

四. 圖的概念 (Graph Concepts)

1. 基本性質

一個圖 G 可以看成是 (V, E) 或 $(V(G), E(G))$ ，其中 V 為一個集合也稱為是圖 G 的點集合， $E \subseteq \binom{V}{2}$ 為圖 G 的邊集合；在 E 中的元素為圖 G 的邊。由定義具有 p 個點的圖最多有 $\binom{p}{2}$ 個邊；因此，具有最多邊的圖稱為是完全圖 (Complete Graph)。具有 p 點的完全圖以 K_p 表示。反過來，沒有邊的 p 點圖以 O_p 表示。

對於一個圖究竟有多少邊，我們可以利用點與點之間連接的關係來說明。

定義 1.1. 令 $G = (V, E)$ 為一圖，其中 $\{u, v\} \in E$ 或簡寫為 $uv \in E$ ，則我們稱 u 與 v 相鄰 (adjacent) 或 u 為 v 的一個鄰點 (Neighbor)；此時 v 也是 u 的一個鄰點。一般也用 $u \sim_G v$ 來表示 G 中相鄰的兩點 u 與 v 。

定義 1.2. $N_G(v) = \{u \in V(G) | u \sim_G v\}$, $N_G[v] = \{v\} \cup N_G(v)$, $N_G(v)$ 稱為是 v 的鄰域。

定義 1.3. 我們定義 G 中一個點 v 的度 (Degree) 為 $\deg_G(v) = |N_G(v)|$ 。

定理 1.4. $\sum_{v \in V(G)} \deg_G(v) = 2|E(G)|$ 。

推論 1.5. 在一個圖中，具有奇數度的點數為偶數。

我們把奇數度的點稱為是奇點，而偶數度的點即為偶點。

定義 1.6. 所有點都是奇點的圖為奇圖 (Odd Graph)，都是偶點的圖稱為是偶圖。

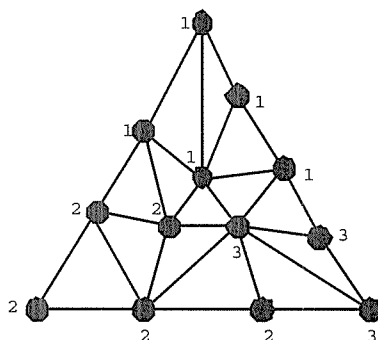
Quiz：有奇數點的奇圖嗎？

(註) Euler 在 18 世紀所發表的第一篇圖論論文就是探討偶圖的特性。

推論 1.5 是一個很容易了解的概念，卻具有很多漂亮的應用，以下我們證明 Sperner's 引理。

定理 1.7. (Sperner's Lemma)

令 T 為一個封閉的三角形，在 T 中利用加入一些點而形成大三角形細分成很多的小三角形 (Subdivision)，如下圖。令 T 的三頂點分別標示 1,2,3；1,2 邊上點標示 1 或 2；1,3 邊上點標示 1 或 3；2,3 邊上點標示 2 或 3，內部點標示 1 或 2 或 3。證明在 T 的內部必存在奇數個三角形它的三個頂點分別 1,2,3。



證明. 在每一個細分出來的三角形中各選一點當代表，並且在三角形外選一點形成一個圖的點集合 V 。現在，把 V 中代表相鄰三角形的兩點 u 與 v ，在共用邊的兩端點具有不同標示的數字時，連接成邊，於是得到一個圖 G 。由於三角形 T 邊上的點有規定的標示方式，我們可以推得在 G 中，代表 T 外部的那一點是奇點 (?)，因此，在內部代表小三角形的點中有奇數個奇點。由於奇點的度不會是 "1"，而且最大為 3，所以，全部都是度數為 3 的點。現在，度數為 3 的點所代表的三角形就是我們所需要的三角形。

利用上述定理，我們可以證明近代數學中最重要的一個定理。 ■

定理 1.8. (Brouwer's Fixed-point Theorem)

令 f 為由 D 對應到 D 的一個連續函數，其中 D 為一個封閉的 n 維單位原盤 ($n-disc$)。則在 D 中存在一點 x 使得 $f(x) = x$ 。

(註) 在 $n = 2$ 時， $n-disc$ D 與封閉的三角形 T 同態 (Homomorphic)，所以， $f : D \rightarrow D$ 可以想成 $f : T \rightarrow T$ 。

證明的概要.(平面)

- (1) 令 T 的三頂點 (座標) 為 x_1, x_2, x_3 , 則 $\forall x \in T, x = a_1x_1 + a_2x_2 + a_3x_3$, $a_i \geq 0, i = 1, 2, 3$, 且 $a_1 + a_2 + a_3 = 1$ 。令 $x = (a_1, a_2, a_3)$ 。
- (2) 令 $f(a_1, a_2, a_3) = (a'_1, a'_2, a'_3)$, $\sum_{i=1}^3 a_i = \sum_{i=1}^3 a'_i = 1$ 。
- (3) 令 $S_i = \{(a_1, a_2, a_3) | a'_i \leq a_i\}, i = 1, 2, 3$ 。(標示 i 的點在 S_i 中, $i = 1, 2, 3$)
- (4) 證明 $S_1 \cap S_2 \cap S_3 \neq \phi$ 。(證明 S_1, S_2, S_3 皆為 closed set。)
- (5) 如果 $(a_1, a_2, a_3) \in S_1 \cap S_2 \cap S_3$, 則 $f(a_1, a_2, a_3) = (a'_1, a'_2, a'_3) = (a_1, a_2, a_3)$ 。

作業1. 完整寫出上述的證明。

作業2. 利用圖的概念, 證明對於所有和為 n 的正整數 $n_1, n_2, \dots, n_t$, $\binom{n}{2} \geq \sum_{i=1}^t \binom{n_i}{2}$ 。

作業3*. 如下圖所示, A, B 由海拔高度相同的兩地出發, 證明 A, B 可以在一直維持相同高度的條件下最後在某處相遇。



補充說明. 如何將一個問題轉換成圖的形式再加以解決就是所謂的圖模式 (Graph Modeling)。



2. 子圖 (Subgraph)

基本上， G 的子圖代表著 G 的一部分；因此，如果 $H = (V', E')$ 為 $G = (V, E)$ 的子圖時，代表 $V' \subseteq V$ ，而且 $E' \subseteq E$ ，這敘述也是子圖的定義。例如圖 2-2 即為圖 2-1 的一個子圖。反過來稱 G 為 H 的擴充圖 (Supergraph)。

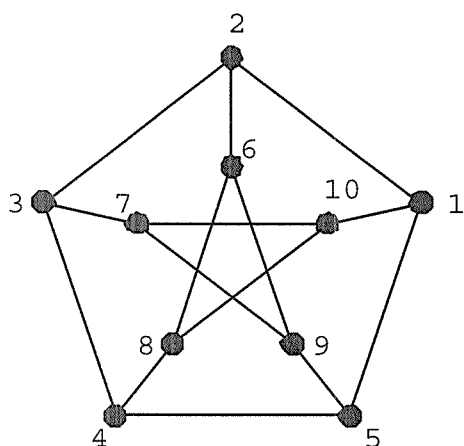


圖 2-1 : Petersen Graph(P)

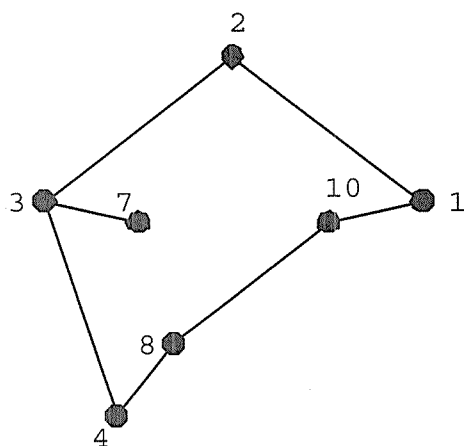


圖 2-2 : P 的一個子圖

顯然，「子圖」的關係和「子集」的關係都具有部份序 (Partial Order) 的性質，而且任一個 G 的子圖都可以經由去掉點或去掉邊而獲得，不過，「只利用去掉點的方式就不一定了」(?)。

定義 2.1. 令 e 為圖 $G = (V, E)$ 中的一個邊，亦即 $e \in E$ ，則 $G - e = (V, E \setminus \{e\})$ 。

定義 2.2. 令 v 為圖 $G = (V, E)$ 中的一個點，亦即 $v \in V$ ，則 $G - v = (V \setminus \{v\}, E')$ 其中 $E' = \{e \in E \mid v \notin e\}$ 。

定義 2.3. 我們稱 H 為 G 的一個導出子圖 (Induced Subgraph) 如果 H 可以由 G 經由去掉點的方式獲得；如果 $G = (V, E)$ ，而去掉的點所成的集合為 S' ，則 H 可以用 $\langle V \setminus S' \rangle_G$ 表示，或是 $\langle S \rangle_G$ ，其中 $S = V \setminus S'$ 。

從這個定義也可以看出， H 基本上是利用 S 中的點所建構出來的圖，而它所用的邊也就是 G 的邊，所以也可以說 H 是由 S 所生成 (Generate) 的圖。值得注意的是，子圖不一定要如上述的方式產生，例如圖 2-2 就不是圖 2-1 的導出子圖。

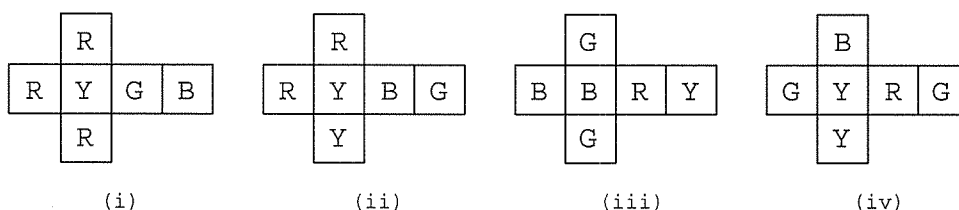
上述的圖是由點集合所形成，我們也可以用邊集合來生成一個圖。

定義 2.4. 令 $G = (V, E)$ 為一圖， $E' \subseteq E$ 。則 H 稱為是 G 的邊導出子圖 (Edge-induced Subgraph) 若且唯若 $H = (V', E')$ ，其中 $V' = \{v \in V \mid v \text{ 是 } e' \text{ 中某一邊的點}\}$ 。 H 也可以表成 $\langle E' \rangle_G$ 。

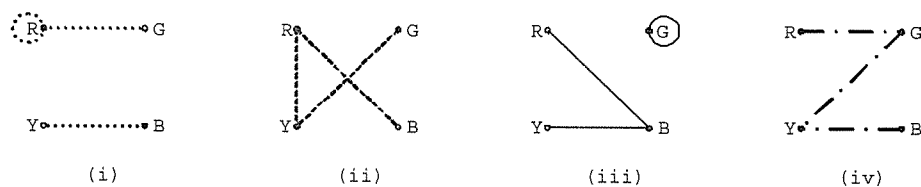
子圖的結構對於原圖的了解有很大的幫助之外，適當地利用它也可以解決重要的問題。

以下是一個圖論中有趣的問題，但是介紹這個問題之前我們需要一些預備知識：以下我們把自己連到自己的邊稱為是弧圈 (Loop)，兩點連很多次所產生的數個邊稱為是重邊 (Multi-Edge)。

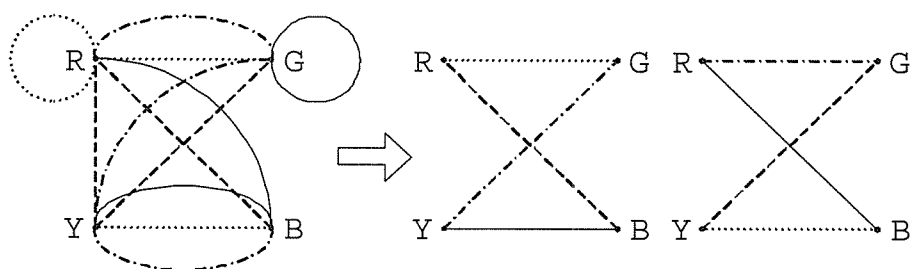
問題. 我們可否把下面的四個正六面體疊成四層，使得不管由哪一面看過去都具有四種不同的顏色？



(解) 令 R, Y, G, B 分別代表四點，然後就 (i), (ii), (iii), (iv) 的形狀分別把正六面體“對”面的顏色連接起來，於是四個圖分別如下：



再將四個圖合成一個大圖，這個圖有兩個子圖分別有四邊而且來自不同的正六面體：利用它們即可找到正確的排法。(?)



(註) 找到的兩個子圖沒有共用邊，而且每個點的度數皆為 2。

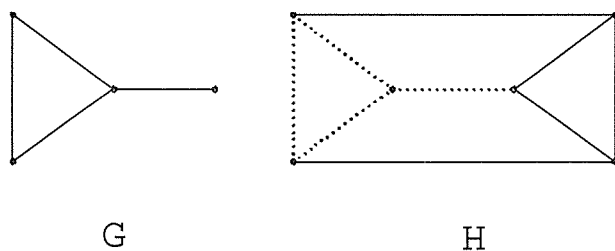
作業 4. 可以辦到的充要條件為何？

作業 5. 如果每個顏色均個出現六次，是否一定可以辦到？

定義 2.5. 一個圖中，如果每個點的度數都是一樣，則此圖被稱為是正則圖 (Regular Graph)；當每個點的度數都是 r ，則此圖為 r -正則圖。

問題. 任給一個圖 G ，是否一定存在一個 $\Delta(G)$ -正則圖 H ，使得 G 為 H 的導出子圖，這裡的 $\Delta(G)$ 代表圖 G 中出現的最大度數。

例.



作業 6. 證明 H 一定可以找到，而且可以找到一個 H 它的點數不大於 $2|V(G)|$ 。

3. 圖的運算 (Graph Operations)

前面所提到的去掉一邊，或去掉一點，基本上就是一種運算 (Operation)，以下所要介紹的是將兩個圖放在一起，也許用二元運算 (Binary Operation) 來說明比較恰當。

定義 3.1. (聯集)

$G_1 = (V_1, E_1)$ 與 $G_2 = (V_2, E_2)$ 的聯集以 $G = (V, E)$ 表示，其中 $V = V_1 \cup V_2$ ， $E = E_1 \cup E_2$ 。

定義 3.2. (交集)

$G_1 = (V_1, E_1)$ 與 $G_2 = (V_2, E_2)$ 的交集為 $(V_1 \cap V_2, E_1 \cap E_2)$ 。

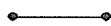
定義 3.3. (卡迪遜積，Cartesian Product)

令 G 與 H 為兩個圖， G 與 H 的卡迪遜積為 $G \times H = (V, E)$ ，其中 $V = V(G) \times V(H)$ ， $E = \{\{(u, x), (v, y)\} | u \sim_G v \text{ 且 } x = y \text{ 或 } u = v \text{ 且 } x \sim_H y\}$ 。

一般說到兩個圖的乘積指的就是這種乘積。

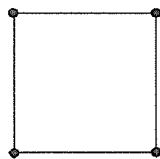
例1. n 維方塊 (n -cube)

$n=1$



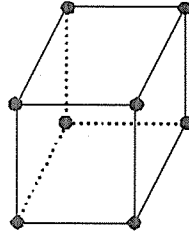
$$(K_2) = Q_1$$

$n=2$



$$(K_2 \times K_2) = Q_2$$

$n=3$



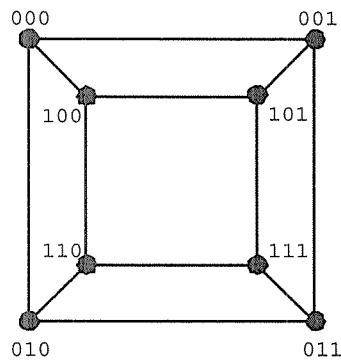
$$(Q_2 \times K_2) = Q_3$$

n 維方塊 (n -cube)

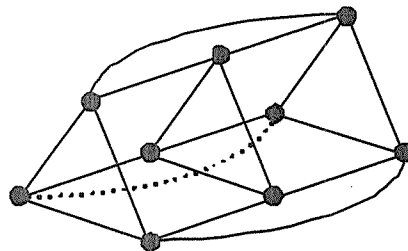
$$Q_n = (Q_{n-1} \times K_2)$$

n 維方塊具有很多漂亮的性質，其中一個是它的點集合可以用 $\mathbb{Z}_2^n$ 來表示，而且兩點相連的充要條件為兩點的座標恰有一個位置不相等。

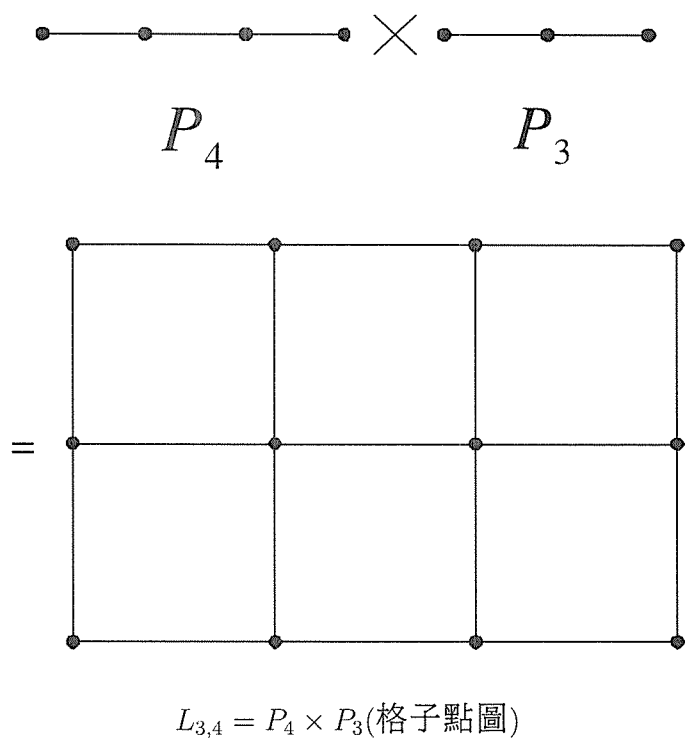
例2.



例3. $K_3 \times K_3$

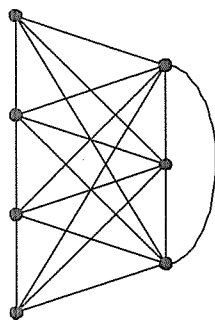


例4. $P_4 \times P_3$



定義 3.4. 兩個不相交的圖 G 與 H 相連之後所得到的圖為 $G \vee H = (V, E)$, 其中 $V = V(G) \cup V(H)$, $E = E(G) \cup E(H) \cup \{(u, v) | u \in V(G), v \in V(H)\}$ 。

例5. $P_4 \vee K_3$



例6. $O_4 \vee K_3$ (完全分裂圖 , Complete Split Graph)

證明. 將圖上的邊逐一去掉, 但是要維持連通性。

(註) 樹可以看成是不含任何圈 (Acyclic) 的連通圖。

定義 5.18. (森林, Forest)

不含任何圈的圖稱為是森林。

定義 5.19. (部分, Component)

以 " $\leq$ " 為關係的最大連通子圖稱為是一個圖的部分。一個圖 G 的部分的個數以 $c(G)$ 表示。 $c(G) = 1$ 的圖 G 就是一個連通圖, 孤立點也可以看成是一部分。

定義 5.20. (橋, Bridge (狹義))

在一個圖 G 中, 若 $c(G - e) > c(G)$, 則 e 稱為是 G 的一個橋。

在圖論中, 橋可以定義成另一種比較廣義的圖; 但是就上述的定義而言, 實際上若 e 是橋, 則 $c(G - e) = c(G) + 1$ 。

引理 5.21. G 為樹圖, 若且唯若 G 為連通圖而且每一邊都是橋。

定理 5.22. 下列的敘述彼此等價:

- (1) G 為樹圖。
- (2) G 是不含任何圈的連通圖。
- (3) G 是恰有 $|V(G)| - 1$ 邊的連通圖。
- (4) G 是不含任何圈而且具有 $|V(G)| - 1$ 邊的圖。
- (5) G 中的任兩點皆有唯一的一條路徑相連。
- (6) G 為連通圖而且每一邊都是橋。
- (7) 對於 G 中不相連的兩點 u, v , $G + \{u, v\}$ 必含唯一的一個圈。

證明. 作業 13.

作業 14. 已知 G 為一圖, 它的度數列 $(d_1, d_2, \dots, d_p)$ 滿足 $\sum_{i=1}^p d_i = 2(p-1)$ 。

証明由這個數列可以畫出一個樹圖它的度數列恰為 $(d_1, d_2, \dots, d_p)$ 。



6. 有向圖 (Digraph)

有向圖的點和一般圖一樣，它的邊則用點與點之間所形成的有序對來取代；所以有向邊 (u, v) 與 (v, u) 是不同的兩邊，爲了區分有向邊及一般邊，有向邊一般以弧 (arc) 稱之。圖 D 的弧所成的集合就用 $A(D)$ 表示。

定義 6.1. (內度與外度, Indegree vs. outdegree)

一個點 v 在有向圖 $D = (V, A)$ 中的內度 $id_D(v) = |\{u | (u, v) \in A\}|$ ，而外度 $od_D(v) = |\{w | (v, w) \in A\}|$ 。

所以，下面的結果就不難推得。

引理 6.2. 在一個有向圖 D 中， $|A(D)| = \sum id_D(v) = \sum od_D(v)$ 。

一個有向圖之連通性一般分成兩種，一種是根據圖的結構 (不考慮方向)，如此一來這個圖可能是一種重邊圖 (最多二重邊)；我們稱此有向圖爲弱連通圖 (Weakly Connected)，若且唯若在不考慮方向 (即 $(u, v) \rightarrow \{u, v\}$) 的情況下此圖爲連通圖。另一方面，如果圖中的任兩點 u, v ，均存在一個有向路徑， $u = v_0 - v_1 - v_2 - \cdots - v_t = v$ 連接 u, v ，即對於所有的 $0 \leq i \leq t-1$ ， (v_i, v_{i+1}) 爲圖上的弧，則此圖稱爲是強連通圖 (Strongly Connected)。顯然，在網路上這種圖最受歡迎的。值得注意的事：在有向圖中，一般討論有路徑相連是指有向路徑，而連通與否也決定於是否任兩點都存在有向路徑相連。

作業 15. 令 D 爲一有向圖它滿足任意 $V(D)$ 中的兩點 u, v ，不是 $(u, v) \in A(D)$ 就是 $(v, u) \in A(D)$ 。証明在 D 中必存在一條有向路徑，由一點出發到另一點而且通過 $V(D)$ 中的其它所有點。

(註) 上述圖 D 一般稱爲競賽圖 (Tournament)。

定義 6.3. 在一個有向圖 D 中，如果由 $(u, v), (v, w) \in A(D)$ 可以推得 $(u, w) \in A(D)$ ，我們稱 D 爲遞移圖或遞移有向圖 (Transitive Digraph)。

定義 6.4. 在一個有向圖 D 中，我們稱點的外度爲該點的分數 (Score)。

定義 6.5. D 中具有最大分數的點稱爲國王 (King)。

定理 6.6. 在一個遞移競賽圖 D 中，國王唯一存在。

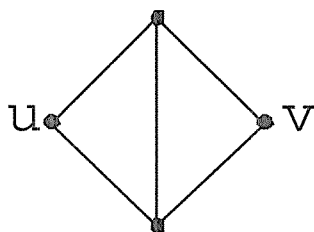
証明. 假如有兩個以上。令 u 與 v 皆為國王。由於 D 為競賽圖，所以 $(u, v) \in A(D)$ 或是 $(v, u) \in A(D)$ ，但是不會同時成立。不失一般性，假設 $(u, v) \in A(D)$ ，於是 $od_G(u) \geq od_G(v) + 1$ ，這是因為當 $(v, w) \in A(D)$ 時 (u, w) 也會在 $A(D)$ 中（遞移性質）。所以 u 與 v 不可能同時是國王。這與假設矛盾。

推論 6.7. 在一個競賽圖中，如果有兩點它們具有相同的外度，則此圖不具遞移性質。

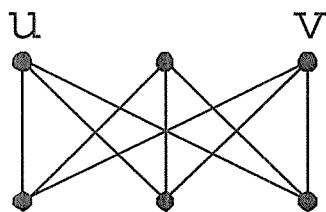
在這一節的最後，我們介紹一個遊戲 (Game)，它具有非常好的圖結構概念。

GameA. 給定一個圖 G ，及 G 上的兩點 u 與 v 。現在有兩位 Players 甲與乙他們分別扮演進攻與防守的角色，甲希望由 u 到 v 建構一條有向的路徑通到 v ，但是每次只能在一個尚未指定方向的邊上指定方向（把邊改成弧），而乙則與甲輪流在未指定方向的邊上指定方向，而他的目標是讓 u 到 v 找不到有向路徑。（甲贏：甲找到路徑，乙贏：甲找不到路徑。）

例. 甲先選會贏的例子。



例. 甲先選也不會贏。



問題. 那一類圖會是先選先贏？那一類圖是甲必贏？那一類圖是乙必贏？

GameB. 把 GameA 找到有向路徑改成路徑，而甲，乙分別輪流選邊來形成子圖。若是甲所選出來的邊可以形成一條由 u 到 v 的路徑則甲贏，否則乙贏。

問題. 在什麼情況下 (何種圖) 甲必定會贏 (先選或後選都可以)? 在什麼情況下 (何種圖) 乙必定會贏 (先或後)? 在什麼情況下先選會贏?
作業 16**. 回答上述兩個問題。



7. 加權圖 (Weighted Graph)

一個加權圖是指一個一般圖再加上定義於 $V(G) \cup E(G)$ 上的一個加權函數 w 。在應用上，大部分的加權都是在邊上，所以 w 可以看成是由 $E(G)$ 映至某個集合的函數；這個時候，如果有所有比較大小的加權，則 $w: E(G) \rightarrow R$ ，不然的話，其它如 $\{+, -\}$ 亦可。以下我們就探討一個與大小有關的問題，令 (G, w) 為一加權圖，其中 $w: E(G) \rightarrow R^+$ 。現在考慮以下的問題：

問題. (最小懸掛樹, Minimum Spanning Tree)

在一個連通的加權圖中，如何找到一個懸掛樹，使得它的邊上之加權總和為最小？

如果我們把邊上的加權看成是在網路上維持該邊的成本，則最小懸掛樹是指一種在維持連通的前提之下使得成本最低的連接方法。由於圖的變化很大，要找出一個定性的方法直接計算加權或直接求得答案並不容易，於是演算法成為最有力的工具。

求最小懸掛樹的演算法 (Kruskal)

任給一個連通的加權圖 (G, w) 。

步驟 1. 令 $i = 0$ 以及 $E_0 = \phi$ 。

步驟 2. 在圖 G 中選一邊 $e_i \notin E_{i-1}$ 而且 $w(e_i)$ 為最小，使得 $T_i = \langle E_{i-1} \cup \{e_i\} \rangle$ (邊導出子圖) 不含任何的圈，同時定義 $E_i = E_{i-1} \cup \{e_i\}$ 。若是找不到這種 e_i ，則令 $T = \langle E_{i-1} \rangle$ 並且停止演算法。

步驟 3. 用 $i+1$ 取代 i ，同時回到步驟 2。

綜合上述三個步驟，基本上演算法的過程是不斷地加入加權最小 (未用) 的邊，同時維持圖上沒有圈即可。

定義 7.1. 上述 Kruskal 的演算法的確可以在任意的連通加權圖中找到最小的懸掛樹。

証明. 令 (G, w) 為給定的加權圖，由演算法不難看出最後求出的答案 T 必定是 G 的懸掛樹，所以 $V(T) = V(G)$ ，而且 $|E(T)| = |V(G)| - 1$ 。

令 $E(T) = \{e_1, e_2, \dots, e_{p-1}\}$ ，同時 $w(T) = \sum_{i=1}^{p-1} w(e_i)$ 。現在，假設 T 不是最小的懸掛樹，於是在 G 中可以找到其它的最小的懸掛樹，它的總加權小於 $w(T)$ 。令 H 是其中一個最小懸掛樹它與 T 有最多的共用邊。由

於 $E(H) \neq E(T)$ ，令 $e_i, 1 \leq i \leq p-1$ ，為一個在 $E(T) \setminus E(H)$ 的邊（由 $1, 2, \dots$ ，看起）。由樹的性質，我們知道 $H \cup \{e_i\}$ 必含一圈，而且在這一圈上必含一邊 $e_0 \notin E(T)$ 。令 $T' = (H \cup \{e_i\}) \setminus \{e_0\}$ ，則 T' 也是 G 的一個懸掛樹，同時 $w(T') = w(H) + w(e_i) - w(e_0)$ 。

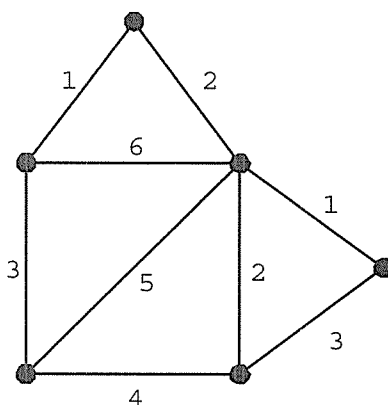
由於 H 為最小懸掛樹，所以 $w(H) \leq w(T')$ ，因此 $w(e_i) \geq w(e_0)$ ，另一方面，由演算法 $w(e_i) \leq w(e_0)$ （？），所以 $w(e_i) = w(e_0)$ ，這麼一來 $w(H) = w(T')$ ， T' 也是最小懸掛樹。然而 $|E(T') \cap E(T)| > |E(H) \cap E(T)|$ ，這與假設矛盾，所以 T 應該就是最小的懸掛樹。

除了 Kruskal 的演算法之外，尚有其它類似的演算法也可以求出最小的懸掛樹。

方法 2. 去掉加權最大的邊只要連通性可以維持即可。

方法 3. 任選一點 x_1 ，然後在 x_1 的鄰域中選一點 x_2 使得 $w(x_1, x_2)$ 為最小，接著再看 $N(x_1) \cup N(x_2) \setminus \{x_1, x_2\}$ ，再選能加入的最小加權邊；再繼續擴充考慮尚未選過的“點”，最後獲得一最小懸掛樹。

作業 17. 求下面加權圖的最小懸掛樹。



8. 圖的表示法

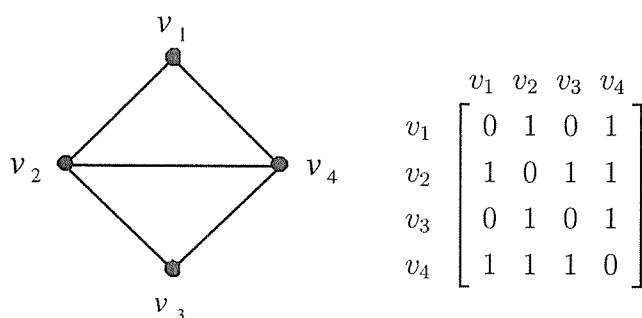
圖固然可以很容易地畫出來，但是畫法很多而且儲存不便，因此，適當地將圖轉換成其它形式來表示出來，也很重要。

定義 8.1. (相鄰矩陣, Adjacency Matrix)

一個圖 G 相鄰矩陣 $A(G)$ 為一 $p \times p$ 的方形矩陣，其中 $V(G) = \{v_1, v_2, \dots, v_p\}$ ，同時在 $v_i \sim v_j$ 時 $A(i, j) = A(j, i) = 1$ ，而 $v_i \not\sim v_j$ 時 $A(i, j) = A(j, i) = 0$ 。

於是，同構的圖自然就會有對應的相鄰矩陣在適當的換行（對稱）之後得到相同的矩陣。

例 1.



圖的相鄰矩陣提供了相當多研究圖性質的資訊。由於矩陣的行列都是對應到固定的點，所以第 i 列表示的就是與 v_i 相關，同理第 j 行與 v_j 相關。

引理 8.2. 令 A 為 G 的相鄰矩陣，則 $A^n(i, j)$ ，矩陣 A^n 的 (i, j) 位置所得到的值就是在圖 G 中由 v_i 到 v_j 不同的 n -步步行方法數。

證明. 我們對 n 歸納，顯然當 $n = 1$ 時，敘述成立。現在假設當 $n = k$ 時，敘述成立，亦即 $A^k(i, j)$ 為由 v_i 到 v_j 走 k 步可以到達的方法數。現在考慮 $A^{k+1}(i, j)$ 。我們要證明由 v_i 走 $k+1$ 步到 v_j 的方法數恰為 $A^{k+1}(i, j)$ 。但是 $A^{k+1}(i, j)$ 可以看成是 A^k 的第 i 列與 A 的第 j 行的內積（把列與行看成向量）。這個內積的值代表著 v_i 走 k 步到 v_k ， $k = 1, 2, \dots, n$ ，然後再走一步到達 v_j ，於是它代表由 v_i 到 v_j 走 $k+1$ 步的總方法數，定理得証。

定理 8.3. 令 $C_n(G)$ 代表在圖 G 中不同 n -圈之個數，則

$$(1) C_3(G) = \frac{1}{6} \text{tr}(A^3)。$$

$$(2) C_4(G) = \frac{1}{8} \left[\text{tr}(A^4) - 2q - 2 \sum_{i \neq j} A^2(i, j) \right]。$$

$$(3) C_5(G) = \frac{1}{10} \left[\text{tr}(A^5) - 5 \text{tr}(A^3) - 5 \sum_{i=1}^p \left(\sum_{j=1}^p A(i, j) - 2 \right) A^3(i, i) \right]。$$

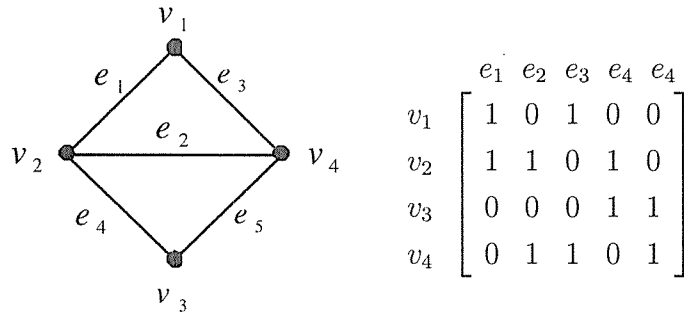
証明. $A^3(i, i)$ 代表著從 i 經過 3 步 (邊) 回到原地 i 的相異步行數。由於同一個三角形將被算到 6 次這樣的步行數，所以可以求出 (1)。在 (2) 的部分，4 步回到原地的方法有很多，不形成 4 邊形都需要扣掉；譬如一個邊來回各走了兩次，或者是有兩個邊相鄰，一邊走兩次，另一邊也走兩次，所以求得 (2)；(3) 就要更複雜一些，細節在此省略。

另外，我們也可以看到利用相鄰矩陣本身的性質，例如求出該矩陣的特徵值譜 (Spectrum)，來刻劃一個圖，這種概念我們留在後面再詳細介紹。除了相鄰矩陣之外，相接 (Incidence) 矩陣也是很重要的表示法。

定義 8.4. (相接矩陣, Incidence Matrix)

令 $G = (V, E)$ ，其中 $V = \{v_1, v_2, \dots, v_p\}$ ， $E = \{e_1, e_2, \dots, e_q\}$ ，則相接矩陣 $B(G)$ 為一 $p \times q$ 矩陣，其中若是 v_i 為 e_i 的一個端點則 $B(i, j) = 1$ ，不然的話 $B(i, j) = 0$ 。

例 2.



顯然，在 $B(G)$ 中任一行都恰有兩個 1，而每一列 1 的個數，恰為對應點的度數，因此，以下的性質就不難看出來。

引理 8.5. 令 $L(G)$ 為圖 G 的邊圖 (Line Graph)，則 $A(L(G)) = B(G)^T B(G) - 2I_q$ 。

證明. $B(G)^T B(G)$ 中不為零的位置 (i, j) 表示 e_i 與 e_j 有共同點, (當 $i = j$ 時有兩點, 不在考慮範圍), 故得証。

作業18. 証明定理 8.3 (3)。



9. 尤拉迴路 (Euler Circuit)

它可以說是圖論研究的第一問題，早在二百多年前就已經開始了。當時，瑞士數學家尤拉 (Leonhard Euler) 因為研究七橋問題而發展出這種圖。所謂七橋問題是發生在現今德國境內：當布魯格 (Pregel) 河流經哥尼斯堡 (Königsberg) 附近時形成如下圖，為了彼此的聯繫在當時共建了七座橋，有不少人試著由一個地方出發，希望在不重複的情況下走完此七座橋，然後又回到原地。問題是可以辦到嗎？

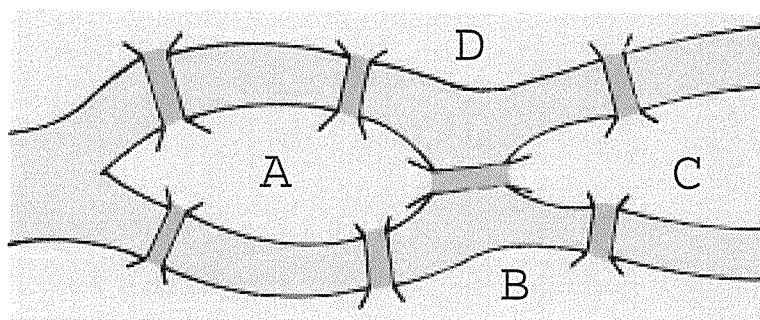


圖 1：哥尼斯堡的七橋

以圖的方式來看圖 1，則可以把 A,B,C,D 看成是點，橋為邊，於是得到一個重邊圖，圖 2。現在七橋問題等於是問在圖 2 中可否找到一個迴路通過所有的邊。

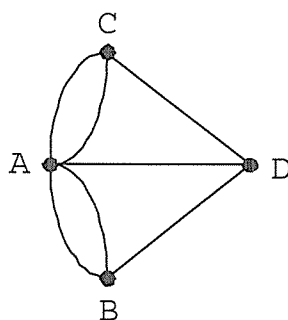


圖 2：七橋問題之對應圖

看出圖 2 不含任何通過所有邊的迴路是不困難的事，因為一旦有這種迴路存在，則圖上的所有點必然都是偶點（進出的現象），但是要證明逆敘述就得花一點力氣了。

定義 9.1. (尤拉圖 , eulerian graph)

一個圖若是存在一個迴路通過所有的邊, 則此圖稱為是尤拉圖, 此時, 該迴路也稱為是圖中的一個尤拉迴路。

定理 9.2. 一個圖 G 是尤拉圖的充要條件是圖 G 為連通圖且圖中的每一點都是偶點。

證明.

($\Rightarrow$) 如上述說明。

($\Leftarrow$) 由歸納法, 顯然當 G 只有一點或兩點時原敘述必成立。

假設當 $|V(G)| \leq n$ 時原敘述成立, 亦即當每個點都是偶點時, G 為尤拉圖。令 H 為一連通圖, 其中 $|V(G)| = n + 1$ 而且 H 中的每一點都是偶點。令 $v \in V(H)$, 而且 $N_H(v) = \{v_1, v_2, \dots, v_{2i-1}, v_{2i}\}$ 。現在考慮 $H' = (H - v) \cup (v_1v_2, v_3v_4, \dots, v_{2i-1}v_{2i})$ 。顯然 H' 具有 n 個點, 而且每個點都是偶點。同時 H' 最多是 i 個部分 (Components) 所形成的圖。由歸納假設, 每個部分都是尤拉圖; 現在把每個部分的尤拉迴路畫出來, 當這些迴路通過 $v_1v_2, v_3v_4, \dots, v_{2i-1}v_{2i}$ 中的任一邊 $v_{2j-1}v_{2j}$ 時, 將此迴路改道成 $-v_{2j-1} - v - v_{2j} -$ (去掉 $v_{2j-1}v_{2j}$), $j = 1, 2, \dots, i$, 則我們求得 H 的一個尤拉迴路, 所以 H 為尤拉圖, 定理得證。 ■

這個定理的證明尚有其它不同的方式, 請自行另證(作業19.)。

雖然我們可以證明定理 1.2, 但是在一個複雜的尤拉圖把尤拉迴路畫出來的確不是一件容易的事; 以下的演算法就針對如何有效地找到尤拉迴路提供一個方法。

任給一個尤拉圖畫尤拉迴路之演算法 (Fleury)

步驟 1. 令 $i = 0$, 並且任選 G 中的一點 v_0 以及定義 $T_0 : v_0$ 。

步驟 2. $T_i : v_0, v_1, \dots, v_i$ 這個步徑已經找到, 接下來再找 v_{i+1} , 這時候 v_jv_{j+1} 不是 $G \setminus \{v_0v_1, v_1v_2, \dots, v_{i-1}v_i\}$ 中的一條橋。若無 v_iv_{i+1} 存在, 則停止。

步驟 3. 定義 $T_{i+1} = T_i + v_iv_{i+1}$ 。

步驟 4. 用 $i + 1$ 取代 i , 然後回到步驟 2。

上述演算法簡單說就是要避開橋, 除非該邊是剩下的唯一邊。

定理 9.3. Fleury 的演算法所找到的就是尤拉迴路。

證明. (扼要說明) 由於尤拉圖中的每一個點都是偶點，所以當這演算法繼續做，最後一定會接回原來出發的點，所以擔心的是無法經過所有的邊。

然而，如果沒有走過全部的邊表示中途在可以選擇的情況下選了橋，那是演算法所避免的；因此不會剩下邊沒走過。(全部是偶點的圖不會有橋存在!(作業20.))。 ■

有些圖，雖然不是尤拉迴路，但是，它可以由一點出發在走完所有邊時到達另一點，也就是有一個經過全部邊的步徑，一般也稱為是一筆畫(eulerian trail) 如下圖。

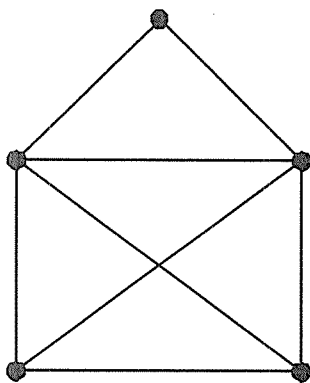


圖 3：一筆畫的圖

定理 9.4. 一個連通圖可以一筆畫的充要條件為該圖最多有兩個奇點，而且這一筆畫的起點與終點 就是可能存在的兩個奇點。

證明. 若是 G 中的所有點皆為偶點，則 G 自然可以一筆畫，因此假設 u, v 為 G 中的兩個奇點。首先，如果 $uv \notin E(G)$ ，則連接 u 與 v 得到 G' ；現在 G' 為一尤拉圖，令 C 為一個尤拉迴路，則 $C - uv$ 就是 G 的一個步徑，由 u 到 v ，而且通過所有 G 中的邊。再來，如果 $uv \in E(G)$ ， G' 為一圖，它滿足 $V(G') = V(G) \cup \{w\}$ 以及 $E(G') = E(G) \cup \{wu, wv\}$ 。同理 G' 為一尤拉圖， C 為一尤拉圖，此時從 C 中扣掉 wu 及 wv 就可以得到 G 的一筆畫。 ■

我們可以把尤拉圖推廣至有向圖，這時候的步徑自然要有方向，而且要一致，所以它的要求就會更多一些。

定理 9.5. 令 D 為一有向的連通圖。則 D 為尤拉圖的充要條件為在 D 中的每一點 v , $od_D(v) = id_D(v)$ 。

作業 21. 證明定理 9.5. (*)

作業 22. 證明若 G 為尤拉圖, 則 $L(G)$ 必為尤拉圖 ; 而它的逆敘述不一定對。

作業 23. 在下圖中要去掉多少邊才會使得剩下的部份圖是尤拉圖。

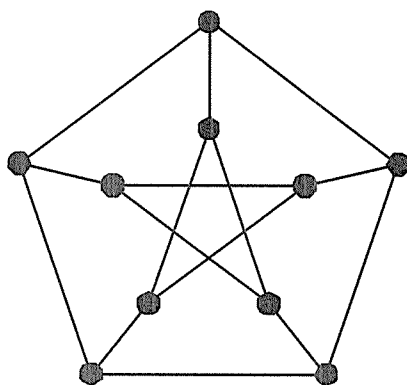


圖 4



10. De Bruijn 數列及郵差問題

這一節，我們將討論兩個尤拉圖概念的應用。

定義 10.1. (De Bruijn Sequence)

一個循環數列 $(a_1, a_2, \dots, a_{2^n})$ 稱為是 $(2, n)$ -de Bruijn 數列如果下列兩個條件滿足：

- (1) $a_i \in \{0, 1\}$, $i = 1, 2, \dots, 2^n$; 且
- (2) $(a_j, a_{j+1}, \dots, a_{j+n-1})$, $j = 1, 2, \dots, 2^n, (\text{mod } 2^n)$, 為相異的 2^n 個 n 維向量。

例 1. $n = 1, 2, 3, 4$; $(2, n)$ -de Bruijn 數列分別為 (括號, 逗號省略)

01, 0110, 01110100, 0000100110101111

對於所有的 n 要建構一個 $(2, n)$ -de Bruijn 數列並不困難，以下是荷蘭數學家 N. de Bruijn 用來尋找這種數列的有向圖。

定義 10.2. ($(2, n)$ -de Bruijn 有向圖, $D_{2,n}$)

$D_{2,n}$ 為一加權有向圖，它滿足下列兩條件：

- (1) $V(D_{2,n}) = (Z_2)^{n-1}$ 及
- (2) $(a_1, a_2, \dots, a_{n-1})$ 連到 $(a_2, a_3, \dots, a_{n-1}, a_n)$ 並且在這個弧上給予加權 $(a_1, a_2, \dots, a_n)$ 。

例 2.

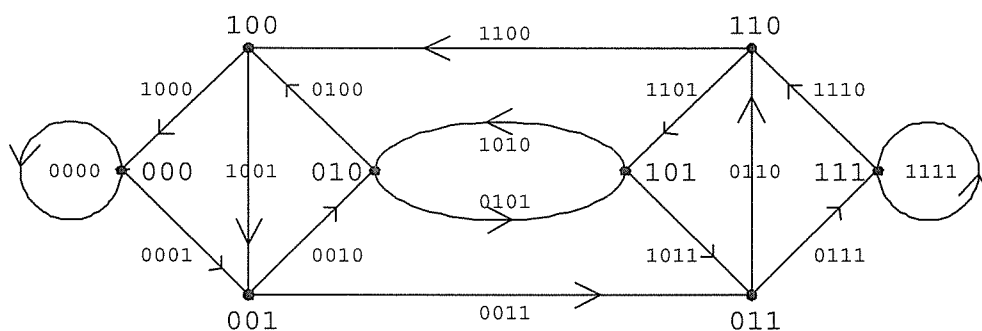


圖 4: $D_{2,4}$

由定義，我們可以證明 $D_{2,n}$ 為一有向的尤拉圖。

引理 10.3. $(2, n)$ -de Bruijn 有向圖為尤拉圖。

證明. 因為 $D_{2,n}$ 為強連通圖而且每一點的內度與外度均為 2, 故得證。 ■

引理 10.4. 在 $(2, n)$ -de Bruijn 有向圖上的加權全部不一樣。

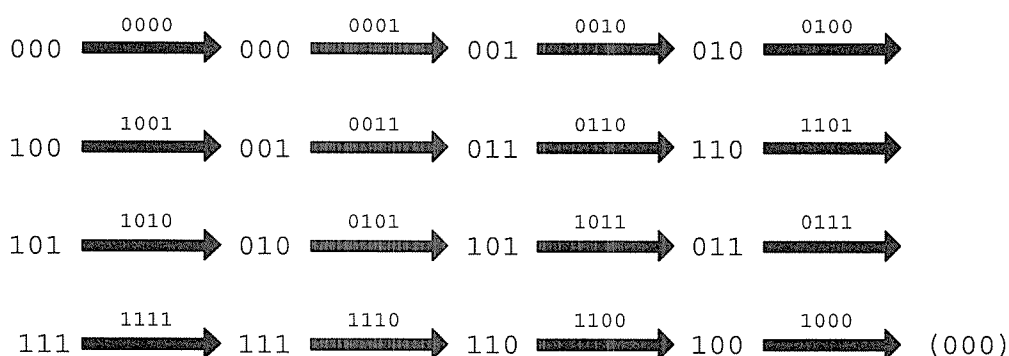
證明. 由定義可得。 ■

引理 10.5. 對於所有的 n , $(2, n)$ -de Bruijn 數列存在。

證明. 首先, 由引理 2.3, 一個 $(2, n)$ -de Bruijn 有向圖存在, 所以存在一個尤拉迴路; 現在, 令此迴路經過的邊為 $e_1, e_2, \dots, e_{2^n}$; 同時對於所有的 i 令 $l(e_i) = a_i$ 為 e_i 邊上加權的最左邊那個數字, 於是我們得到一個數列 $(a_1, a_2, \dots, a_{2^n})$, 這個數列就是一個 $(2, n)$ -de Bruijn 數列。 ■

以下的例子也許有助於了解上述的證明。

在 $D_{2,4}$ 中的尤拉迴路可以是



$(2, 4)$ -de Bruijn 數列為 0000100110101111, 而它所產生的 16 個不同之向量依次為 16 個加權向量。

De Bruijn 數列最出名的應用是在旋轉鼓 (Rotating Drum)。它可以利用連續的位置來判斷不同的輸入 (機械原理), 如下圖所示。

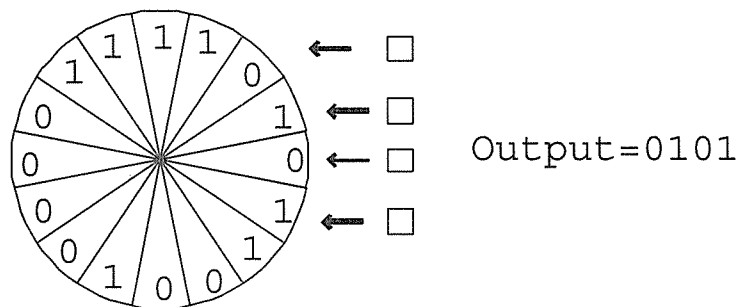


圖 6：旋轉鼓

另一個出名的應用是由華裔數學家管梅各 (Meigu Guan) 所提出來, 後來 J. Edmonds 稱它為中國郵差問題。

定義 10.6. (郵差路線, Postman Tour)

任給一個圖 G , 所謂郵差路線是指在 G 中的一個封閉步行 (Closed Walk), 它通過 G 中的每一個邊至少一次。

定義 10.7. (最佳的郵差路線, Optimal Postman Tour)

在一個加權圖中, 一個郵差路線所經過的邊加權總和為最小時, 我們稱這路線為最佳郵差路線。

這個問題看起來十分複雜。因為所考慮的圖千變萬化, 然而, 在提出這個問題 (尋找最佳郵差路線) 之後不到十年, 即由 Edmonds 和 Johnson 所解決, 他們提供了一個多項式時間 (Polynomial-time) 的演算法來找出最佳路線。

在沒有介紹演算法之前, 我們先看一個例子。在 G 中有 4 個奇點, 現在利用這 4 點 建構一個加權的 K_4 , 在邊上的加權為兩點在 G 中的最短距離 (加權)。接著在 K_4 中找到不相鄰的兩邊 (配對) 使得它們的加權和最小, 此例中的 bf 即 dh , 最後再將原圖 G 改為 G^* , 如圖 7, G^* 的尤拉迴路即為所求。

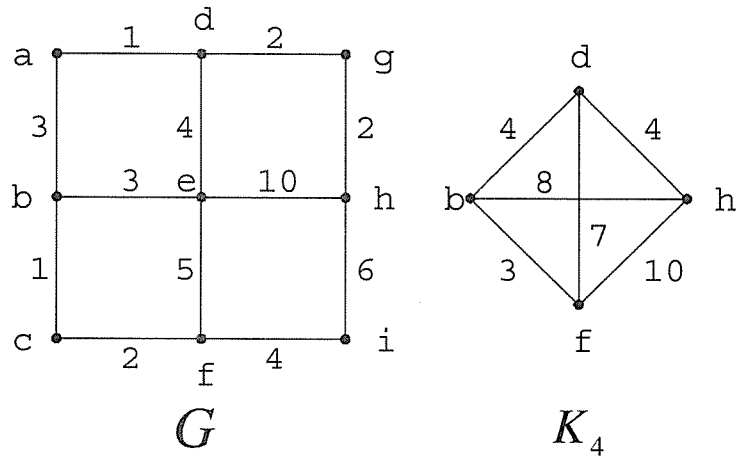


圖 7：最佳郵差路線

演算法 (Edmonds 及 Johnson)(摘要)

1. 令 S 為 G 中奇點所成的集合。
2. 利用 S 的點建構一個加權完全圖 $K_{|S|}$ ， ab 邊上的加權為由 a 到 b 的最短距離（加權路徑）。
3. 在 $K_{|S|}$ 中選出 $|S|/2$ 個獨立邊，使得它們的加權總和為最小。
4. 對應於 3 中的獨立邊，將所經過的邊（最短路徑）全部加上相同加權的邊。
5. 由 4 所得到的圖之尤拉迴路即為所求。

證明. 3 的找法是多項式時間，所以這個演算法也是，詳細證明在此省略。



其它方面尚有不少應用，例如在掃街路線的安排，要怎樣行駛才最省經費；在 RNA 的重組方面如何利用片斷資訊來得到完整的 RNA；以及在資訊傳遞方面如何編碼才更有效率地把資料表現出來，都可以利用有向的尤拉圖來達到目的；由於篇幅有限在此省略，請自行參考由 Gross 及 Yellen 所寫的書 "Graph Theory and its Application"。

一個人真正的價值，是覺得自己有用並且快樂。



11. 哈密爾頓圖 (Hamiltonian Graph)

一個哈密爾頓圈是通過圖上所有點的圈，而具有哈密爾頓圈稱為是哈密爾頓圖 (Hamiltonian Graph)。這種圖有此名字據說是因為愛爾蘭數學家哈密爾頓 爵士在 1857 年發明了一種環遊世界的遊戲；它是在一檜木製的正十二面體的 20 個角落填入當時著名的 20 個都市，如倫敦、紐約、... 等，然後讓 參加遊戲者任意由一個都市出發，在不重覆都市的情況下走完全部都市並回到 起點。(圖 8)

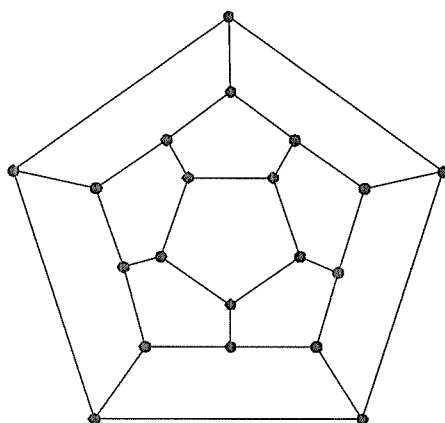


圖 8：正十二面體

不像尤拉迴路的尋找那麼幸運，決定一個圖是否有哈密爾頓圈是非常困難的問題，儘管圖論學家在一百多年來不斷地努力，目前仍然停留在提供具有哈密爾頓圈的充分條件，而無法刻劃出何種圖是哈密爾頓圖。我們先從判斷一個圖不是哈密爾頓圖開始。

觀察：

由於哈密爾頓圈要通過圖上的所有點，所以每個點至少要碰到兩個邊；因此 (1) 如果有度數為 2 的點，則與該點相接的兩邊就毫無疑問的是圈上的兩邊，於是保留這些邊，再繼續看需要那些邊；此時 (2) 所加入的邊不能和已經有的邊形成一個點數不到全部的”小”圈，同時 (3) 一旦一個點的附近已經確定有兩邊是必要的邊，則其餘的邊都可以刪去。

以上的 (1), (2), (3) 成為是判斷一個圖它不含哈密爾頓圈的一個簡單方法，不過，當每個點的度數都不小時，要用上述法仍有相當大的難度。

例 1. 判斷下圖是否具有哈密爾頓圈。(作業 24.)

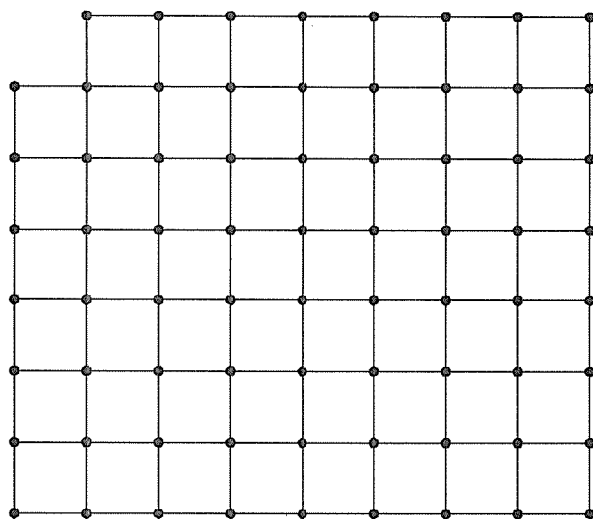


圖 9

接下來，我們探討什麼樣的圖會有哈密爾頓圈。直覺上，如果圖中的每一個點度數都不小，則出現哈密爾頓圈的機會就比較大；所以，在 1952 Dirac 即證明了以下的定理。

定理 11.1. 當 $p \geq 3$ 時，一個 p 點的簡單圖中，如果每個點的度數至少為 $p/2$ ，則這個圖中有哈密爾頓圈。

這個結果在數年後被 Ore 的定理所涵蓋，在這裡就對 Ore 的結果作一簡要證明。其它有不少充分條件，無法在此一一說明。

定理 11.2. (Ore, 1960)

在一個 p 點的簡單圖 G 中， $p \geq 3$ ，如果對於不相連的任意兩點 u 與 v ， $\deg_G(u) + \deg_G(v) \geq p$ 則 G 為哈密爾頓圖。

證明. 假設定理不對。則一定存在一個極大的非哈密爾頓圖 G ，它滿足 G 中不相連的任意兩點之度數和至少為 $|V(G)|$ ，同時，只要加入任一邊 uv ，則 $G + uv$ 為哈密爾頓圖。於是 G 中必存在一個 $u - v$ 路徑，它通過所有的點： $u = v_1, v_2, \dots, v_p = v$ 現在考慮與 u 相連的點集合 $N_G(v_1)$ ；如果 $v_i \in N_G(v_1)$ ，則 v_{i+1} 必然不與 v_p 相連，否則 $(v_1, v_2, \dots, v_{i-1}, v_p, v_{p-1}, \dots, v_i, v_1)$ 為 G 的一個哈密爾頓圖，與假設矛盾。但是，因為當 $\deg(v_1)$ 為 t 時，則 $\deg(v_p) \leq p - t - 1$ ，所以 $\deg_G(v_1) + \deg_G(v_p) = \deg_G(u) + \deg_G(v) < p$ ，這又與假設矛盾，所以定理得證。 ■

推銷員問題 (Traveling Salesman Problem)

在哈密爾頓圖方面，最有名的應用問題非推銷員問題莫屬。它是要找一個最便宜的旅行方式，到過他計畫前往的每一個城市之後，再回到原住處；由於安排旅行行程的不同，所花費的經費自然會有所不同，更何況所使用時間也會不一樣，以下，我們用加權圖的方式來表示推銷員所要訪問的城市，以及城市與城市之間旅行所需要的費用；目標是要找到一個哈密爾頓圈來作為他旅行的過程，而在圖上的加權總和則為研究的對象，越小越好。

由於討論的城市都是要拜訪的城市，所以加權圖的底圖 (Background graph) 可以讓它是一個完全圖；我們同時假設 (i) 如果城市甲與城市乙無路可通，則費用加權為 $+\infty$ ，(ii) 由甲到乙所需費用等於由乙到甲的費用。

這個問題，以下簡稱 TSP，最早在 1934 年時由 Whitney 首先提出來它的重要性在 OR 中 (Operation Research) 不斷地被提出來，直到 1954 年，Dontzig, Fulkerson 及 Johnson 有了突破，它們對於美國的 49 個重要城市提供了一個幾近完美的答案，他們不但用到了圖論的觀念，也用到了線性規劃中整數規劃 (Integer Programming)，最主要的兩個工具是 Branch-and-Bound 及 Cutting Planes。到了 1980 年，Crowder 及 Padborg 有了更重大的突破，他找到了接近最佳答案的 318 個城市，他們一共花了 6 分鐘而已，至於苦力式地把所有可能的路線都比較過一次的方法，就是每秒鐘能做 10 億次，也要花上 10^{639} 年哩！這兩位學者的貢獻令人欽佩，他們除了上述提及的 Branch-and-Bound 之外，又用到了 Facet-defining Inequalities。

在這一節中，主要介紹幾個不錯的演算法，以時間，花費力氣的考慮，它們都是“相當好”的方法。

我們已經知道 TSP 是一個 NP-hard 的問題，所以要找到一個最好 (Optimal) 的答案絕非易事，尤其是在點數很多的時候，更是困難。因此，對於由一個城市到下一個城市應該如何選擇，就要發揮研究者的想像力了。一般而言，從比較可能的幾個選擇找一個城市然後繼續下去是勢在必行，也許不見得會得到答案（最好），但是，希望如此選擇可以達到相當好的效果。這一類在幾個比較可能的情況下繼續選擇一做下去的方法一般稱為是啟發式的演算法 (Heuristic Algorithm)。這種演算法可能求不出最優解，但是，肯定可以在比較短的時間內選到一個“不錯”的答案。以下

就提供三個這種類型的演算法，細節的討論在此省略。

演算法 A : (最近鄰居法 , Nearest Neighbor)

選擇最近但是尚未去過的城市。

演算法 B : (兩倍樹法 , Double Tree)

(1) 先求最小的懸掛樹 (Minimum Spanning Tree, MST)。

(2) 把上述得到的樹擴大一倍 (邊畫兩次)。

(3) 利用圖中的尤拉迴路再加上找捷徑的概念求得哈密爾頓圈 (參考圖 12)。

(即跳過尤拉迴路已經通過的城市。)

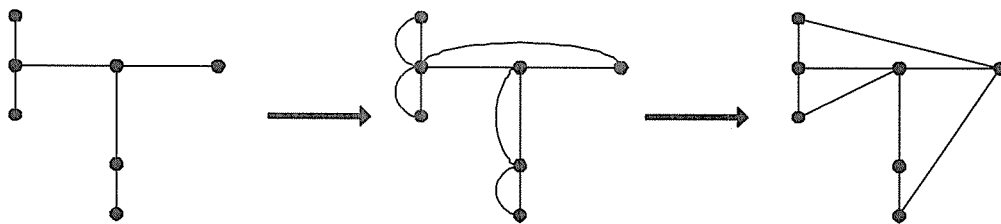


圖 12 : 兩倍樹法

演算法 C : (樹與配對 , Tree and Matching)

(1) 找一個 MST, T 。

(2) 找出由上述 MST 的奇點所導出的子圖 H 。

(3) 在 H 中找到一個完全配對 M 它的加權總和為最小。

(4) 把 M 的邊加到 T 上, 於是得到 T^* 。

(5) 利用 T^* 中的尤拉迴路找到哈密爾頓圖。

TSP 的變化形式有很多, 包括找到最短的哈密爾頓路徑 (通過全部點的路徑), 迴路設計, 行車路線安排, 工作安排 等, 都是同類型的問題, 這也更凸顯這個問題在應用領域上的實用價值。

作業 25. 試求在一個 10 點的完全圖中不同的哈密爾頓圖 (兩圖至少有一邊不一樣)。



pm 5:00
It's over!



pm 5:30
See you tomorrow!