



要利用程式設計做什麼？

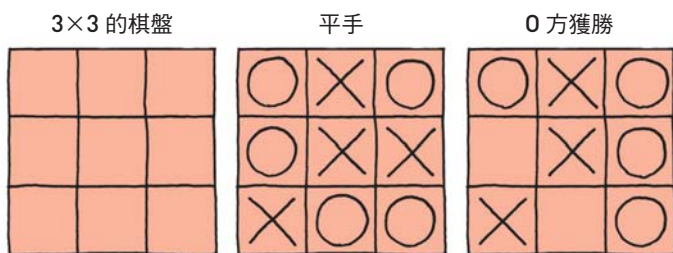
程式設計的必要元素是什麼？

學習程式設計的時候，應該會發現程式語言的文法很無聊。市面上有許多從輸出「Hello World」開始介紹的書，但我都懷疑這些書真的有用嗎？

學校的程式設計課程通常偏重理論，很難讓學生知道這些理論該怎麼使用，所以有些人就會對程式設計失去興趣而不再繼續學習。

反過來說，如果有想要寫的程式，而且還能真的寫出來的話，一定覺得很有成就感。要是這個程式還具有一定的難度，會讓人更有興趣挑戰。

雖然這類主題不多，但在此讓我們一起思考「井字棋」的程式，也就是在 3×3 的棋盤裡，不管是垂直、水平還是斜向，只要連續三個 O 或 X 排成一線就能獲勝的遊戲。



這個井字棋有許多學習程式設計必須知道的元素。例如，下列這些的功能都是必要的元素。

- 輸入 O 或 X 的「接收輸入」功能
- 顯示棋盤的「於螢幕輸出內容」的功能
- 於處理資料過程中，「暫時儲存」資料的功能
- 與電腦對戰所需的「AI (人工智慧)」功能

其實大部分的程式都需要這些功能。不管是哪種程式，都需要輸入與輸出（沒有輸入功能的話，每次的結果都一樣，沒有輸出功能的話，不知道處理是否完成）。



此外，要儲存處理中的資料就必須具備**資料結構**的知識，就算用不到 AI 這麼高深的處理，也需要能思考處理步驟的**演算法**（演算法將於 Part 2 解說）。

接著讓我們透過井字棋的程式，按部就班學習程式設計的基本知識，寫出上述這些功能。

建立 Python 的程式設計環境

本書使用的程式語言是「Python」。這是適合資料分析與統計的語言，最近也很常用於開發機械學習這類人工智慧。除了內建於 Raspberry Pi 這類小型電腦使用的作業系統，也很常用來開發網路程式，算是近來相當受到程式設計師歡迎的程式語言。



本書提供 Ruby、JavaScript 的題目（Part 3）解答（參考 P.10）。有機會請試著利用其他的程式語言撰寫相同的程式。

Python 是 macOS 內建的標準程式設計語言，但 Windows 環境須另行安裝。較簡單的安裝方式就是從官方網站下載 Anaconda（<https://www.anaconda.com>），再依照畫面說明安裝即可。請大家試著安裝看看。

為了簡化建構開發環境的步驟，這次要使用只有網頁瀏覽器也能執行程式的「Google Colaboratory」 (<https://colab.research.google.com/>)。只要有 Google 帳號就能免費使用這個服務，還請大家試用看看。

進入網站後，點選「新增筆記本」，就會顯示下列的畫面。



在這個輸入欄位輸入**程式碼**，再點選輸入欄位左側的「▶」按鈕，就能**執行**輸入的程式。若想增加輸入欄位可點選「+ 程式碼」。

若想儲存執行過的內容可點選「檔案」，再點選「儲存」與替檔案命名，之後開啟這個檔案就能執行同樣的程式。

練習題

星座的判斷（題目）

許多人都會拿星座占卜，也有許多人會在看完每天早上的星座占卜節目之後才出門吧？只要知道自己的生日，就能知道自己的星座。

不過，要記住其他人的星座可就沒那麼容易，而且星座的日期是從每月的幾號到幾號，每本書講的都不一樣，所以，我們不如寫一個只要輸入生日的月與日，就自動顯示星座的程式。12 個星座各自的日期可參考下列表格的內容。

星座	生日	星期	生日
水瓶座	1月20日～2月18日	獅子座	7月23日～ 8月22日
雙魚座	2月19日～3月20日	處女座	8月23日～ 9月22日
牡羊座	3月21日～4月19日	天秤座	9月23日～10月23日
金牛座	4月20日～5月20日	摩羯座	10月24日～11月22日
雙子座	5月21日～6月21日	射手座	11月23日～12月21日
巨蟹座	6月22日～7月22日	白羊座	12月22日～ 1月19日

提示

也可利用多組條件分歧判斷生日期間，但只要先將星座的資訊放進列表，就能快速判斷星座（解答請參考 Part 2 最後一頁）。



「樹狀構造」與演算法

搜尋的重點

雖然前面的 Part 1 撰寫了井字棋的程式，但如果要讓電腦「懂得思考」，從多個方案選出最佳方案的話，就必須撰寫更多處理。其實我們的日常生活也是一連串的「選擇」。

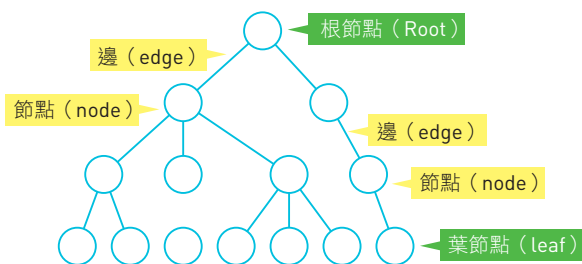
- 早上要幾點起床？
- 早餐要吃什麼？
- 要走哪條路去公司？
- 要從哪項工作先做？…

有些當然是例行公事，有些則是在經過多次選擇之後得出的最佳結論，但一開始應該都是從多個方案之中，挑選某個方案才對。

那麼，有哪些方法可選出最佳方案呢？如果能先檢視所有方案的話，當然就能從中挑出最佳方案。

不過，就實務而言，方案通常會多得無法一一檢視，通常都是先進行篩選，再從中找出最佳方案。

解謎也是一樣，最佳方案當然是能全部瀏覽一遍，否則就需要利用一點巧思縮減範圍，再從中找出最佳方案。最常見的方法就是下圖這種樹狀構造。



Point 樹狀構造很像電腦管理檔案的資料夾，不過卻是上下顛倒的樹，頂點的「根節點」則被視為起點。主要的目的是從這個起點出發，試著找出走向哪邊的「樹枝」才能得到最佳結果。





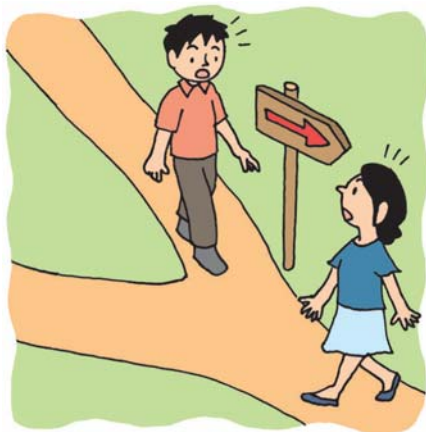
讓搜尋變得更快速

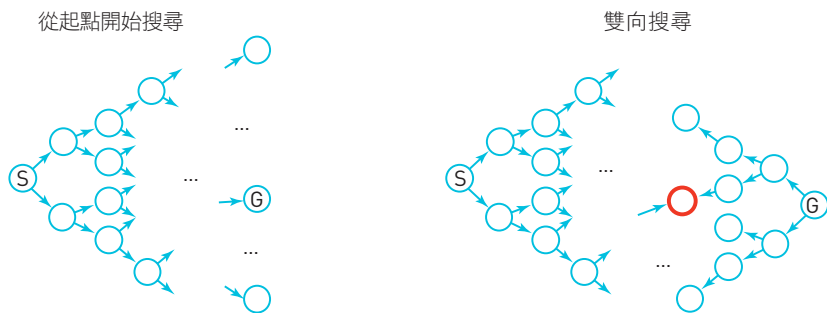
反向思考的「雙向搜尋」

若想找到其他的搜尋方法，最常想到的就是「反向思考」的搜尋方式。讓我們從終點反推這次的迷宮吧！因為只要是能從起點走到終點的迷宮，當然也能從終點走到起點。

不過，若只是從終點開始搜尋，搜尋所需的時間跟從起點開始搜尋其實差不多，因為若只是將起點與終點換個位置，搜尋到的路徑數量會一樣多，搜尋所需的時間也差不多。

讓我們試著做**雙向搜尋**吧！
也就是同時從起點與終點開始搜尋，當雙方於某處相會時，這條路徑就是最短路徑。



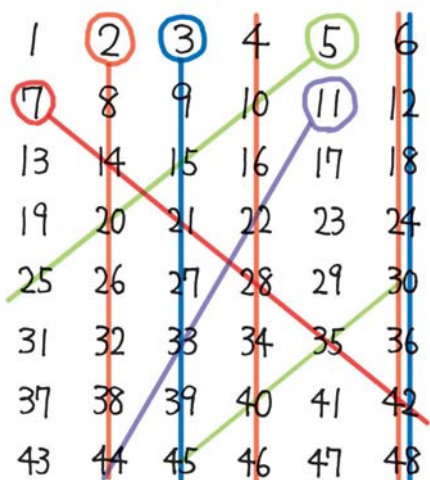


以上圖而言，雙向搜尋的路徑似乎比單向搜尋來得少，稍微計算一下也會發現，雙向搜尋的搜尋量比較少。假設所有的分歧點都有兩條路可走，而這種分歧點總共有 12 處的話，單向搜尋的搜尋量就會是 2 的 12 次方，也就是 4096 種。

假設改以雙向搜尋的方式搜尋，單邊就只需要搜尋 6 個分歧點，所以搜尋量就是 2 的 6 次方，由於是雙向搜尋，所以總搜尋量只會是 $64 \times 2 = 128$ 。換言之，兩種搜尋方式的搜尋量約有 30 倍的差異，而且分歧點越多，這個差異就會越明顯。

由於雙向搜尋無法使用深度優先搜尋，所以會以寬度優先搜尋的方式進行，而且也不會只是從佇列的開頭取得位置，而是讓兩個方向的搜尋深度遞增，藉此找到最短距離。

接下來會將從起點開始搜尋的過程稱為 fw（forward 的簡寫），並將從終點開始搜尋的過程 bw（backward 的簡寫）。



問題 找出小於等於 10 萬的所有質數。

思考邏輯

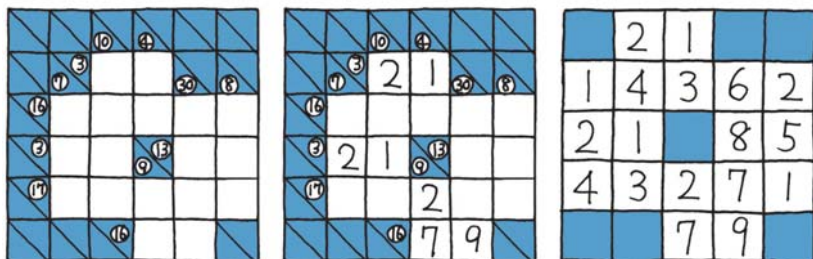
先將要尋找的數字放入列表，再從列表之中較小的數字開始除，看看能否整除，再將除不盡的數字留在列表裡，就能打造出只剩質數的列表。

提示

若要让除不盡的數字留在列表裡，可使用列表推導式。

Q16 以乘法打造的數謎

「數獨」是一種只要一枝筆就能進行的紙上遊戲，也受到許多人的喜愛，但其實還有一種數學紙上遊戲，那就是所謂的「數謎」。這項遊戲會將特定的數字分解成 1~9 的數字，之後在空格裡填入這些數字，當這些數字加總之後，會與直排或橫列裡的特定數字一致。要注意的是，在填入數字時，不能使用相同的數字。



(節錄 例題 自 <https://www.nikoli.co.jp/ja/puzzles/kakuro/> 解到一半的情況 答案
「數獨」、「KAKUKO」是株式會社 NIKORI 的註冊商標。

這次要思考的是「乘法版」的數謎。比方說，將 16 拆成 3 格時，只有 $1 \times 2 \times 8$ 這種答案 ($2 \times 2 \times 4$ 則是因為使用了相同的數字所以不行，而 $1 \times 1 \times 16$ 則是因為使用了 1~9 以外的數字所以不行)。不過，若將 18 拆成 3 格，就能得到 $1 \times 2 \times 9$ 以及 $1 \times 3 \times 6$ 這兩種答案。

可拆成三格、三種數字的模式共有下列三種。

分解前	模式 1	模式 2	模式 3
24	$1 \times 3 \times 8$	$1 \times 4 \times 6$	$2 \times 3 \times 4$
48	$1 \times 6 \times 8$	$2 \times 3 \times 8$	$2 \times 4 \times 6$
72	$1 \times 8 \times 9$	$2 \times 4 \times 9$	$3 \times 4 \times 6$

問題 在 $1 \sim 362880$ ($1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9$) 的範圍之內，能拆成 3 種 6 格組合的數字共有幾個？

思考邏輯 雖然可從頭到尾算一次，但這種算法太浪費時間，所以不妨換個角度，從拆解之後的模式回推拆解之前的數字有幾個，就能大幅減少計算量。

提示 拆解數字的時候，只會用到 $1 \sim 9$ 這幾個數字，而且數字不能重複。