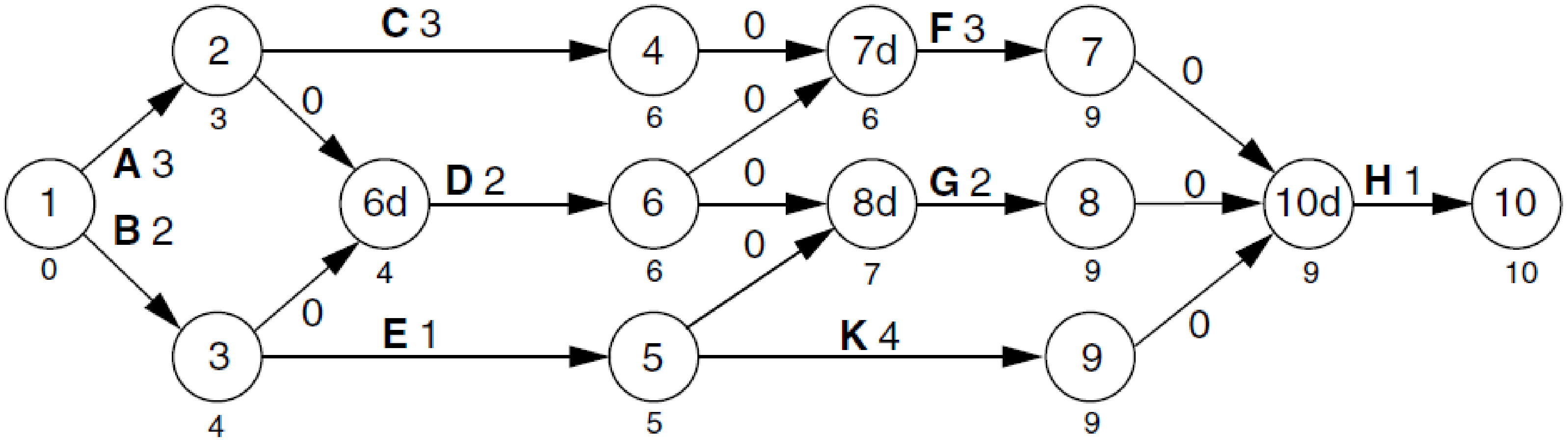
- LC_i = latest completion time event i can finish without affecting the final completion time
- Nodes not on the critical (longest) path can finish later than their earliest date without delaying the entire project



LATEST COMPLETION TIMES – HOW TO COMPUTE
(WEISS FIG. 14.36)

- Start with the last (ending) node and work backwards in the diagram
- For the last node, LC_{last} = the critical path size, or $LC_N = EC_N$ (= LC_{10} = 10)
- To compute LC for each previous/predecessor of node ‘w’ subtract from node w’s LC the cost of moving from the predecessor node to node ‘w’
- For predecessor nodes having more than one successor node use the minimum value as its LC

$$LC_N = EC_N \quad \text{and} \quad LC_v = \mathrm{Min}_{(v,\,w)\,\in\,E}(LC_w - c_{v,\,w})$$



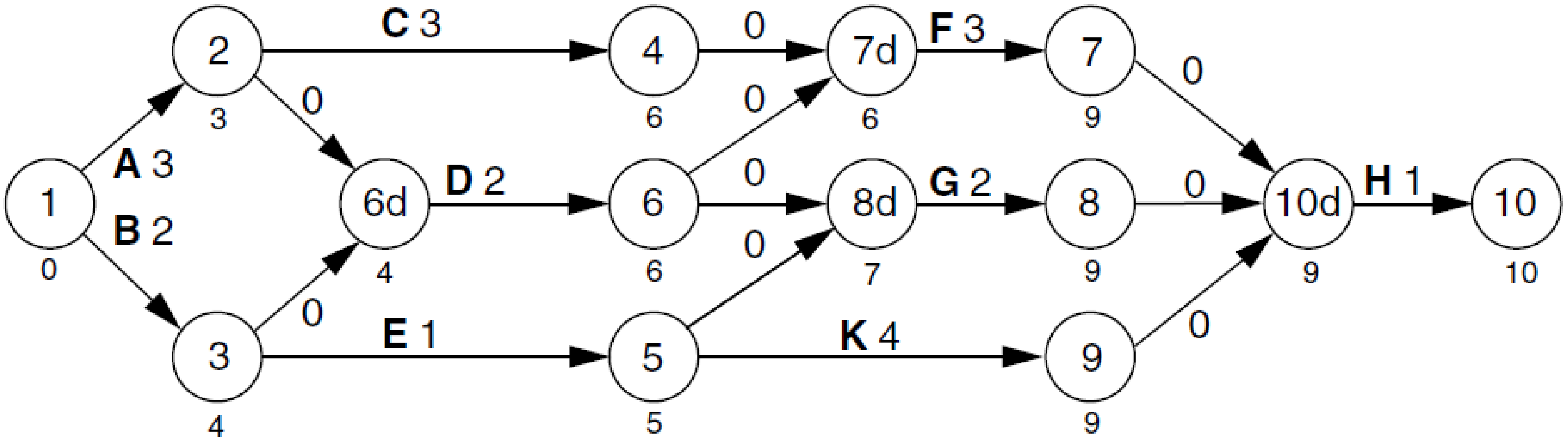
LATEST COMPLETION TIMES – HOW TO COMPUTE
(WEISS FIG. 14.36)

- Record each LC_i below the node's circle in the diagram, example computations:

$LC_{10d} = LC_{10} - c(10d, 10) = 10 - 1 = 9$

$LC_5 = \text{Min}\,[(LC_{8d} - c(5, 8d)), (LC_9 - c(5, 9))] = \text{Min}[(7 - 0), (9 - 5)] = [7, 5] = 5$

$$LC_N = EC_N \quad \text{and} \quad LC_v = \text{Min}_{(v,\,w)\,\in\,E}(LC_w - c_{v,\,w})$$

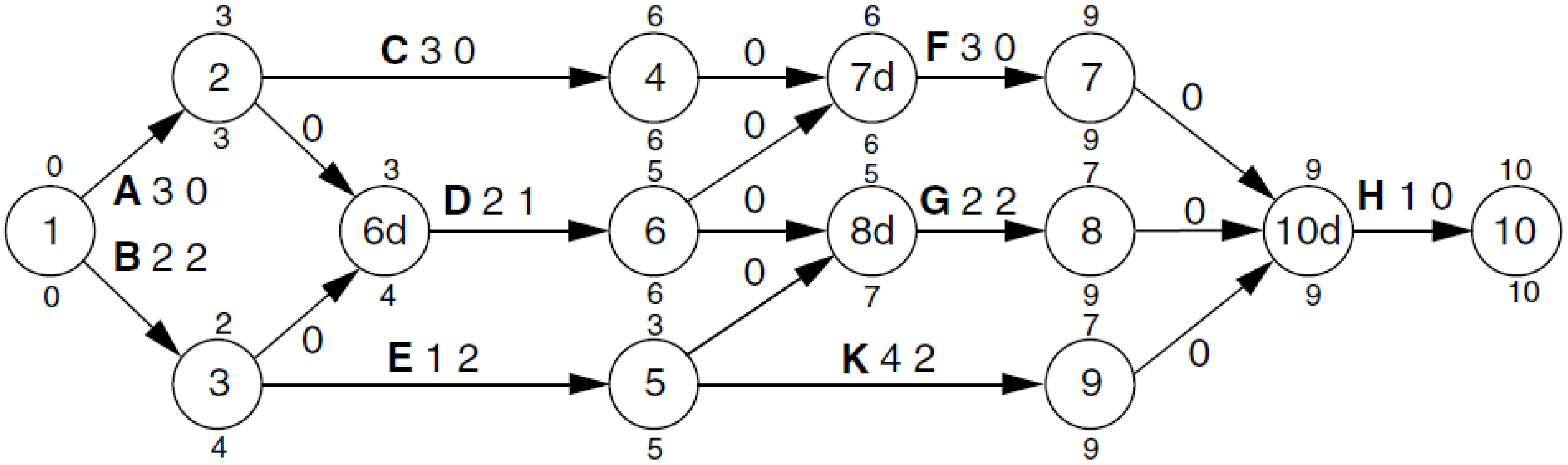


EARLIEST/LATEST COMPLETION AND SLACK TIMES

(WEISS FIG. 14.37)

- Slack time for an edge (activity) is the amount of time the completion of that activity can be delayed without delaying the overall project completion
- It’s the difference between the activity’s lattest and earliest completion dates

$$\text{Slack}_{(v,\, w)} = LC_w - EC_v - c_{v,\, w}$$

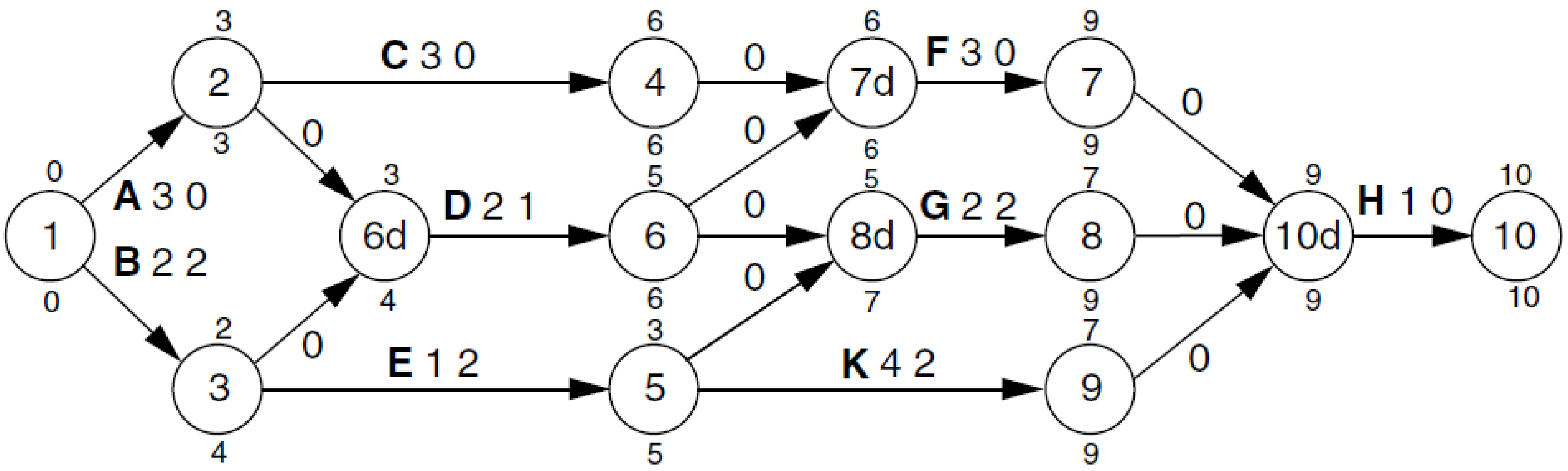


EARLIEST/LATEST COMPLETION AND SLACK TIMES

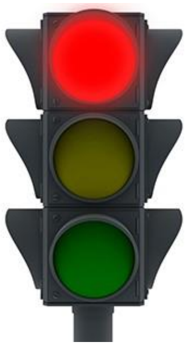
(WEISS FIG. 14.37)

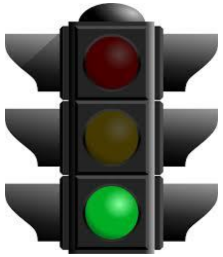
- EC is recorded above the diagram, LC below it
- The slack for node 9 is 9 – 7 = 2, for node 6 it's 6 – 5 = 1
- Slack for all nodes on the critical path is 0 since they cannot be delayed without delaying the entire project
- Record the amount of slack as the second number after the task/activity id

$$\text{Slack}_{(v,\,w)} = LC_w - EC_v - c_{v,\,w}$$



End of Critical Path Analysis





Please proceed to the next course activity now or at a later time