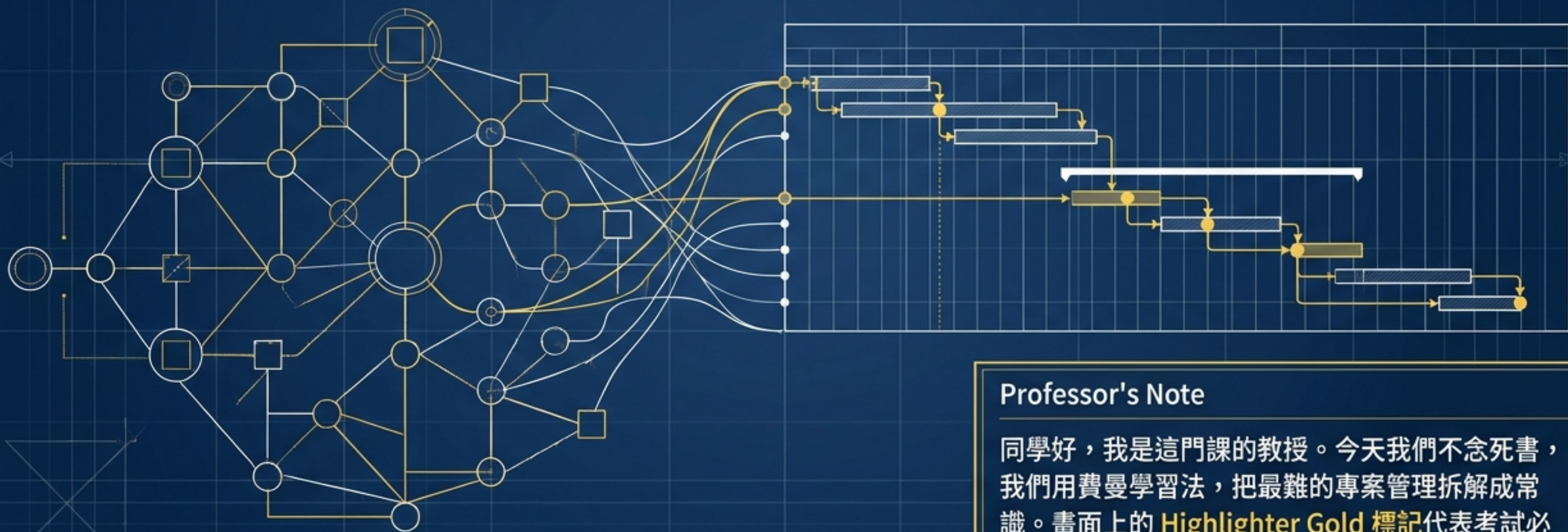


軟體專案管理完全掌握

從零基礎到精通：專案章程、Gantt Chart 與 PERT Chart 要徑解析



Professor's Note

同學好，我是這門課的教授。今天我們不念死書，我們用費曼學習法，把最難的專案管理拆解成常識。畫面上的 **Highlighter Gold** 標記代表考試必考的英文專有名詞，請務必記熟。

專案管理的生命週期 (Project Management Phases)

Initiation
(啟動)



定義專案的初始概念與商業價值。

Planning
(規劃)



制定詳細計畫 (本課程核心，包含 Gantt 與 PERT 排程)。

Execution
(執行)



團隊開始實際動工編寫程式。

Control
(控制)



監控進度、預算與軟體品質。

Enclosure/Closure
(結案)



完成測試並正式交付系統。

Professor's Note

專案生命週期就像蓋房子：(1) Initiation 是決定要蓋房子，(2) Planning 是畫設計圖，(3) Execution 是請工人來蓋，(4) Control 是監工確保沒偷工減料，(5) Closure 是交屋給你。

選擇開發方法 (Selecting the Development Methodology)



Clarity of User Requirements (使用者需求清晰度)

需求不明確時選 Agile；
明確時可選 Waterfall。



Familiarity with Technology (技術熟悉度)

團隊對技術越陌生，越需要
迭代測試。



System Complexity (系統複雜度)

複雜系統需要更嚴謹的
架構規劃。



System Reliability (系統可靠度)

高可靠度需求(如醫療)，
需完整的文件與測試階
段。



Short Time Schedules (專案時程緊迫性)

時間極短時，可能需採用
快速原型開發。



Schedule Visibility (進度可見度)

客戶是否需要每週看到
進度？



Exam Focus Box

必考題！ 考試常問：如果「使用者需求不清晰」或「使用不熟悉的技術」，該用什麼方法？
答案通常是 Agile (敏捷開發) 或 Extreme Programming (極限編程)。

專案的基石：專案章程 (The Project Charter)



Professor's Note

這是專案的『憲法』。在任何一行程式碼寫下去之前，這份文件確保所有人的期待是一致的。

時間的視覺化：甘特圖 (Gantt Chart Basics)

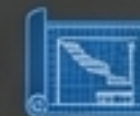
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	T	U	V	W	X	Y	Z	A	S	T
3	[Project Name]																											
4	[Company Name]																											
5	Today's Date: <u>2/24/2009</u> Tuesday (vertical red line)																											
6	Project Lead: <u>John Doe</u>																											
7	Start Date: <u>1/5/2009</u> Monday																											
8	First Day of Week (Mon=2): 2																											
					Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining	05 - Jan - 09	12 - Jan - 09	19 - Jan - 09	26 - Jan - 09	02 - Feb - 09	08 - Feb - 09	16 - Feb - 09	23 - Feb - 09	03 - Mar - 09	08 - Mar - 09	18 - Mar - 09	23 - Mar - 09	30 - Mar - 09	06 - Apr - 09	13 - Apr - 09	20 - Apr - 09	27 - Apr - 09	04 - May - 09	
9	WBS	Tasks	Task Lead	Start	End	Duration	% Complete	Working Days	Days Complete	Days Remaining																		
10	1	Task Category 1	John	1/03/09	3/33/09	75	73%	53	52	23																		
11	1.1	Sub Tasklevel 2		1/03/09	1/20/09	18	100%	12	18	0																		
12	1.2	Sub Tasklevel 2		1/21/09	2/19/09	80	95%	22	28	2																		
13	1.2.1	Sub Tasklevel 3		1/22/09	1/31/09	10	20%	7	2	8																		
14	1.2.2	Sub Tasklevel 3		1/22/09	2/01/09	10	20%	6	2	8																		
15	1.3	Sub Tasklevel 2		1/22/09	2/09/09	10	95%	13	18	1																		
16	1.4	Sub Tasklevel 2		2/10/09	3/18/09	37	80%	27	18	19																		
17	2	Task Category 2	Jane	3/01/09	5/12/09	73	13%	53	0	64																		
18	2.1	Sub Tasklevel 2		3/01/09	3/17/09	17	50%	12	8	9																		
19	2.2	Sub Tasklevel 2		3/18/09	4/27/09	17	30%	12	8	12																		
20	2.3	Sub Tasklevel 2		3/19/09	4/25/09	28	0%	28	0	28																		
21	2.4	Sub Tasklevel 2		4/25/09	5/12/09	23	0%	20	0	28																		
22	3	Task Category 3	Bill	4/25/09	8/22/09	100	0%	70	0	100																		
23	3.1	Sub Tasklevel 2		4/25/09	5/11/09	17	0%	11	0	17																		
24	3.2	Sub Tasklevel 2		5/12/09	5/28/09	17	0%	13	0	17																		
25	3.3	Sub Tasklevel 2		5/29/09	7/05/09	39	0%	26	0	38																		
26	3.4	Sub Tasklevel 2		7/05/09	8/29/09	29	0%	20	0	29																		
27	4	Task Category 4	Bill	4/25/09	8/22/09	100	0%	70	0	100																		
28	4.1	Sub Tasklevel 2		4/25/09	5/11/09	17	0%	11	0	17																		
29	4.2	Sub Tasklevel 2		5/12/09	5/28/09	17	0%	13	0	17																		
30	4.3	Sub Tasklevel 2		5/29/09	7/25/09	38	0%	26	0	38																		
31	4.4	Sub Tasklevel 2		7/05/09	8/25/09	28	0%	20	0	28																		

Definition: A horizontal bar chart illustrating a project schedule against time (以水平長條圖顯示任務在時間軸上的進度).

Task Relationships (任務關係定義)

Sequential (循序): 必須有嚴格的先後順序。

Parallel (平行): 可以同時進行的任務。



Professor's Note

循序就像：必須先打地基，才能蓋牆壁。

平行就像：客廳刷漆和臥室鋪地板可以同時請兩批工人進行。

甘特圖的優缺點 (Pros & Cons of Gantt Charts)

Advantages (優點)

1. **Global Understanding (容易理解):**
畫面直觀，高階主管一眼就能看懂進度。

2. **Clear Current Status (狀態清晰):**
透過 Today Line 能立刻知道誰落後。

Disadvantages (缺點 - 考試重點)

1. **Poor Dependency Visibility (無法顯示任務依賴性):**
看不出一個任務延遲會如何『連鎖影響』其他任務。

2. **Unwieldy for Large Projects (大型專案難以管理):**
超過 30 個活動畫面就會變得極度混亂。

3. **Misrepresents Workload (無法反映實際工作量):**
兩條一樣長的橫槓，可能一個需要 10 人，一個只要 1 人。



Professor's Note

甘特圖的問題在於它是一張死圖表。它告訴你專案何時發生，但沒有告訴你任務之間『誰綁死誰』的數學邏輯。這就是為什麼我們需要 PERT。

核心引擎：PERT Chart 與 AON 節點 (The PERT Engine & AON Node)

PERT (Program Evaluation and Review Technique):

A statistical tool to analyze tasks and find the minimum time needed.

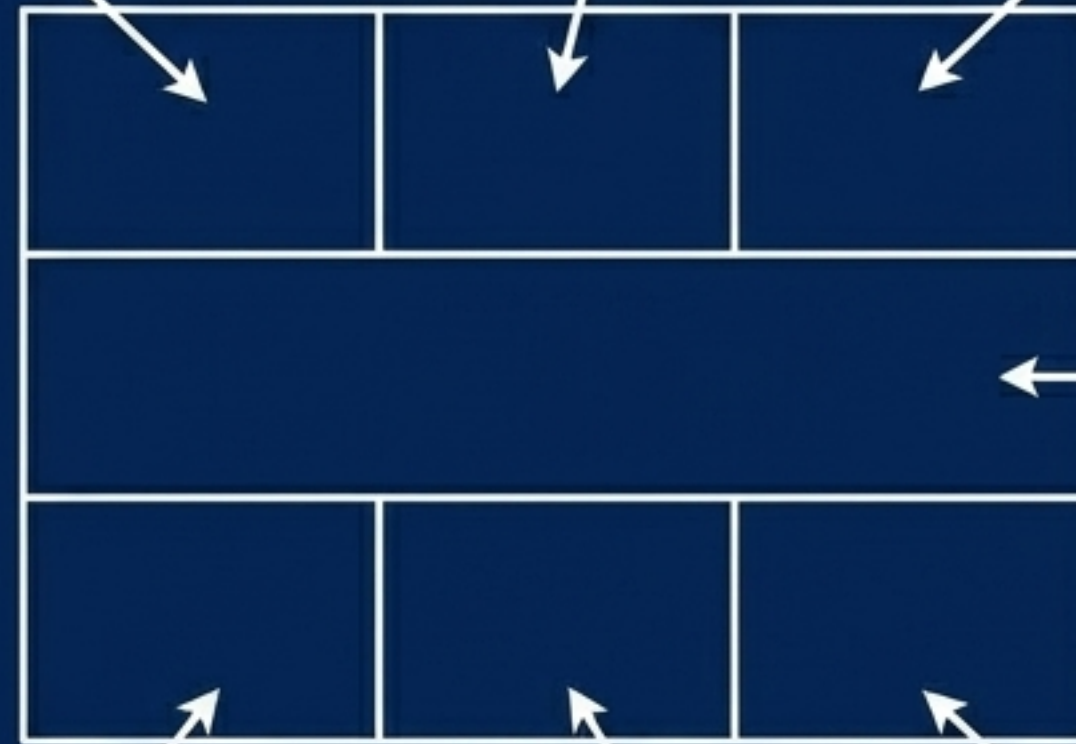
AON (Activity on Node):

任務寫在節點上。

ES (Early Start):
最早開始時間

Duration: 工期

EF (Early Finish):
最早結束時間



Task Name:
任務名稱

LS (Late Start):
最晚開始時間

Slack:
浮動時間

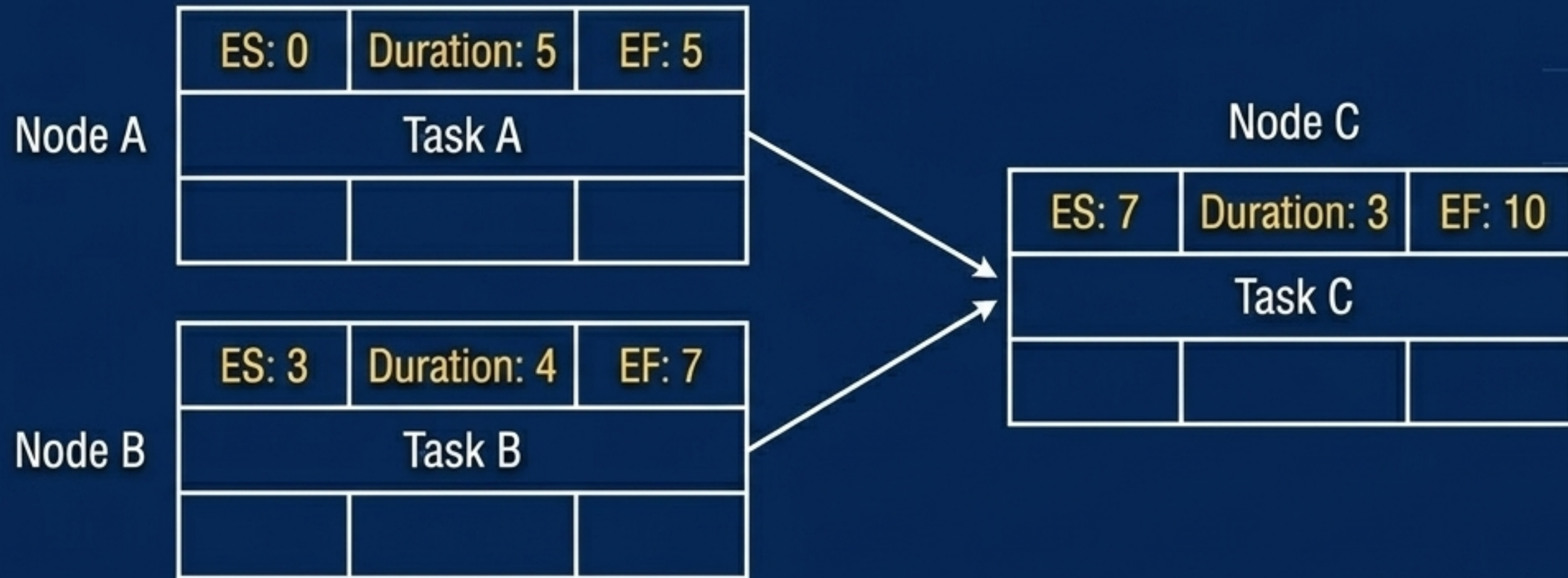
LF (Late Finish):
最晚結束時間



Professor's Note

把這個方塊當成火車時刻表。上面那一排是你『最理想、最早』能到達的時間；下面那一排是為了不讓整條鐵路大誤點，你『最底線、最晚』必須到達的時間。

PERT 運算第一步：順向推算 (Forward Pass)



Rule 1:
 $ES + \text{Duration} = EF$
(最早開始 + 工期 = 最早結束)

Rule 2:
前一個任務的 EF，
就是下一個任務的 ES。

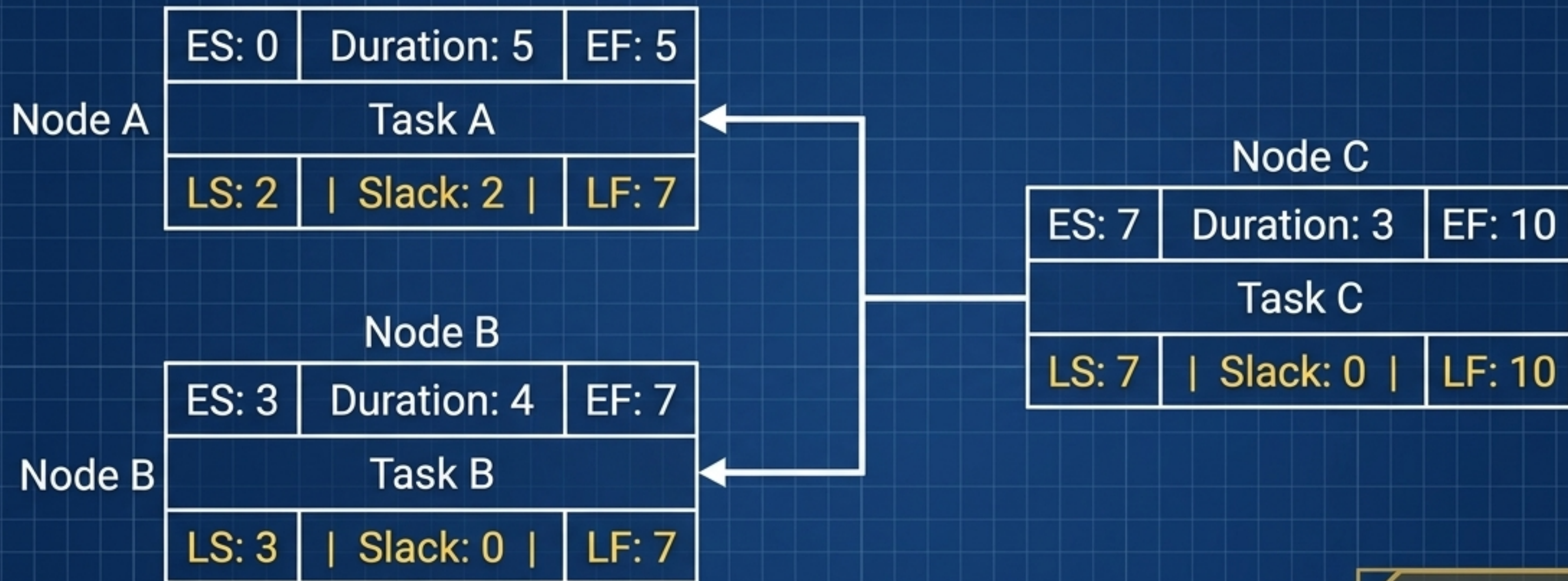
Rule 3 (The Merge Rule):
如果前面有多個任務匯
集，取**最大的 EF**！



Professor's Note

為什麼匯流時要取『**最大值**』？想像你要出門，必須等『**最慢**』化好妝的家人準備好，全家才能一起出發。你必須等所有前置任務都做完！

PERT 運算第二步：逆向推算 (Backward Pass)



Start Point:
最後一個任務的
LF = 它的 EF。

Rule 1:
 $LF - \text{Duration} = LS$
(最晚結束 - 工期
= 最晚開始)

Rule 2:
後一個任務的 LS，
變成前一個任務的
LF。

Rule 3
(The Split Rule):
逆推遇到分岔時，
取**最小的 LS**！

Professor's Note
為什麼逆推分岔時要取『**最小值**』？因為你身為主管，不能害後續『**任何一個**』任務遲到，所以你要配合要求最嚴格（最早需要開始）的那個任務！

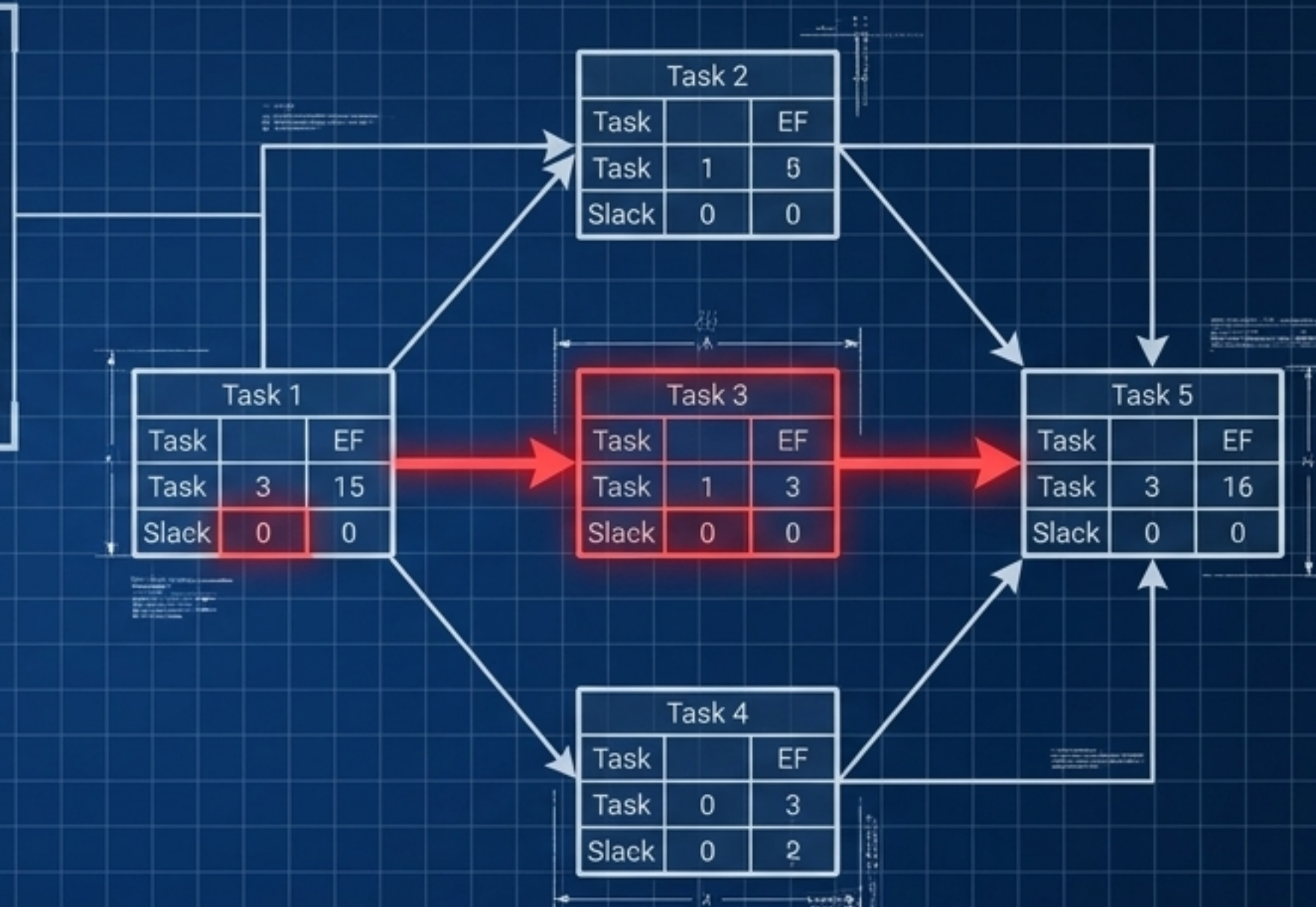
尋找要徑 (Identifying the Critical Path & Slack Time)

Free Float / Slack Time (浮動時間)
任務可以拖延卻不影響總工期的時間。

Formula: Slack = LF - EF

■ Critical Path (CPM)

- 由 Slack = 0 的任務組成的路徑。
- 專案中『最長』的路徑 (Longest path)。
- 決定專案『最短』完成時間的路徑 (Shortest completion time)。



Exam Focus

考試必默寫：Activities with Slack=0 forms the Critical Path.

Professor's Note: 想像你要趕上最後一班火車，關鍵路徑上的任務就是你『完全沒有浪費時間』的步驟。任何一個步驟慢了，你就會錯過火車（專案延誤）。其他有 Slack 的任務就像是你有額外時間買咖啡或去廁所，不影響最終趕上火車。

實戰演練：PERT 要徑完整解析 (Exam Walkthrough)

ES	Duration	EF
A		
4	Slack: 4	7

ES	Duration	EF
D		
7	Slack: 4	LF

16	Duration	19
F		
16	Slack: 0	19

8	Duration	16
E		
8	Slack: 0	16

0	Duration	5
B		
0	Slack: 0	5

5	Duration	8
C		
5	Slack: 0	8

16	Duration	17
H		
16	Slack: 0	17

17	Duration	19
I		
17	Slack: 0	19

8	Duration	11
G		
13	Slack: 5	16

Step 1: Calculate ES/EF (Forward Pass).
Node E takes largest EF from C & D.

Step 2: Calculate LS/LF (Backward Pass).

Step 3: Calculate Slack (LF - EF).

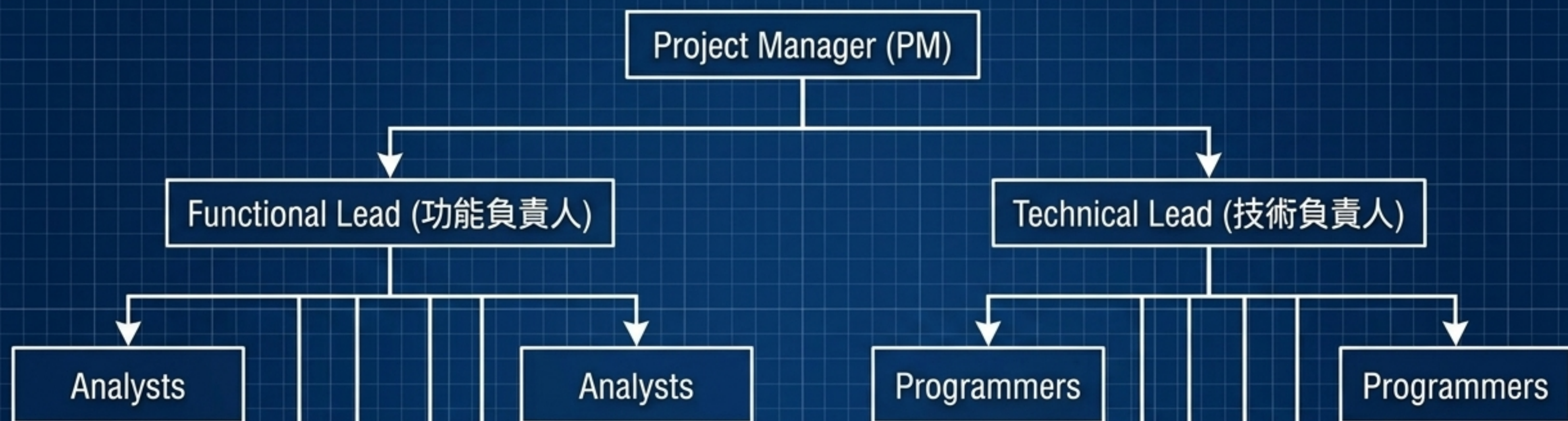
Result: Nodes B, C, E, F, H, I have
Slack = 0.

Exam Insight: A project can have
MULTIPLE critical paths!

Final Answer: Critical paths are B-C-E-F
and B-C-E-H-I. 專案最短完成時間為 19
週。

Professor's Note: 關鍵路徑的本質是「時間瓶頸」。當多條路徑同時成為瓶頸，任何一條上的延誤都會導致專案總時程變長。就像兩條塞車的車道同時到達終點，哪條都不能再慢了。

專案的人力配置 (Staffing the Project)



Staffing Plan (人力配置計畫)

描述專案所需人員的角色與數量。

Ideal Team Size (理想團隊規模)

8-10 people.
階層化管理降低溝通複雜度。

Key Roles (關鍵角色)

Functional Lead
管理分析師，對業務負責

Technical Lead
管理工程師，對技術負責

Professor's Note: 注意！『加人不一定變快』。在專案後期加入新人，反而會因為增加培訓與溝通成本，導致進度更慢 (軟體工程著名的 Brooks's Law)。

激勵與衝突管理 (Motivation & Conflict Handling)



Motivation (激勵團隊)

- Caution with Money: 謹慎使用金錢獎勵，效果短暫。
- **Intrinsic Rewards** (內在獎勵 - 效果最好)：
 - Recognition (認可)
 - Achievement (成就感)
 - Responsibility (賦予責任)
 - Advancement (晉升機會)
 - Learn new skills (學習新技能)



Handling Conflict (處理衝突)

- Clearly define plans (明確定義計畫)。
- Use the **Project Charter**: 利用專案章程作為團隊的最高法規與行為準則。
- Forecast priorities ahead of time: 提前預測優先順序，避免資源搶奪。

Professor's Note:

最好的 PM 不只是懂數學算要徑，更懂人心。用章程防範衝突，用成就感驅動進度。

總結：排程工具大比拼 (Gantt vs. PERT Matrix)

Comparison Matrix	Gantt Chart (甘特圖)	PERT/CPM Chart (網狀圖)
Best For (最佳用途)	Executive presentations, simple visual tracking. (向高階主管報告、進度追蹤)	Mathematical analysis, identifying bottlenecks. (數學分析、尋找瓶頸與死線)
Time Visualization (時間視覺化)	Excellent. Uses a real timeline and calendar dates. (優秀，有真實時間軸)	Poor. Nodes don't physically scale to time. (極差，方塊大小不代表時間)
Dependency Tracking (任務依賴性追蹤)	Poor. Hard to see the domino effect of delays. (極差，看不出連鎖反應)	Excellent. Strict logical flow and critical path. (優秀，要徑關係一目了然)

Professor's Note: 掌握章程定目標，甘特圖看時程，PERT圖算要徑。這三步，就是滿分專案經理的必備武器。下課，祝各位考試順利！