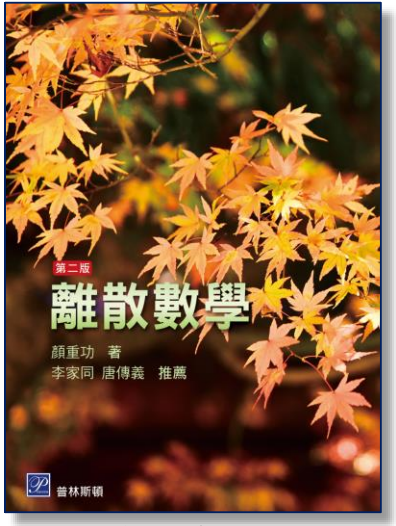
Discrete Mathematics



顏重功 著

- 課程目標：離散數學是許多資訊處理與演算法的基礎，課程內容包含基本計數、邏輯、集合、函數、關係、圖論等數學觀念及演算技巧，使同學奠定修習計算機科學之學習基礎。

- 評分方式：
  - Mid-term 20%
  - Final-term 20%
  - Others 60% (Attendance, Exercise, Quiz, Notes)

- 授課老師：吳汶涓 (wenn@niu.edu.tw)

資訊工程學系
Department of Computer Science and Information Engineering

課程進度

1) Fundamentals and Introduction
2) Counting
3) Set Theory
4) Set Theory
5) Theory of Integers
6) Logic and Boolean Algebra
7) Logic and Boolean Algebra
8) Relations and Functions
9) Midterm exam
10) Relations and Functions
11) Relations and Functions
12) Recurrence Relations
13) Recurrence Relations
14) Graph Theory
15) Graph Theory
16) Trees
17) Optimization problem
18) Final exam

圖上最佳化問題與樹

本章大綱

8-1　距離

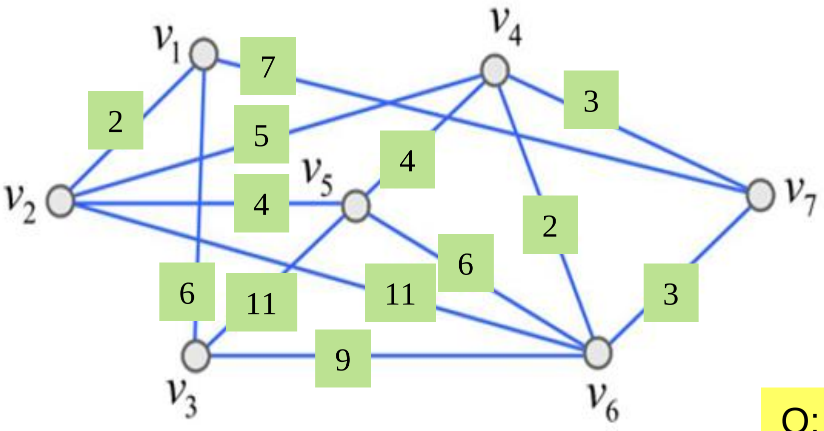
- 單一起點最短路徑
- 兩兩頂點最短路徑

8-2　樹與有根樹

- 有根樹
- 二元樹與k元樹

8-1 Distance

路途長度 (Lengths of Walks)



v_2 到 v_6 的路途

W_1 ： $v_2 — v_6$ (長度1)

W_2 ： $v_2 — v_5 — v_2 — v_5 — v_6$ (長度4)

W_3 ： $v_2 — v_6 — v_2 — v_6 — v_2 — v_6$ (長度5)

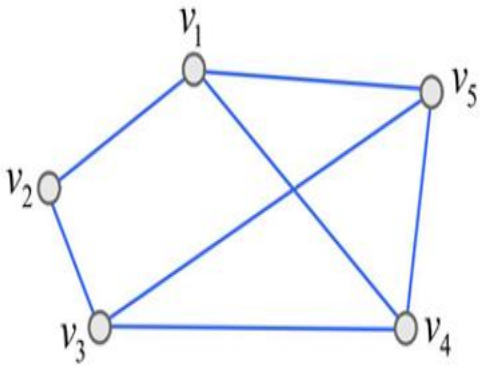
Q: 若圖很大，如何算出任兩頂點長度為*k*的路途?

加權圖(Weighted graph) → W1: 11, W2: 18, W3: 55

8-1 Distance

計算任兩頂點長度為k的路途

範例8.2



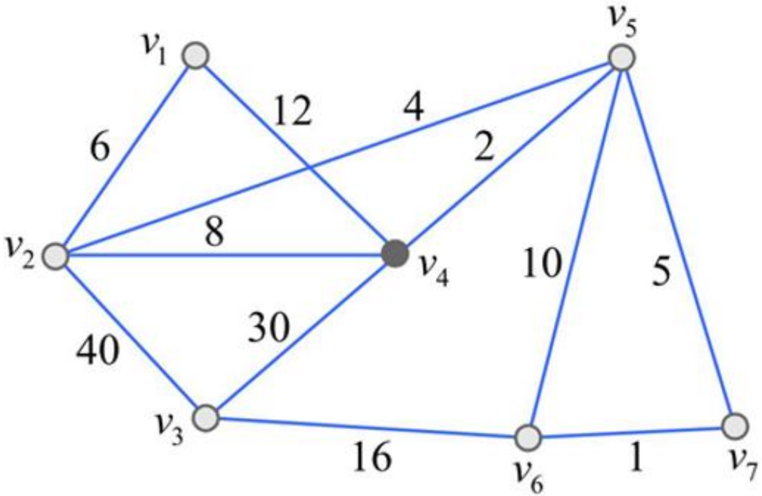
$$A = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 1 \\ 4 & 1 & 0 & 1 & 0 & 1 \\ 5 & 1 & 0 & 1 & 1 & 0 \end{array} \Rightarrow A^2 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 3 & 0 & 3 & 1 & 1 \\ 2 & 0 & 2 & 0 & 2 & 2 \\ 3 & 3 & 0 & 3 & 1 & 1 \\ 4 & 1 & 2 & 1 & 3 & 2 \\ 5 & 1 & 2 & 1 & 2 & 3 \end{array}$$

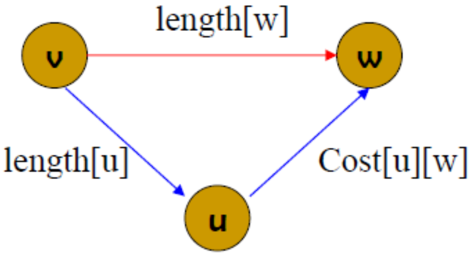
$$A^3 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 2 & 6 & 2 & 7 & 7 \\ 2 & 6 & 0 & 6 & 2 & 2 \\ 3 & 2 & 6 & 2 & 7 & 7 \\ 4 & 7 & 2 & 7 & 4 & 5 \\ 5 & 7 & 2 & 7 & 5 & 4 \end{array}$$

A^2=A*A: 表示任兩點間長度為2的路途個數
A^3=A*A*A:表示任兩點間長度為3的路途個數
…

8-1 Distance

Single Source Shortest Paths (Dijkstra's Algorithm)

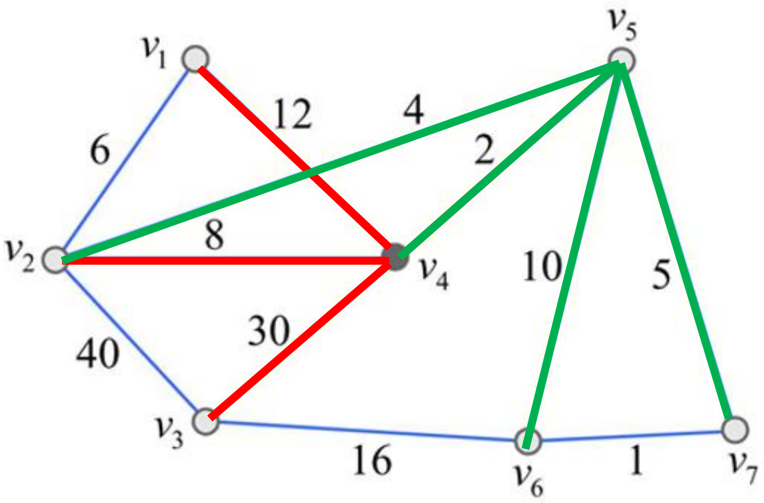




$$\text{length}[w] = \min(\text{length}[w], \text{length}[u] + \text{Cost}[u,w])$$

8-1 Distance

Single Source Shortest Paths (Dijkstra's Algorithm)



頂點 x	v_1	v_2	v_3	v_4	v_5	v_6	v_7
距離值 $d(x)$	∞	∞	∞	0	∞	∞	∞

解：Step1: 選擇v4 → 更新連接的頂點

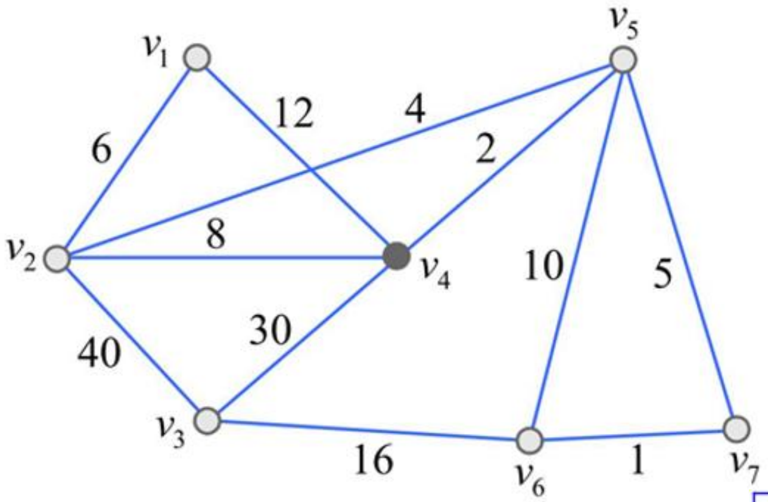
頂點 x	v_1	v_2	v_3	v_4	v_5	v_6	v_7
距離值 $d(x)$	12	8	30	0	2	∞	∞

Step2: 選擇v5 → 更新連接的頂點

頂點 x	v_1	v_2	v_3	v_4	v_5	v_6	v_7
距離值 $d(x)$	12	6	30	0	2	12	7

8-1 Distance

Single Source Shortest Paths



頂點 x	v_1	v_2	v_3	v_4	v_5	v_6	v_7
距離值 $d(x)$	12	6	30	0	2	12	7

Step3: 選擇v2 → 更新連接的頂點

頂點 x	v_1	v_2	v_3	v_4	v_5	v_6	v_7
距離值 $d(x)$	12	$\boxed{6}$	30	0	2	12	7

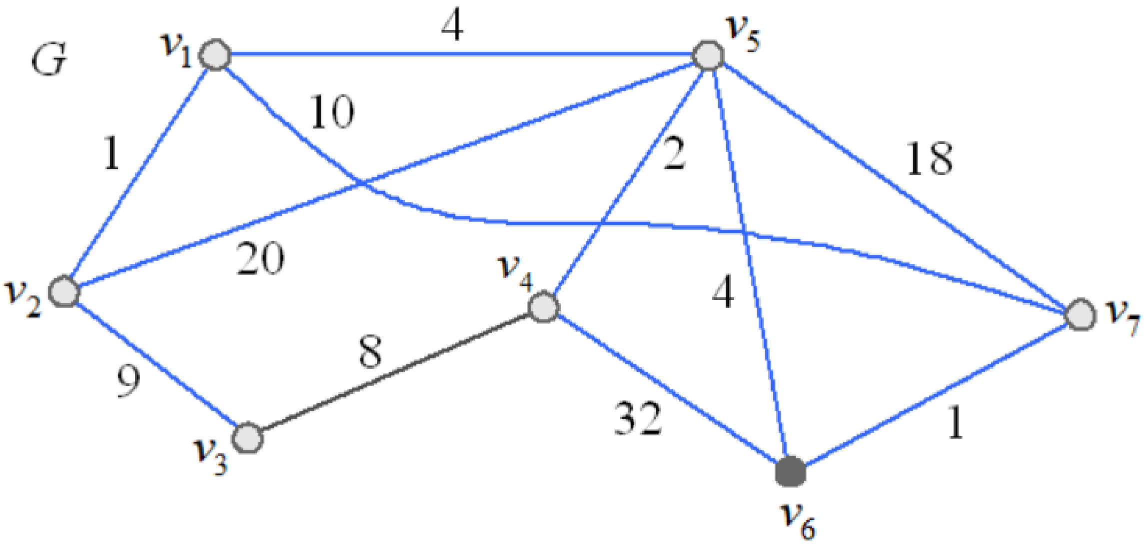
Step4: 選擇v7 → 更新連接的頂點

步驟	頂點 x	v_1	v_2	v_3	v_4	v_5	v_6	v_7
7	距離值 $d(x)$	12	6	24	0	2	8	7

共執行Step 七次

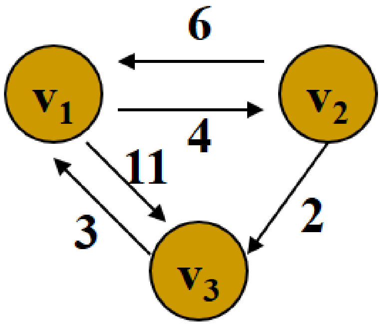
8-1 Distance

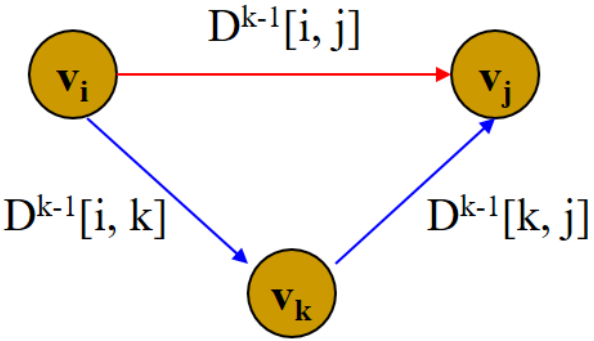
請試看看使用Dijkstra方法求 v6到各頂點的最短路徑



8-1 Distance

All-Pairs Shortest Paths (Floyd's Algorithm)



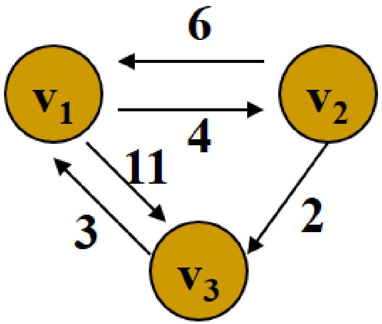


此為n*n矩陣，紀錄vi到vj頂點的最短路徑，其中只經過{v1,v2,…,vk}為中介點

$$D^k[i,j] = \begin{cases} W[i,j], & \text{if } k = 0 \\ \min(D^{k-1}[i,j], D^{k-1}[i,k] + D^{k-1}[k,j]), & \text{if } k > 0 \end{cases}$$

8-1 Distance

All-Pairs Shortest Paths (Floyd's Algorithm)



解： Step1

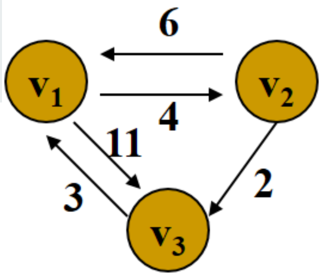
$$D^0 = W = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix} \end{array}$$

Step2

$$D^1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

$$\begin{cases} 2 \to 3 : 2 \\ 2 \to 1 \to 3 : 17 \end{cases}$$

$$\begin{cases} 3 \to 2 : \infty \\ 3 \to 1 \to 2 : 7 \end{cases}$$



8-1 Distance

Step3. $D^1 \rightarrow D^2$

$$D^1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array} \Rightarrow D^2 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

$$\begin{cases} 1 \rightarrow 3 : 11 \\ 1 \rightarrow 2 \rightarrow 3 : 6 \end{cases}$$

$$\begin{cases} 3 \rightarrow 1 : 3 \\ 3 \rightarrow 2 \rightarrow 1 : \infty \end{cases}$$

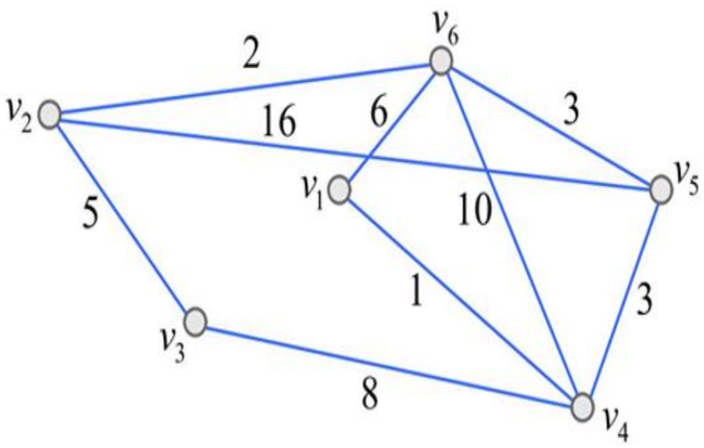
Step4. $D^2 \rightarrow D^3$

$$D^2 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array} \Rightarrow D^3 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \end{array}$$

$$\begin{cases} 1 \rightarrow 2 : 4 \\ 1 \rightarrow 3 \rightarrow 2 : \infty \end{cases}$$

$$\begin{cases} 2 \rightarrow 1 : 6 \\ 2 \rightarrow 3 \rightarrow 1 : 5 \end{cases}$$

8-1 Distance



步驟 1： $D^{(0)} = A$

$$D^{(0)} = A = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & \infty & \infty & 1 & \infty & 6 \\ 2 & \infty & 0 & 5 & \infty & 16 & 2 \\ 3 & \infty & 5 & 0 & 8 & \infty & \infty \\ 4 & 1 & \infty & 8 & 0 & 3 & 10 \\ 5 & \infty & 16 & \infty & 3 & 0 & 3 \\ 6 & 6 & 2 & \infty & 10 & 3 & 0 \end{array}$$

步驟 2：求 $D^{(1)}$

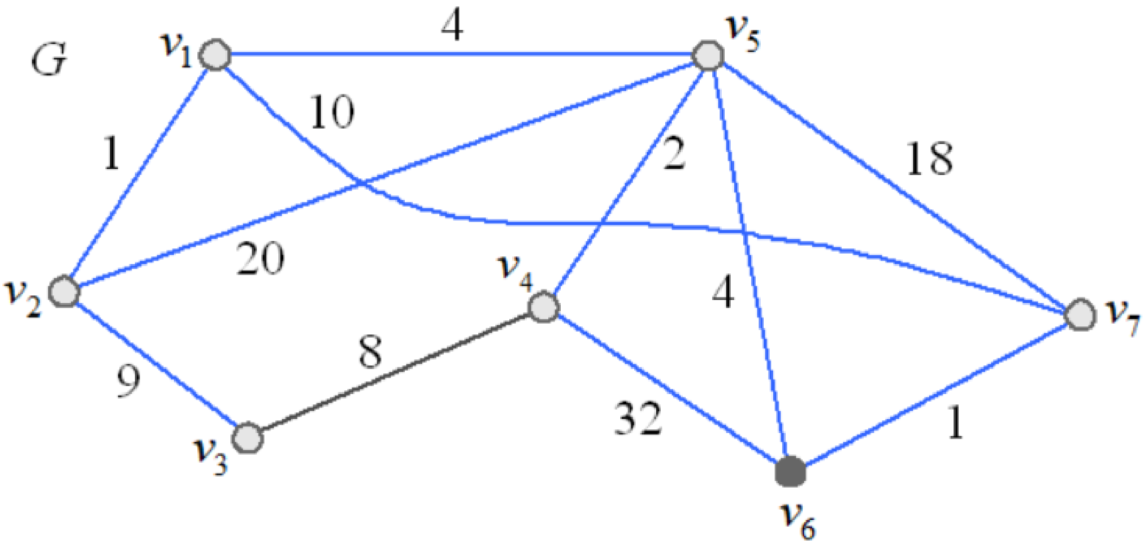
$$D^{(0)} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & \infty & \infty & 1 & \infty & 6 \\ 2 & \infty & 0 & 5 & \infty & 16 & 2 \\ 3 & \infty & 5 & 0 & 8 & \infty & \infty \\ 4 & 1 & \infty & 8 & 0 & 3 & 10 \\ 5 & \infty & 16 & \infty & 3 & 0 & 3 \\ 6 & 6 & 2 & \infty & 10 & 3 & 0 \end{array} \rightarrow D^{(1)} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & \infty & \infty & 1 & \infty & 6 \\ 2 & \infty & 0 & 5 & \infty & 16 & 2 \\ 3 & \infty & 5 & 0 & 8 & \infty & \infty \\ 4 & 1 & \infty & 8 & 0 & 3 & 7 \\ 5 & \infty & 16 & \infty & 3 & 0 & 3 \\ 6 & 6 & 2 & \infty & 7 & 3 & 0 \end{array}$$

步驟 3：求 $D^{(2)}$

$$D^{(1)} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & \infty & \infty & 1 & \infty & 6 \\ 2 & \infty & 0 & 5 & \infty & 16 & 2 \\ 3 & \infty & 5 & 0 & 8 & \infty & \infty \\ 4 & 1 & \infty & 8 & 0 & 3 & 7 \\ 5 & \infty & 16 & \infty & 3 & 0 & 3 \\ 6 & 6 & 2 & \infty & 7 & 3 & 0 \end{array} \rightarrow D^{(2)} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & \infty & \infty & 1 & \infty & 6 \\ 2 & \infty & 0 & 5 & \infty & 16 & 2 \\ 3 & \infty & 5 & 0 & 8 & 21 & 7 \\ 4 & 1 & \infty & 8 & 0 & 3 & 7 \\ 5 & \infty & 16 & 21 & 3 & 0 & 3 \\ 6 & 6 & 2 & 7 & 7 & 3 & 0 \end{array}$$

8-1 Distance

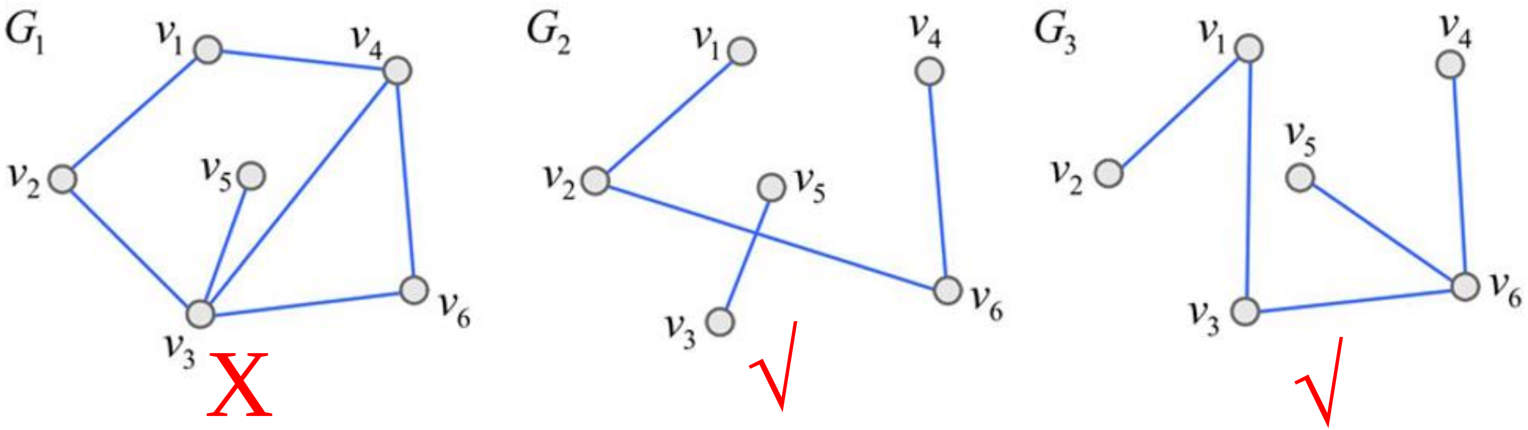
請試看看使用Floyd's Algorithm求 各頂點間的最短路徑 (需求到$D^{(7)}$)



8-2 Tree and Rooted Tree

T(V,E)為**沒有循環**的連通簡單圖，稱樹形圖(Tree Graph)

範例8.6 以下何者為樹？

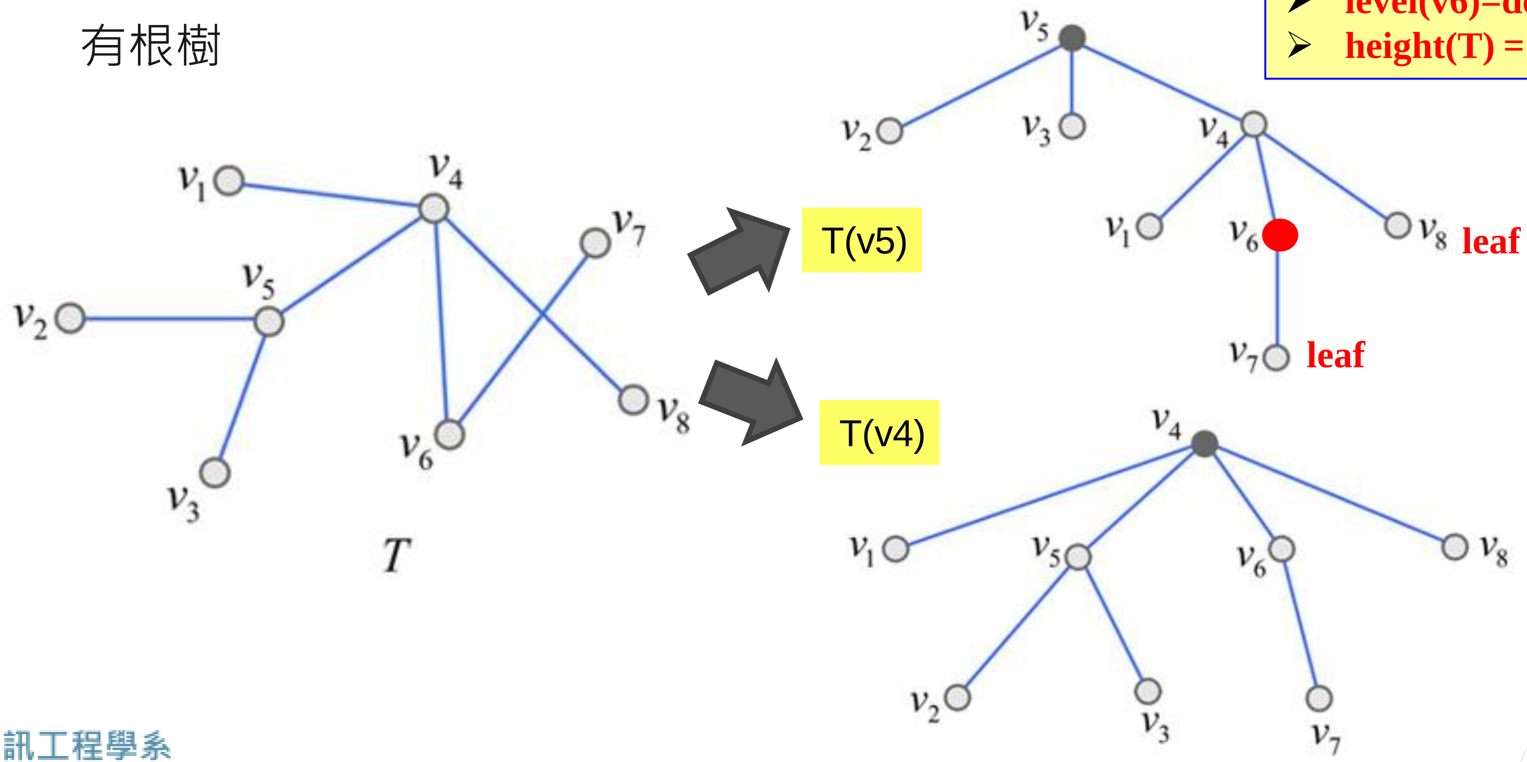


v6而言

- child:v7, parent:v4
- anscestor:v4, v5
- sbiling:v1, v8
- level(v6)=depth(v6)=2
- height(T) = 3

8-2 Tree and Rooted Tree

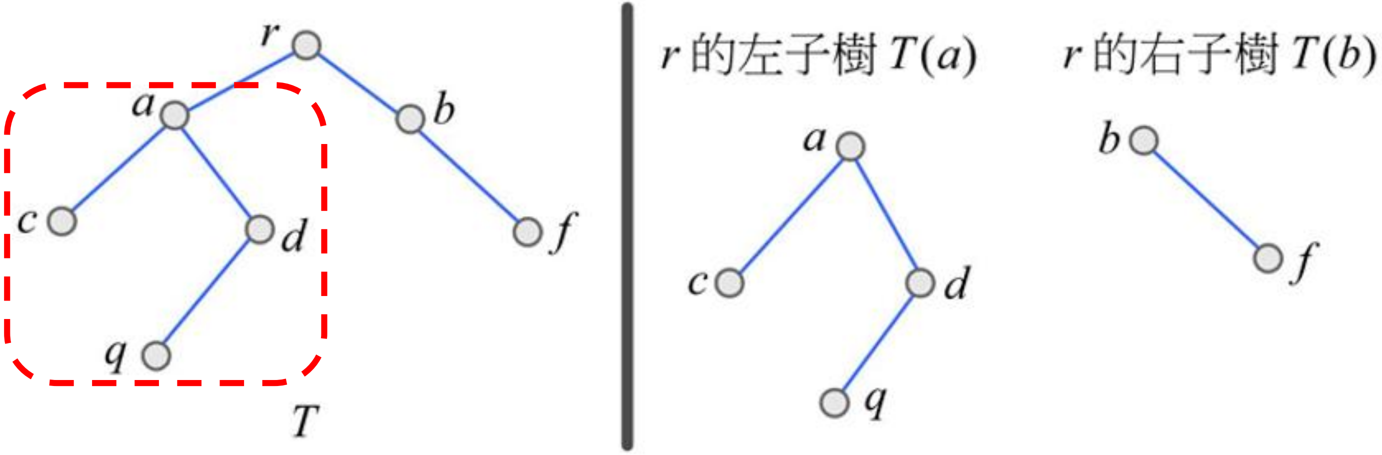
有根樹



資訊工程學系
Department of Computer Science
and Information Engineering

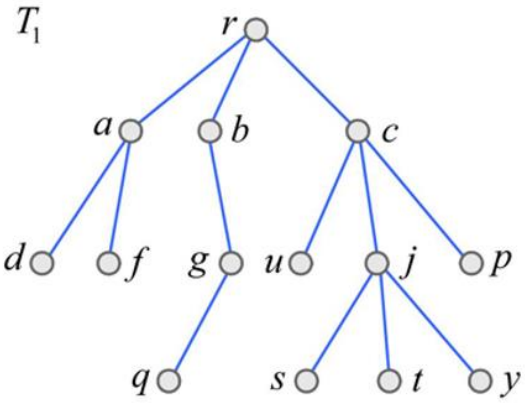
8-2 Tree and Rooted Tree

Binary Tree：T為一個有根樹且每個內部節點最多有兩個children.

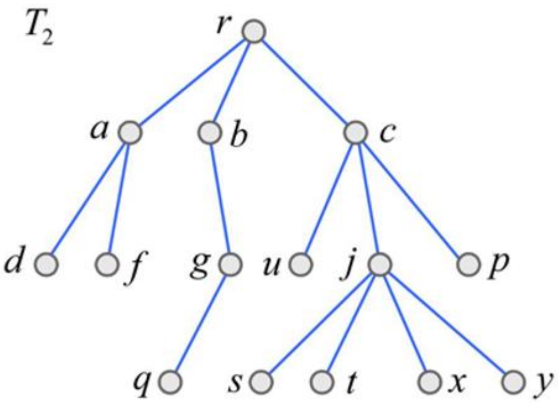


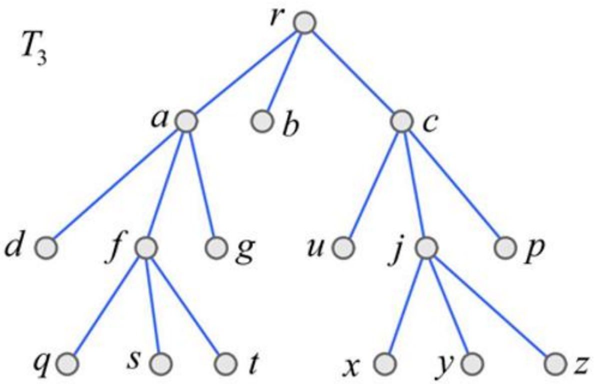
8-2 Tree and Rooted Tree

k*-ary Tree**：T為一個有根樹且每個內部節點最多有 ***k 個children.



三元樹

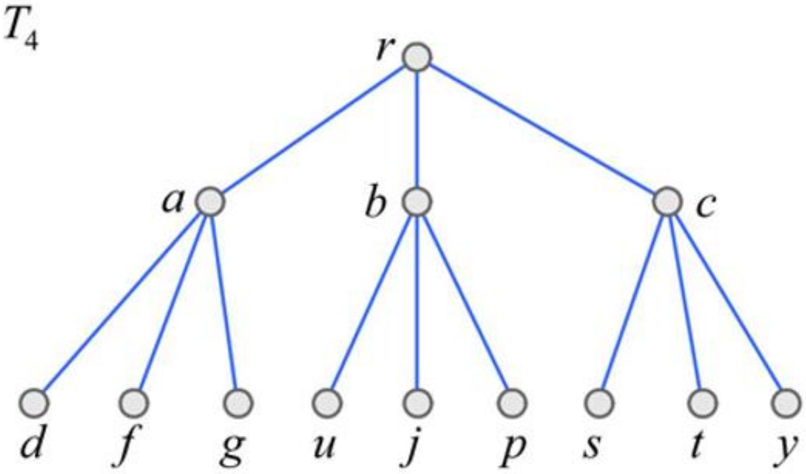




完全三元樹
(complete 3-tree)

每個內部節點的分支都是 3

8-2 Tree and Rooted Tree



$l=0,\ 3^0=0$

$l=1,\ 3^1=3$

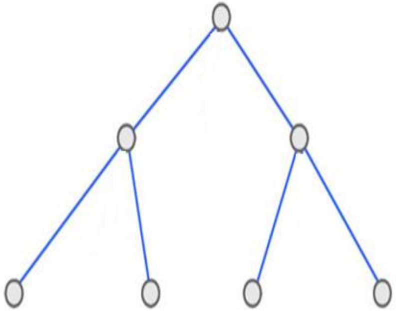
$l=2,\ 3^2=9$

完滿三元樹
(Full 3-tree)

T為完全樹且height(T)= 2
即所有 leaf 的深度都是 2

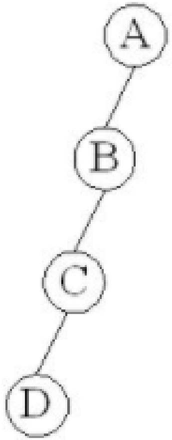
完滿k元樹的節點總個數
$= k^0+k^1+k^2+\ldots.+k^{(h)}$

完滿 2元樹且高度為2的節點總個數
$= 2^{2+1} - 1 = 7$

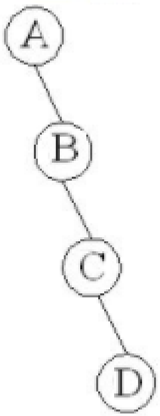


8-2 Tree and Rooted Tree

Skewed Binary Tree: 所有頂點都只有單一節點 (左節點或右節點)



左歪斜二元樹



右左歪斜二元樹

- ✓ 二元樹最大的高度 $h = n - 1$
- ✓ 節點總個數 $n = h + 1$
- ✓ 邊的總個數 $m = n - 1$

8-2 Tree and Rooted Tree

Presentation

(1) Array

[0]	A
[1]	B
[2]	C
…	…
[7]	H

(2) Linked list