一個燈也不留

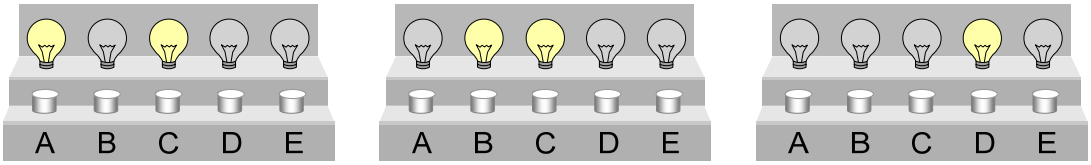
許介彥
大葉大學電機工程學系
chsu@mail.dyu.edu.tw

你知道全台灣當兵當最久的男生是誰嗎？這是一個老笑話了，答案是「阿榮」，因為他出現在一個與軍中生活有關的電視廣告中，而該廣告一播就播了十幾年。隨著役期的逐漸縮短，年輕一輩大概越來越難體會過去的人所津津樂道的那種軍中生活了；到軍中服役對過去在台灣成長的男生來說真是人生中一段相當奇特的經歷，過程儘管辛苦，之後回想起來卻似有數不完的趣事。

如果你問一位曾經服過兵役的男生他當兵時每天最想聽到的一句話是什麼？答案很可能是「熄燈！」，因為不論白天的出操多麼累人，只要晚上就寢時間一到，阿兵哥們在寢室裡躺平後，隨著班長的一聲令下，所有電燈開關同時被扳下，寢室裡頓時漆黑一片，白天的種種辛苦也就被拋到了腦後，緊張的軍中生活於是得以獲得暫時的抒解。

一般的電燈開關有 ON 與 OFF 兩個狀態，分別用來控制電燈的亮與暗；一個開關有可能同時控制好幾盞燈，因此可以同時切換好幾盞燈的狀態。如果某個電路出了問題使得一個開關所切換的電燈的狀態有可能不一致（即不是全亮或全暗），那麼問題就大了——但也變好玩了。

我們來玩個遊戲吧。假設在你面前有五顆燈泡排成一排，每顆燈泡下方都有一個按鈕（如圖 1(a)所示），每當某個按鈕被壓一下，位於這個按鈕正上方的燈泡的狀態將被切換（由亮變暗或由暗變亮），而且與這顆燈泡相鄰的燈泡的狀態也將被切換。例如圖 1(a)中，從左邊算起的第一與第三顆燈泡是亮的，如果此時按鈕 A 被壓一下，這五顆燈泡的狀態將變成如圖 1(b)；如果接著按鈕 C 被壓一下，五顆燈泡的狀態將如圖 1(c)。



(a) (b) (c)

圖 1

我們的問題是：如果一開始這五顆燈泡全是亮的，要如何按壓這些按鈕才能使得燈泡全部熄滅呢？或者更具一般性地說，對任意一種燈泡的初始狀態（總共

有 $2^5 = 32$ 種初始狀態），要如何按壓按鈕才能使燈泡全部熄滅呢？

如果你的瀏覽器可執行 Java 程式的話，不妨用這個小程式[1]實際體驗一下這個遊戲；用滑鼠在畫面中的任一小格內部點一下即可模擬將按鈕壓一下的動作。請多玩幾次，看看能不能找出有系統地解決這個問題的方法。

五顆燈泡的情形其實還蠻簡單的，你也許不用花兩分鐘就找到竅門了，不過如果燈泡數增多的話就沒那麼容易了，不信的話請試試 10 顆燈泡的情形[2]。

「線性代數」(Linear Algebra) 是大學裡相當重要的一門課，許多科系都將其列為必修，它所探討的主題之一是聯立線性方程式的求解，這部分的許多觀念和技巧其實在中學數學裡就提過了。如果我告訴你，我們剛才的遊戲可以用聯立方程式來解決，你會不會覺得有點意外呢？

既然要借助數學方程式來解決我們的問題，第一步當然是要想辦法將遊戲中所涉及的資訊（如燈泡的狀態及按鈕的動作等）盡量用數學的方式來表示；由於每顆燈泡的狀態非明即暗，讓我們用數字 1 和 0 分別代表亮和暗，那麼這五顆燈泡在任何時刻的狀態都可以用一個長度為 5(也就是有五個分量)的向量來表示，例如圖 1(b)的五顆燈泡的狀態就對應到這個向量：

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$，也就是 $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \end{bmatrix}^T$

當某顆燈泡的狀態須被切換，它的狀態將由 1 變成 0 或由 0 變成 1；從數學的角度來說，切換的動作相當於將這顆燈泡原來的狀態加 1，然後再除以 2 取餘數，理由如下：

原狀態		新狀態
0	+ 1 = 1，除以 2 的餘數為	1
1	+ 1 = 2，除以 2 的餘數為	0

我們的遊戲中，當按鈕 A 被壓一下將導致最左邊兩顆燈泡的狀態被切換，因此在數學上相當於將代表原來五顆燈泡狀態的向量與下面這個向量相加：

[1] http://www.dyu.edu.tw/~chsu/Java/02lights/p1/p1.htm
[2] http://www.dyu.edu.tw/~chsu/Java/02lights/p2/p2.htm

$$\mathbf{a}=\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}$$

然後將所得向量的每個分量除以 2 取餘數。舉例來說，圖 1(a)中的五顆燈泡的狀態為

$$\mathbf{s}=\begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix}$$

如果此時壓一下按鈕 A，結果將是

$$\mathbf{s}+\mathbf{a}=\begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix}+\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}2\\1\\1\\0\\0\end{bmatrix}\xrightarrow[\text{除以 2 取餘數}]{\text{每個分量}}\begin{bmatrix}0\\1\\1\\0\\0\end{bmatrix}$$

這正是圖 1(b)的五顆燈泡的狀態。因此我們僅透過純粹的數學運算就能完全預測燈泡的狀態因按鈕的按壓而起的變化。

為了敘述簡單起見，本文以下假設所涉及的加法都隱含了「將相加的結果除以 2 取餘數」的動作，因此 1 + 1 = 0，1 + 1 + 1 = 1 等；亦即所有的偶數都相當於 0，所有的奇數都相當於 1。數學家將這類須將運算結果除以某數取餘數的算術稱作「模算術」(modular arithmetic)。

如同按鈕 A，其他四個按鈕也各依其所切換的燈泡分別對應到一個向量：

$$\mathbf{b}=\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix},\quad \mathbf{c}=\begin{bmatrix}0\\1\\1\\1\\0\end{bmatrix},\quad \mathbf{d}=\begin{bmatrix}0\\0\\1\\1\\1\end{bmatrix},\quad \mathbf{e}=\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}$$

對五顆燈泡的任意一種初始狀態 **s**，如果我們依序按壓按鈕 D, B, D, A, B, D，所得的結果對應到的向量將是

$$\mathbf{s}+\mathbf{d}+\mathbf{b}+\mathbf{d}+\mathbf{a}+\mathbf{b}+\mathbf{d}\ =\mathbf{s}+\mathbf{a}+2\mathbf{b}+3\mathbf{d}$$

$$= \mathbf{s} + \mathbf{a} + \mathbf{d}$$

上式等號右邊最後一步的簡化是怎麼來的呢？別忘了，在我們的加法裡，所有的偶數相當於 0，所有的奇數相當於 1，因此 2 相當於 0，3 相當於 1；這其實很容易用日常生活中的經驗來理解，因為將電燈開關扳動一次、三次、五次、……的效果都相當於扳動一次，而將開關扳動零次、兩次、四次、……的效果都相當於沒有扳動。

因此我們發現按壓 D, B, D, A, B, D 六個按鈕的效果等同於只按 A 和 D 兩個按鈕，而且由於加法滿足交換律，先按 A 再按 D 與先按 D 再按 A 就結果而言並無差別。更進一步來說，對五顆燈泡的任意一種初始狀態 **s** 而言，如果真有某一種按鈕順序可讓燈泡全部熄滅，既然過程中按鈕的順序對結果而言沒有差別，我們可以先將 A 全部集中按完之後才按 B，將 B 全部集中按完之後才按 C，……，又由於對同一個按鈕連續按偶數次相當於沒有按，按奇數次相當於按一次，因此每個按鈕其實最多只須被按一次即可（而且和按鈕的順序無關）。

我們剩下的問題是：對任意一種初始狀態 **s**，要讓燈泡全部熄滅，五個按鈕中的哪幾個按鈕須各被按一次？如果我們將燈泡全部熄滅的狀態記作向量 **0**（由五個 0 組成），那麼我們相當於希望找到常數 $x_1, x_2, \ldots, x_5$ 使能滿足下式：

$$\mathbf{s} + x_1\mathbf{a} + x_2\mathbf{b} + \cdots + x_5\mathbf{e} = \mathbf{0}$$

其中的每個 x_i 不是 0 就是 1；上式經過移項得

$$x_1\mathbf{a} + x_2\mathbf{b} + \cdots + x_5\mathbf{e} = -\mathbf{s} = \mathbf{s} \quad \text{（因為 } -1 \text{ 是奇數）}$$

這已經具備了聯立方程組標準的形式。

舉例來說，如果一開始的燈泡狀態是五顆全亮，那麼上式就是

$$x_1\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix} + x_2\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix} + x_3\begin{bmatrix}0\\1\\1\\1\\0\end{bmatrix} + x_4\begin{bmatrix}0\\0\\1\\1\\1\end{bmatrix} + x_5\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} = \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}$$

也就是

$$\begin{bmatrix}1&1&0&0&0\\1&1&1&0&0\\0&1&1&1&0\\0&0&1&1&1\\0&0&0&1&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix} = \begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}$$

接下來就是聯立方程式的求解了；這個工作有一個很有名的做法叫作「高斯消去法」(Gauss-Jordan elimination)，其第一步是將上式等號左邊的5×5係數矩陣與右邊的向量合併成一個「增廣矩陣」(augmented matrix)：

$$\left[\begin{array}{ccccc|c} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right]$$

接著透過一系列的列運算（例如將某兩列交換、將某一列加到另一列等），上面的增廣矩陣可被簡化成

$$\left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

我們由結果的最下面一列全為 0 可知方程組有不只一組解（因為 x_5 是 free variable)；將 x_5 分別設為 0 和 1 可解得以下兩組解：

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad 或 \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

因此如果五顆燈泡一開始時全亮，我們只須將按鈕 A 和 D 各按一次或將按鈕 B 和 E 各按一次就能讓所有燈泡熄滅；沒錯吧？

請注意上述高斯消去法過程中所涉及的算術運算都要用前面提到的模算術來作，因此與一般的化簡過程稍有不同。在電腦這麼發達的時代，解聯立方程式的工作其實有許多數學軟體可以代勞，其中一個很有名的軟體是 Matlab，它擅長處理跟矩陣相關的計算。如果你熟悉 Matlab 的話可下載 prog1.m[3] 和 rrefmod2.m[4] 這兩個檔案，將它們儲存在一個能讓 Matlab 找到的資料夾內，然後在 Matlab 的命令視窗內鍵入“prog1”即可看到上述化簡後的結果；你還可透過修改 prog1.m 來求得初始狀態不是五顆燈泡全亮時的解（修改向量 s 即可），或

[3] **http://www.dyu.edu.tw/~chsu/Java/02lights/prog1.m**
[4] **http://www.dyu.edu.tw/~chsu/Java/02lights/rrefmod2.m**

讓程式能處理燈泡數比五顆還多的情形。

前面的5×5係數矩陣經化簡後最下面一列全為 0，這表示對任意初始狀態 s，我們的聯立方程組其實未必有解。舉例來說，當燈泡的初始狀態如圖 1(a)時，增廣矩陣為

$$\left[\begin{array}{ccccc|c} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right]$$

經過列運算化簡後的結果為

$$\left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

最下面一列所對應的方程式為

$$0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 = 1$$

這顯然無解；因此如果遊戲的初始狀態是如圖 1(a)，我們透過分析即知要讓這五顆燈泡全部熄滅是不可能的，不論再怎麼努力都註定勞而無功。

我們還可以將遊戲稍作變形，從一維變成二維，例如將 25 個格子（燈泡）排成一個5×5的方陣，一開始時有些是亮的，有些是暗的，例如：

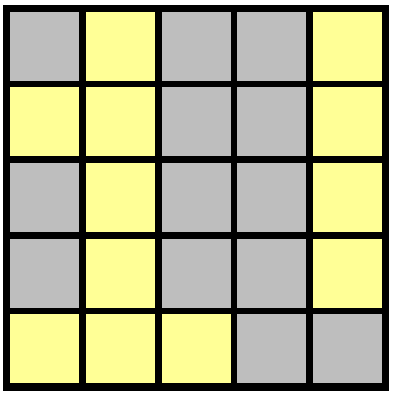


圖 2

當某個格子被壓一下時，我們假設除了該格的狀態將被切換外，與該格左右相鄰及上下相鄰的格子也將一起被切換。我們的目標還是一樣，希望透過一系列按鈕

的動作能讓所有的燈泡熄滅。你不妨用這個 Java 程式[5]實際玩玩看。

這個新遊戲看起來似乎比一維的情形難了不少，不過用數學的方式處理起來並沒有很大的差別。我們現在用一個長度為 25 的向量來表示這 25 個格子的狀態，格子的位置與向量的分量之間的對應關係如下圖所示：

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

圖 3

例如圖 2 的狀態就對應到這個向量：

$$[0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0]^T$$

如同一維的情形，每個按鈕現在也分別對應到一個向量，例如圖 3 中的 1 號按鈕所對應的向量是

$$[1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

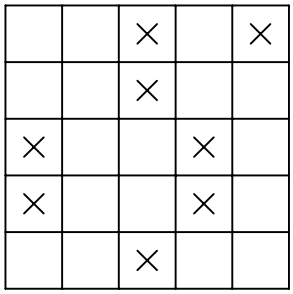
（因為 1 號按鈕可切換編號 1, 2, 6 等三個格子，因此向量中這三個位置是 1，其他為 0）。將某個按鈕壓一下的效果還是相當於將代表當時燈泡狀態的向量與該按鈕所對應的向量相加。對任意初始狀態 s，我們也都可以仿照前面的做法，列出聯立方程式，寫出增廣矩陣（大小為 25×26），然後利用高斯消去法來求解；如同前面的情形，每個格子最多也只須被按一次，而且和按鈕的順序無關。

舉例來說，如果我們以圖 2 做為初始狀態來求解，總共可得四組解，其中一組為

$$[0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0]^T$$

這意味著只要我們將下圖中打叉的格子各按一次就能讓圖 2 中的燈泡全部熄滅：

[5] **http://www.dyu.edu.tw/~chsu/Java/02lights/p3/p3.htm**



你有興趣的話不妨用紙筆驗證一下是否真是如此。

如果你下載 prog2.m[6]和 rrefmod2.m[7]這兩個檔案，然後在 Matlab 的命令視窗內鍵入“prog2”，就可以看到以圖 2 做為初始狀態的解；同樣地，只要修改程式中的向量 s 就能求得其他初始狀態時的解。由於此時的 25×25 係數矩陣經化簡後最下面兩列全為 0，因此也有某些初始狀態將導致方程組無解；不過本文提供的三個 Java 程式所自動產生的各種初始狀態都是有解的。

只要留心觀察，不難發現生活中處處有數學；透過數學的眼光觀看世界不僅能讓我們認清許多事物的本質，也讓我們的生活更加豐富。

[6] **http://www.dyu.edu.tw/~chsu/Java/02lights/prog2.m**
[7] 同 [4]