CSE307: Software Engineering

Project Scheduling and Earned-value Management

Project Scheduling

Planning

- The *Bad News* is - time and tide waits for none :(
- The *Good News* is - you can be the pilot in your project :)
- Planning must start immediately from the first day of a project.
- But, we may not have enough information to devise a good plan in the beginning.
- What we can do is - we can make a working plan and later, revise that plan as the project progresses.

Planning: Things to Know

- While making a plan for a software project, we need to take into account the following scopes.
    - Requirements from client for the product.
    - Scale and features of the product.
    - Performance criteria of the product.
- These scopes should be clearly and easily understandable to different participants in the project.

Planning: Things to Do

- After defining the scopes of a software project, we need to decompose the tasks associated into smaller subtasks.
- This decomposition process continues until all the subtasks are properly and clearly defined.
- That is, each of the subtasks must have a clear description and an achievable goal, given the limited resources and time.

Scheduling

- Scheduling is one of the first and foremost things to do in any project.
- After the tasks decomposition, we need to make a schedule for the project.
- Each subtasks in a project should be scheduled with adequate information - including starting date, duration, assigned resources, and dependencies.
- We may use historical data from past projects to make a schedule.

Scheduling: Steps

- Scheduling involves the following activities.

1. Identify all the tasks in a project.
2. Identify all the *dependencies* among those tasks.
3. Break down all the large tasks into smaller subtasks, if necessary.
4. Assign resources to each of the tasks.
5. *Level resources*, if necessary.

Scheduling: Rules of Thumb

- Only one person from a project team should always edit the schedule.
- Schedule should be simple and useful with appropriate amount of information.
- We always should *level resources* while making schedule.

Scheduling: Classic Mistakes

- Making an overly optimistic schedule.
- Failing to monitor and update schedule.
- Adding human resources (aka people) to a late project.
- Leaving out a task while scheduling.

Scheduling: Terminology

- **Critical Path:** Sequence of tasks in a project that forms the longest path, in terms of completion of the project. Delay in any of the tasks in this sequence will delay the overall completion of the project.

- **Slack:** Amount of time a task can be delayed without delaying the overall completion of the project.

- **Milestone:** An important date in the project schedule.

- **Dependency:** Relationship among different tasks in a project.

Scheduling: Dependencies

- There may exist 4 (four) different categories of dependency between a pair of tasks in a project.

1. <u>Finish to Start (FS):</u>

- We need to *finish* a task (A) in order to *start* another task (B).
- Depicted by the expression **A FS B**, in this case.
- Example can be - **building wall FS painting wall**.
- This is the most common dependency that may exist between a pair of tasks in a project.

Scheduling: Dependencies

- There may exist 4 (four) different categories of dependency between a pair of tasks in a project.

  2. <u>Finish to Finish (FF):</u>
  - We need to *finish* a task (A) in order to *finish* another task (B).
  - Depicted by the expression **A FF B**, in this case.
  - Example can be - **stopping engine FF stopping car**.

Scheduling: Dependencies

- There may exist 4 (four) different categories of dependency between a pair of tasks in a project.

  3. <u>Start to Finish (SF):</u>
  - We need to *start* a task (A) in order to *finish* another task (B).
  - Depicted by the expression **A SF B**, in this case.
  - Example can be - **starting night-shift SF ending day-shift in a hospital**.

Scheduling: Dependencies

- There may exist 4 (four) different categories of dependency between a pair of tasks in a project.

  4. <u>Start to Start (SS):</u>
  - We need to *start* a task (A) in order to *start* another task (B).
  - Depicted by the expression **A SS B**, in this case.
  - Example can be - **starting engine SS starting car**.

Resource Leveling

- Resource leveling is a process to evaluate a project for an unbalanced usage of project resources and resolve conflicts as well as overallocation.

- Conflict or overallocation may happen when multiple tasks are scheduled at the same time for the same person.

- Resources may be levelled by introducing fake dependency between a pair of parallel tasks, to make them sequential, or by splitting resource usage among different concurrent tasks.

- But, these resolutions may delay the overall completion of the project.

Critical Path Method (CPM)

- Critical path method (CPM) is a method that allows us to identify essential tasks for the completion of a project.

- Critical path is the longest sequence of tasks in a project that must be completed on time in order to finish the overall project within a strict deadline.

Critical Path Method (CPM)

- On the other hand, a PERT (Project Evaluation and Review Technique) chart, also known as PERT diagram, is a project management tool used by a project team to schedule and coordinate tasks within a project.

- It is a graphical representation of the timeline of a project, enabling a project team to further analyze the project schedule.

- Now, we will see how we can construct a PERT chart, from given information on a particular project, and further analyze the diagram, by applying CPM, to get useful information about the project.

Critical Path Method (CPM)

- There are 6 (six) steps involved in critical path method to analyze a project.

1. Specify the individual tasks in a project.

2. Determine the FS dependencies among different tasks.

3. Construct a PERT diagram.

4. Compute *early start* (ES) and *early finish* (EF) of each of the tasks **(forward pass)**.

5. Compute *late finish* (LF) and *late start* (LS) of each of the tasks **(backward pass)**.

6. Determine the critical path of the project.

CPM: Forward Pass

- **Early start (ES)** means the earliest time at which a particular task in a project can be started given that all of its preceding tasks have already been completed.

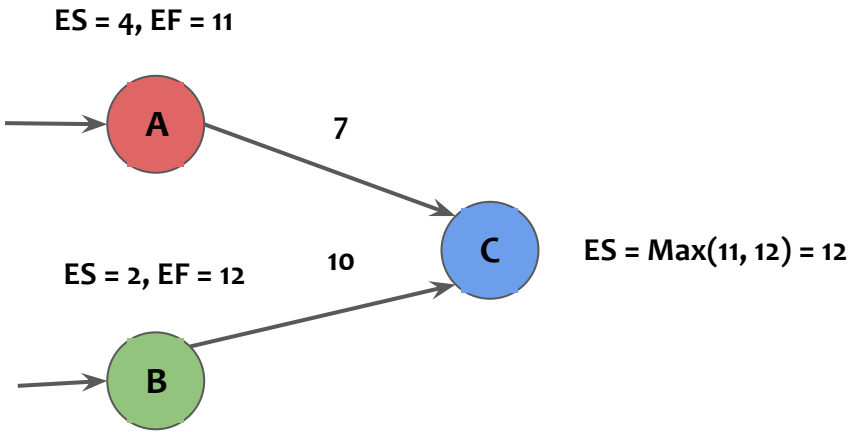
- **Early finish (EF)** means the earliest time at which a particular task in a project can be completed.

- EF of a task = ES of that task + Duration of that task.

- We determine the ES and EF of different tasks in a PERT diagram from left (source nodes) to right (sink node) in forward pass, *assuming there may be one or multiple starting tasks but only one ending task in a project*.

CPM: Forward Pass

- ES = 0 (zero) for starting tasks (corresponding to source nodes) in PERT diagram.
- ES = Max(EF of preceding tasks) for other tasks in PERT diagram.

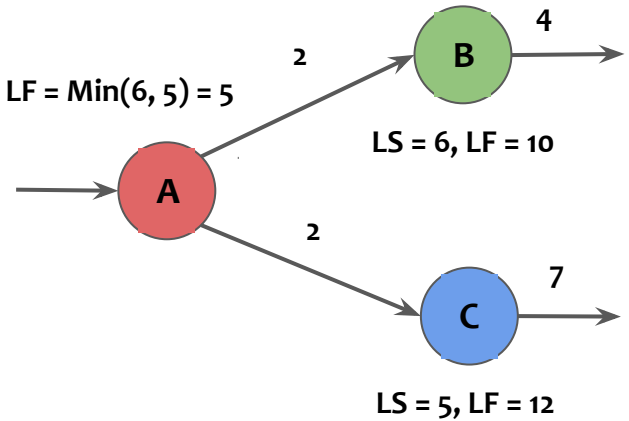


CPM: Backward Pass

- **Late finish (LF)** means the latest time at which a particular task in a project can be completed without delaying the overall completion of the project.
- **Late start (LS)** means the latest time at which a particular task in a project can be started without delaying the overall completion of the project.
- LS of a task = LF of that task - Duration of that task.
- We determine the LF and LS of different tasks in a PERT diagram from right (sink node) to left (source nodes) in backward pass.

CPM: Backward Pass

- LF = EF for ending task (corresponding to sink node) in PERT diagram.
- EF, as well as LF, of ending task is the *strict deadline* of a project.
- LF = Min(LS of succeeding tasks) for other tasks in PERT diagram.



CPM: Determining Slack and Critical Path

- We can determine the slack of each of the tasks in a PERT diagram either by subtracting its ES from its LS or by subtracting its EF from its LF.
- That is, slack of a task = LS of that task - ES of that task.
- Or, slack of a task = LF of that task - EF of that task.
- Critical tasks in a PERT diagram are tasks with 0 (zero) slack.

CPM: Determining Slack and Critical Path

- The path from source node to sink node on which each of the nodes corresponds to a critical task is called a critical path in PERT diagram.

- There should be at least 1 (one) critical path in a PERT diagram of a project.

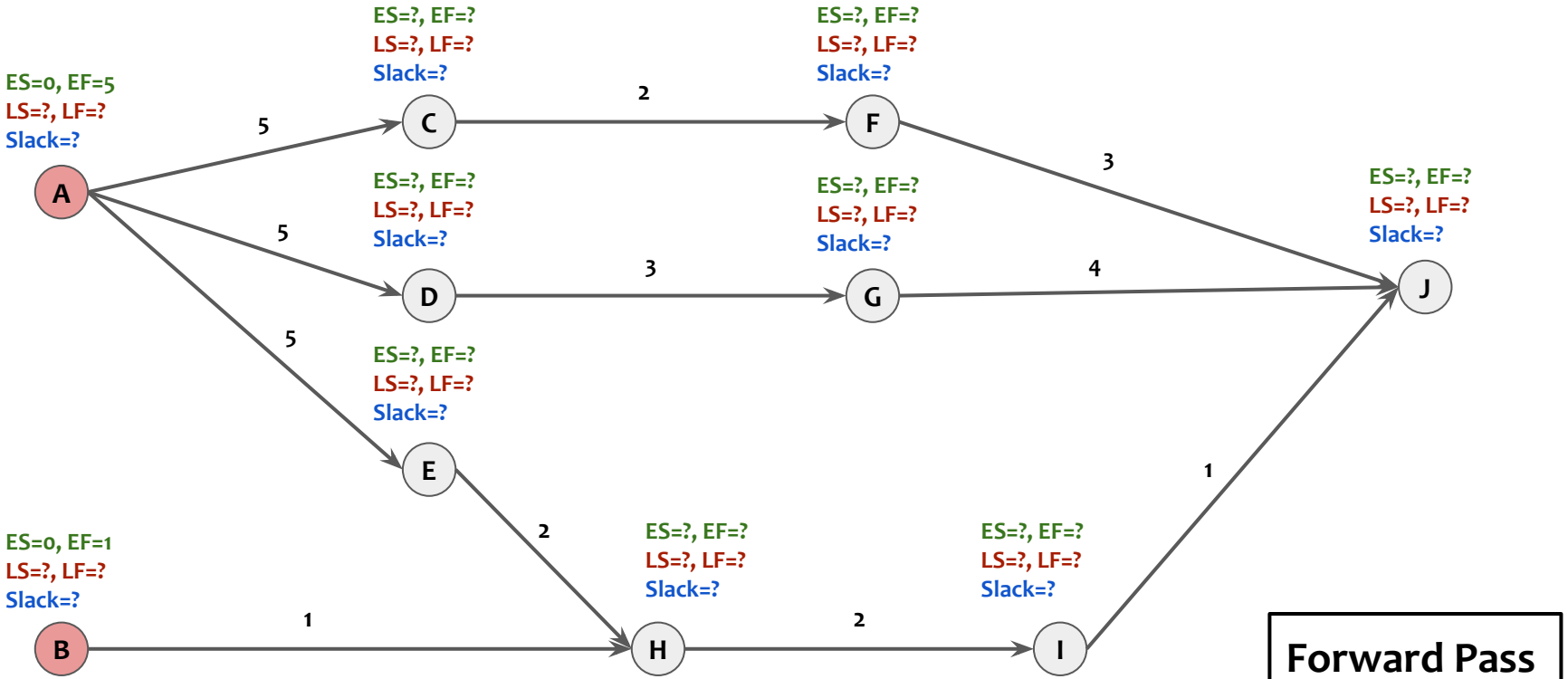
- Critical path gives us the set of tasks in a project which can not be delayed, even for one day, to complete the overall project within its strict deadline.

CPM: Example Problem

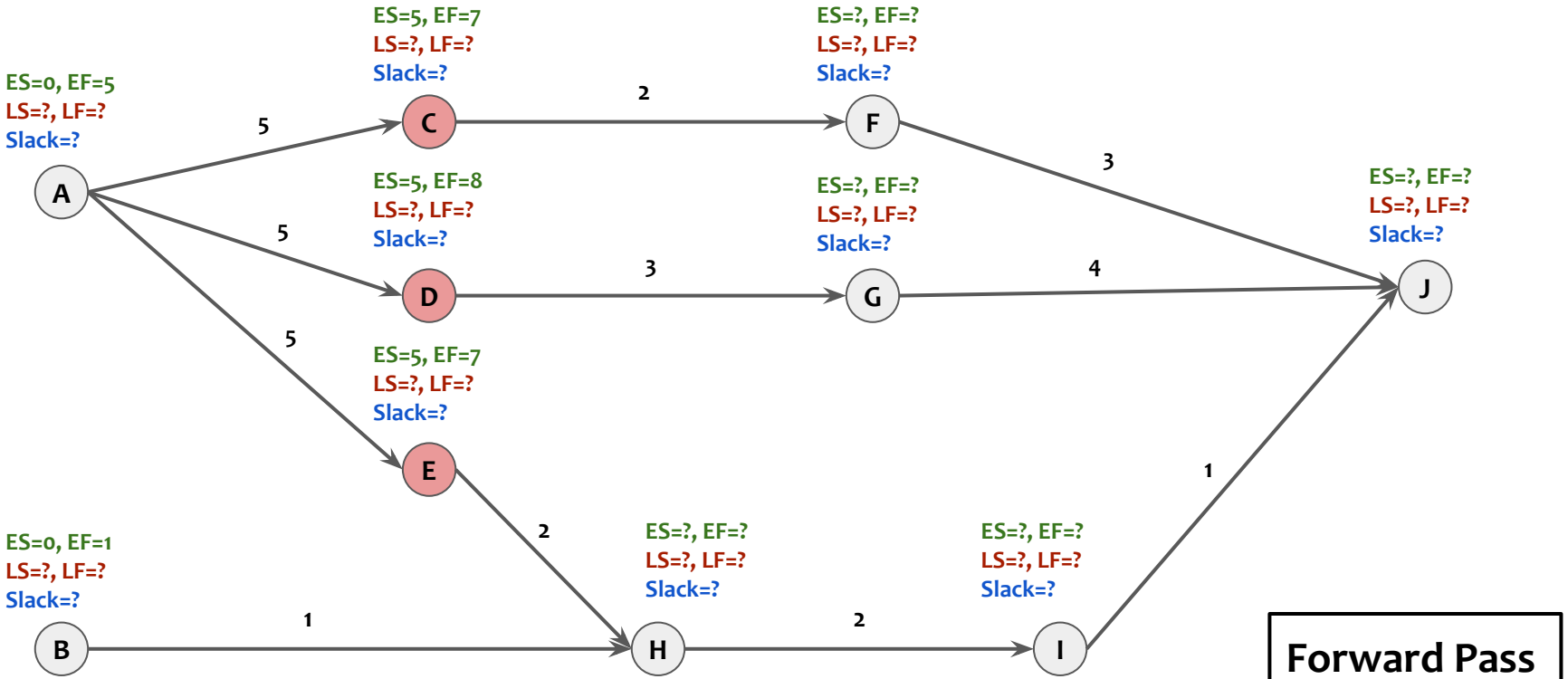
- Consider the following information about a particular project. Now, determine the strict deadline, slack of different tasks, and critical path of this project by applying Critical Path Method (CPM).

Task	Predecessor(s)	Duration (in Day)
A	-	5
B	-	1
C	A	2
D	A	3
E	A	2
F	C	3
G	D	4
H	B, E	2
I	H	1
J	F, G, I	1

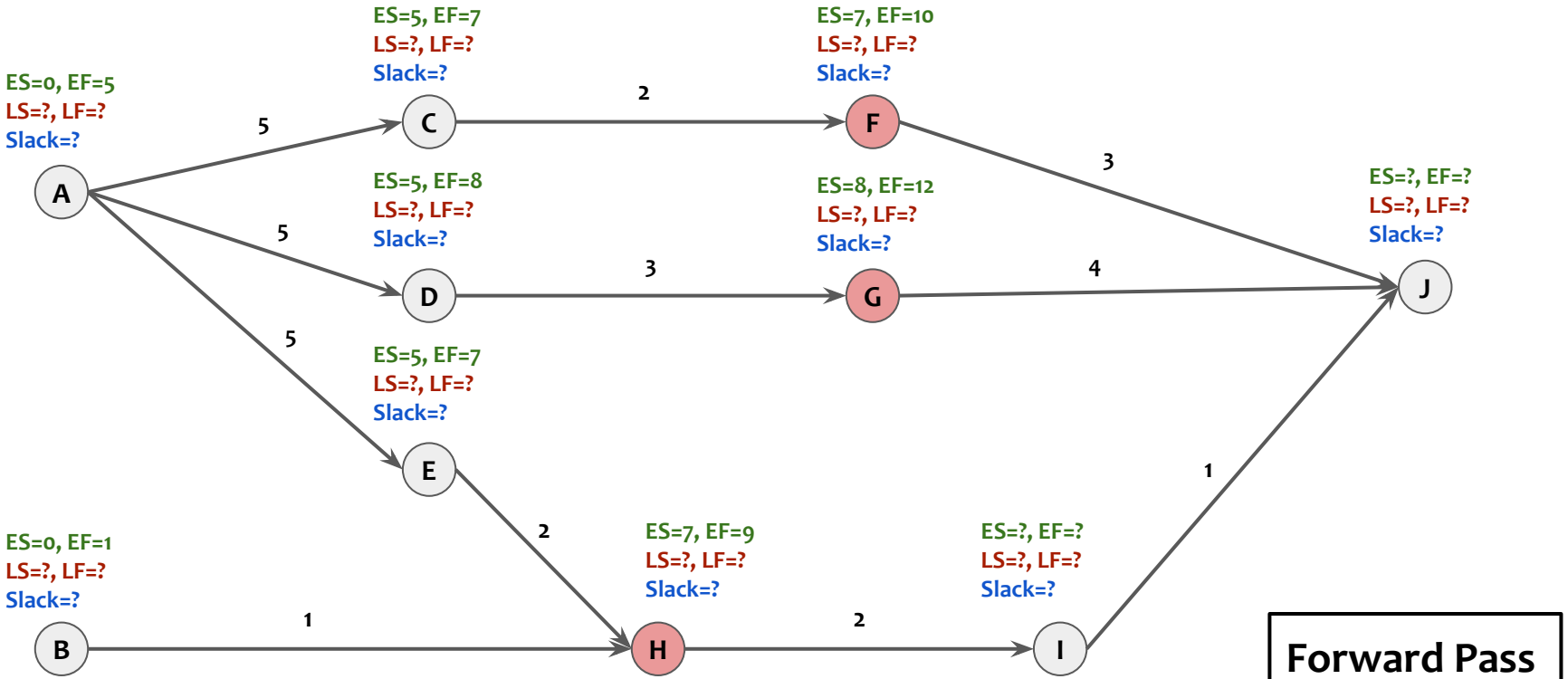
CPM: Example Problem



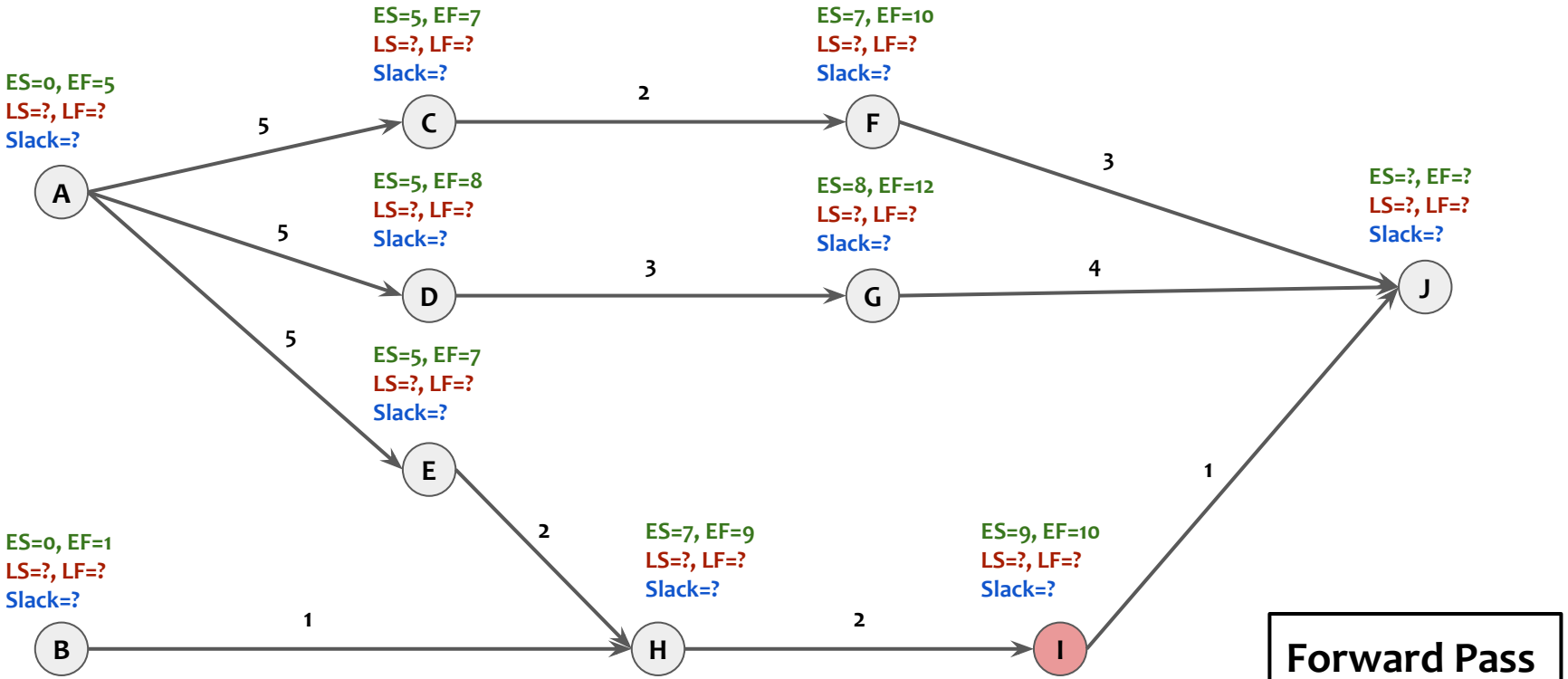
CPM: Example Problem



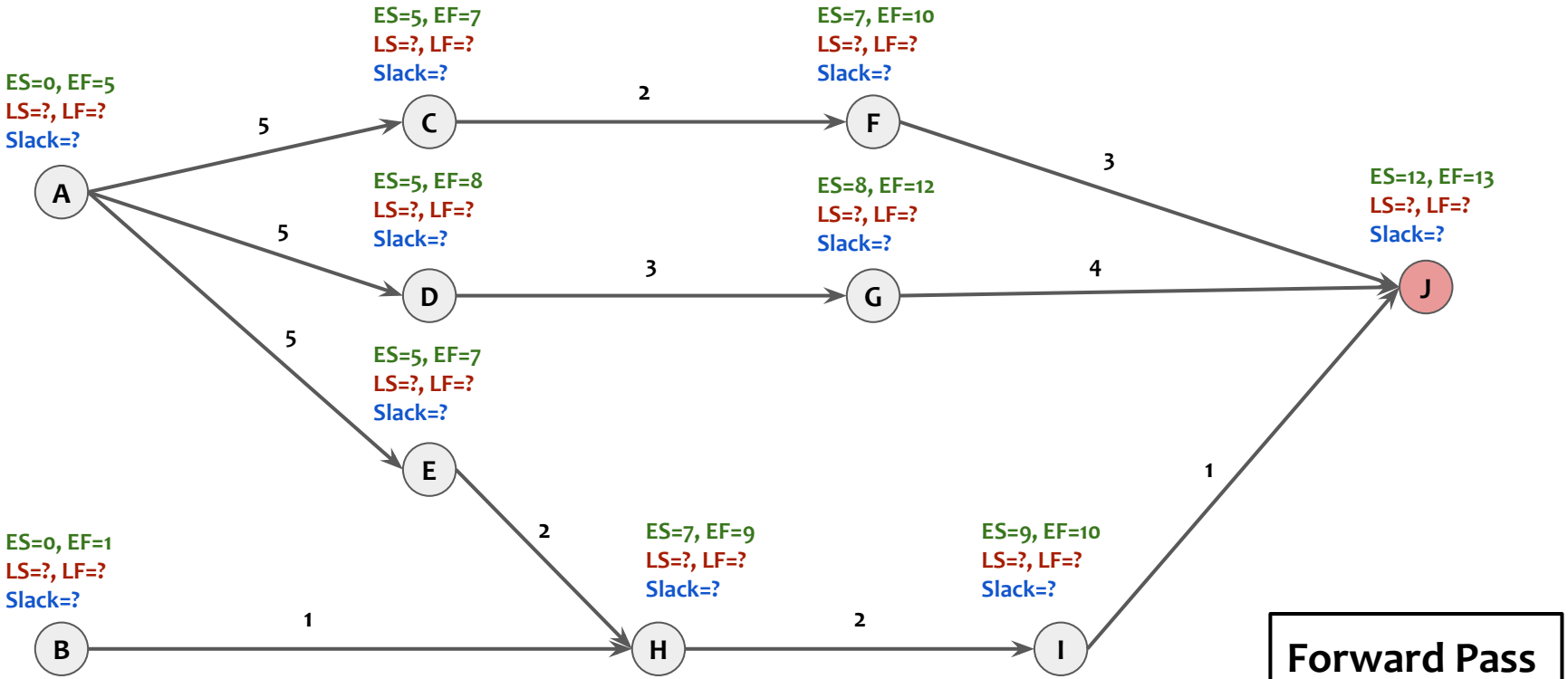
CPM: Example Problem



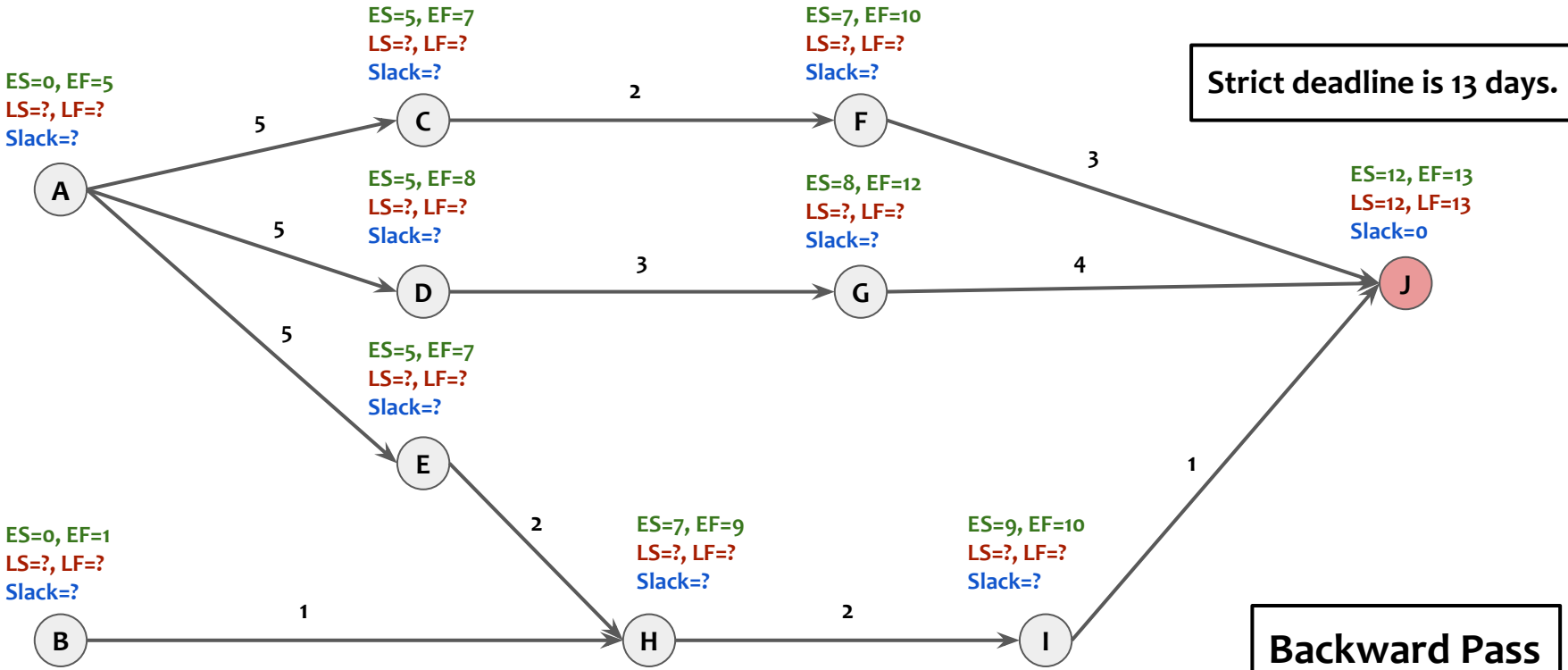
CPM: Example Problem



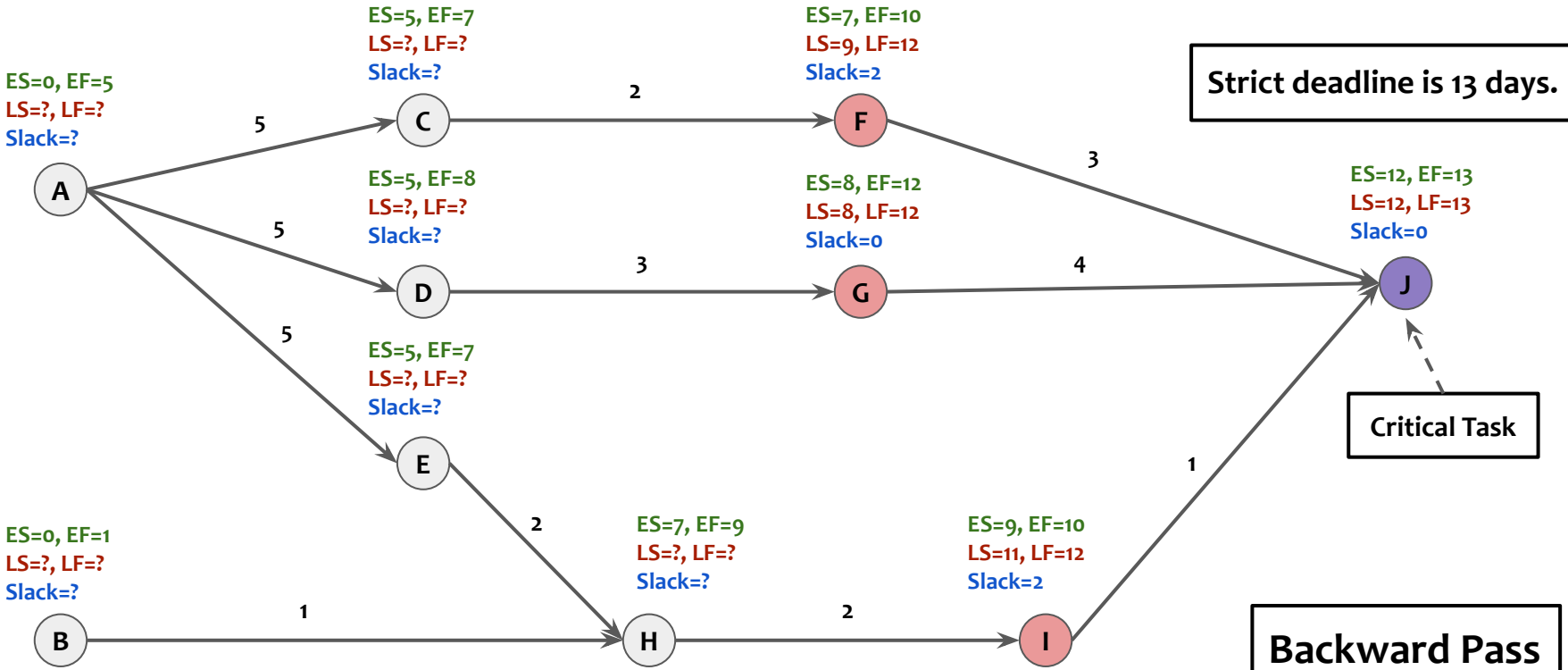
CPM: Example Problem



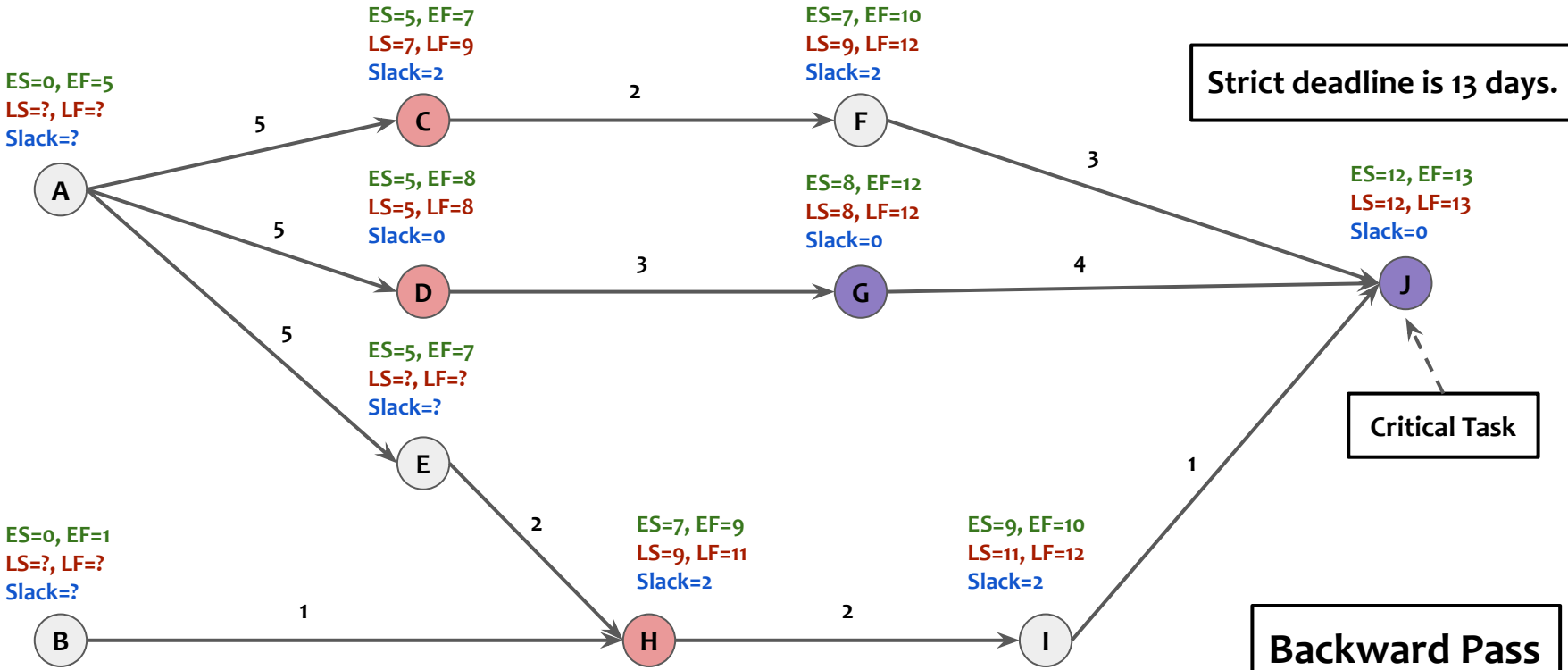
CPM: Example Problem



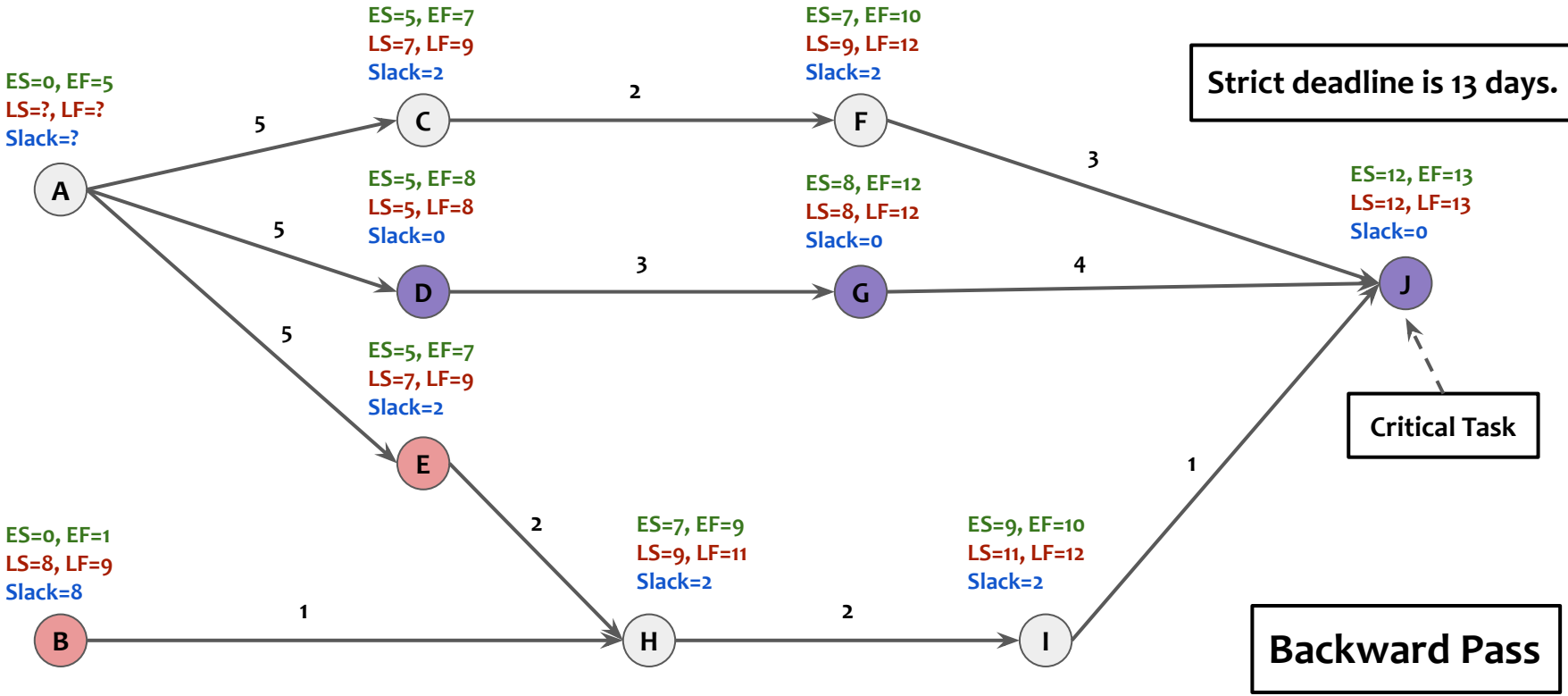
CPM: Example Problem



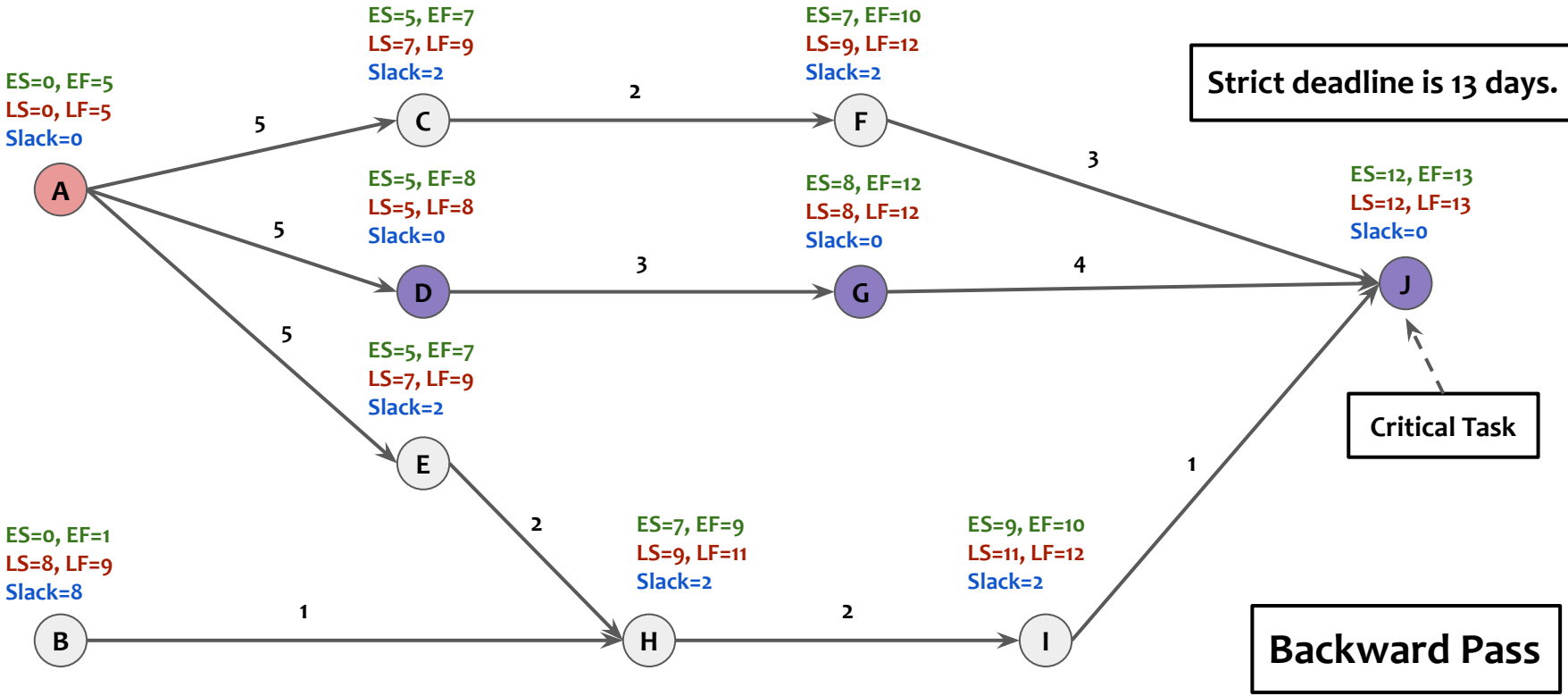
CPM: Example Problem



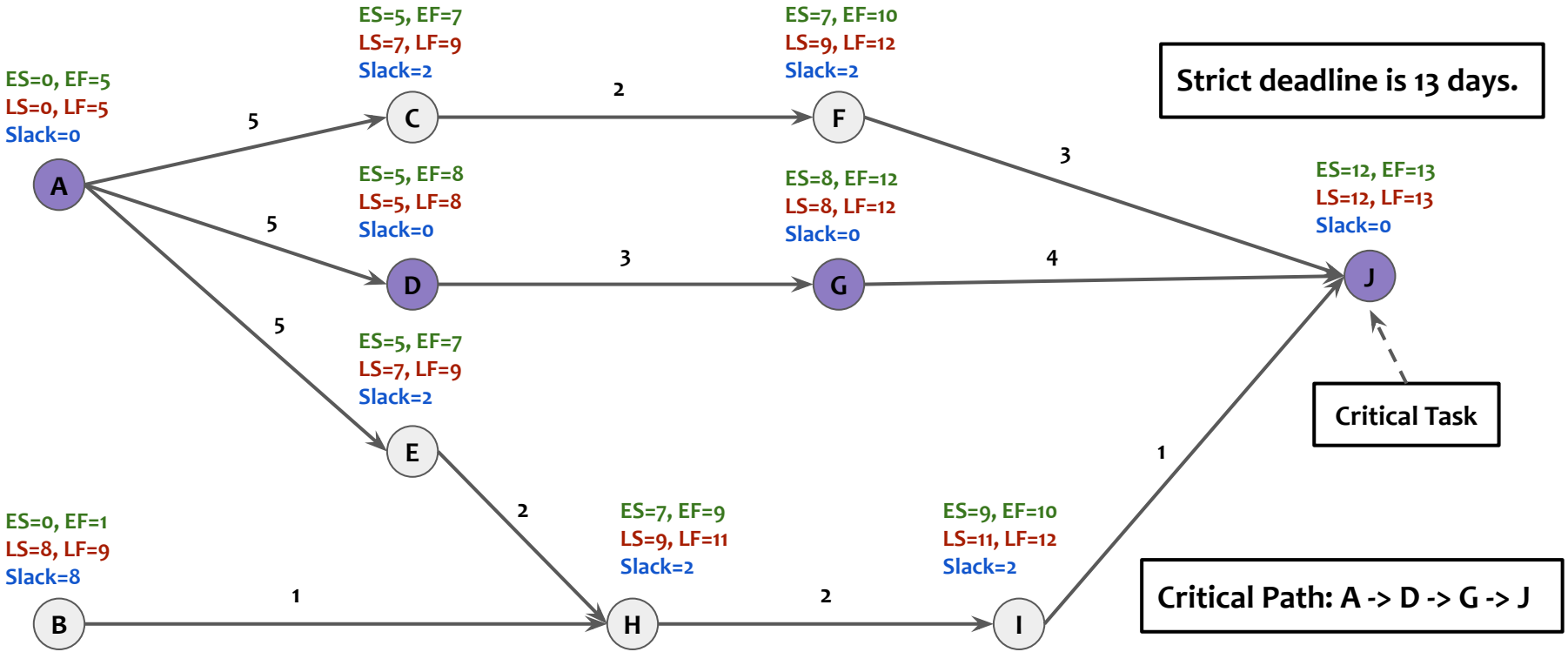
CPM: Example Problem



CPM: Example Problem



CPM: Example Problem



CPM: Practice Problem

- Consider the following information about a particular project. Now, determine the strict deadline, slack of different tasks, and critical path of this project by applying Critical Path Method (CPM).

Task	Predecessor(s)	Duration (in Day)
A	-	14
B	I	5
C	J	5
D	A	10
E	B, C, D	31
F	E	7
G	-	14
H	G	21
I	-	14
J	-	9
K	F, H	21

Earned-value Management

Earned-value Management (EVM)

- The amount of work you have planned to have accomplished by now, in monetary term or time, is called the ***planned value***.
- The amount of monetary resource or time you have actually spent by now is called the ***actual cost***.
- The amount of work, in terms of your baseline budget, you have actually accomplished by now, in monetary term or time, is called the ***earned value***.
- Idea, here, is to link schedule and cost together to monitor both in the same *units* of value, for gathering more information about a project than usual.

Earned-value Management (EVM)

- **Planned value (PV)** is the value of all resources needed to meet the objectives of a project. Each objective of a project has an associated planned value.
- **Budgeted cost at completion (BAC)** is the sum of all the planned values in a project.
- **Earned value (EV)** is the amount of value completed or earned at any point during the project.
- **Actual cost (AC)** is the actual amount of money you have spent so far in a project. In a perfect project, EV and AC are equal.

EVM: Example Problem

- Suppose that we have budgeted $200 to buy, setup, test, and operate a new machine in a project.

- Here, our planned values (PVs) of each objective or task include $50 for buying, $75 for setting up, $25 for testing, and $50 for operating the machine. So, our BAC is $200 in this part of the project.

- At a certain point in time, we have completed buying the machine and thus, obtained $50 in earned value (EV). But, we have spent $60 to buy that machine. Therefore, our actual cost (AC), so far, is $60.

EVM: Example Problem

- Given the information, we can determine the following progress measures of the project.

- ***Schedule performance index (SPI)*** = EV/PV = $50/$50 = 1 (on schedule)

    - SPI < 1 means project is *behind schedule*, SPI = 1 means project is *on schedule*, and SPI > 1 means project is *ahead of schedule*.

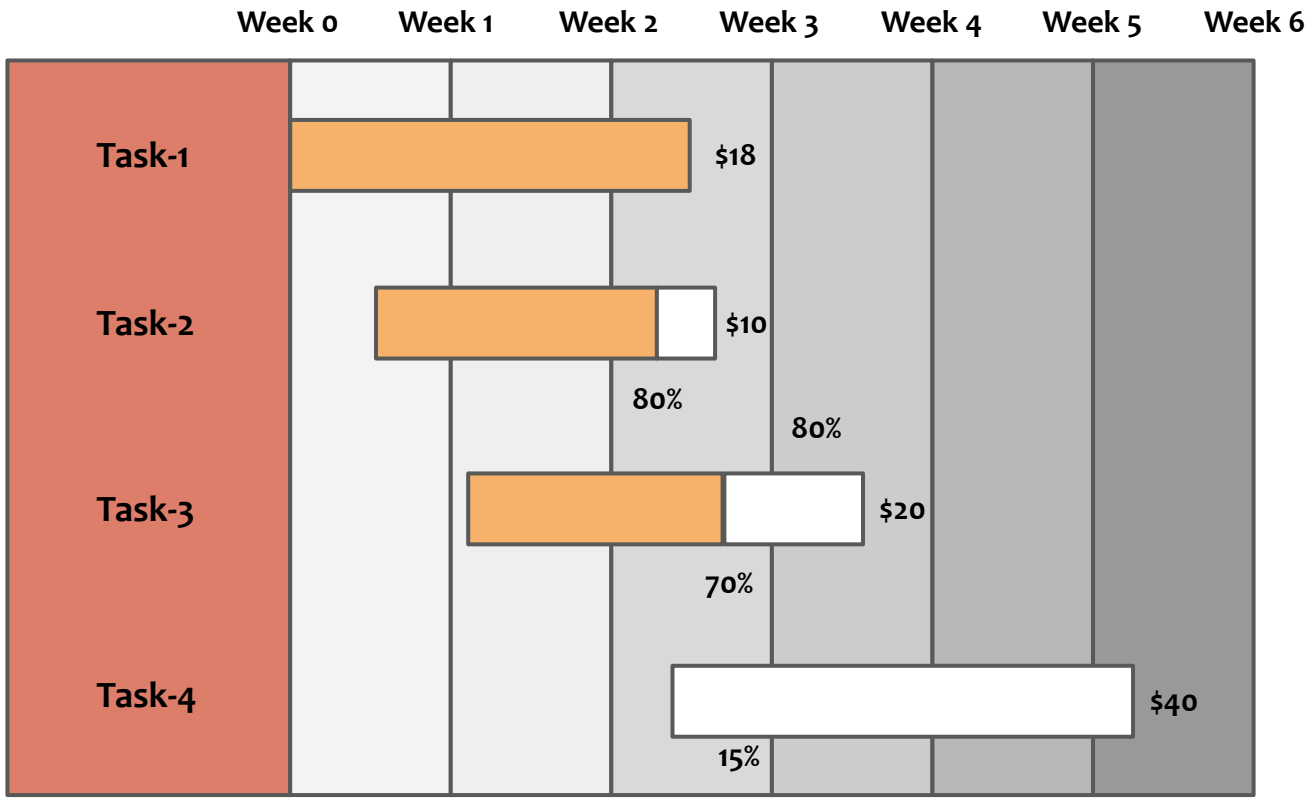
- ***Cost performance index (CPI)*** = EV/AC = $50/$60 = 0.83

    - That means, we get 83 cents worth of work for every dollar spent.

EVM: Example Problem

- ***Estimated cost at completion (EAC)*** = BAC/CPI = $200/0.83 = $240.96
- ***Schedule variance (SV)*** = EV - PV = $50 - $50 = $0
- ***Cost variance (CV)*** = EV - AC = $50 - $60 = -$10
- *Negative variance* is usually bad, whereas *positive* and *zero variances* are usually good.

EVM: Yet Another Example Problem



EVM: Yet Another Example Problem

- Given the information, we can compute the progress measures of this particular project *on week 3*.
- Planned value (budgeted cost of the scheduled work) is computed as follows.
    - Sum(PVs on week 3) = $18 + $10 + 80% of $20 + 15% of $40 = $50
- Budgeted cost at completion (BAC) = $18 + $10 + $20 + $40 = $88

EVM: Yet Another Example Problem

- Earned value (budgeted cost of the performed work) is computed as follows.
    - Sum(EVs on week 3) = $18 + 80% of $10 + 70% of $20 + 0% of $40 = $40
- Suppose that actual cost (of the performed work) is $45 on week 3.

EVM: Yet Another Example Problem

- Schedule performance index (SPI) = EV/PV = $40/$50 = 0.8
    - That means, the project is behind schedule.
- Cost performance index (CPI) = EV/AC = $40/$45 = 0.89
- Estimated cost at completion (EAC) = BAC/CPI = $88/0.89 = $98.88
- Schedule variance (SV) = EV - PV = $40 - $50 = -$10
- Cost variance (CV) = EV - AC = $40 - $45 = -$5