CH7、樹

樹

目錄：

7.1 樹的介紹

定義：

1. Tree 為不含 Cycle 之連通無向圖
2. 只含一點之 Tree 稱為 Degeneration Tree(退化)或 Trivial Tree

Note：

Tree 一定為 Bipartite，Tree 的任一邊皆為 Bridge

定理：

G=(V, E)：Nontrivial Tree ⇔ G 中任 2 點間恰有一條 Path 相連

⇔G 為 Connected 且去掉一邊，變成 Disconnected

⇔G 為 Acyclic，且加入一邊則恰含一個 Cycle

定理(18 個)：

G=(V, E)

1. G is a tree ⇔
2. G 為 Acyclic, $|E|=|V|-1$ ⇔
3. G 為 Connected 且$|E|=|V|-1$

證明：

1.⟹2.：Claim $|E|=|V|-1$

by Induction on $|V|$, $|V|=1$：$|E|=0$，成立

設$|V|<n$ 成立，consider $|V|=n$

任取一邊 e，則 G-e 形成 2 個 components, G1=(V1, E1), G2=(V2, E2)

則 G1, G2 皆為 Tree，且$|V1|, |V2| < n$

by IH, $|E|1=|V1|-1, |E2|=|V2|-1$

$\Longrightarrow |E|=|E1|+|E2|+1 = |V1|-1 + |V2|-1 + 1 = |V1|+|V2|-1 =|V|-1$

2.⟹3. 令 k=k(G)

設 G1=(V1, E1), …, Gk=(Vk, Ek)為 G 之 k 個 Component

因為 G 為 Acyclic，G1 為 Acyclic，所以 Gi 為 Tree，由 1⟹2 之結果

$|Ei|=|Vi|-1, |E|=|E1|+\ldots+|Ek| = |V1|+\ldots+|Vk|-k = |V|-k$

3⟹1：Claim G is acyclic

設 G 含 Cycle，則 G 中去掉 Cycle 之一邊形成 G'=(V', E')使 G'仍 Connected，

因此$|E'|\geq|V|-1 \Longrightarrow |E|-1 \geq |V|-1 \Longrightarrow |E|\geq|V|$ →←

例(97 清大)：G=(V, E)：Tree， 10 個點 degree=2, 10 個點 degree=3, 10 個點 degree=4, 1 個點 degree=5，其他點 degree=1，求|V|為何？

設 degree 為 1 個有 x 點 $\Longrightarrow V=31+x$

$20+30+40+5+x = 2(30+x)$

$\Longrightarrow x=35, |V| = 31+35 = 66$

Note：

Forest：不含 Cycle 之無向圖，$|E|=|V|-k(G)$

定義：

$G=(V, E), v\in V$，若 $degree(v)=1$，稱為 pendant(懸吊點)

定理：

$G=|V|-k(G)$：Nontrivial Tree $\Longrightarrow$ G 中至少 2 個 pendant

證明：

設 pendant 個數為 k $\Longrightarrow 2|E| = \sum deg(a) \geq k+2(n-k)$

$\Longrightarrow k\geq 2$

7.2 有根樹

定義：

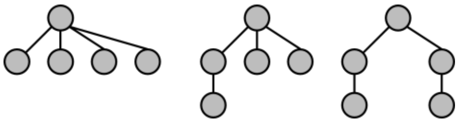
G=(V, E)：Digraph，若將 G 視為無向圖為 Tree，則稱 G 為 Diected Tree (Root=level 0)

1. 若$\exists!\ r \in V, \ni id(r)=0$，稱 G 為 Rooted Tree
2. 若每個內部節點至多 m 個子點，則稱 G 為 m-ary Tree
3. 若改成恰含 m 個子點，稱 G 為 Full m-ary Tree
4. 若再加上所有 leaf 具相同 level，稱 G 為 Complete m-ary Tree
5. 若所有 leaf 具 level h 或 level(h-1)，稱 G 為 Balance Tree

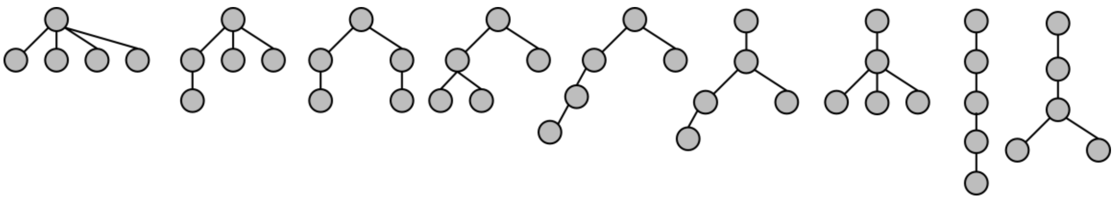
例(96 清大)：

1. 5 個點 nonisomophic free tree 個數
2. 5 個點 nonisomophic rooted tree 個數？

1.



2.



Note：

T=(V, E)：m-ary Tree, n=|V|, l=leaf 個數, i=internal node

1. $n \leq mi+1$
2. $n=l+i$
3. $l+i \leq mi+1 \Longrightarrow i \geq (l-1)/(m-1)$
4. 當 T 為 Full m-ary Tree，則 $n=mi+1$

例(96 元智)：502 台 PC，利用 Extension Cord(4 outlets)，至少 n 條可連接所有 PC？

$i \geq (l-1)/(m-1) = 501/3 = 167$

Note：

T：Full m-ary Tree with height h

1. $(m-1)(n-1)+m \leq l \leq m^h$
2. $mh+1 \leq n \leq (m^h-1)/(m-1)$

Note：

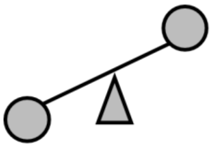
T：Full m-ary Tree with height h

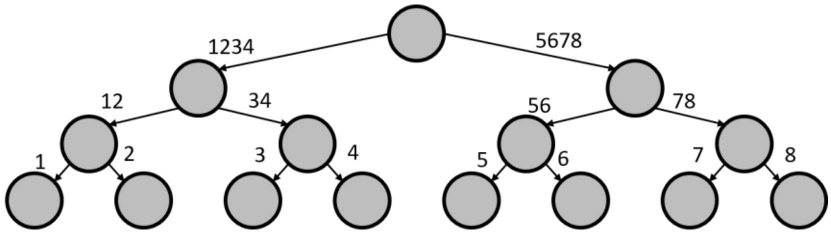
$\Longrightarrow l \leq m^h$

$\Longrightarrow h \geq [\log_m l]$

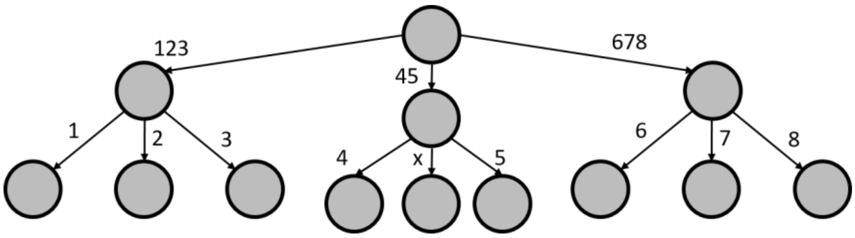
例：8 個 Coins，一個偽 Coin 較重，利用天平至少秤幾次可找出來？

$h \geq [\log_2 8] = 3$





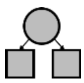
若改成為一個偽 Coin 不知輕重，則：
$l = 16, m=3$，$height \geq [\log_3 16] = 3$



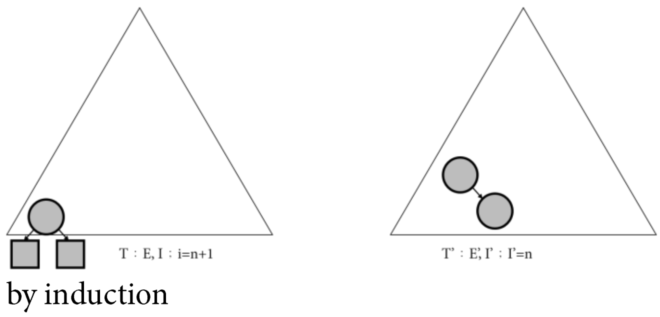
定理(98 清大)：
E=I+2i

證明：
by induction on i

1. i=1 時，成立



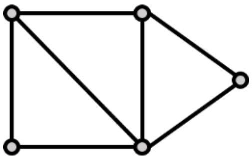
2. 設 i=n 時成立
3. Consider i=n+1



by induction
E' = I' + 2i' = I' + 2n
另外：E=E'+2h-(h-1) = E'+h+1
I = I' + (h-1)
所以：E= E'+h+1 = (I'++2n)+h+1 = (I-h+1)+2n+h+1 = I+2(n+1) = I+2i

推廣(98 清大)：
T：Full m-ary Tree ⟹ E=(m-1)I+mi

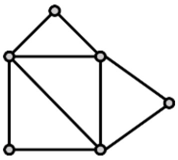
7.3 生成樹



定義：

G=(V, E)：Connected，若 T 為 G 之 Spanning Subgraph，且 T 為 Tree，稱 T 為 G 之一 Spanning Tree

定理(3 個)： G=(V, E) G 為 Connected ⇔ G 含 Spanning Tree 證明： ⇐：成立 ⇒： 若 G 不含 Cycle，則 G 為 G 之 Spanning Tree 否則，G 中去掉 Cycle 之一邊，仍 Connected，再檢查是否含 Cycle，若有，再去掉 Cycle 之一邊仍 Connected，再檢查是否含 Cycle，以此類推，直到不含 Cycle 時，即為 G 之一 Spanning Tree



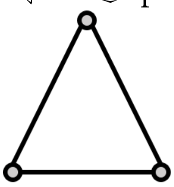
V=6, e=8, e'=v-1 = 5

砍 e-e' = e-(v-1) = e-v+1

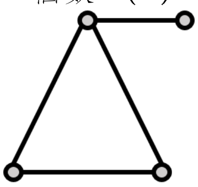
Regin：1+(e-v+1)

定理(13 個)：歐拉公式 Euler Formular G=(V, E)：Connected Planar $\Rightarrow$ v-e+r=2 令 T 為 G 之一 Spanning Tree $\Rightarrow r_T = 1$，且 e_T=V-1，T 中每加入一邊，將增加一個 Region，T 中加入 $e-e_T$=e-v+1 個邊，可還原回到 G $\Rightarrow$ r=1+(e-v+1) $\Rightarrow$ v-e+r = 2

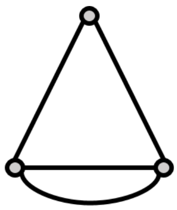
*求 G 之 Spanning Tree 個數 N(G)：暴力法



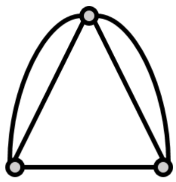
3



3

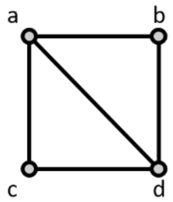


5



8

定理：Matrix Tree
G=(V, E), V={v_1, …, v_n}
M=[mij]：n×m
mij = degree(v_{ij}) , if i=j
= -1 , if i≠j，且 vi 與 vi 有相連
= 0 , if i≠j，且 vi 與 vi 不相連
則 N(G)為 M 之各項 cofactor



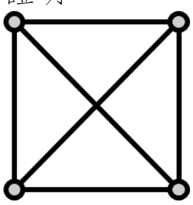
M=

$$\begin{vmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{vmatrix}$$

cof(m_{ii}) = (-1)1+1(矩)=8
cof(m_{12})=(-1)1+2(矩)

定理：(Cayley Number)
N(K_n)=n^{n-2}

證明：

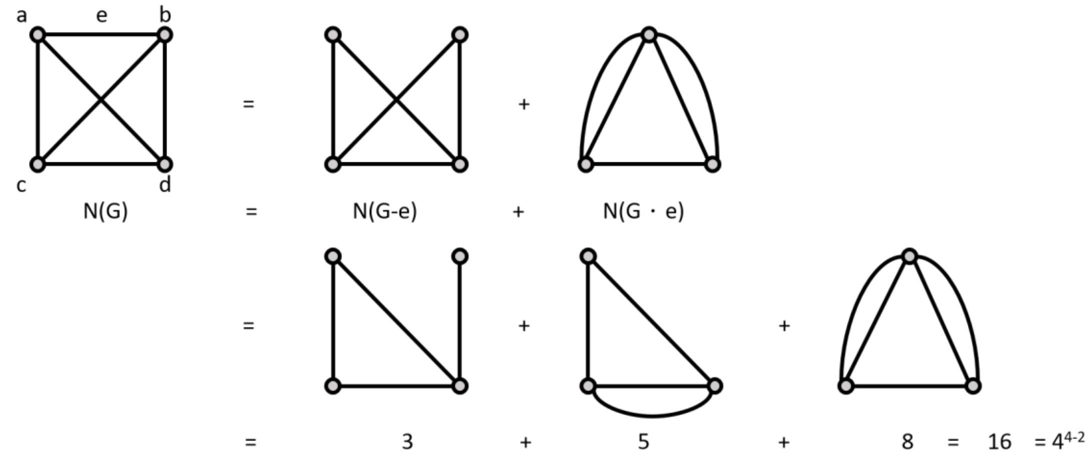


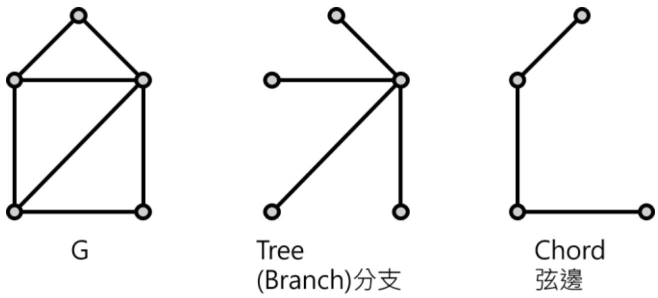
K_4：M=

$$\begin{vmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{vmatrix}$$

N(K_n)=cof(m_{11})
eigenvalues：n, …, n(共 n-2 個 n), 1
N($K_{m,n}$)= m^{n-1}×n^{m-1}

Note：





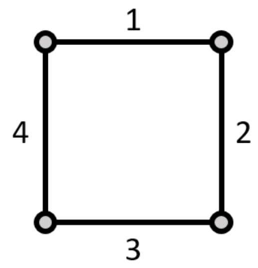
加一弦、生一 Cycle⟹幾個弦幾個 Cycle，每一個 Tree 邊一個 Cut Set

Note：

G：Connected，T：Spanning Tree，C：Cycle，E：Cut Set

1. E 與 T 至少含 1 共同邊
2. T 與 C 至少含 1 共同邊
3. C 與 E 含偶數共同邊

7.4 最小生成樹

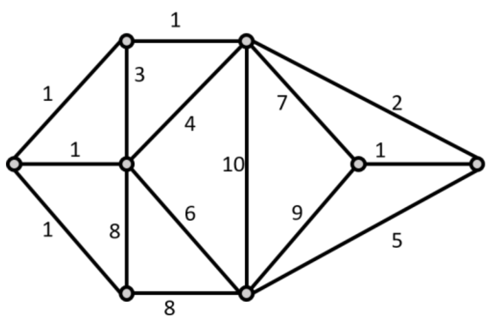


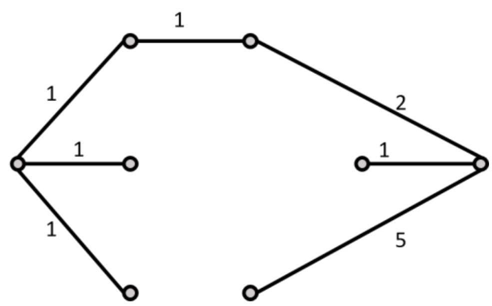
定義：
G=(V, E)
w：E→R weight function，稱 G=(V, E, w)為 Weighted Graph

定義：
G=(V, E, w)：Connected
G 之 Spanning Tree 中，具有邊之 Weight 總和最小者，稱為 G 之一 Minimum Cost Spanning Tree(MCSTT)

Kruskal's 演算法
由邊開始
為 Greedy Algorithm

例(99 清大)：





Prim's 演算法：
由點開始
Keep Connected
Greedy Algorithm
Complexity：$O(n^2)$, n=|V|

Note：

1. Minimum Cost Spanning Tree 未必唯一
2. 當所有邊之 Weight 皆相異時，Minimum Spanning Tree 唯一

Greedy 演算法，共 6 例：

1. Huffman
2. Fractional Knapsack Problem
3. Kruskal's
4. Prim's
5. Sollin's
6. Dijkstra's

Dynamic Programming，共 6 例：

1. LCS
2. 0/1 Knapsack Problem
3. Matrix Chain
4. Bellman-Ford
5. Floyd-Warshall
6. OBST

7.5 前置碼

Encode：固定長/變動長

(A, B, C, D) = (00, 01, 10, 11)

AABC → 00 00 01 10 → AABC

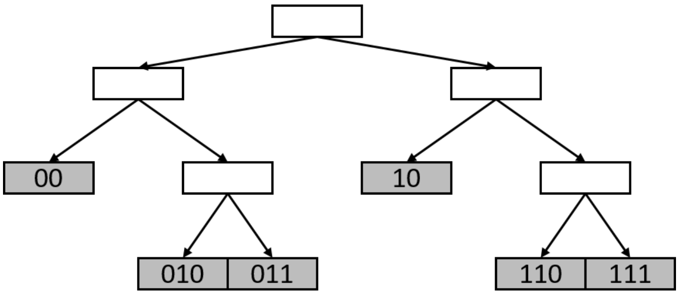
(A, B, C, D) = (0, 1, 00, 01)

AABC → 00100

定義：

G：Set of code word

若∀ x≠y∈G, x 不為 y 之 prefix，稱 G 為 Prefix Code

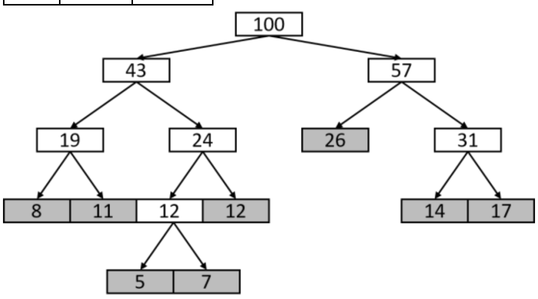


W(T) = ∑ w(a)l(a)   *//leaf 與 level*

求 T ∋ w(T)為 min，稱 T 為 Optimal Tree

例(99 中央)：

字	freq	code
A	0.17	111
B	0.08	000
C	0.26	10
D	0.12	011
E	0.05	0100
F	0.07	0101
G	0.14	110
H	0.11	001



*平均 bit 數 = 0.17*3 + 0.08*3 + … = 2.86*