

思考的演算：

跟著電腦學思考，你也可以成為計算思考大師

作者：保羅·科松、彼得·馬克歐文

出版社：八旗文化

出版日期：2018年4月3日



文／白榮銓

十二年國民基本教育課程綱要草案，新增「科技領域」，學習內涵包含「生活科技」與「資訊科技」。資訊科技課程是以「運算思維」(computational thinking, CT)為主軸，學習重點強調「透過電腦科學相關知能的學習，以培養學生的運算思維能力；並藉由資訊科技之設計與實作，增進學生運算思維的應用能力與解決問題的能力」；教學實施則強調「以問題解決或專題製作之方式進行，鼓勵學生進行自主性、探索式的學習，以實踐運算思維的課程理念」。到底何謂運算思維（本書譯為「計算思考」）？包含那些技能？如何應用來解決問題？

本書作者科松（Paul Curzon）與馬克歐文（Peter McOwan），都是倫敦瑪麗皇后大學（Queen Mary University of London）的電腦科學教授，共同創辦了「電腦科學好好玩」（Computer Science for

Fun）網站，致力於提供教師專業發展課程。他們在青少年時代，熱愛閱讀美國數學科普大師加德納（Martin Gardner, 1914-2010）撰寫的娛樂性數學（recreational mathematics）讀物，深受加德納的啟發和影響。本書內容來自作者重新改寫他們在「電腦科學好好玩」網站、雜誌、以及在用來協助程式寫作與資訊融入教學的網站「Teaching London Computing」上所發表的文章，以讀者容易理解的方式，搭配遊戲（game）、謎題（puzzle）、魔術以及電腦科學家在研究的真實難題，解釋何謂計算思考，以及這種問題解決方法中的各種技巧，包括：演算法思考（algorithm thinking）、評估（evaluation）、模式比對（pattern matching）、通化（generalization）、抽象化（abstraction）以及拆解（decomposition）等，幫助讀者了解與學習電腦科學家的思考方式。

計算思考

1980年，美國數學家暨教育家派普特（Seymour Papert, 1928-2016）出書《心智衝擊：兒童、電腦與充滿活力的創意》（*Mindstorms: Children, Computers, and Powerful Ideas*），首次提出「計算思考」（computational thinking）一詞。計算思考是「人類解決複雜問題的多元技能之集合」，適用於電腦科學、人文科學、數學與所有科學領域，計算思考有助於學生了解學科與學科、以及課堂與生活之間的關係。

2007年，改編自真人實事的法國電影《潛水鐘與蝴蝶》（*The Diving Bell and the Butterfly*, 圖1）上映，劇情描述法國時尚雜誌《ELLE》總編輯鮑比（Jean-

Dominique Bauby, 1952-1997），於1995年12月突然中風，陷入昏迷，雖然三星期後甦醒，卻被診斷患有閉鎖症候群（locked-in syndrome），全身癱瘓，完全喪失發音能力，全身僅剩左眼可以自由活動，在接下來的兩年時間內，神奇地完成他生命中唯一的一本書《潛水鐘與蝴蝶》。鮑比在書中記錄自己中風後的日常生活點滴。假設你是他的語言治療師杜蘭（Henriette Durand），請問要如何為他設計出一套與外界溝通，而且能完成寫作的方法。

此時，你需要和他達成共識，找出一套能將「眨眼動作轉為字母」的方法，你的第一想法可能是告訴他：「眨一次眼代表字母A，眨兩次眼代表字母B」，以此類推；接著，手上拿著紙筆的杜蘭，只要數他眨了幾次眼，記錄下相對應的字母，就像網路世界中，不同電腦之間用於資料編碼與解碼的「通訊協定」（protocol）。當我們設計出一系列可以遵循及解決問題的步驟，能正確地溝通病患所想的字母，我們就已在進行像電腦科學家一樣的思考，這就是計算思考的核心，稱為「演算法思考」。透過演算法思考，人們不只能想出單一答案，更能想出一套可以依照一定步驟，甚至更有效解決問題的方案。

鮑比在書中的確有更好的方法，他描述：杜蘭依序大聲地朗讀出字母A、B、C……，當他腦中所想的那個字母被朗讀出來時，他就眨一次眼，接著，杜蘭就會寫出那個字母，然後再重新開始朗讀一個又一個的字母。問題是需要用來溝通的元件，不只是字母，還需要空格、數字、逗號及句號等；另一個問題是：萬一不小心眨了眼怎麼辦。像這樣，檢查一個演算法在理論及實際上，是否都行得通，是計算思考的重要部分之一，稱為「評估」。



圖 1.《潛水鐘與蝴蝶》電影海報（圖片來源：<http://collider.com/exclusive-video-interviews-the-diving-bell-and-the-butterfly/>）

為了與鮑比加速溝通流程，最好是拼一個字的一部分字母，就能正確地猜出這個字，例如杜蘭已經寫下 **a-n-t-e-l**，就能猜出鮑比腦中想的是 **antelope**（羚羊）。因此可以改變規則，讓杜蘭以這種方式猜猜看，這種演算法類似於手機與電腦在「聯想字詞」時所使用的運作方式。當人們在搜尋引擎輸入一個想要搜尋的關鍵詞時，它所做的事情和上面情況類似，這是計算思考的另一個重要部分，稱為「模式比對」。

能夠運用「聯想字詞」這種解決方法，是因為杜蘭和電腦所面臨的問題是類似的，即杜蘭必須一個字母接著一個字母，認出閉鎖症患者想要表達的單字；而電腦必須一個字母接一個字母，拼出使用者想輸入的單字。更理想的情況是，我們已有一個適用於許多類似問題的解決方法，並從一開始就為該演算法創造一個描述（**description**），這樣只要類似情況一出現，就能使用這種演算法，這是計算思考的重要技巧之一，稱為「通化」。

身為法國女性雜誌《ELLE》的總編輯，鮑比對於語言有很深的造詣，他知道這種 **ABC** 演算法，還可以用不同方式來加以改善，有些字母比起其他字母出現的頻率更高，例如 **E** 是最常出現的字母（在英語和法語中皆然），因為鮑比使用的是法語，所以在電影中，他請杜蘭以法文常出現的頻率順序「**E-S-A-R……**」（圖 2）來朗讀字母，就可以使杜蘭加速念到單字內所含的字母。像這樣，使用頻率分析（**frequency analysis**）的方法，就是運用了計算思考中的「模式比對、通化」等技巧。

運用計算思考的技巧，能得到比最初「針對每一個字母，分別眨眼不同次數」更好的溝通方法，對於「寫完整本書要多久的時間、是否為最佳方式」，進行評估之

後，我們是否能設計出更理想的演算法，幫助鮑比完成這本書。我們可以利用「分析式思考」（**analytical thinking**），用簡單的數學來計算，只要將字母的數量，乘上念一個字母所需的時間，如此不但能知道想要完成的工作量，更能知道這是否為最佳方式。像這樣，過濾或忽略不必要的特徵，集中在重要的特徵上（例如字母的數量、念一個字母的時間），是計算思考的重要技巧之一，稱為「抽象化」。

使用頻率分析能得到最佳的溝通方式，但最糟糕的情況依然要念完 **26** 個字母；於是我們還要思考：如何再減少杜蘭念的字母數量。**19** 世紀時，美國各地風行「**20 個問題**」（**Twenty Questions**）遊戲，遊戲規則：選擇一名玩家作為「回答者」（**answerer**），這個人選擇一個「主題」（**subject**）；其他的玩家都是「提問者」（**questioner**），每個人輪流問一個問題，回答者只能用簡單的「是、否」做答。遊戲中，回答者不允許說謊，提問者須在 **20** 個問題內，正確猜中「主題」，猜中者將成為下一輪遊戲的回答者，例如「想像我是一位名人，猜猜我是誰？」提問者與回答者的對答如下：「是女的嗎？不是」、「還活著

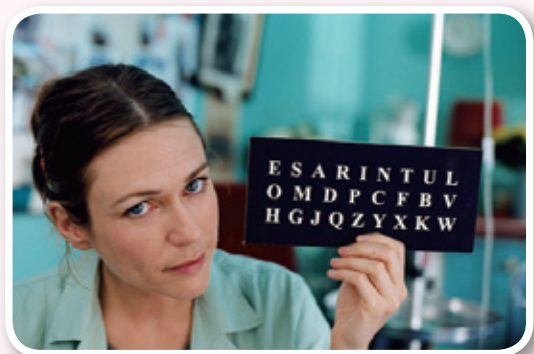


圖 2. 劇中的法文字母出現頻率順序卡（圖片來源：<https://tenyearsago.wordpress.com/2017/12/13/ten-years-ago-the-diving-bell-and-the-butterfly/>）

嗎？不是」、「是電影明星嗎？不是」、「來自英國嗎？是的」、「是作家嗎？是的」、「20世紀的人物嗎？不是」、「19世紀的人物嗎？不是」、「是莎士比亞嗎？是的」。

玩「20個問題」的秘訣：無論答案是什麼，每次提問都要是「能排除一半人選」的問題，即每次都能提出「二分法」的問題。像這樣，將一個複雜的問題，分解成很多的小問題，使問題能更容易解決，是計算思考的重要技巧之一，稱為「拆解」。依照「模式比對與通化」的技巧，猜鮑比腦中的字母，與猜「回答者」腦中的主題，應屬於類似的問題。杜蘭一開始可以提問「這是一個A到M之間的字母嗎？」(圖3)如果答案為「是」，則可以再問「這是一個A到F之間的字母嗎？」如果答案為「否」，則改為問「這是一個N到S之間的字母嗎？」以此類推，這樣一定能在5個問題內，得到鮑比要的字母，我們還可以藉助運算法的聯想字詞，這樣將更能提升解決問題的效率。

猜字母的決策樹 (decision tree)，能夠衍生出一個截然不同的解決方案。如果

我們將圖3的「yes (是)、no (否)」以及「眨眼、不眨眼」，定義成「1、0」，這棵決策樹就能夠定義出字母的「二進位序列」(binary sequence, 表1)。進行溝通時，只要使用此序列，就能找出對應的字

表1 以二進位序列代表字母的對照表

編碼	字母	編碼	字母	編碼	字母
11111	A	10100	J	0100	S
11110	B	10011	K	00111	T
1110	C	10010	L	00110	U
11011	D	1000	M	00101	V
11010	E	01111	N	00100	W
1100	F	01110	O	00011	X
10111	G	0110	P	00010	Y
10110	H	01011	Q	00000	Z
10101	I	01010	R		

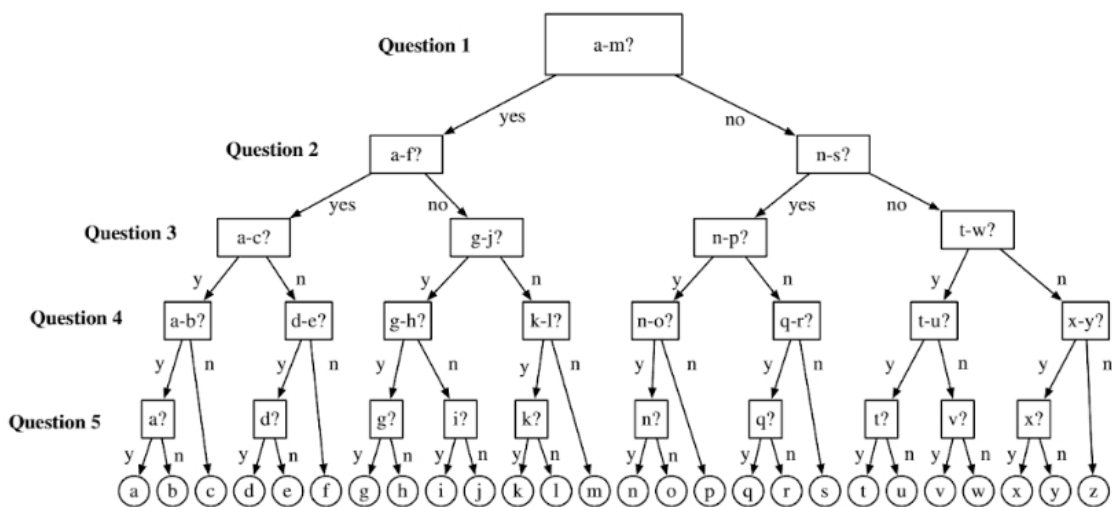


圖3. 決策樹上用來得出每一個字母的密碼

母，例如密碼 1100（眨眼、眨眼、不眨眼、不眨眼），代表 F 字母，亦即利用演算思考，我們最終能得到溝通的編碼(code)。

由上述可知，「演算法思考」是計算思考的核心，能幫助我們開發出解決問題的不同方法及規則；「評估」則能讓我們知道各種演算法是否可行，甚至發展出最佳的解決方法。「模式」讓問題簡化，「模式比對」則能找出類似問題的共同特徵；「通化」能讓我們使用相同的解決方法，去解決類似的問題。「抽象化」讓我們忽略不必要的特徵，將問題具體化。「拆解」幫助我們將一個複雜的問題，分解成許多小問題，使問題更容易被理解及解決。

導遊謎題與騎士之旅

底下以導遊謎題及騎士之旅，做為運用「計算思考」解決問題的實例。假設你是某旅館的導遊，住在該旅館的遊客，希望參加一個包含該城市所有景點的一日遊行程，你有一張標示市內所有景點，及不同景點之間的地鐵路線圖（圖 4）。因為

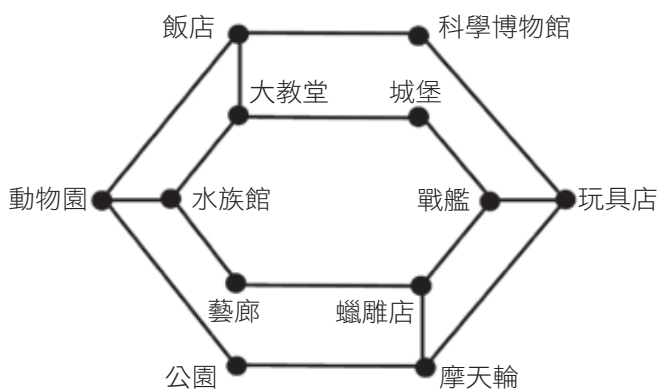


圖 4. 導遊手上的城市導覽地圖（圖片來源：<https://teachinglondoncomputing.files.wordpress.com/2014/12/activity-knightstour.pdf>）

旅客只在這城市待一天，為了節省時間，他們不想經過同一景點兩次，所以你必須設計一條路線，帶著城市旅遊團，從飯店出發，在一天內，乘坐地鐵，遊遍地圖內的所有觀光景點，每一個景點只能經過一次，當天最後必須回到飯店，你能畫出符合旅客需求的導覽路線圖嗎？

為了探究導遊謎題的解決方法，每名學生各發一份：棋盤(圖 5)、騎士(knight)棋子的移動規則(圖 6)、解決方案空白紙。

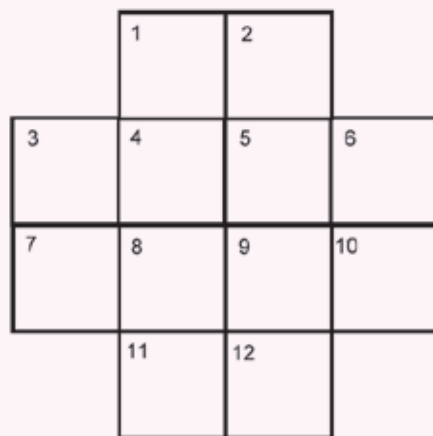


圖 5. 騎士之旅的棋盤（圖片來源：<https://teachinglondoncomputing.files.wordpress.com/2014/12/activity-knightstour.pdf>）

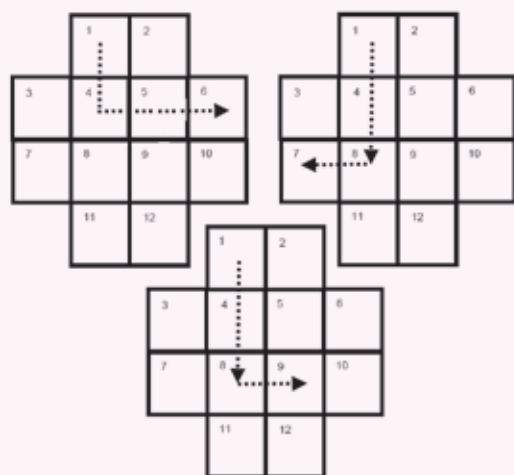


圖 6. 騎士棋子的移動規則（圖片來源：<https://teachinglondoncomputing.files.wordpress.com/2014/12/activity-knightstour.pdf>）

騎士棋子的移動規則：朝一個方向移動 2 格，然後轉 90 度，再走 1 格；或者朝一個方向移動 1 格，然後轉 90 度，再走 2 格，即走 L 形路線。因此，騎士自方格 1 出發，可以分別走至方格 6、7、9。對學生提出問題：依照騎士的移動規則，在騎士之旅的解決方案空白紙上，填寫從方格 1 開始的移動順序，讓騎士經過各個方格一次，最後回到方格 1。

騎士之旅和導遊謎題非常類似，兩個問題在本質上有些相似點：(1) 旅程都從一個指定地點開始，(2) 旅程必須造訪每一地點，(3) 旅程不可經過之前已造訪過的地點，(4) 旅程必須要回到一開始的起點。我們以圖解對「騎士之旅」做進一步的分析，以圓點表示每一個方格，就像地鐵圖上以圓圈表示觀光景點一樣。當騎士依照移動規則，從任何一個圓點，走到另一個圓點時，我們就在兩個圓點之間畫上一條線，這些線就是圖解的邊。從方格 1 分別能夠移到方格 6、9、7，我們就分別在兩者之間畫上一條線（圖 7）；接著同樣依照移動規則，畫出所有從方格 6 延伸出去的所有線、從方格 9 延伸出去的所有線，以及從方格 7 延伸出去的所有線。以此類

推，這種方法稱為圖解的「廣度優先搜尋」（breadth-first search, BFS）演算法，這種演算法，會先將同一層的所有節點，全都拜訪（visit）之後，再拜訪下一層的節點。

另一種選擇是探索從源頭，直到終點的一條路線，例如沿著 1-9-3-11-5-7……，分別在兩者之間畫上一條線，直到終點 1，如果失敗，則回頭嘗試不同路線，這種方法則稱為圖解的「深度優先搜尋」（depth-first search, DFS）演算法，這種演算法總是先拜訪最深的節點（deepest node）。廣度優先搜尋演算法和深度優先搜尋演算法，是兩種不同的「圖解遍歷演算法」（graph traversal algorithms）。

以圖解法繪製騎士的路線圖，圖上有許多交叉的線條，看起來有點混亂（圖 8）。為了不讓任兩條線交叉，重新整理成比較整齊的圖形，繪製出相同對應關係，並以箭頭標示「騎士之旅」的正確路線圖（圖 9），請注意正確路線不只一種。仔細觀察這個版本的圖形，發現它與導遊手上的城市旅遊地圖相同，不同的只是標記在圓圈上的標籤是數字，而不是景點名稱，只要你破解了其中一個問題，另一題也就解出來了（圖 10）。

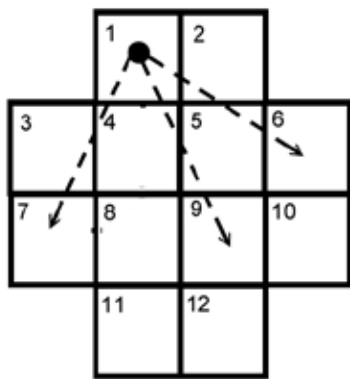


圖 7. 「騎士之旅」在棋盤上的移動可能性

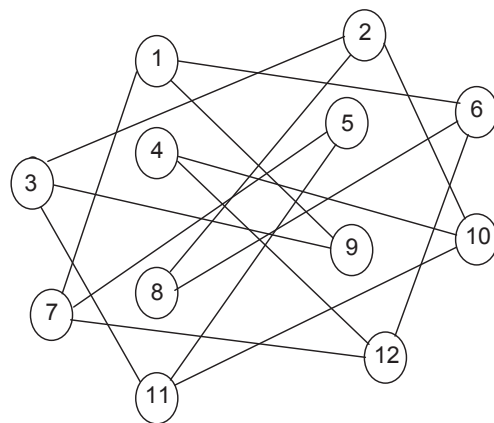


圖 8. 騎士之旅的圖解

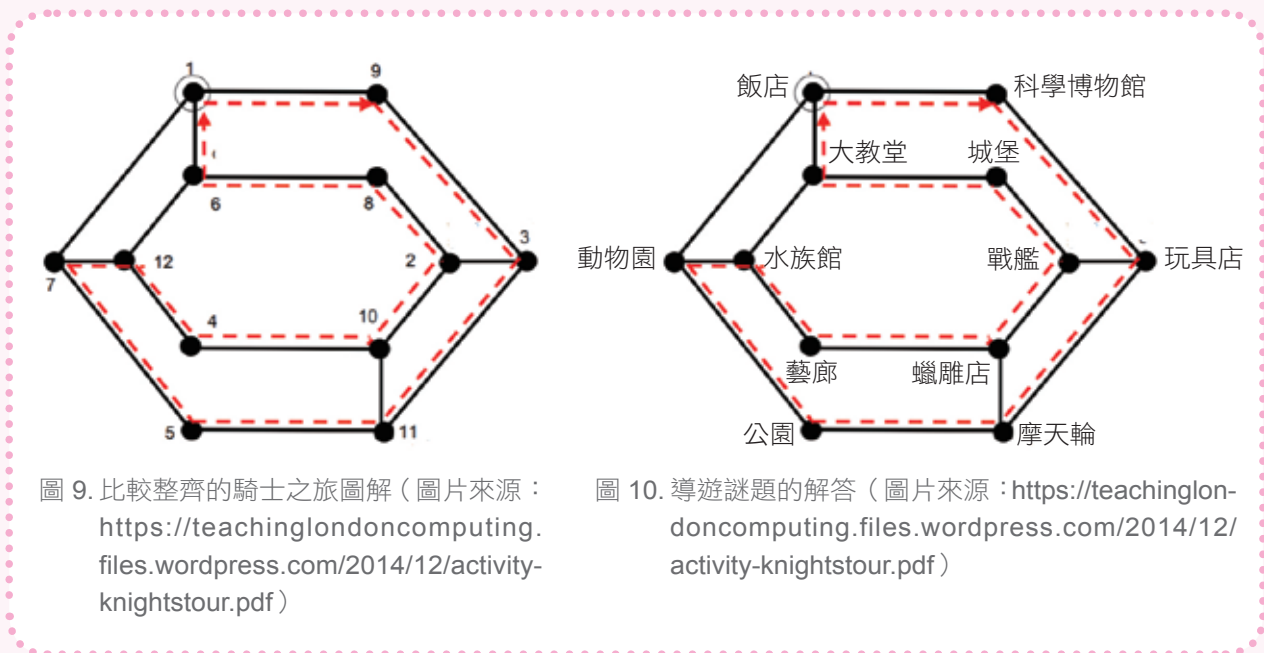


圖 9. 比較整齊的騎士之旅圖解 (圖片來源：<https://teachinglondoncomputing.files.wordpress.com/2014/12/activity-knightstour.pdf>)

圖 10. 導遊謎題的解答 (圖片來源：<https://teachinglondoncomputing.files.wordpress.com/2014/12/activity-knightstour.pdf>)

著名的「柯尼斯堡七橋」(Seven Bridges of Königsberg, 圖 11) 問題, 是另一個可供思考的謎題。問題如下:「18 世紀時, 柯尼斯堡市區橫跨歐洲的普瑞格爾河 (Pregel River) 兩岸, 河中心有座小島, 島上有 7 座古橋, 分別連接河流的兩岸, 以及河流中的兩座小島, 當時城鎮的

居民開始思考: 如何在只通過各座橋一次的前提下, 走遍 7 座古橋?」, 充滿趣味性的生活化題目, 吸引無數市民踴躍嘗試, 卻始終無法挑戰成功。

1735 年, 瑞士數學家暨物理學家尤拉 (Leonhard Euler, 1707-1783), 對於「柯尼斯堡七橋」深感興趣, 展開研究。尤拉



圖 11. 柯尼斯堡七橋問題 (圖片來源：<https://simonkneebone.com/2011/11/29/konigsberg-bridge-puzzle/>)

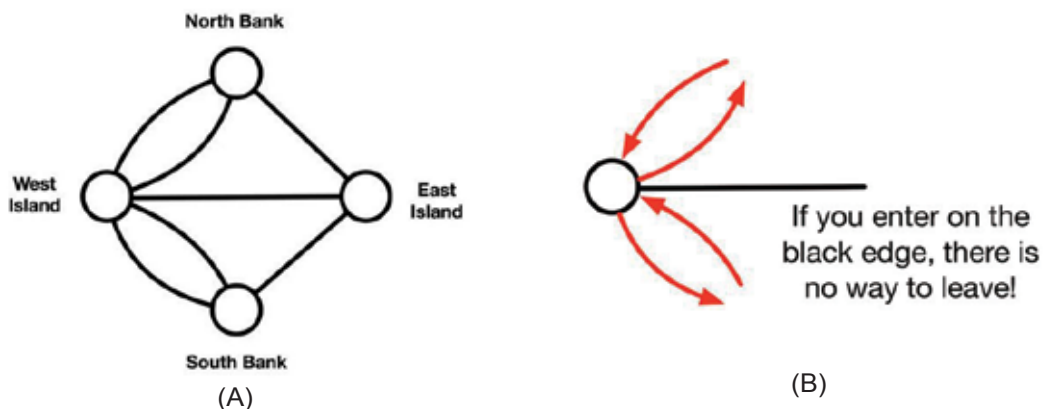


圖 12. 柯尼斯堡七橋問題的圖解說明（圖片來源：<http://www.cs4fn.org/ada/puzzlingdoodlesanswer.php>）

利用圖解，將問題簡化為平面上的圓圈與線條之組合（圖 12A），每一座橋以一條曲線或直線表示，7 座橋所連接的地區，包含北岸（north bank）、南岸（south bank）、東島（east island）、西島（west island），各以圓圈表示，視為一個節點（node）。如果一個節點有偶數條線聚集，則這個節點能夠讓人從一邊進，另一邊出，符合「只經過各座橋一次」的規則；如果一個節點有奇數條線聚集，則不符合「只經過各座橋一次」的規則（圖 12B）。因此，奇數條線聚集的節點必須是 0 或 2 個，若是 2 個，則其中一個節點作為起點，另一個作為終點。因為柯尼斯堡圖解上的 4 個節點，全都是奇數條線聚集，所以「柯尼斯堡七橋」無法讓人在只經過各座橋一次的前提下，走遍 7 座古橋。

由上述可知，導遊手上的城市導覽地圖，將所有景點以及地鐵路線無關的細節，其餘不相關的資訊全都隱藏起來，這就是整座城市的「抽象化」表現。導遊謎題與騎士之旅，都是由指定的起點前往指定的終點，途中經過其他的節點，而且只經過一次，透過「模式比對與通化」，我們

能判定它們是同一類的問題，其實，電腦科學家將這種路線的圖解，稱為「漢密頓循環」（Hamiltonian cycle）。

綜合上述，計算思考不是電腦思考的方式，而是人類讓電腦完成任務時，所必須做的各種思考方法。計算思考是由許多不同的技巧所構成，電腦科學家常將這些豐富有用的技巧，以相互連結的方式，靈活地組合起來解決問題；最終，電腦科學家能以這些技巧為基礎，創造出以機器為基礎的解決方案。當演算法變為程式時，我們的生活、工作和遊戲方式都會發生改變，尤其是在發展「人工智慧」時，電腦科學家已逐漸能寫出讓機器進行計算思考的程式。至於如何運用計算思考，讓數學魔術萬無一失？人類的觀看能力來自於我們能在影像中找到邊界，那麼電腦要如何「觀看」到邊界呢？如何藉助演算法及電子設備，將科學以及數學，轉變為拯救生命的醫療科技？這些都有待您進一步的閱讀與思考！🔗

白榮銓
臺中市居仁國中退休教師