

Project Management: Part 2

Scheduling

- One of the most important things you can do is schedule.
- Also one of the first things you should do!
- Tools help
 - Microsoft Project
 - OpenProj.org
 - OpenWorkbench.org



Planning

- The bad news: time flies
- The good news: you're the pilot!
- You must begin planning immediately
 - Given limited information
 - Plan anyway and then revise



Creating a plan: Things to know

■ Scope

- **Context.** How does the software to be built fit into a larger system, product, or business context and what constraints are imposed as a result of the context?
- **Information objectives.** What customer-visible data objects are produced as output from the software? What data objects are required for input?
- **Function and performance.** What function does the software perform to transform input data into output? Are any special performance characteristics to be addressed?

■ Software project scope must be unambiguous and understandable at the management and technical levels.

--	--

Creating a plan: Things to do

- Problem Decomposition: Sometimes called *partitioning* or *problem elaboration*
- Once scope is defined ...
 - It is decomposed into constituent functions
 - It is decomposed into user-visible data objects
or
 - It is decomposed into a set of problem classes
- Decomposition process continues until all functions or problem classes have been defined (this won't be far at the beginning of your project)

--	--

Schedule

- List of tasks
 - With dates
 - With assigned resources (people)
 - With durations
 - With predecessors and successors
- Use historical data from past projects
- Use an incremental model that will deliver critical pieces of information

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	Don't Care	Don't Care	Don't Care	Don't Care About You	I'm in Love	Wait
Always Comes Too Late	You can fall apart	Break My Heart	Break My Heart	Doesn't Even Start	I'm in Love	Wait
Always Comes Too Late	Don't Care	Heart Attack	Heart Attack	Never Looking Back	I'm in Love	Wait
Always Comes Too Late	Hold Your Head	Stay in Bed	Stay in Bed	Watch the Walls Instead	I'm in Love	Wait
Always Comes Too Late	30	31				

GraphJam

Schedule Terms

- Critical path
 - Sequence of tasks that form the longest path to completion of the project. Any delay on any of these will make the overall completion date move.
- Slack
 - Amount of time a task can be delayed without affecting the overall completion date.
 - Start slack - amount before task needs to start
 - Finish slack - amount before task needs to finish
- Milestone - An important date in the schedule
- Dependencies - relationship between tasks

--	--

Schedule Dependencies

- **FS - Finish to start (most common)**
 - A FS B. B doesn't start until A is finished
 - Build wall FS Paint wall
- **FF - Finish to finish**
 - A FF B. B doesn't finish before A is finished
 - Write final chapter FF Complete Index
- **SS - Start to start**
 - A SS B. B doesn't start until A has started
 - Project funded SS project management activities begin
- **SF - Start to finish**
 - A SF B. B doesn't finish before A has started
 - Once A starts, B is allowed to finish
 - B=Baby sit a child, A=parent comes home

Resource Leveling



- A process to examine a project for an unbalanced use of people and to resolve over-allocations or conflicts
- Happens when multiple tasks are scheduled at the same time for the same person
- Solution:
 - Make tasks sequential by introducing “fake” dependencies
 - Split resource usage among tasks (50% on task 1, 50% on task 2)

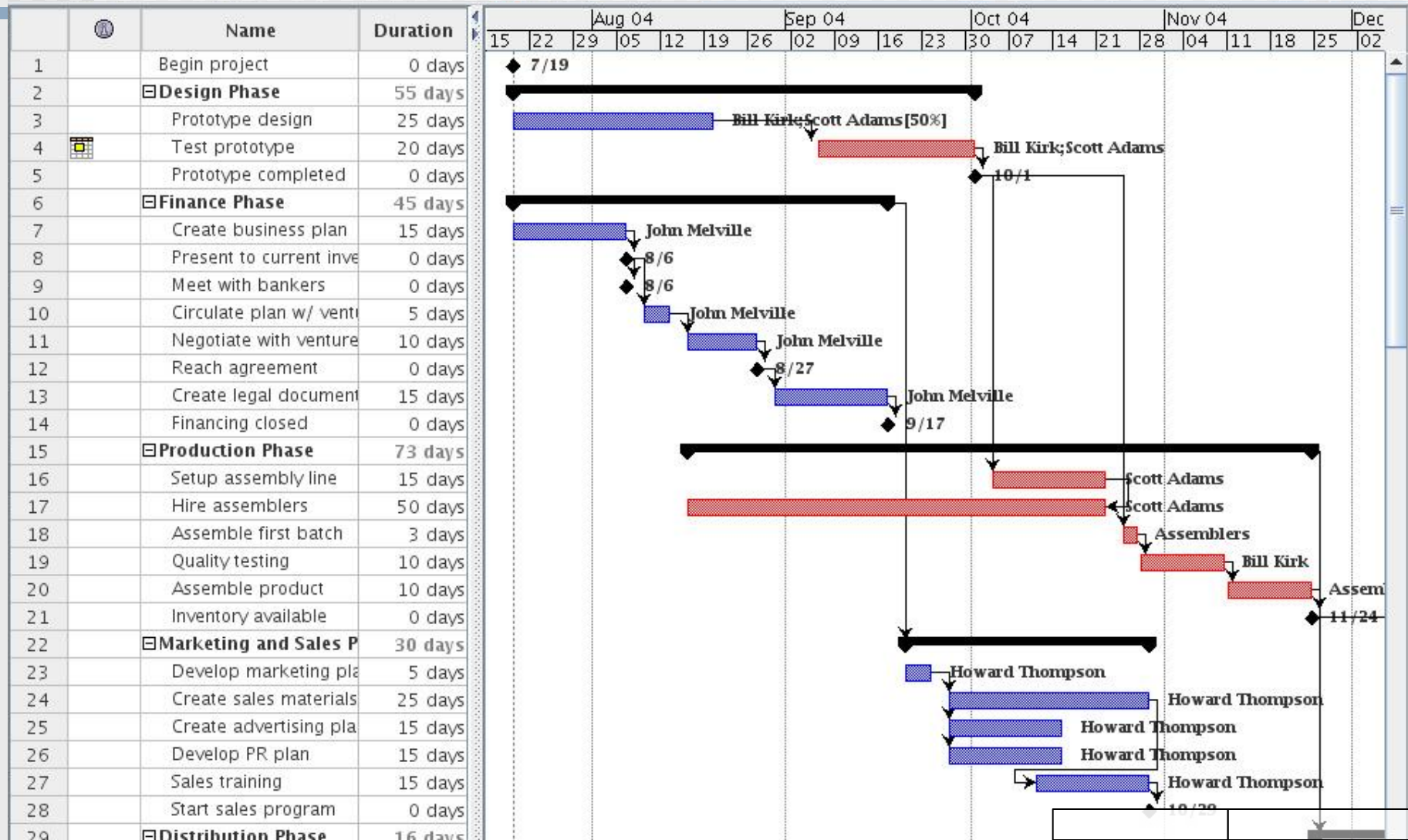
--	--

Auto Resource Leveling

- Some tools (not Open Project) provide auto resource leveling
- Tool automatically ensures no person works over 100% of the time (automatically makes tasks sequential)
- Advantageous because this does not introduce “fake” dependencies

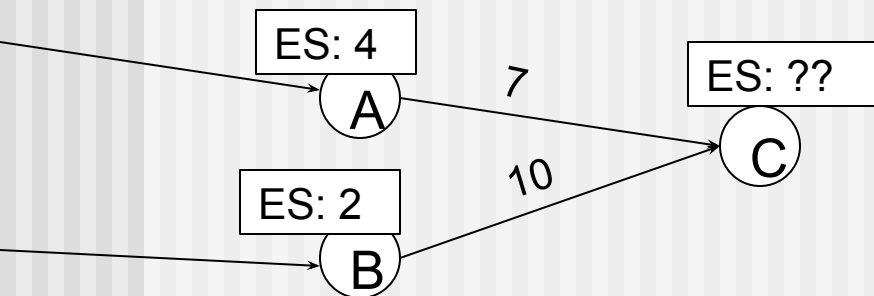
--	--

Gantt Chart



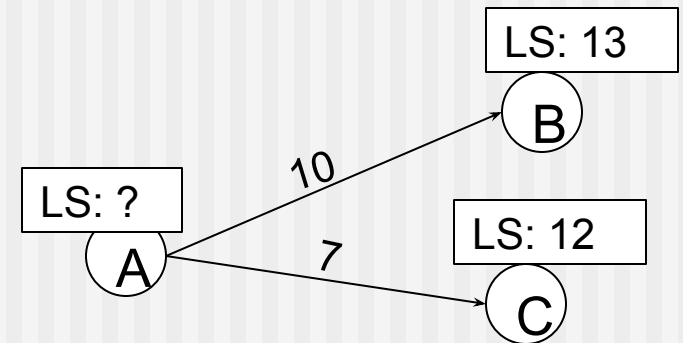
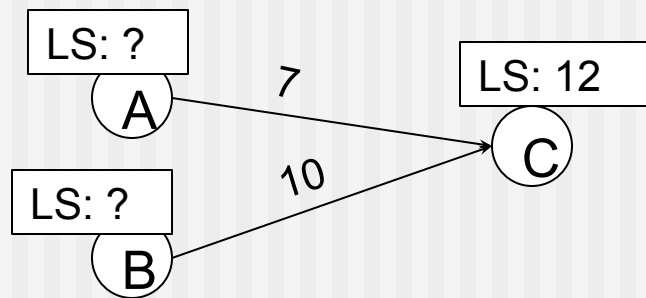
Finding Critical Path

- Draw a network diagram of the activities
- Determine the Early Start (ES) of each node. Work from beginning node (ES=0) to final node
- ES - earliest time the activity can start
- $ES = \text{Max}(\text{ES}_{\text{prevNode}} + \text{Duration}_{\text{PrevNode}})$



Finding Critical Path

- Determine the Late Start (LS) of each node. Work from the final node to the beginning node.
 - The latest time the activity can start without changing the end date of the project
 - $LS = \text{MIN}(\text{LS}_{\text{next}} - \text{Duration}_{\text{Node}})$
 - For the last node $LS = ES$

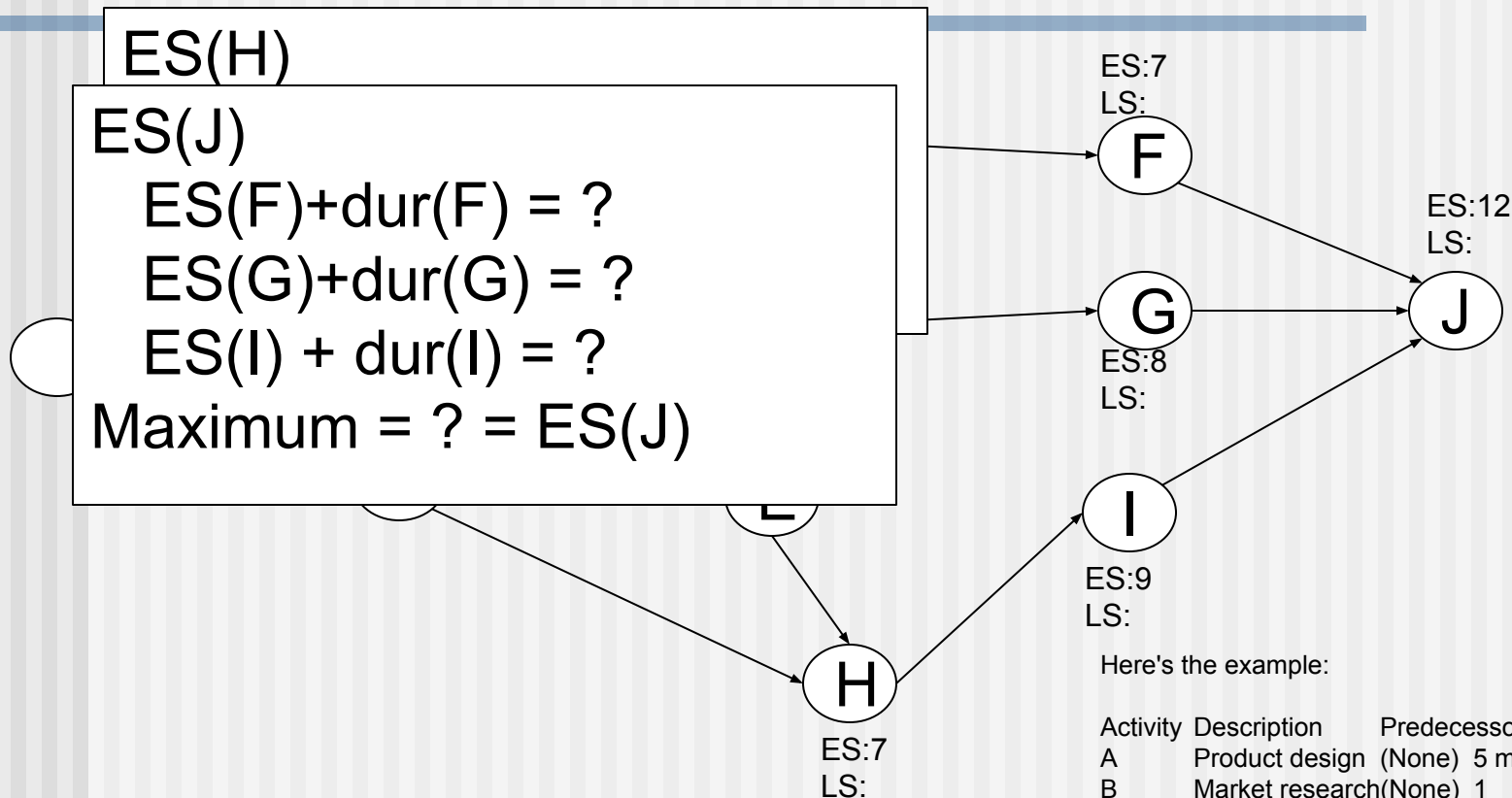


Example

Here's an example:

Activity	Description	Predecessor	Duration
A	Product design	(None)	5 months
B	Market research	(None)	1
C	Production analysis	A	2
D	Product model	A	3
E	Sales brochure	A	2
F	Cost analysis	C	3
G	Product testing	D	4
H	Sales training	B, E	2
I	Pricing	H	1
J	Project report	F, G, I	1

Example Node Network



Here's the example:

Activity	Description	Predecessor	Duration
A	Product design	(None)	5 months
B	Market research	(None)	1
C	Production	A	2
D	Product model	A	3
E	Sales brochure	A	2
F	Cost analysis	C	3
G	Product testing	D	4
H	Sales training	B, E	2
I	Pricing	H	1
J	Project report	F, G, I	1

Example Node

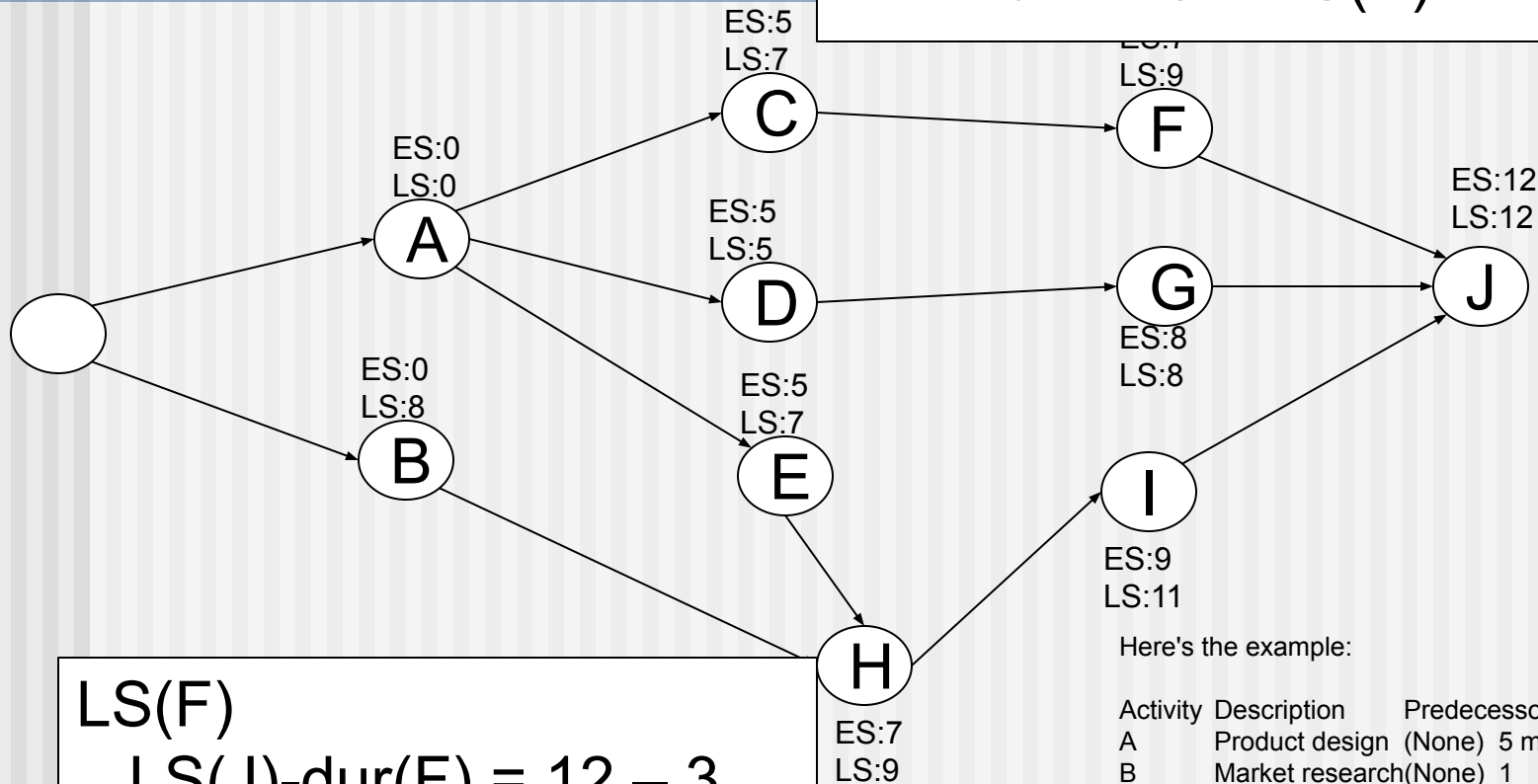
$$LS(A) =$$

$$LS(C) - \text{dur}(A) = 7 - 5 = 2$$

$$LS(D) - \text{dur}(A) = 5 - 5 = 0$$

$$LS(E) - \text{dur}(A) = 7 - 5 = 2$$

$$\text{Minimum} = 0 = LS(A)$$



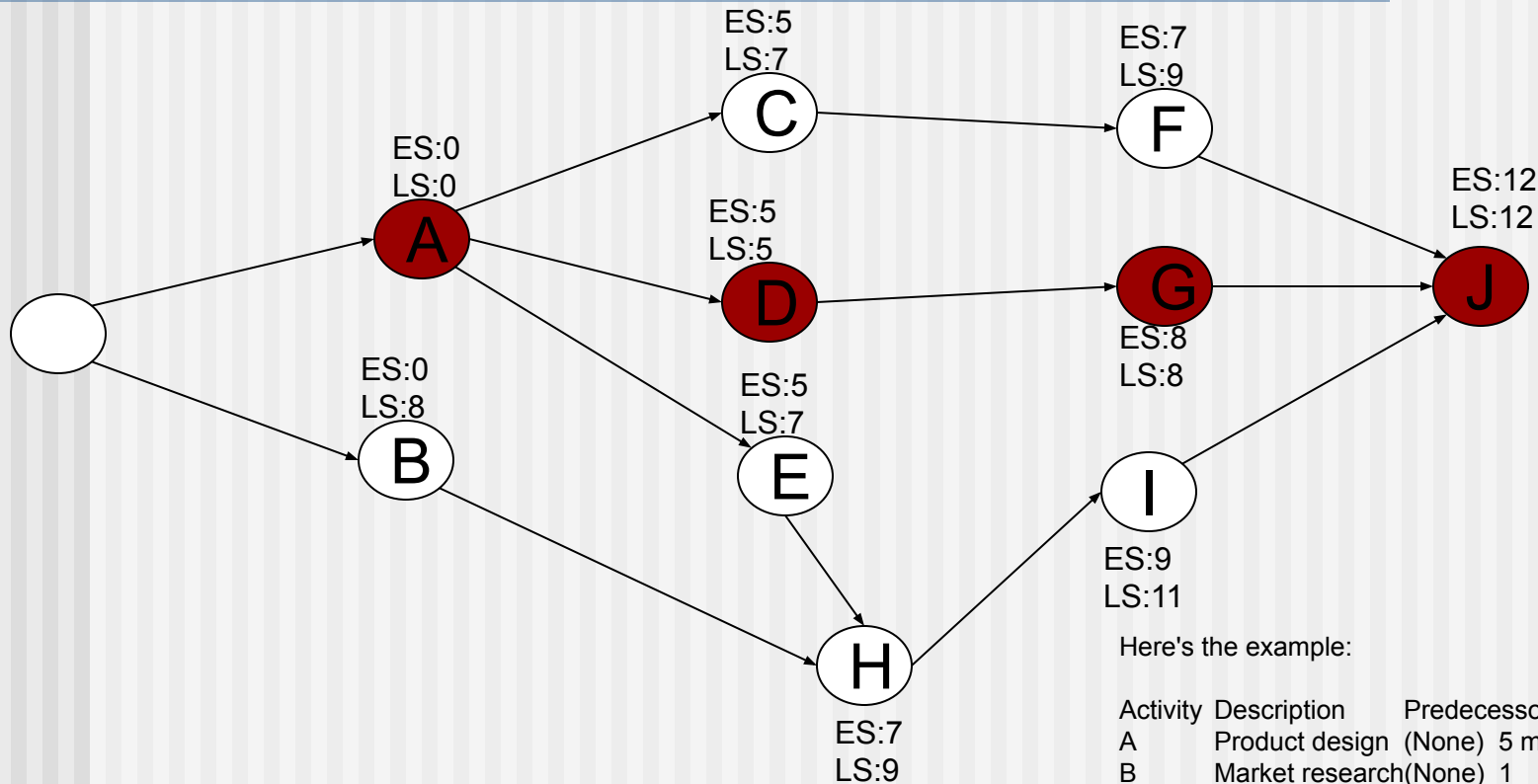
$$LS(F)$$

$$LS(J) - \text{dur}(F) = 12 - 3 = 9$$

Here's the example:

Activity	Description	Predecessor	Duration
A	Product design	(None)	5 months
B	Market research	(None)	1
C	Production	A	2
D	Product model	A	3
E	Sales brochure	A	2
F	Cost analysis	C	3
G	Product testing	D	4
H	Sales training	B, E	2
I	Pricing	H	1
J	Project report	F, G, I	1

Example Node Network



Here's the example:

Activity	Description	Predecessor	Duration
A	Product design	(None)	5 months
B	Market research	(None)	1
C	Production	A	2
D	Product model	A	3
E	Sales brochure	A	2
F	Cost analysis	C	3
G	Product testing	D	4
H	Sales training	B, E	2
I	Pricing	H	1
J	Project report	F, G, I	1

Game Development Exercise

Find the critical path

TASK	DURATION (days)	PREDECESSORs
A Graphics Engine	14	
B Sound Engine	5	I
C Music Engine	5	J
D Input Engine	10	A
E Gameplay/general programming	31	B, C, D
F Physics	7	E
G 2D Artwork	14	
H 3D Artwork	21	G
I Sound Effects	14	
J Music	9	
K Level Design	21	F, H

Scheduling Rules of Thumb

- One person should always edit the schedule (you!)
 - If you have two people that need to, create two files and link them together
- Keep it simple and useful
- Level your resources
- Share the schedule with your team
- 40-20-40; coding is 20% of the effort

--	--

Schedule Example

- Lets try to schedule this work among our three developers “John, Mary, Carl”

TASK	DURATION (days)	PREDECESSORs
A Graphics Engine	14	
B Sound Engine	5	I
C Music Engine	5	J
D Input Engine	10	A
E Gameplay/general programming	31	B, C, D
F Physics	7	E
G 2D Artwork	14	
H 3D Artwork	21	G
I Sound Effects	14	
J Music	9	
K Level Design	21	F, H

Scheduling Steps

- Add in all the tasks (preferably in a hierarchy)
- Add in all the dependencies
- Break down large tasks into smaller tasks. Optimally (in Dan Fleck's opinion) you want to schedule so the duration of each smallest task is at most 3-5 days
- Assign people (resources) to tasks
- Level your resources

--	--

Classic Mistakes

- Overly optimistic schedule
- Failing to monitor schedule
- Failing to update schedule
- Adding people to a late project
- Failure to manage expectations of others
- Leaving out a task

