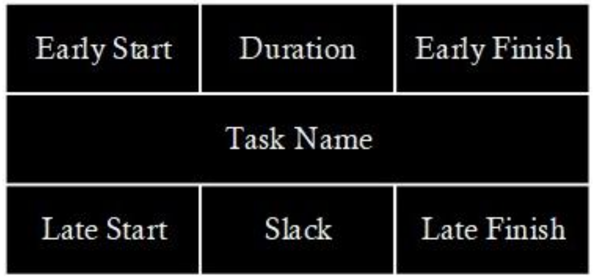
Job planning and schedule

Creating a Network Diagram: By knowing each activity’s individual duration, we can determine the early start, early finish, late start, and late finish of each activity. Not only that, but you can also determine exactly how much slack you have,i.e., the amount of time an activity can be delayed without affecting a successor activity.



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

Figure 11: Node compounds definitions

- Early Start (ES): The earliest time we can start an activity.
- Duration: How long it takes to finish all the tasks in an activity. It can be hours, days, weeks, or months.
- Early Finish (EF): The earliest time we can finish an activity.
- Late Start (LS): The latest time we can start an activity. Some activities may have some flexibilities (slacks or floats) that allow us to have some delay to start without affecting the overall project duration and other activities.
- Late Finish (LF): The latest time we can finish an activity. Based on the slacks (floats), we can finish an activity later than its scheduled completion time.
- Slack or float : It is the difference between LS and ES, or between LF and EF. Both subtractions generate the same result.
- As a result E. F=E.S+ Duration and, L.S= L.F – Duration. So we can have the total float as T.F= L.F - E.F.

Job planning and schedule

Creating a Network Diagram

Here is the below activities where we will use the **finish-to start dependency** between activities.

Activity	Duration (week)	Predecessors
A	1	–
B	2	–
C	2	A
D	4	A
E	1	B
F	2	C, D
G	3	E
H	1	G
I	4	G
J	1	F
K	3	J, H
L	4	I
M	1	K, L

We should place each activity node on the diagram by adhering to their precedence relationships with other activities from the above table. After we connect all the nodes, we can type the duration for each activity as can be seen in Figure 12.

All other parts in the nodes are zero for now. Besides, as all the dependencies are finish-to-start, the arrows (connectors) start **from the right side of a predecessor activity and finish on the left side of a successor activity**. For instance, when we finish Activity A, we can start both Activity C and Activity D. When we finish both, we can start Activity F.

Job planning and schedule

Creating a Network Diagram

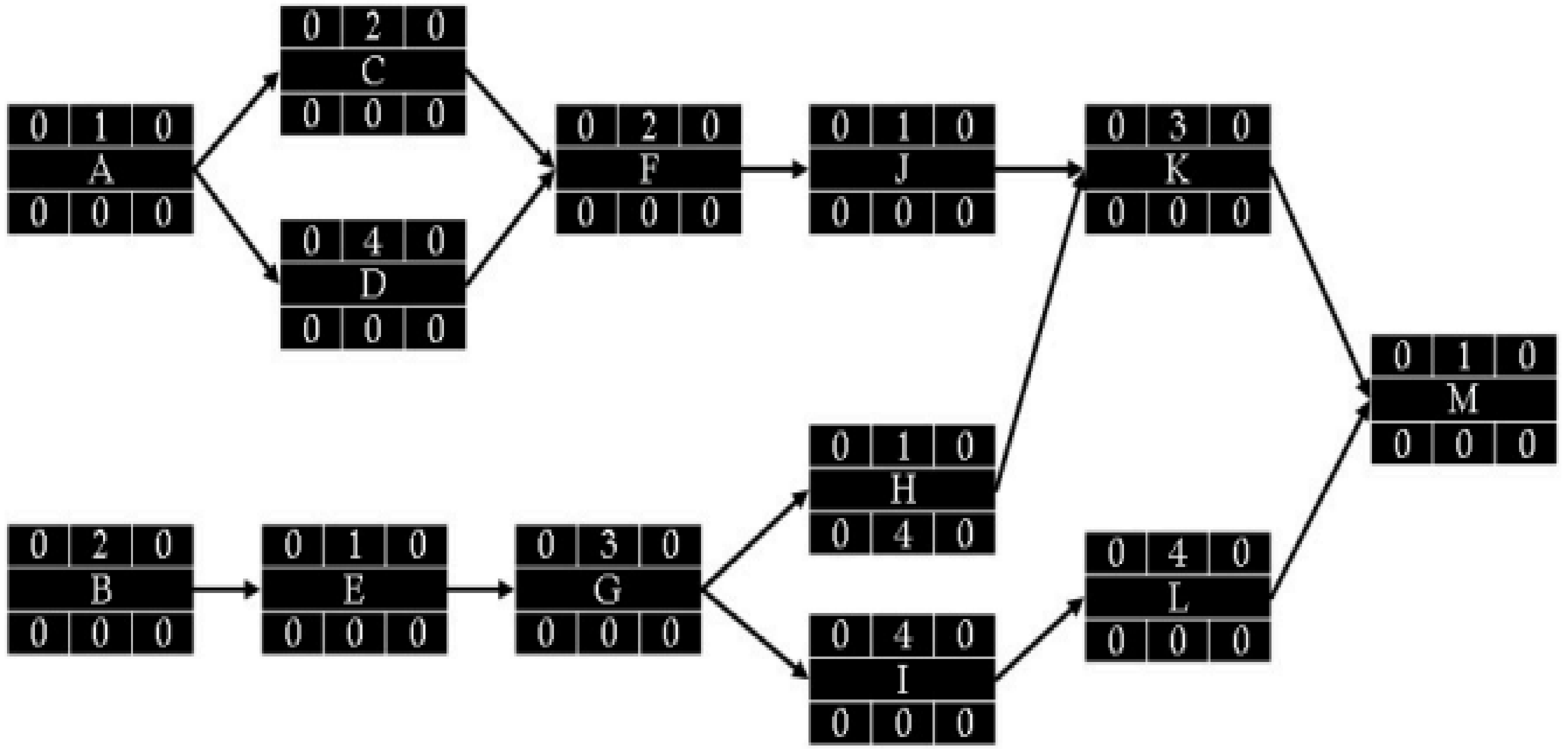


Figure 12 : Activity Network Diagram with Durations

To determine the dates we should go forward through the network for early start and early finish then for late start and late finish we should backward. We will get the overall time to finish the whole project.

Job planning and schedule

Creating a Network Diagram (Forward pass)

For two starting activities (A and B), ES is marked zero, which means that it is the very first day of the project (Figure 13)

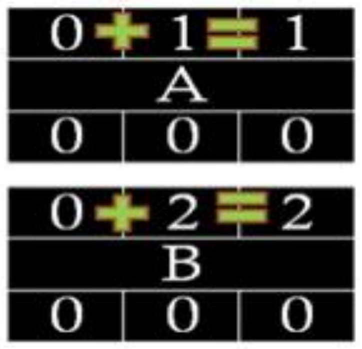


Figure 13: ES and EF times

We add ES to the duration for each activity to find EF.

For A, EF is (0+1) = 1 week, and for B, it is (0+2) = 2 weeks. It means that we can finish A at the end of the first week, and finish B at the end of the second week.

Then, we carry the EF time to the nodes immediately succeeding the recently completed nodes (predecessors). C and D inherit 1 (EF) from A, and it becomes ES for both successor activities. For E, we pass 2 (EF for B) to E as the ES time.

Then, we add new ES times to the duration of activities to find the EF for new successors (Figure 14).

Job planning and schedule

Creating a Network Diagram (Forward pass)

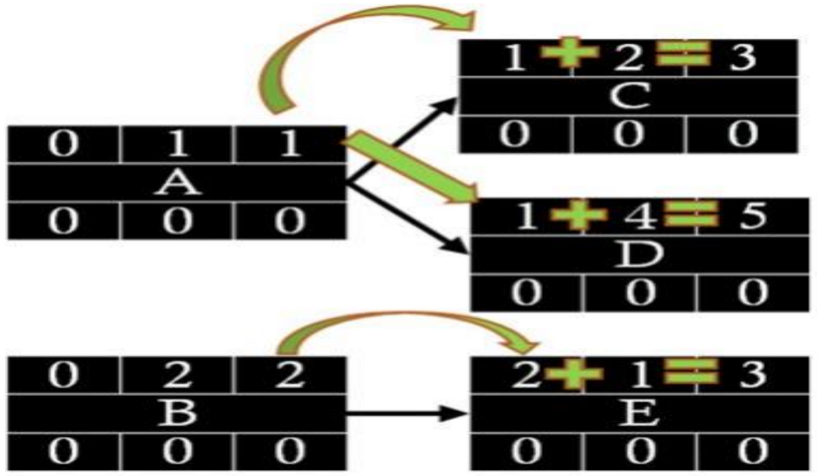


Figure 14: Passing predecessor ES times to successors as EF times

At a merge point, as is the case when C and D merge at F, we pass the highest EF time of predecessors (C and D) to the successor activity (F) (Figure 15). EF time of D becomes ES time for F.

Job planning and schedule

Creating a Network Diagram (Forward pass)

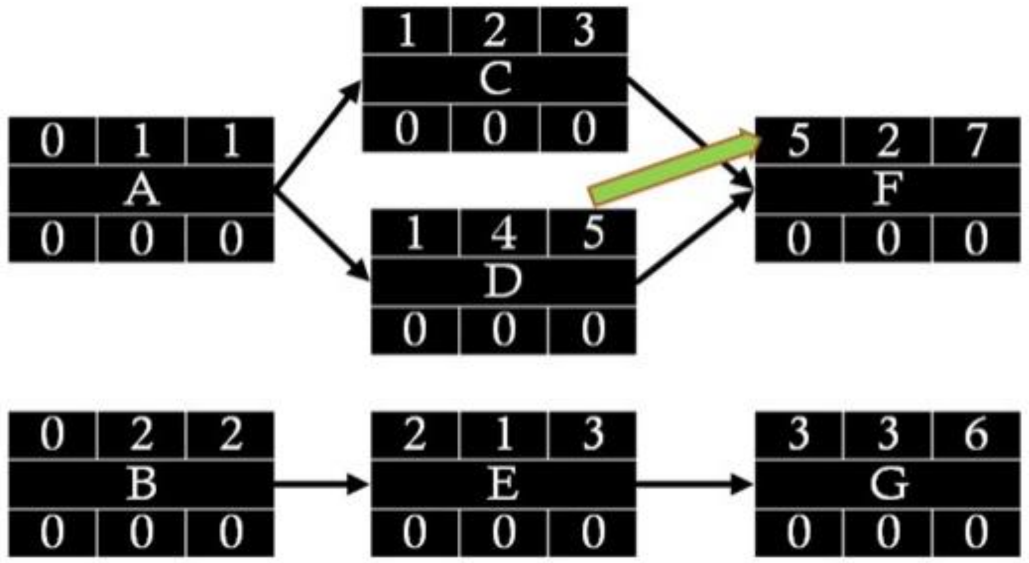


Figure 15: Passing predecessor ES times at merge points

When the forward pass is done, we can generate all ES and EF times for all the activities. The EF of the last activity (M) gives us the overall duration of the project which is 15 weeks (Figure 16)

Job planning and schedule

- Network Diagram: **A**ctivity **O**n **N**ode (Precedence Diagramming Method)

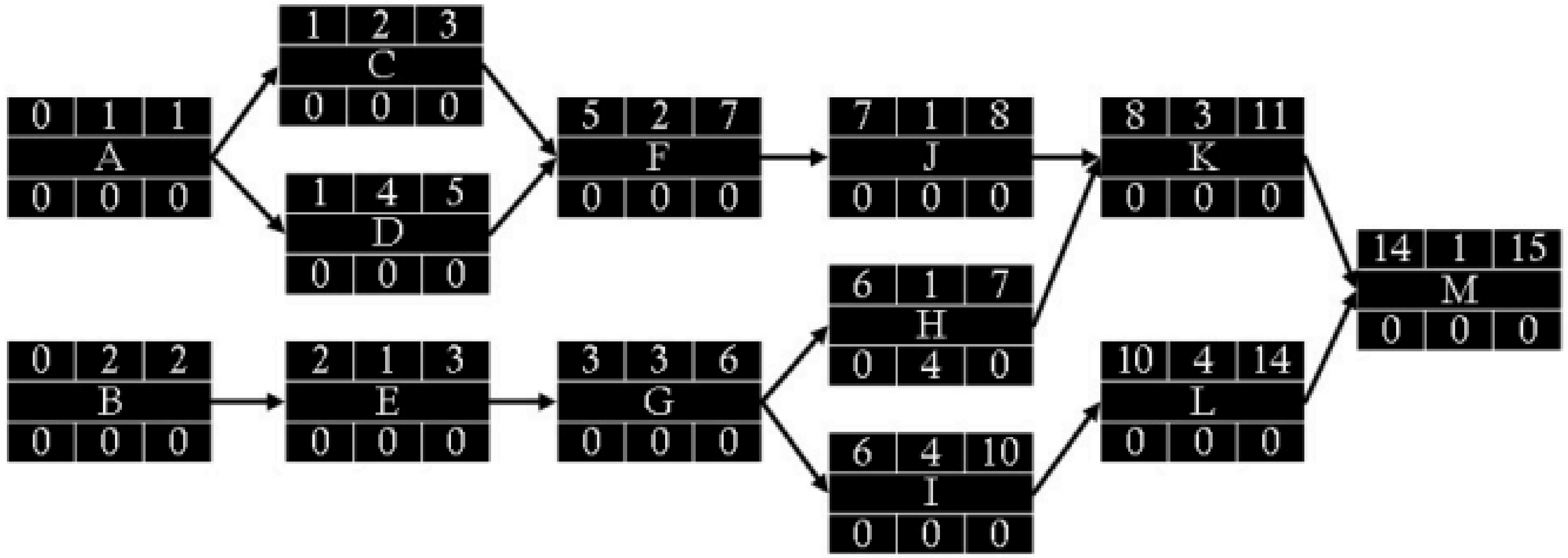
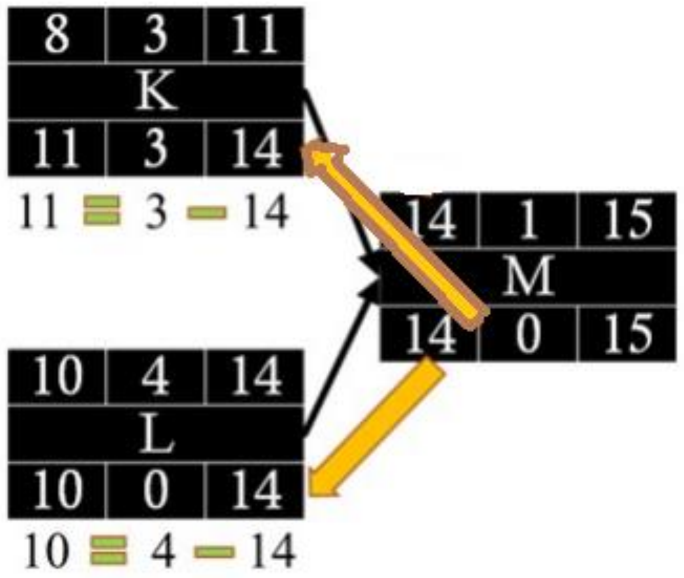


Figure 16: Activity Network Diagram filled with Forward Pass method

Job planning and schedule

Network Diagram: **A**ctivity **O**n **N**ode (Precedence Diagramming Method)

- Using the backward pass method we can determine the slack/float for each activity and we can define the critical path easily.
- **Slack is the difference between LS and ES, or between LF and EF, if Slack is Zero this means the activity is critical.**



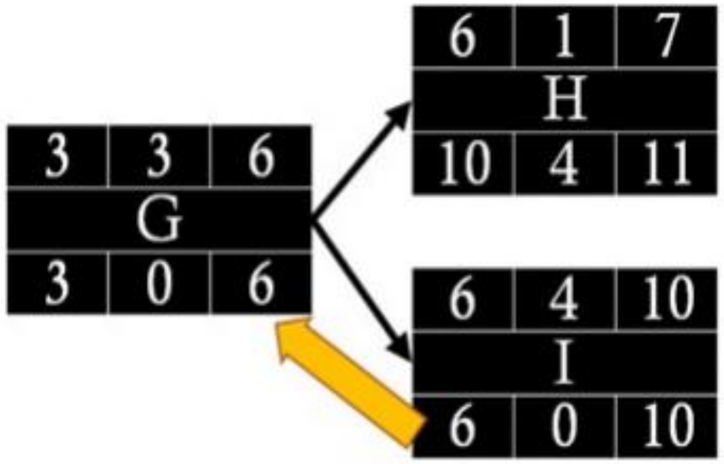


Figure 17:Passing LS times to successors as LF times

Job planning and schedule

Activity **O**n **N**ode (Precedence Diagramming Method)

- By applying the backward pass method we can have the below network:

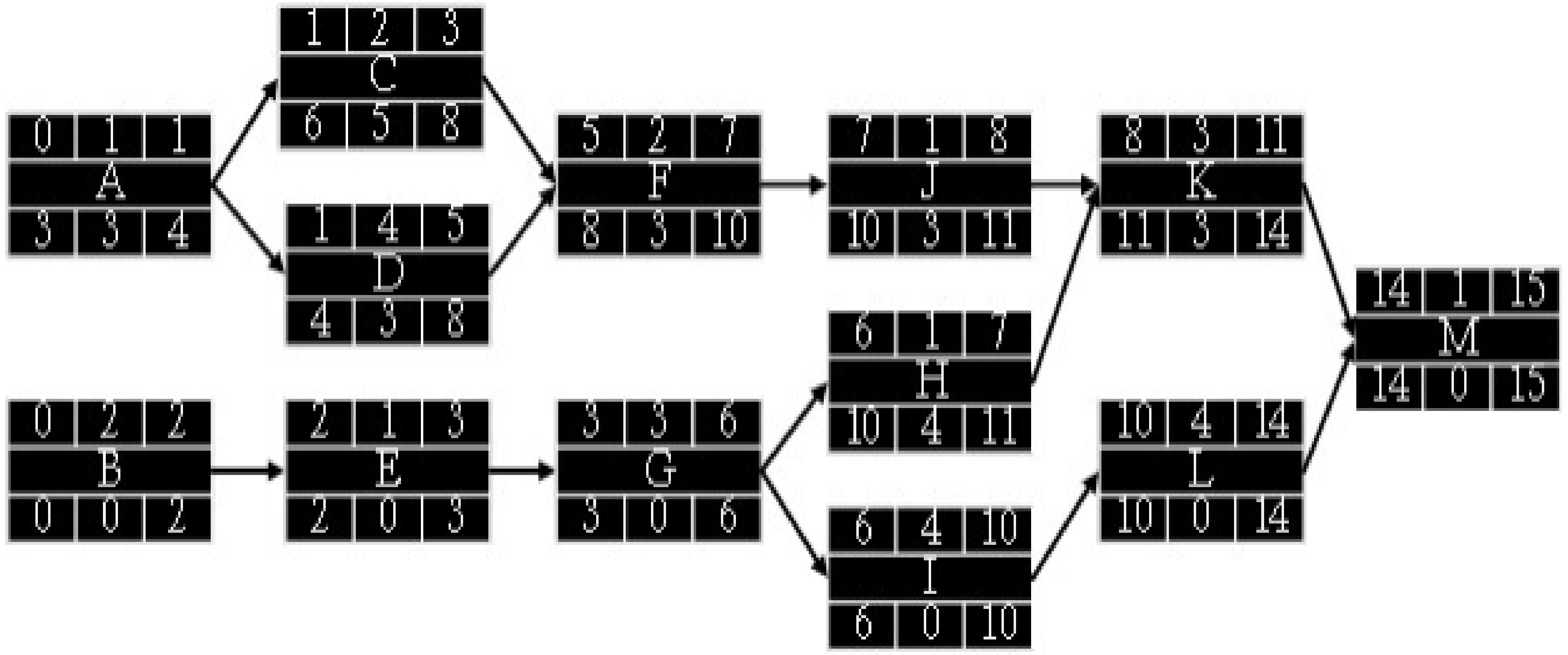


Figure 18: Activity Network Diagram filled with backward Pass method

Job planning and schedule

Critical Path: By definition it is the path with the longest duration possible in the network diagram which equal to the project duration; as consequence no delay is acceptable on any of the activities belonging to this critical path.

From the previous figure 18 we can easily define the critical path for this AON; by connecting the activities with **Zero** slack which are critical and no delay is accepted, So **B-E-G-I-L-M** is the critical path.

If we go for the path with longest duration determination we will have the below:

A – C – F – J – K – M:	1+2+2+1+3+1=10 weeks
A – D – F – J – K – M:	1+4+2+1+3+1=12 weeks
B – E – G – H – K – M:	2+1+3+1+3+1=11 weeks
B – E – G – I – L – M:	2+1+3+4+4+1= 15 weeks

So, the last path is the longest network path, as a result **B – E – G – I – L – M** is the critical path for this project.