

網圖

NETWORK

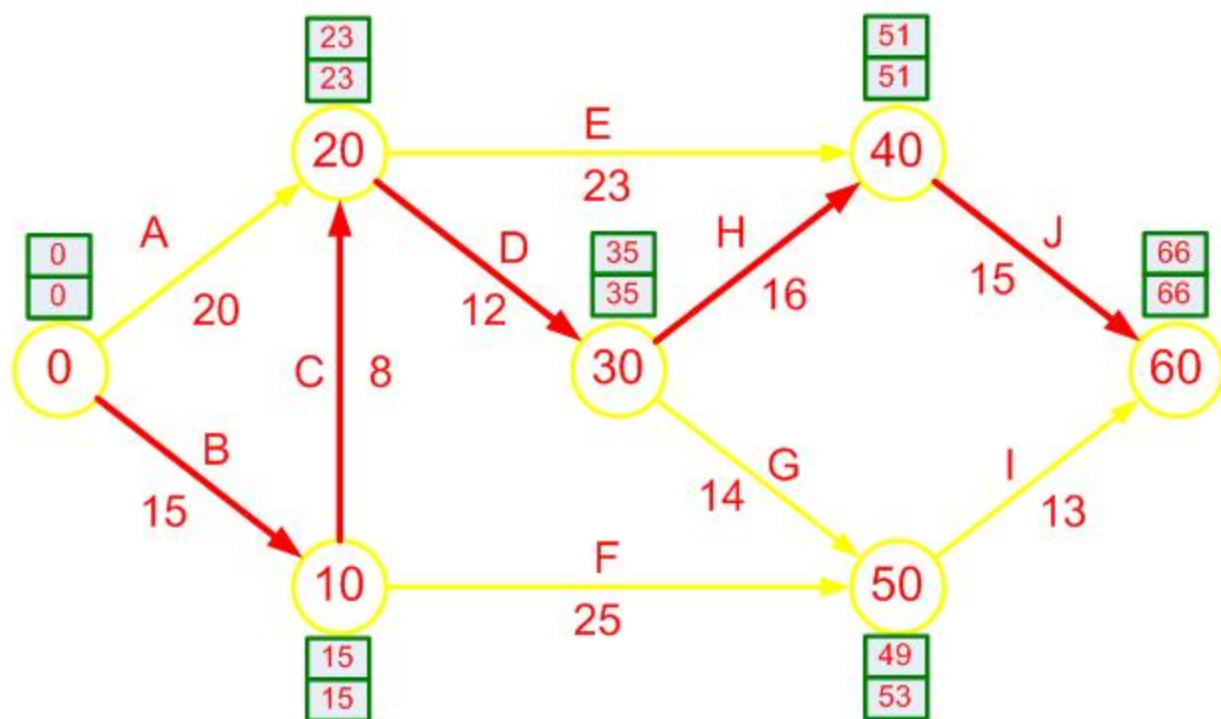
箭線圖法

〈 Arrow Diagramming
Method 〉

網圖〈Network〉

在營建管理上的意義

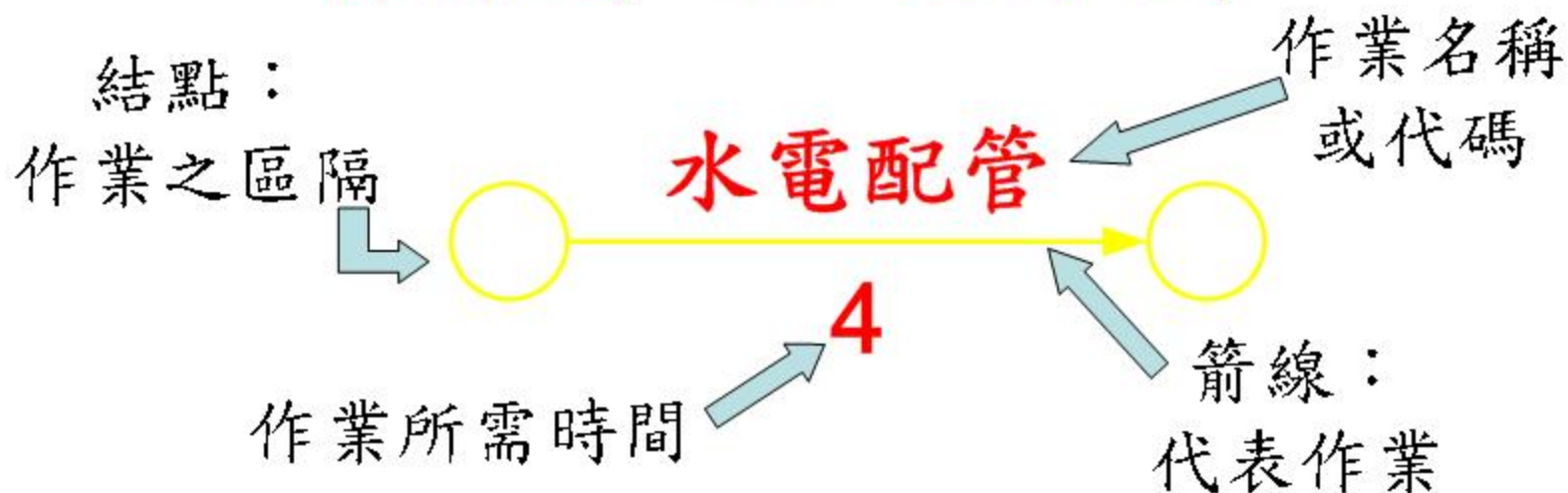
- 將專案的各作業及其前後順序
- 以圖形方式的表示
- 製作簡單化、清楚化、資訊化的施工進度計劃



箭線圖法 (Arrow Diagramming Method) AOA (Activity-on-Arrow)

- 箭線：代表作業項目
- 結點：區隔作業之先後邏輯關係
代表網圖上的一個時間點

箭線圖之註記方式



箭線圖法邏輯上之基本規則

- 每一條箭線均代表一個作業，而且只代表一個作業
- 網路繪製之方向須從左至右，箭頭可以向上、下及向右邊推行
- 結點為一個作業之開始和結束的時間點，也就是兩箭之交合點
- 除非在時間標尺的狀況下，箭線的長度並不代表時間的長短

箭線圖法邏輯上之基本規則

- 所有的箭線均由箭頭經由結點連接至後續作業之箭尾
- 所有的箭線圖僅有一個開始及一個結束結點
- 賦予結點識別碼時，以較小的號碼放在尾端，而箭頭端則為較大的號碼，且網圖上的結點識別碼不可重複
- 為了方便在網路圖上加入其他作業，結點的編號最好以跳號的方式編製

箭線圖法邏輯上之基本規則

- 箭線圖必須包括所有該計畫的全部作業項目及所有作業的順序關係
- 結點為作業之間結束與開始的邏輯關係，所有的先行作業完成之後，後續的作業才可以開始
- 兩個結點之間不可直接連結兩條以上的箭頭
- 作業名稱及所需資源標記於箭線之上，而作業所需時間記在箭線之下

箭線圖之基本規則_圖例

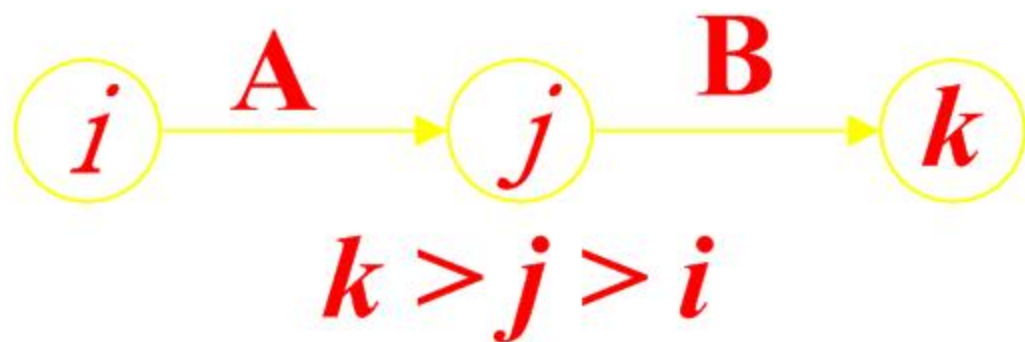
1.作業基本格式：



箭線圖之基本規則_圖例

2.作業順序規則：

A完成後，**B**作業才可開始



箭線圖之基本規則_圖例

3. 平行作業：

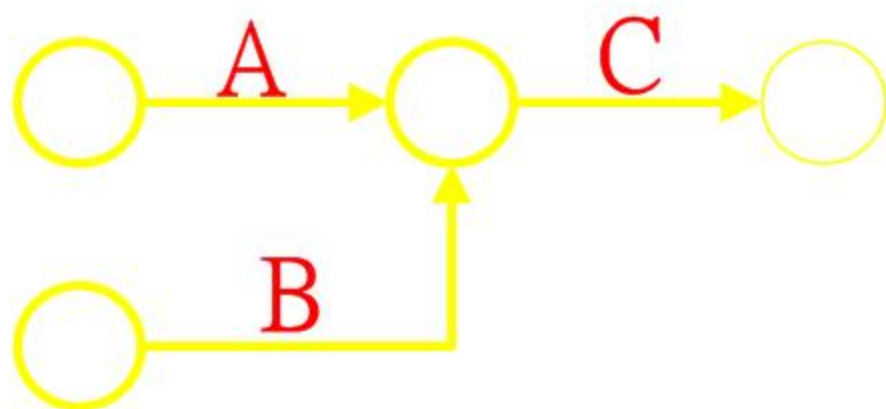
兩作業可同時進行；但不需要同時開始或結束，且此兩作業之間也無先後依附之關係



箭線圖之基本規則_圖例

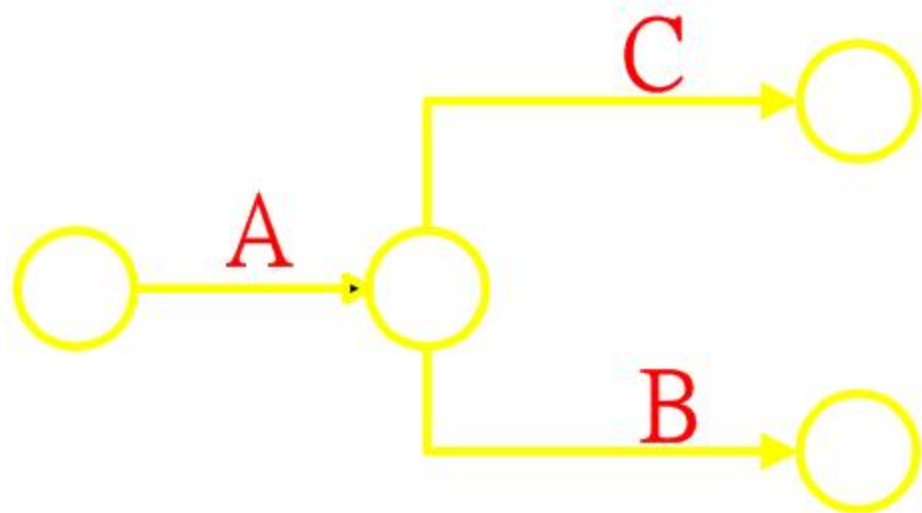
4. 匯集作業：

A和**B**都完成後，**C**才可以開始



箭線圖之基本規則_圖例

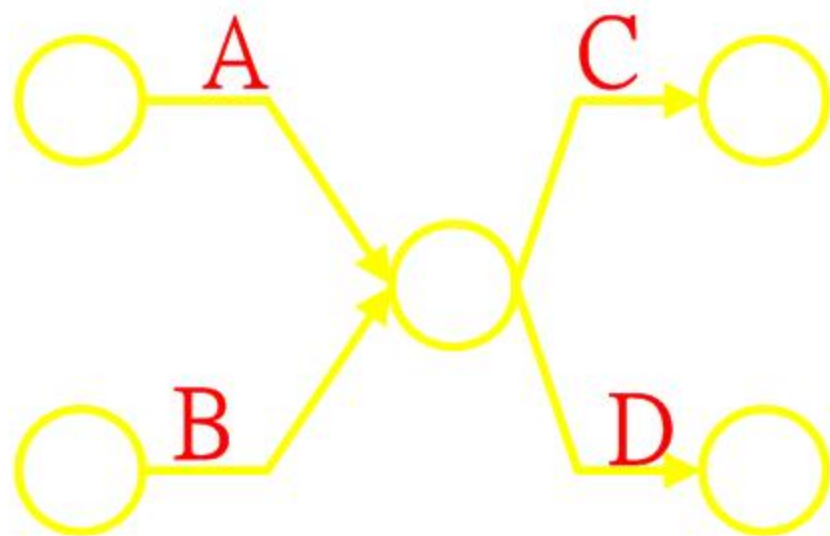
5.分岐作業：A完成後C和B方可開始



箭線圖之基本規則_圖例

6. 十字型作業：

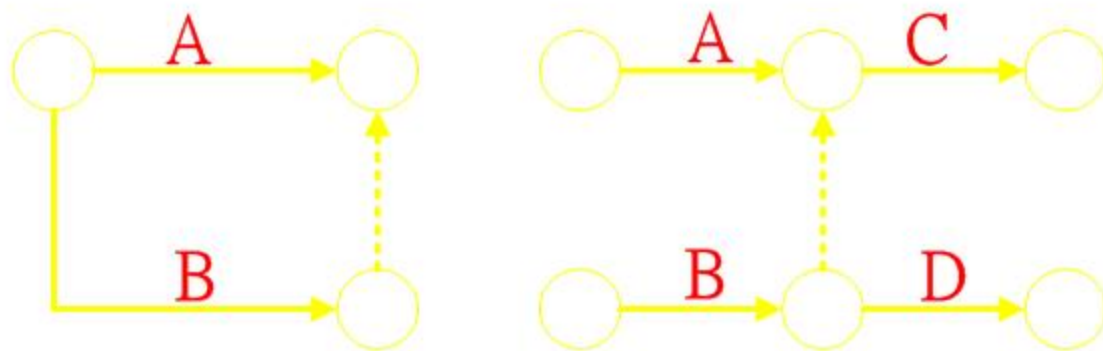
作業**C**及**D**均須等作業**A**及**B**的完成後可開始



箭線圖之基本規則_圖例

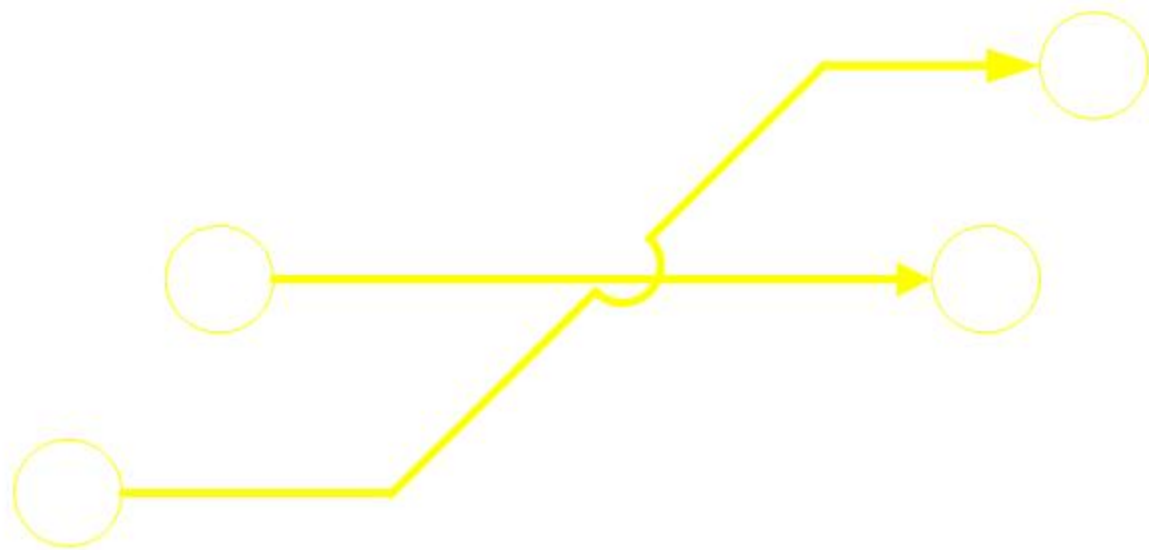
7. 虛作業 (Dummy Activity) :

- 避免兩結點間有二個作業共用同一結點碼。
- 兩個作業路線為交叉相關時；如C須等待A，B皆完成後才能開始，而D只要B完成就可開始。



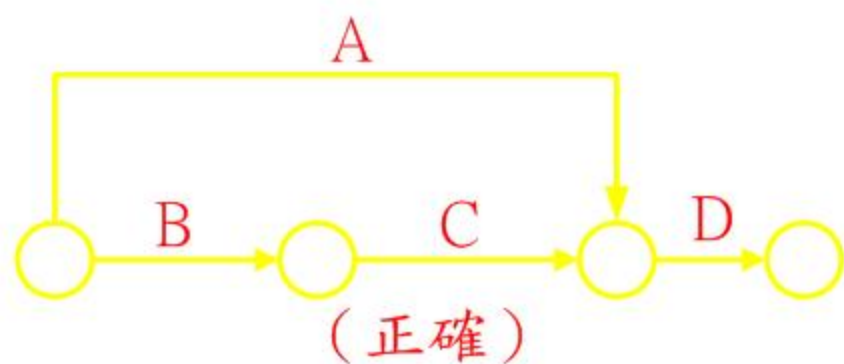
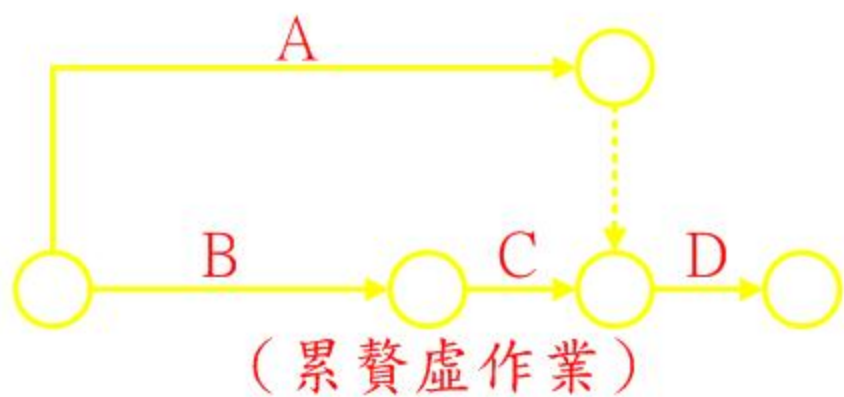
箭線圖之基本規則_圖例

8. 交叉作業：以符號來表示交叉之情形



箭線圖常見的錯誤

- 太多虛作業



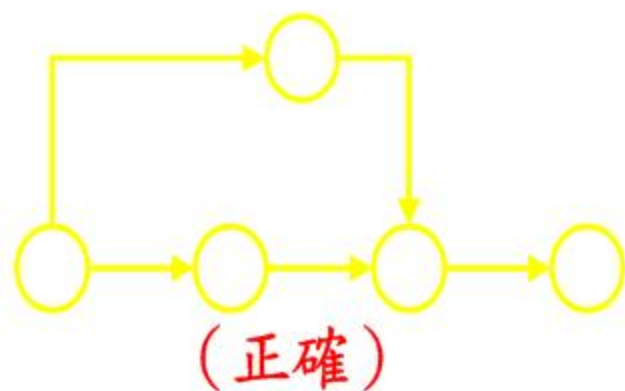
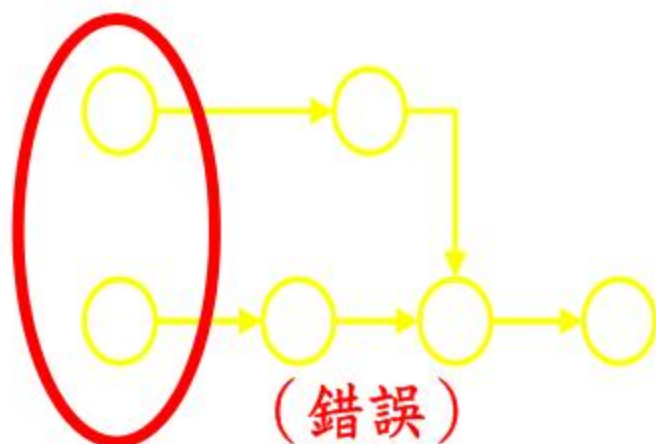
箭線圖常見的錯誤

- 作業箭頭忘了劃
- 作業結點之圓圈未劃
- 最初或最尾之結點圓圈未劃

箭線圖常見的錯誤

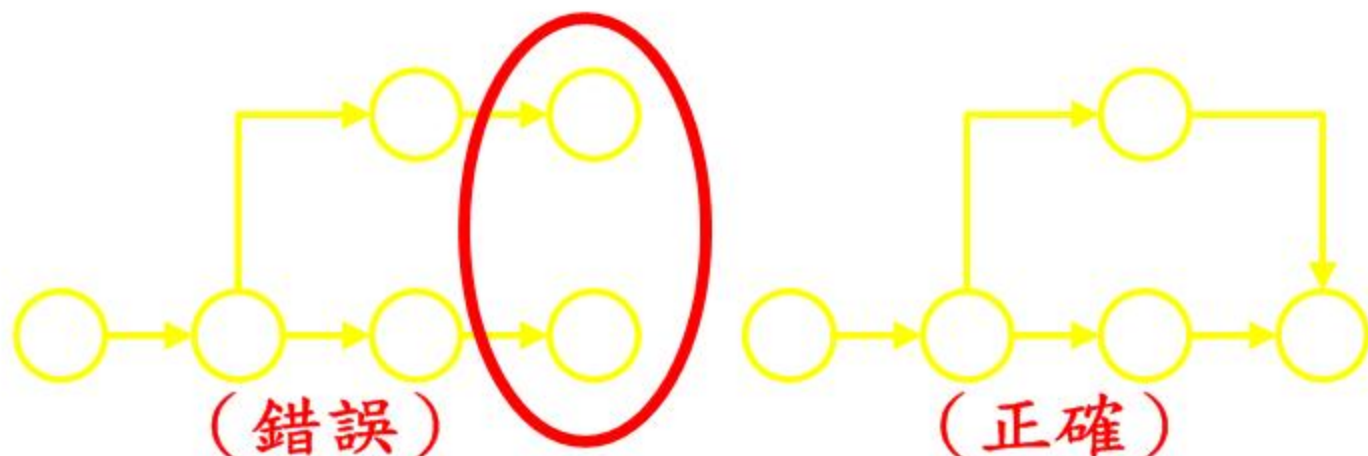
- 雙頭或雙尾

- 雙頭：



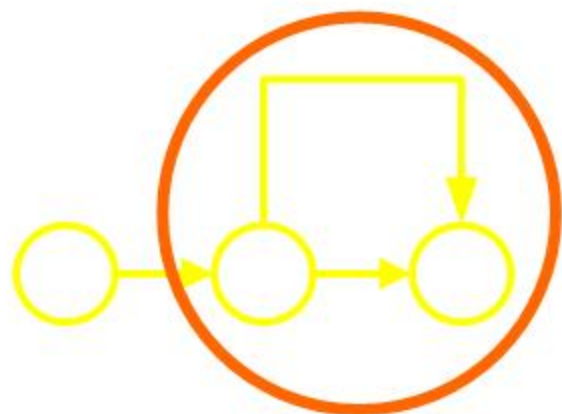
箭線圖常見的錯誤

- 雙頭或雙尾
 - 雙尾：

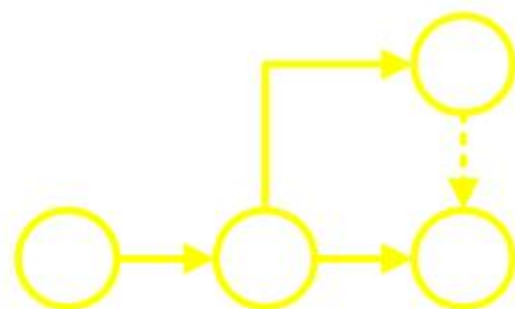


箭線圖常見的錯誤

- 共用編碼



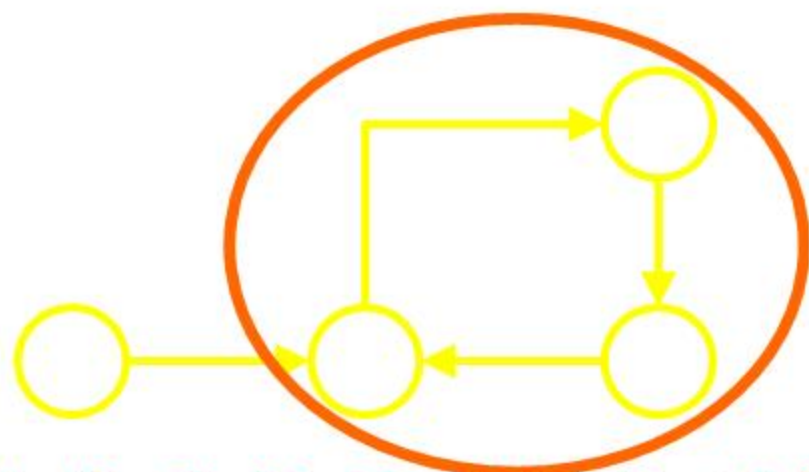
(錯誤)



(正確)

箭線圖常見的錯誤

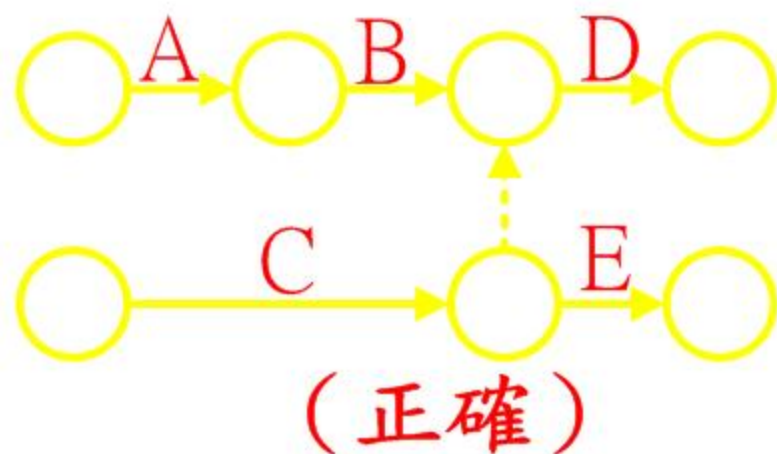
- 迴路



(作業邏輯需重新檢討)

箭線圖常見的錯誤

- 邏輯錯誤
 - 如: A,C為起始作業
 - A在B前
 - D在B,C前
 - E只在C後



箭線圖繪製練習

| 作業 | 後續作業 |
|----|------|
| A | F,G |
| B | F |
| C | D,E |
| D | F |
| E | H,I |
| F | H,I |
| G | — |
| H | — |
| I | — |

ADM日程計算

- 日程計算之階段：
 - 求得結點時間
 - 以結點時間推求作業日程(數字)
 - 將作業日程轉換為日曆天日程

結點時間計算



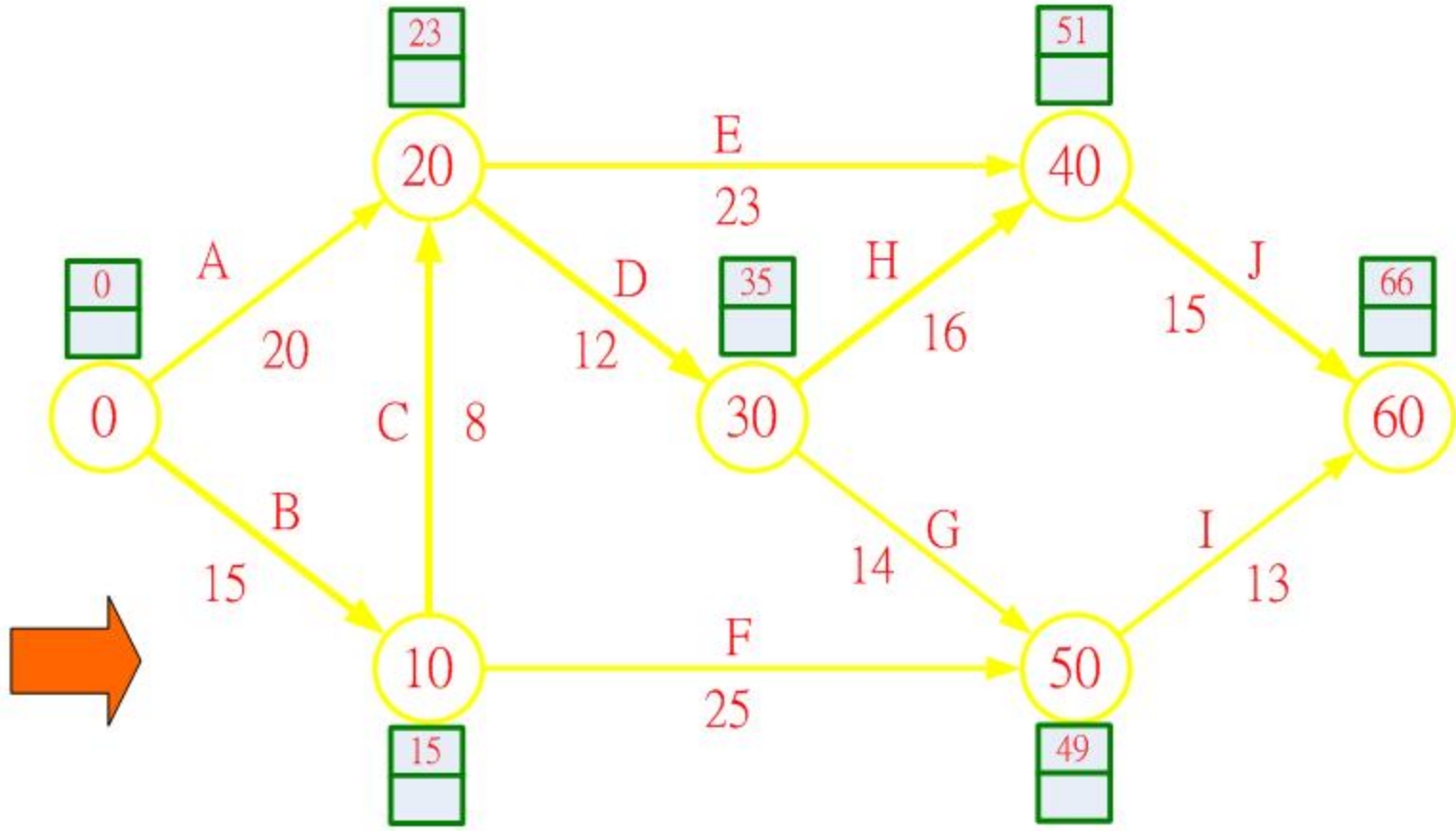
- EET_i : i 的最早結點時間 (Early Event Time)
- LET_i : i 的最遲結點時間 (Late Event Time)

前進計算：求最早結點時間



- $EET_j = EET_i + \text{作業時間 (Dur)}$
- 如有一個以上的作業同時進入結點 j 時，則取其中**最大者**，為該結點的最早結點時間

前進計算：求最早結點時間

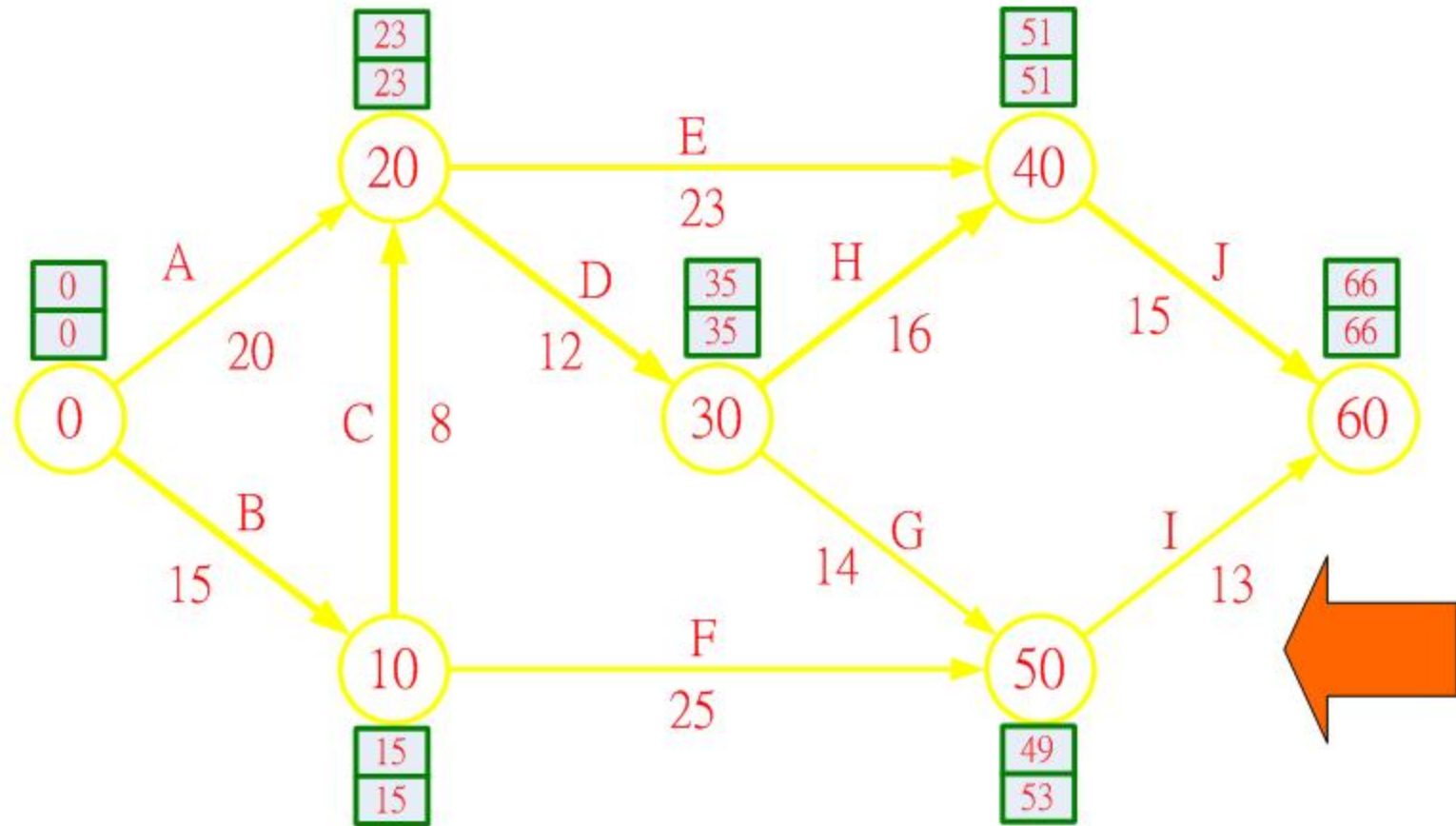


後退計算法:求最遲結點時間



- $LET_i = LET_j - \text{作業時間 (Dur)}$
- 如有一個以上的作業同時由結點 i 離開時，則取其中**最小者**為該結點的最遲結點時間

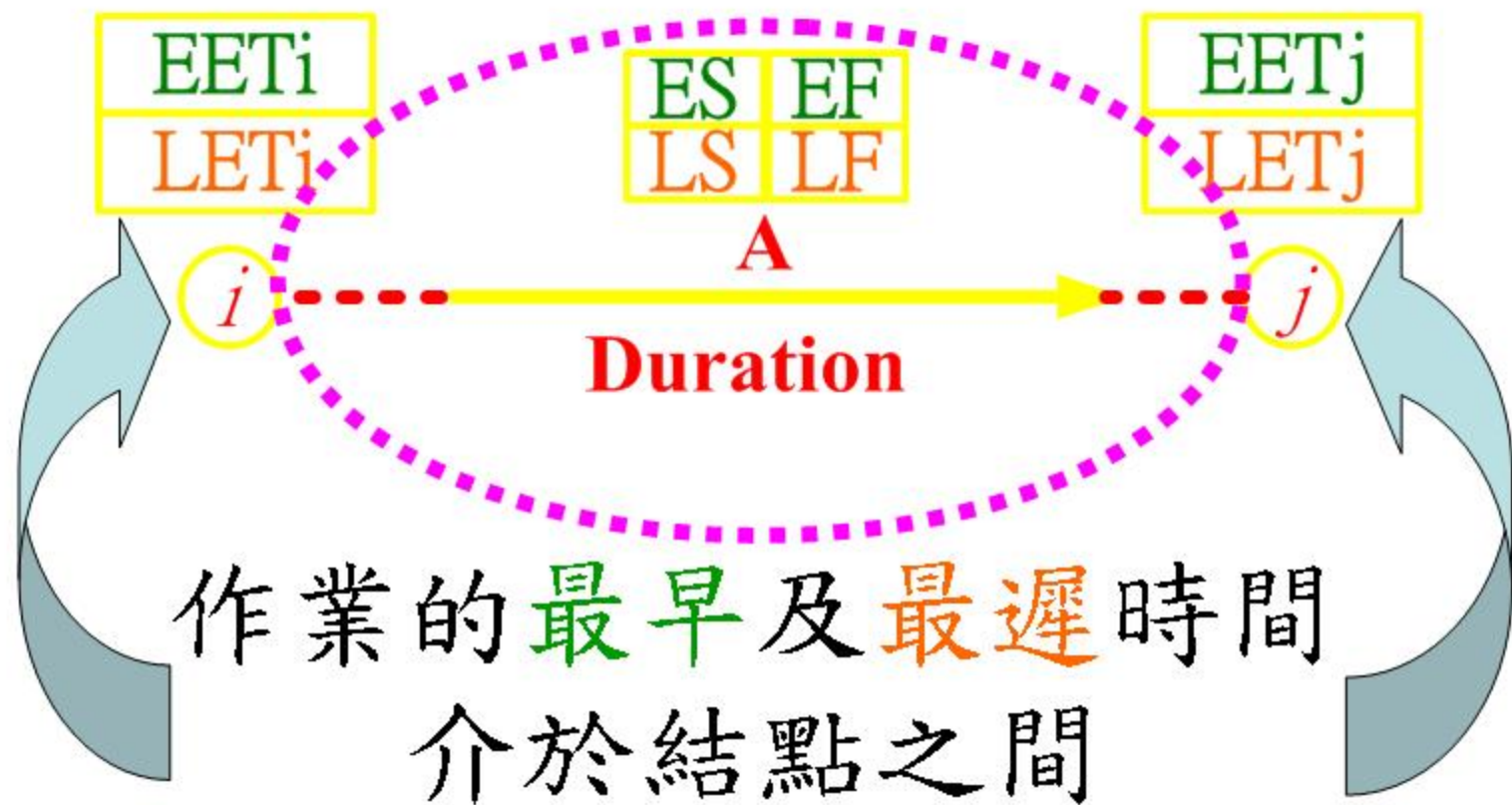
後退計算：求最遲結點時間



作業日程計算 (Activity Time Computations)

- 一般網圖作業的開始和結束日程：（四時）
 - **ES** (Early Start) : 作業最早開始日程
 - **EF** (Early Finish) : 作業最早結束日程
 - **LS** (Late Start) : 作業之最遲開始日程
 - **LF** (Late Finish) : 作業最遲結束日程

作業日程計算 (Activity Time Computations)

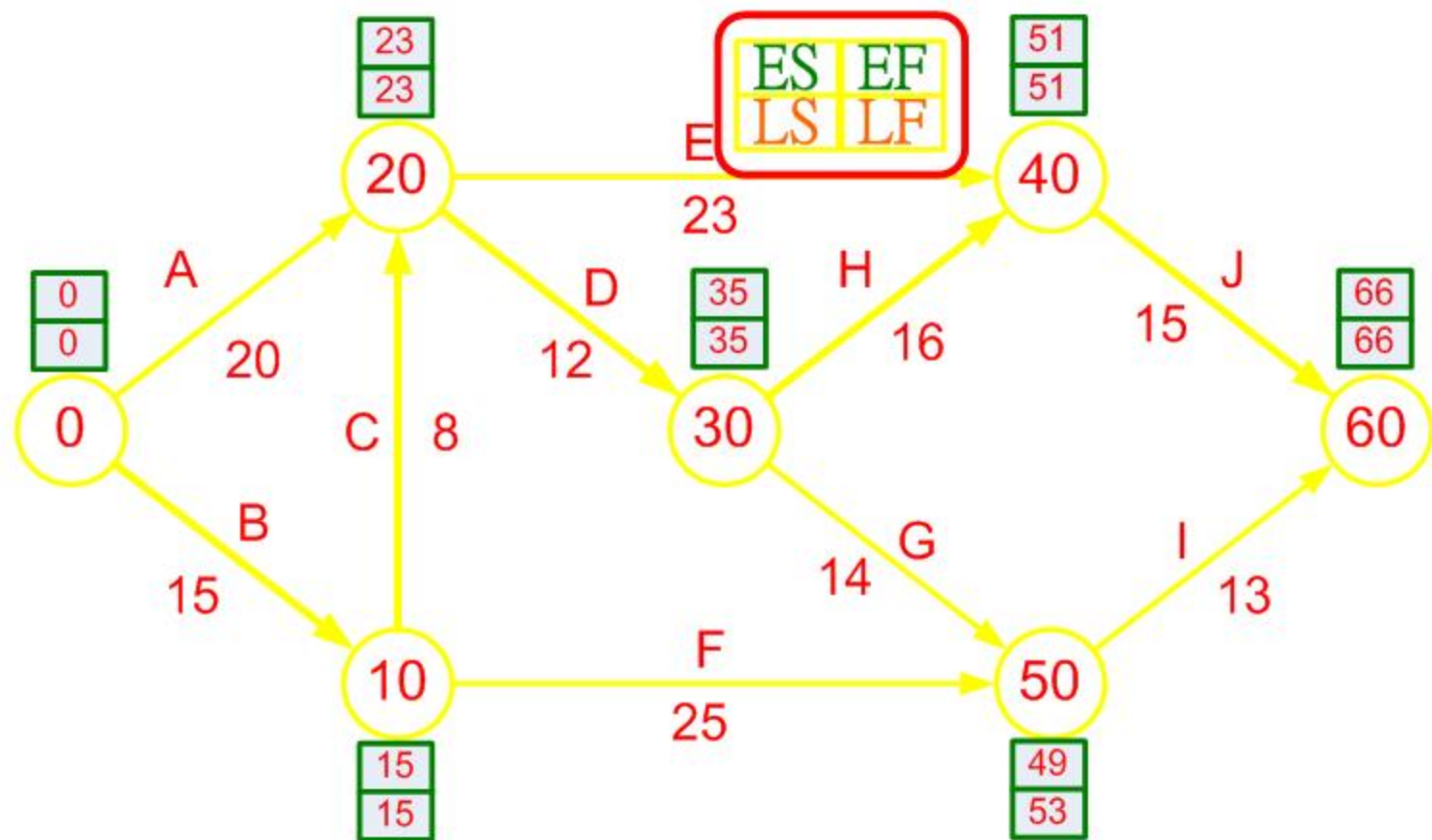


結點時間：作業時間計算之母

作業最早與最遲時程之計算

- **ES (Early Start)**
 - 為結點*i*之最早結點時間 EET_i
- **EF (Early Finish)**
 - $EF = ES + Duration$
- **LF (Late Finish)**
 - 為結點*j*之最遲結點時間 LET_j
- **LS (Late Start)**
 - $LS = LF - Duration$

作業最早與最遲時程之計算



作業之緩衝時間 (寬裕時間、浮時 Float)

- 網圖上有些作業可以比原設定的日程稍緩施做，而不致造成工期延長，或影響後續作業的最早開始時間。這種可以延緩的時間稱為浮時 (Float) 或緩衝時間或寬裕時間。

浮時的種類

- 總浮時 (**Total Float ; TF**)
- 自由浮時 (**Free Float ; FF**)
- 干擾浮時 (**Interfering Float ; IF**)

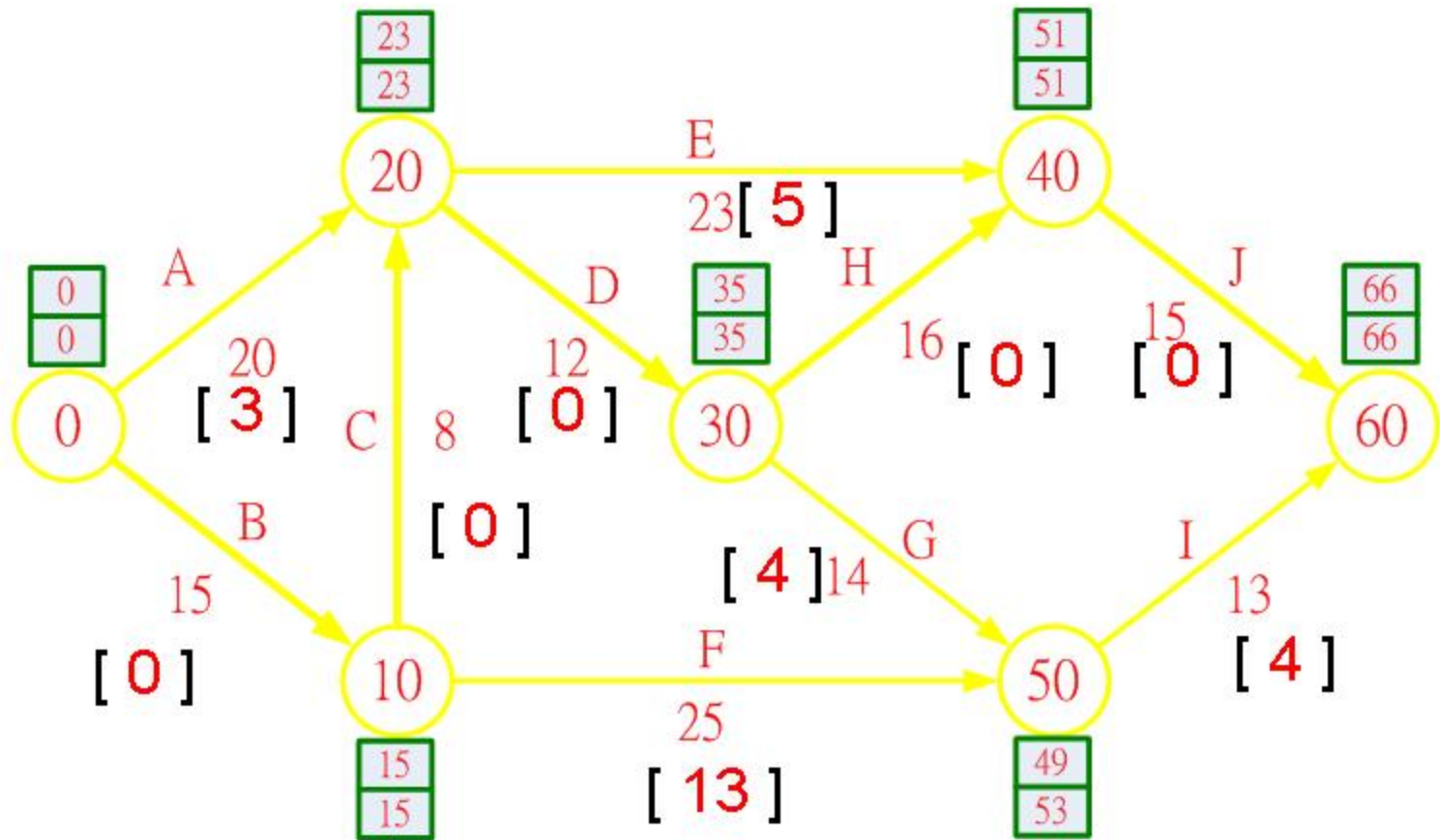
總浮時 (Total Float ; TF)

- 作業以最早開始時間開始，以最遲結束時間完成
- 為作業結點之間的最大寬裕時間
- 這段時間即使被耗用，也不致使整個工程延遲
- 為該作業在進入下一個結點前，所能允許延誤的最長時間

作業總浮時之計算：

$$\begin{aligned} \text{TF} &= \text{LET}(j) - \text{EET}(i) - \text{Dur} \\ &= \text{LF} - \text{ES} - \text{Dur} \\ &= \text{LF} - \text{EF} \end{aligned}$$

總浮時之計算



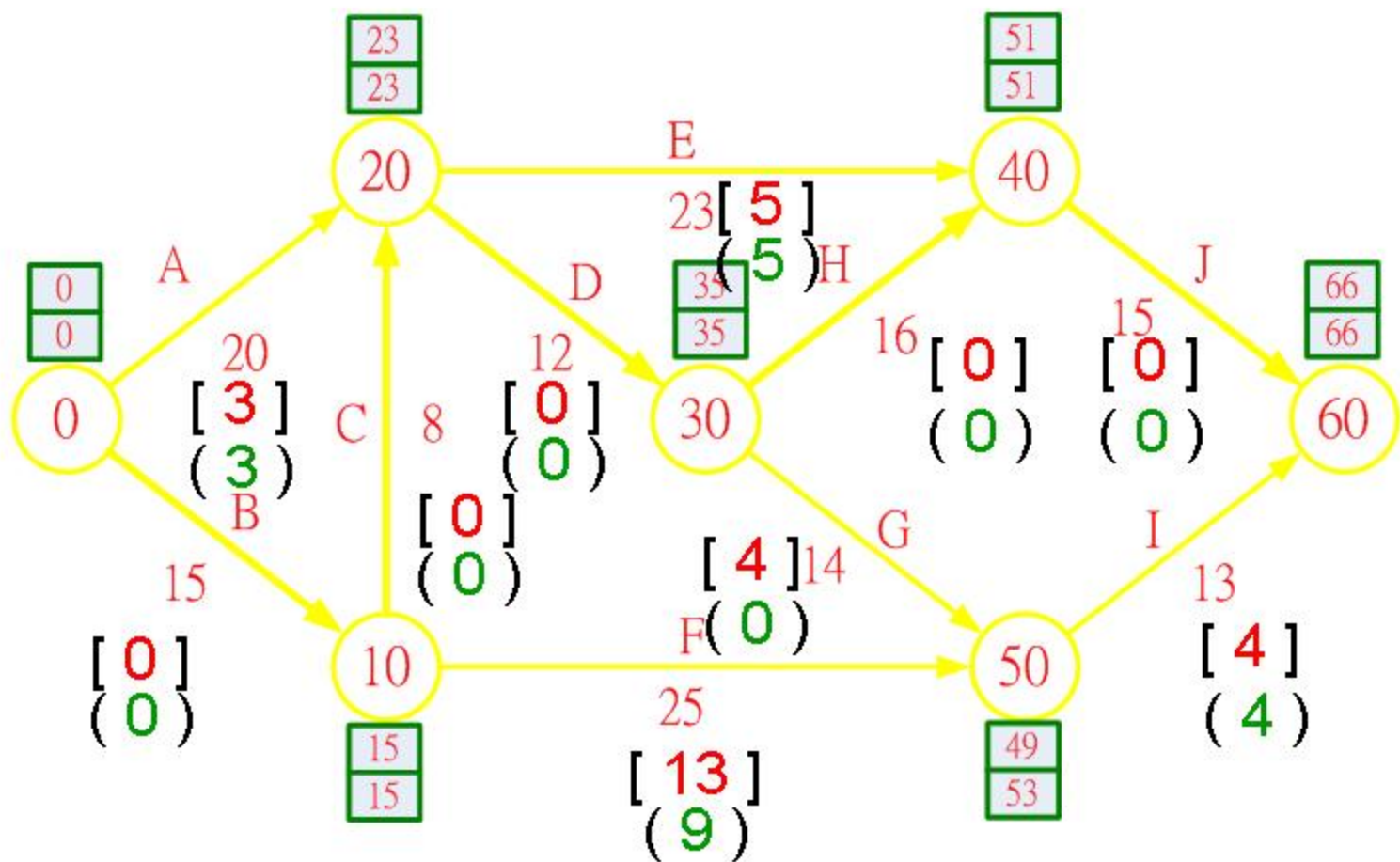
自由浮時 (Free Float ; FF)

- 作業以最早開始時間開始，其後續作業亦以最早開始時間開始時作業可以延遲的天數
- 此延遲不會使整個工程延誤
- 不會影響到後續作業的最早開始時間
- 自由浮時只受同一連鎖關係的先前作業所影響

作業自由浮時之計算：

$$FF = EET(j) - EET(i) - Dur$$

自由浮時之計算



干擾浮時 (Interfering Float ; IF)

- 一個作業可以延遲的時間
- 使用了干擾浮時不致於影響整個工程之完工期限
- 使用了干擾浮時會影響其後續作業的最早開始時間

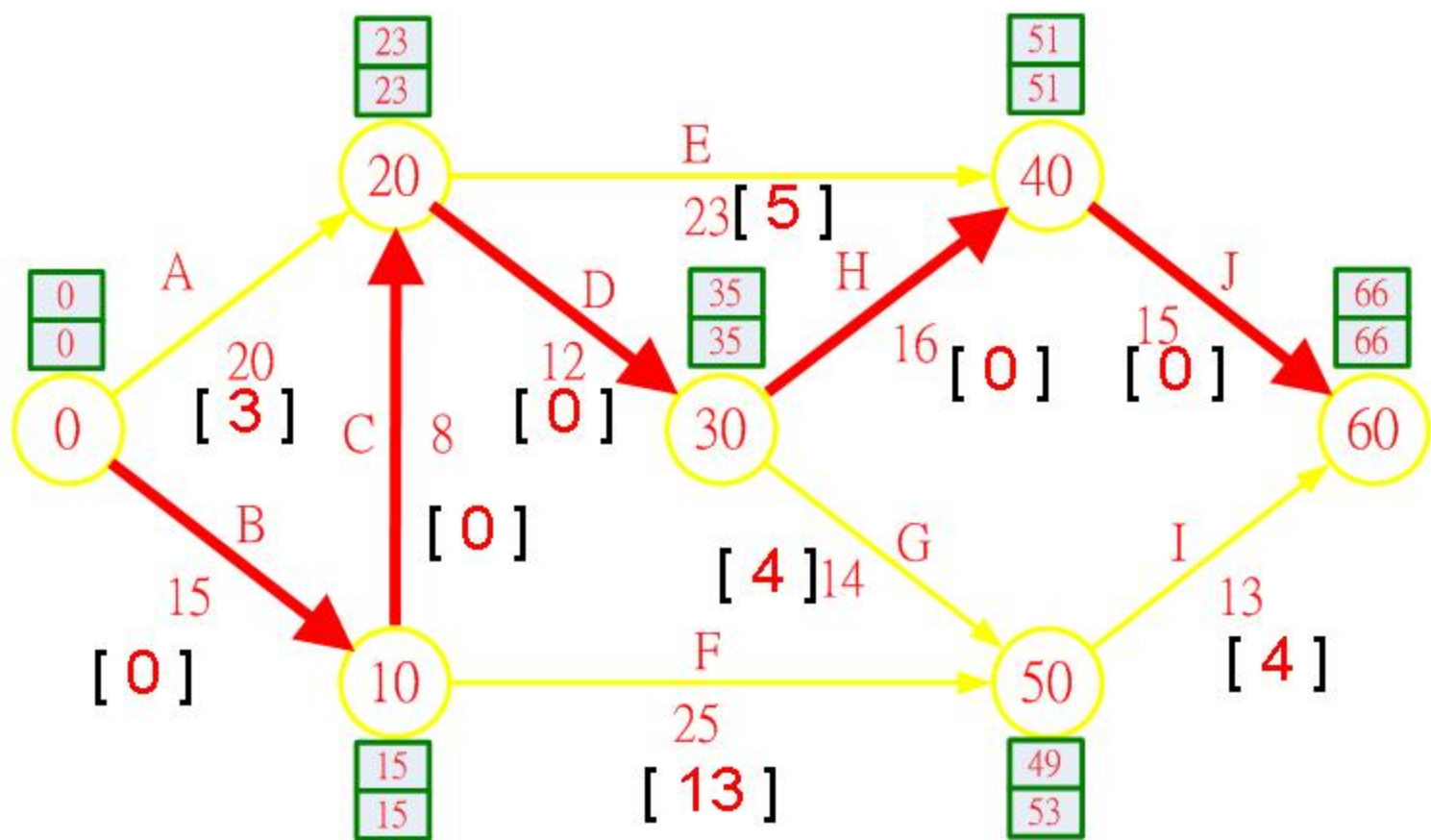
干擾浮時之計算：

總浮時與自由浮時之差

$$IF=TF-FF$$

要徑之決定

要徑：自網圖之開始作業連結浮時為零之
作業至最後一項作業其所形成之路徑



要徑之特性

- 要徑上之作業，其浮時皆為零
- 要徑代表網圖上所有路徑中之最長者，即要徑支配整個工期
- 要徑上任何作業之時程有任何延誤，均將使專案工程之總工期延長
- 需要縮短工期時，要徑為首要之目標

要徑之特性

- 每一網圖至少有一條要徑存在，若干狀況下，常有多條同時存在
- 非要徑作業可能因為浮時之耗盡而轉變成為要徑
- 在工程進行中，首重要徑作業之管制；對於浮時較小之所謂次要徑作業亦須予以密切關注

浮時的特性及應用

- 浮時主要在於辨明要徑之所在、作業可延遲之範圍、大小、管理之重點及趕工之依據
- 浮時若為零，則表示作業時限緊湊，無寬裕時間可彈性運用
- 浮時若為負數，則表示此一作業勢必要縮短其作業時間，才能如期達成任務
- 浮時若為正數，表示作業有寬裕的時間可供彈性運用

浮時的特性及應用

- 有了浮時的觀念，管理者才能認識各作業在時間管制上的輕重程度
- 有了寬裕時間，管理者才能合理地調配人力、資源、工具與經費
- 總浮時為一連串的前後作業所分攤，當其中一個作業使用了總浮時，就會造成其他同系列作業之總浮時相對的減少；故網路中同一路徑的所有作業均會因某一前置作業使用了總浮時而受影響

浮時的特性及應用

- 當浮時被用光時，此作業的路徑可能由非要徑接近為要徑或轉變為要徑，故可能造成另外一條新的要徑
- 當浮時被超用時，可能會有其他作業因此從要徑上被排除(不再是要徑，因為有更長的路徑發生)；所以，浮時的超用會造成原來在要徑上的作業受到直接的影響