

Exercise on Critical Path

www.spmbook.com

Compute the critical path of the following plan:

Task	Duration	Dependency
Task 1	4d	
Task 2	2d	
Task 3	4d	2
Task 4	2d	
Deliverable A		1, 3, 4
Task 6	6d	5
Task 7	2d	5
Deliverable B		6, 7
Task 9	2d	1SS

Solution

The **critical path** is the chain of activities which determines the project end date. Delay any activity in the critical path by an arbitrarily small amount and the plan will be delayed.

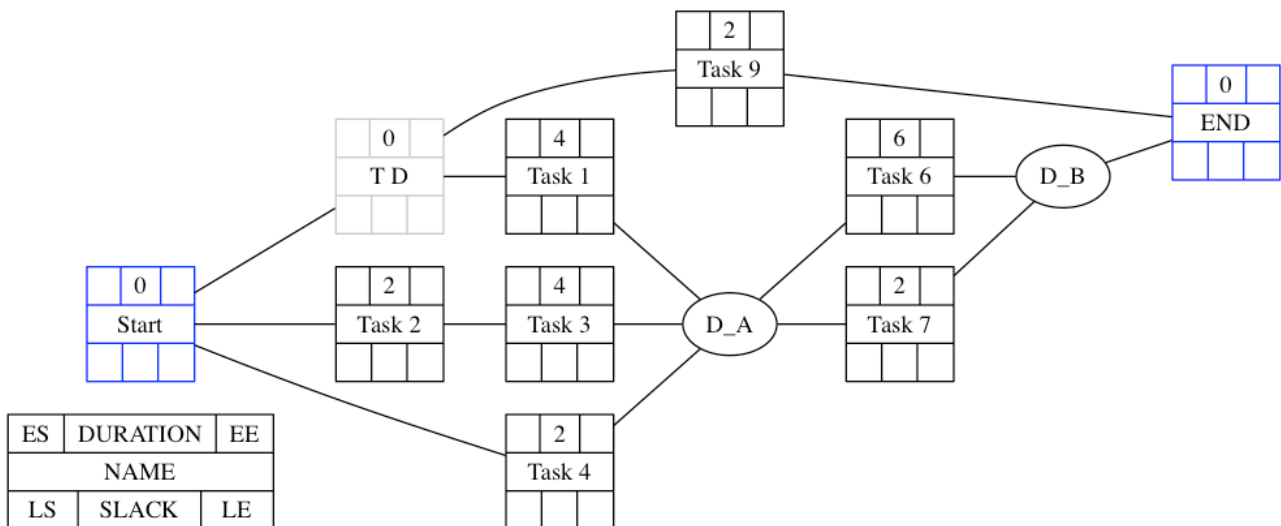
The determination of the critical path is based on the computation of the slack of each activity, computed by determining the earliest and latest starts (or ends) of each task in the plan. This data, in fact, allows us to compute the slack of each activity. The critical path is the path in which all activities have zero slack.

The computation is performed starting from an AON representation of the plan.

The first step is “preprocessing” the plan by:

- Transforming all dependencies into FS dependencies. In the case of the exercise, we have a SS dependency between Task 1 and Task 9. **This is solved by adding a dummy node representing the start of the first task.** In our example we can transform it into an FS dependency by adding a dummy node “**Task 1S**”, on which Task 1 and Task 9 depend with an FS dependency. This is equivalent to promoting the start of Task 1 to a node with zero duration.
- Add a **start node** to the plan and connect with an FS dependency all the activities whose starts do not depend on any other activity
- Add an **end node** to the plan and connect it with an FS dependency to all the activities whose ends do not have any dependency

Thus the following is the network diagram resulting from the specification:

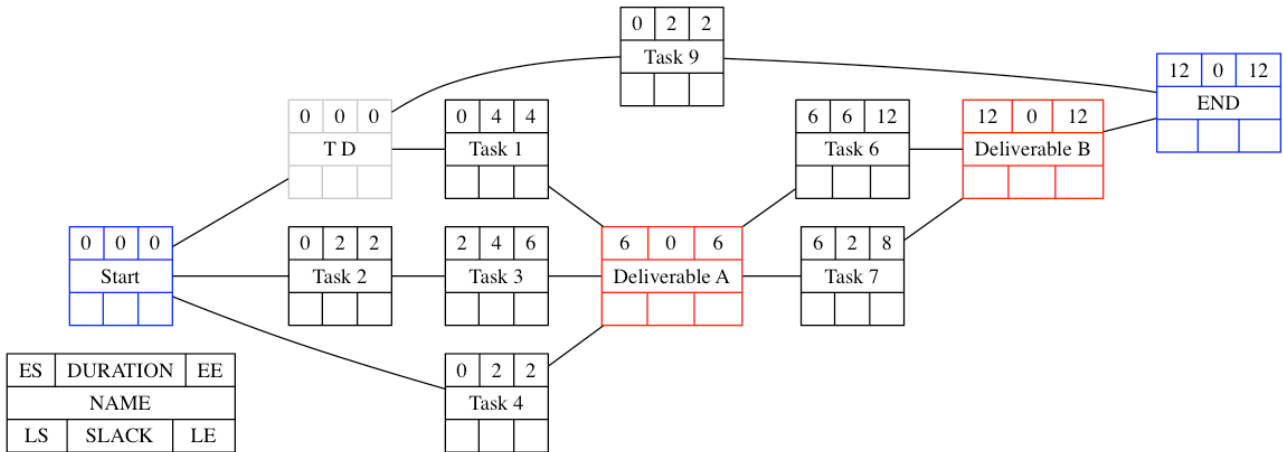


Notice that we have added the dummy node T_D to manage the SS dependency between Task 1 and Task 9; for all tasks we have used a notation which allows us to record all relevant information of a task, namely Earliest Start, Duration, Earliest End, Latest Start, Slack, and Latest End; only for the sake of clarity, we have used a circle for the deliverables—we will have to change it in the next

diagrams.

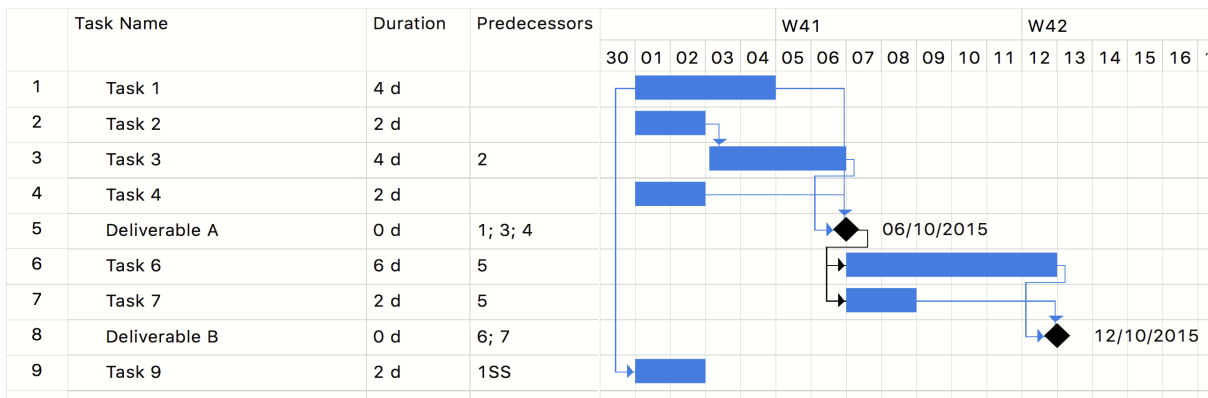
First we perform the **forward pass**, to compute the earliest start and end dates of each task. The Earliest Start and End of the start activity are zero. We compute a task's Earliest Start by taking the highest Earliest End of the predecessors. The Earliest End is computed by summing a task's duration to the Earliest Start (look at the book or at the slides for the details).

The result of the computation is shown below:



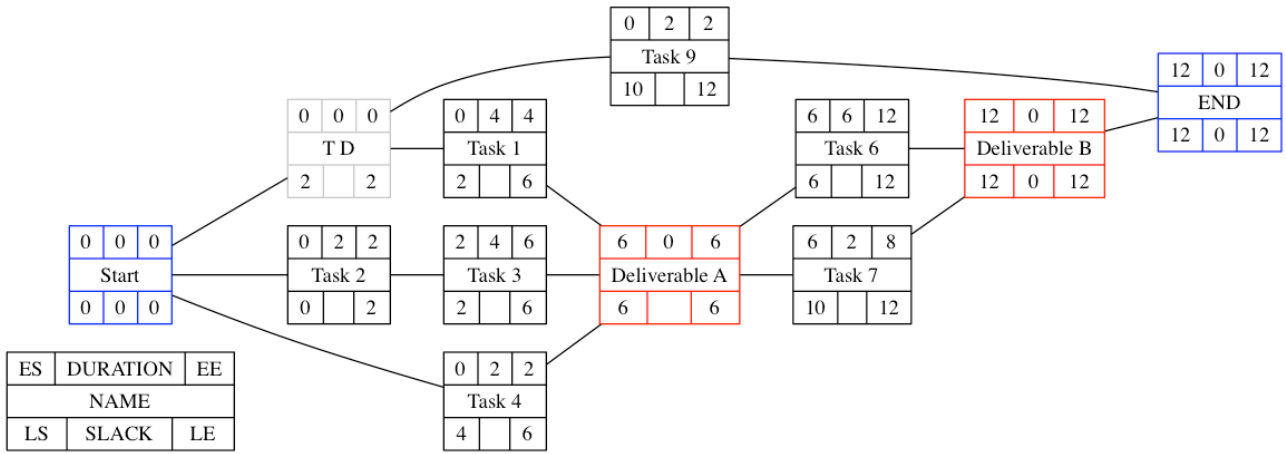
(Notice that we set the duration of deliverables to zero.)

This corresponds to setting all constraints of activities to “As Soon as Possible” (or pulling the tasks to the left), as shown by the corresponding Gantt diagram:

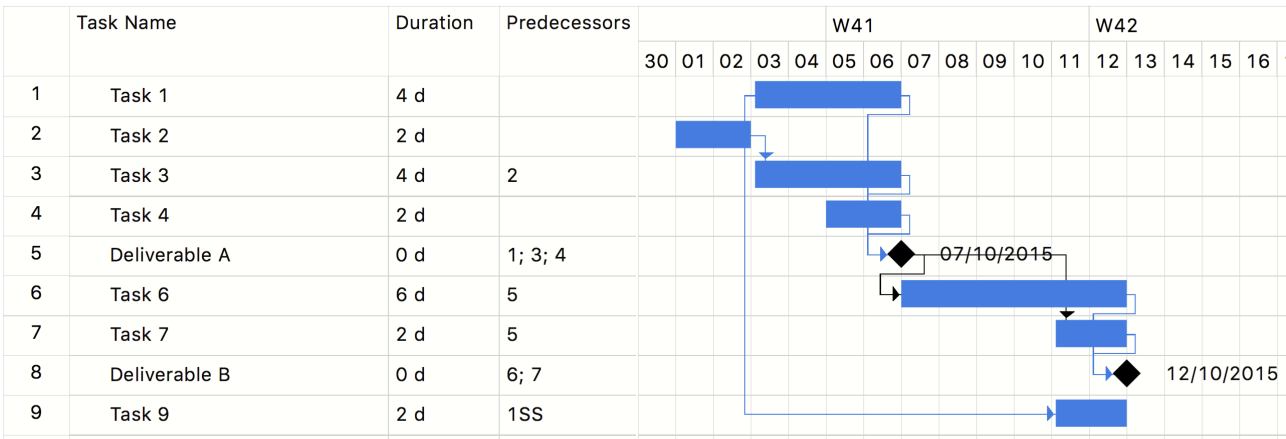


Then we perform the **backward pass**, to compute the latest dates. Starting from the end node, whose latest start and latest end are set to the earliest end, we set a task's latest end by taking the lowest latest end of all the successors. We then compute the latest start by subtracting a task's duration to its latest end.

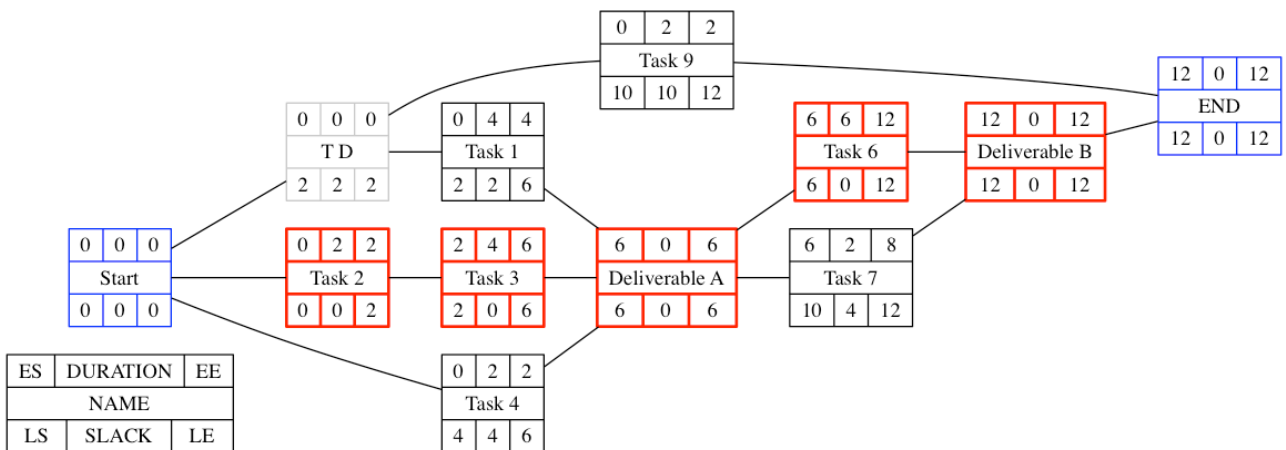
The result is shown in the following diagram:



This corresponds to setting the constraints of all the activities in the plan to “As late as possible” (or pushing the tasks to the right), as shown by the following Gantt diagram:



We can now compute the slack as the different between earliest end and earliest start (or latest end and latest start, equivalently). The **critical path** is made of all the tasks with zero slack (shown with a thickened line and in red in the following diagram):



If we compare the two Gantt charts, the critical path is exactly the set of activities whose start and end dates did not change when we changed the constraints from “As soon as possible” to “As late as possible”.