



Data Analysis

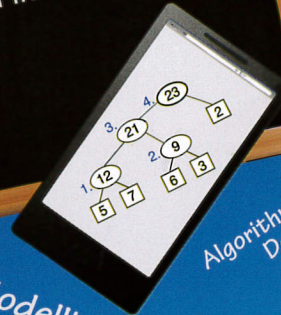
# 2017

# Bebras

## 國際運算思維挑戰賽

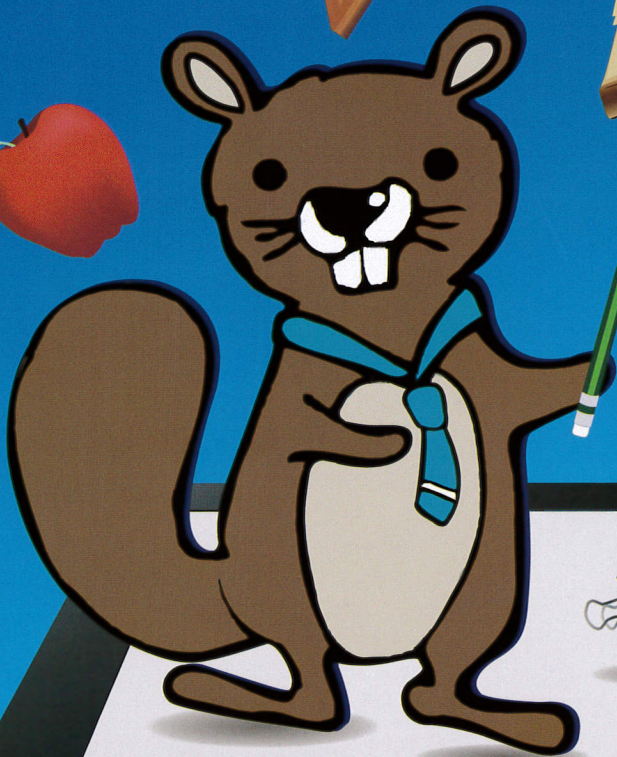
International Challenge on Informatics and Computational Thinking

### 題目解析



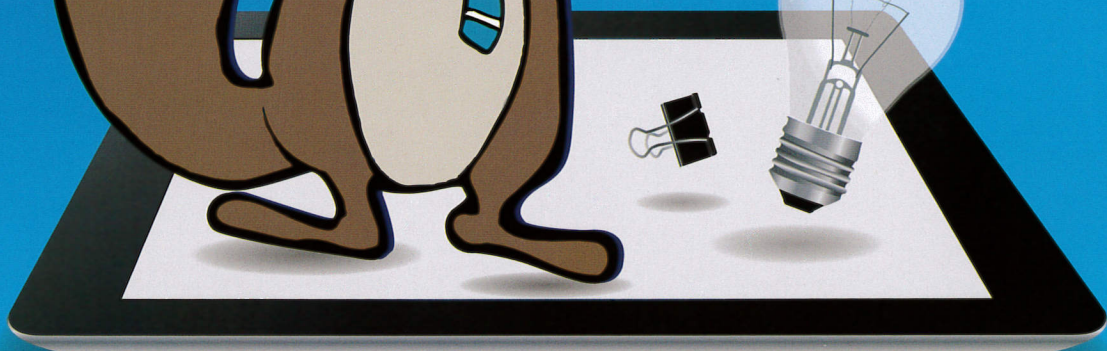
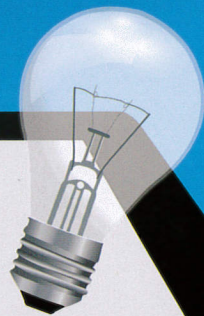
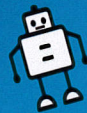
Modelling

Algorithm Design



Abstraction

Decomposition



# 目錄

Benjamin - 國小五、六年級

Cadet - 國中七、八年級

Junior - 國中九年級、高中一年級

Senior - 高中二、三年級

|              | Benjamin | Cadet | Junior | Senior |    |
|--------------|----------|-------|--------|--------|----|
| 1. 括號        | ●        |       |        | ●      | 7  |
| 2. 柵門訊息      | ●        |       |        |        | 9  |
| 3. 向左向右編碼    |          | ●     |        |        | 11 |
| 4. 通勤上班      |          |       | ●      |        | 13 |
| 5. 串聯        |          |       | ●      |        | 15 |
| 6. 狗狗互換      | ●        |       |        |        | 17 |
| 7. 飲料小舖      |          | ●     |        | ●      | 19 |
| 8. 紅燈藍燈      |          | ●     |        |        | 21 |
| 9. 海狸編碼      | ●        |       |        |        | 23 |
| 10. 午餐吃什麼？   |          |       | ●      |        | 25 |
| 11. 水壩地道（簡易） | ●        |       |        |        | 27 |
| 12. 水壩地道（進階） |          | ●     |        | ●      | 29 |
| 13. 卡牌密碼     |          |       | ●      |        | 31 |
| 14. 入侵偵測     |          |       | ●      |        | 33 |
| 15. 不斷線的網路   |          | ●     |        |        | 35 |
| 16. 飛輪趣      |          |       |        | ●      | 37 |
| 17. 新聞編輯     | ●        |       |        |        | 39 |
| 18. 棒棒糖和牙刷   | ●        |       |        |        | 41 |
| 19. 購買專輯     |          | ●     |        |        | 43 |
| 20. 書籍分享俱樂部  |          | ●     |        | ●      | 45 |
| 21. 機器人行走路徑  |          |       |        | ●      | 47 |
| 22. 文字編輯 APP |          |       | ●      |        | 49 |

|              | Benjamin | Cadet | Junior | Senior |     |
|--------------|----------|-------|--------|--------|-----|
| 23. 還書整理     | ●        |       |        |        | 51  |
| 24. 迷路的小海狸   |          | ●     |        | ●      | 53  |
| 25. 熔合金砂     |          |       |        | ●      | 55  |
| 26. 文字加密遊戲   |          |       |        | ●      | 57  |
| 27. 網球賽      |          |       | ●      |        | 59  |
| 28. 國旗       |          |       | ●      |        | 61  |
| 29. 完美洗牌     |          |       | ●      |        | 63  |
| 30. 帽子與腳踏車   |          |       | ●      |        | 65  |
| 31. 降低鐵路營運成本 |          |       |        | ●      | 67  |
| 32. 海狸飯店     |          | ●     |        | ●      | 69  |
| 33. 滾球裝置     |          | ●     | ●      |        | 71  |
| 34. 雪克機器     |          |       | ●      |        | 73  |
| 35. 數字辨認系統   |          |       | ●      |        | 75  |
| 36. 舞者       |          |       |        | ●      | 77  |
| 37. 走出迷宮     |          | ●     |        |        | 79  |
| 38. 環狀城市     |          | ●     |        |        | 81  |
| 39. 太多1了!    |          |       |        | ●      | 83  |
| 40. 拆除障礙牆    | ●        |       |        |        | 85  |
| 41. 棋子回家     | ●        |       |        |        | 89  |
| 42. 五枝牙籤     | ●        |       |        |        | 91  |
| 43. 麵包店      | ●        |       |        |        | 93  |
| 44. 安眠魔藥     |          | ●     |        | ●      | 95  |
| 45. 蘋果裝箱     |          | ●     |        | ●      | 97  |
| 46. 玩具收納     |          |       | ●      |        | 99  |
| 47. 貼壁紙      |          | ●     | ●      |        | 101 |

| Benjamin 組 |      |           |
|------------|------|-----------|
| 易          | 中    | 難         |
| 狗狗互換       | 海狸編碼 | 還書整理      |
| 棒棒糖和牙刷     | 新聞編輯 | 括號        |
| 棋子回家       | 五枝牙籤 | 拆除障礙牆     |
| 麵包店        | 柵門訊息 | 水壩地道 (簡易) |

| Cadet 組 |         |           |
|---------|---------|-----------|
| 易       | 中       | 難         |
| 向左向右編碼  | 蘋果裝箱    | 飲料小舖      |
| 紅燈藍燈    | 貼壁紙     | 海狸飯店      |
| 環狀城市    | 滾球裝置    | 迷路的小海狸    |
| 走出迷宮    | 書籍分享俱樂部 | 安眠魔藥      |
| 購買專輯    | 不斷線的網路  | 水壩地道 (進階) |

| Junior 組 |      |        |
|----------|------|--------|
| 易        | 中    | 難      |
| 國旗       | 玩具收納 | 午餐吃什麼? |
| 文字編輯 APP | 滾球裝置 | 完美洗牌   |
| 帽子與腳踏車   | 貼壁紙  | 雪克機器   |
| 卡牌密碼     | 入侵偵測 | 網球賽    |
| 通勤上班     | 串聯   | 數字辨認系統 |

| Senior 組 |          |           |
|----------|----------|-----------|
| 易        | 中        | 難         |
| 舞者       | 飲料小舖     | 機器人行走路徑   |
| 蘋果裝箱     | 文字加密遊戲   | 迷路的小海狸    |
| 括號       | 書籍分享俱樂部  | 海狸飯店      |
| 太多1了!    | 安眠魔藥     | 熔合金砂      |
| 飛輪趣      | 降低鐵路營運成本 | 水壩地道 (進階) |

國際運算思維挑戰賽 (The International Challenge on Informatics and Computational Thinking) 幫助了解 7 至 18 歲 (一年級至十二年級) 學生的運算思維 (computational thinking) 能力

本挑戰賽自 2004 年開始每年於 11 月中的國際 Bebras 週 (World-Wide Bebras Week) 舉行，我國自 2012 年開始加入，與全球同步舉辦。其旨在激起學生對於資訊科學之興趣，同時了解學生是否具備學習資訊科學之性向。挑戰賽利用淺顯易懂的方式呈現題目 (tasks)，各題皆為情境式任務，讓學習者利用自己既有的知識進行完成任務。



# 國際運算思維挑戰賽

## 測驗目標

### ● 激發學生對資訊科學之學習興趣

本挑戰賽之目的除了瞭解學生是否具備學習資訊科學之性向，更希望藉由情境式的任務，向學生介紹資訊科學或資訊教育的基本概念，激發他們的學習興趣；讓學生瞭解生活中隨處可見資訊科學概念之運用。而解謎推理的題目敘述方式，更可以訓練學生思考，並增進學生高層次思考的能力。

### ● 提升學生利用資訊方法解決問題能力

國際運算思維挑戰賽題目包含家庭生活團體合

作、工作情境等。題組內容多樣化，透過題目讓學生了解生活中的許多問題都能透過資訊科學之概念解決。

### ● 降低學生對資訊科學之恐懼

國際運算思維挑戰賽將抽象的資訊科學題目具體化，以日常生活中會碰到的情境呈現，使未曾受過資訊科學教育的學生亦能利用邏輯、歸納、推理、運算等能力解題。另一方面，題組內容有趣且生動，有助於降低學生對資訊科學的懼怕感。

## 參與對象

國際運算思維挑戰賽每年於 11 月舉辦，參與學生並無特定資格限制，凡是一年級至十二年級之在學學生 (年紀約 7 至 18 歲)，皆能參與。受測學生依就學年段分為五組，分別為：Little Beaver (3、4 年級) 組、Benjamin (5、6 年級) 組、Cadet (7、8 年級) 組、Junior (9、10 年級) 組及 Senior (10、11 年級) 組。每組之考題又分難、中、易三種等級。我國原僅開放 Junior 及 Senior 兩組，於 2016 年起增加 Benjamin 組及 Cadet 組，預計未來視參與情形向下推廣至 Little Beaver 組。

| 組別          | 學生年段        | 臺灣首次實施 |
|-------------|-------------|--------|
| Pre-Primary | 國小一、二年級     | 尚未實施   |
| Primary     | 國小三、四年級     | 尚未實施   |
| Benjamin    | 國小五、六年級     | 2016   |
| Cadet       | 國中七、八年級     | 2016   |
| Junior      | 國中九年級、高中一年級 | 2013   |
| Senior      | 高中二、三年級     | 2013   |

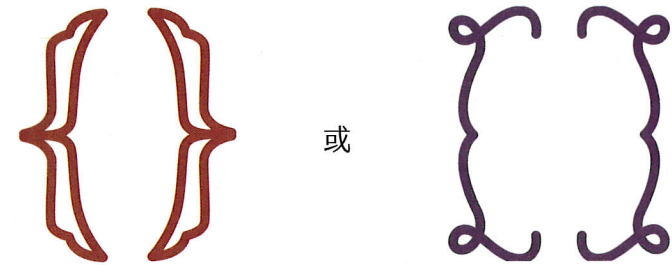
## 計分方式

國際運算思維挑戰賽依題目之難度計分：答對給分、答錯扣分，略過不答則不給分亦不扣分；為了避免負分，挑戰賽之起始分數為各題扣分之總和。Benjamin 組每次挑戰賽共 12 題，各難度平均分配 4 題，其他組別每次共 15 題，各難度平均分配 5 題。各組別分數計算分式如下表所示；各組起始分數為 60 分，最低 0 分，最高 300 分。

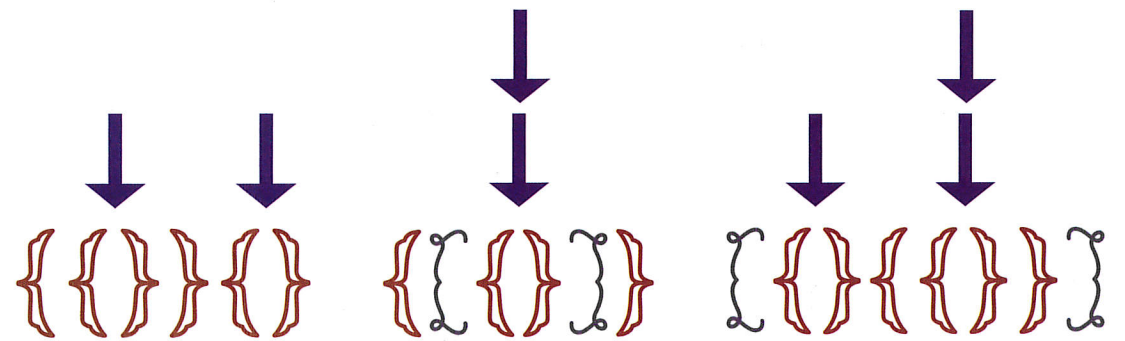
| 年齡組              | 難度 |    |    |    |    |    | 題數 | 起始分 | 最高分 |
|------------------|----|----|----|----|----|----|----|-----|-----|
|                  | 易  |    | 中  |    | 難  |    |    |     |     |
|                  | 正確 | 錯誤 | 正確 | 錯誤 | 正確 | 錯誤 |    |     |     |
| Benjamin (五、六年級) | 16 | -4 | 20 | -5 | 24 | -6 | 12 | 60  | 300 |
| Cadet (七、八年級)    | 12 | -3 | 16 | -4 | 20 | -5 | 15 |     |     |
| Junior (九、十年級)   |    |    |    |    |    |    |    |     |     |
| Senior (十一、十二年級) |    |    |    |    |    |    |    |     |     |

# 1. 括號

一家生產手鐲的珠寶店，店裡的特色商品是用一對括號形狀來當作裝飾花紋。  
而花紋的組成有下列兩種型式（空心或實心）：



接著新的一組括號可以不斷重複地加入裝飾花紋的任何位置，但新插入的左右括號必須相鄰，不能夾著先前插入的裝飾花紋。下面的三個例子示範了如何在第一組括號後插入新的括號：



請問下列哪一種花紋是利用上面提到的方式製作而成的？

- A
- B
- C
- D

## ✔ 正確答案：D

他們從一對實心括號開始，先在第一對實心括號中插入一組空心括號，接者再由空心括號中插入一組實心括號。其他的花紋都不可能由上述的方式製成。選項 A 中第三個位置有錯，因為一對空心括號中不可能只有一個實心左括號。選項 B 中第一個位置就有問題，因為第一個括號不可能為右括號。選項 C 中第二個位置發生問題，因為整個花紋都沒有實心的右括號。

## i 在資訊科學上的意義

題目中的手鐲製作規則與數學中的括號表示法規則完全相同，在資訊領域中很多時候會需要使用成對的符號來代表同一性質的區塊，例如：在製作網頁的 HTML 語法中，會使用 `<b>` 與 `</b>` 的符號將要顯示粗體的文字標示起來，在許多程式語言中，也會使用成對的大括號 `{ }` 來定義一個程式碼區塊。當成對的符號匹配不符合時，就會產生「語法錯誤」(syntax error)，導致電腦無法正確解讀。

當談到語法錯誤的時候，另一個常常會一起提到的是「語意錯誤」(semantic error)，這種錯誤是寫法完全正確，但是無法提供正確的功能。例如我們想將一個變數  $x$  設定為 1 連加到 5，原本的寫法應該是  $x = 1 + 2 + 3 + 4 + 5$ ，但如果漏寫了最後一項，就變成  $x = 1 + 2 + 3 + 4$ ，這兩個式子都是完整且語法正確的，但是第二個式子卻無法滿足「將一個變數  $x$  設定為 1 連加到 5」的要求。

## 💡 關鍵字

符合規則的、違反規則的、語法

## 2. 柵門訊息

海狸們很好客，而且喜歡互相拜訪。當他們外出時，他們會在柵門上留訊息給來訪的客人。

海狸想到用木枝在柵門上的不同擺放方式可表達四種不同的訊息，如下所示：

我們在家喔！請進。



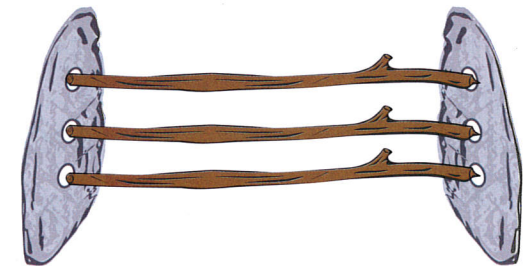
我們中午就會回來了。



我們傍晚會回來。



我們午夜才會回來。



小海狸想改變柵門上木枝的擺放方式，讓留給訪客的訊息能超過四種以上。這些木枝擺放必須滿足下列條件：

- ▶ 木枝只能水平擺放或是完全不放。
- ▶ 木枝的形狀及朝左或朝右方向並不重要。




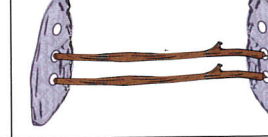

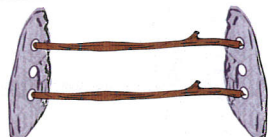

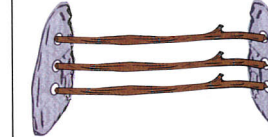
但他不知道最多能設計出幾種不同的柵門訊息。

請問包含原本的四種訊息，最多可以傳遞幾種柵門訊息（僅填入數字。作答範例：4）？

## ✔ 正確答案：8

每根木枝只有兩種可能的狀態：「放」或是「不放」。總共有三根木枝，所以可能的組合有： $2 \times 2 \times 2 = 8$  種。

二進位法是使用兩個不同符號（通常為 0 和 1）來表示的數字系統，以下圖片顯示了所有 8 種可能的組合以及它們對應的二進位代碼示例。

|   |   |  |   |
|---|---|--|---|
|  |  |  |  |
| 二進位代碼：000   | 二進位代碼：001   | 二進位代碼：010  | 二進位代碼：011   |
|  |  |  |  |
| 二進位代碼：100   | 二進位代碼：101   | 二進位代碼：110  | 二進位代碼：111   |

## i 在資訊科學上的意義

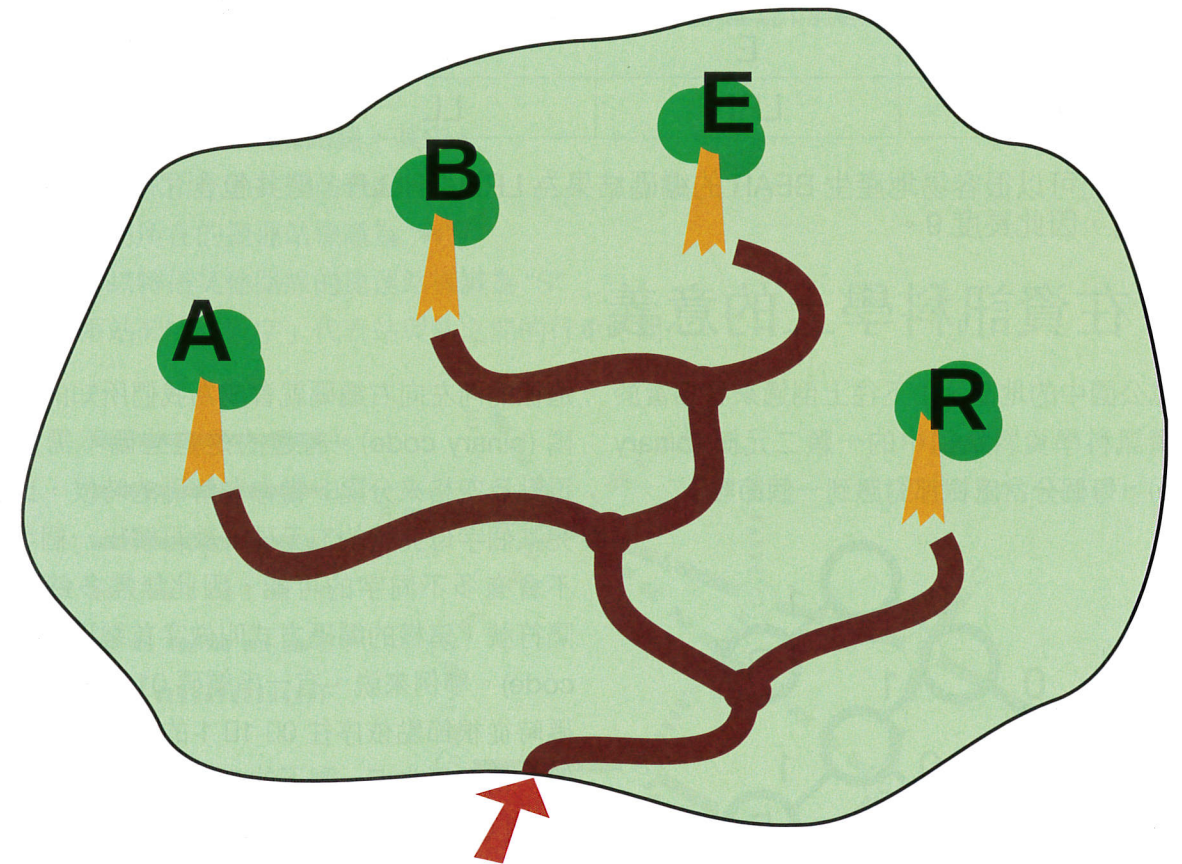
這個任務是處理二進位數字，與二進位系統相關，同時也用到基本的組合概念。二進位表示是電腦中資料表示及運算的最基本概念，也可以由題目了解位元數所能表示的最多幾種不同數字（資料）。

此外，數學是資訊科學中的重要成分之一，所以學生要學習數學概念，例如二進位數、階乘等。雖然這個問題對於較高年級的學生來說可能是一個簡單的工作，但可以幫助較低年級的學生思考組合的觀念。

## 💡 關鍵字

二進位系統、組合學

## 3. 向左向右編碼



海狸先生利用以上地圖制訂了一套英文字母的編碼表示方式：

- ▶ 規則 1：公園中每棵樹都標示一個不重複的英文字母
- ▶ 規則 2：每個英文字母所對應的編碼是根據到達這棵樹所經路線需左轉 (L) 與右轉 (R) 依序組成。
- ▶ 規則 3：要對每個英文字母產生編碼，是從地圖中箭頭標示的公園入口開始走到該字母對應的樹。

範例如下：從公園入口走到標示為 A 的樹，必須左轉兩次，因此 A 字母的編碼是“LL”。一個英文單字的編碼是將其包含字母的編碼依序接起來，因此 BAR 這個字的編碼是 LRLLLR（長度為 6）。

請問以海狸先生設計的編碼方法，要表示 BEAR 這個單字，編碼結果長度為多少（僅填入數字。作答範例：3）？

## ✔ 正確答案：9

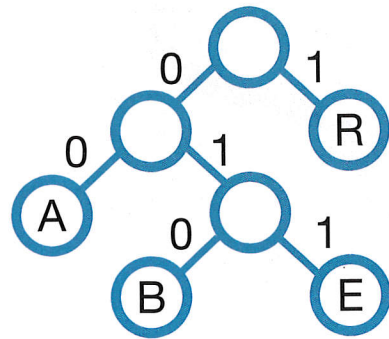
以下表格為圖中每棵樹字母的對應編碼

|     |     |    |   |
|-----|-----|----|---|
| B   | E   | A  | R |
| LRL | LRR | LL | R |

依此我們可以很容易地產生 BEAR 的編碼結果為 LRLRLRLR，總共包含了 3+3+2+1=9 個字碼，因此長度 9。

## i 在資訊科學上的意義

上述公園中的地圖，由下往上倒過來看相當於是資訊科學資料結構中的一顆二元樹 (binary tree)，每個分岔處樹都對應到一個節點：



如果我們把 L 碼表示為 0 而 R 碼表示為 1，

## 💡 關鍵字

二元樹、字首編碼

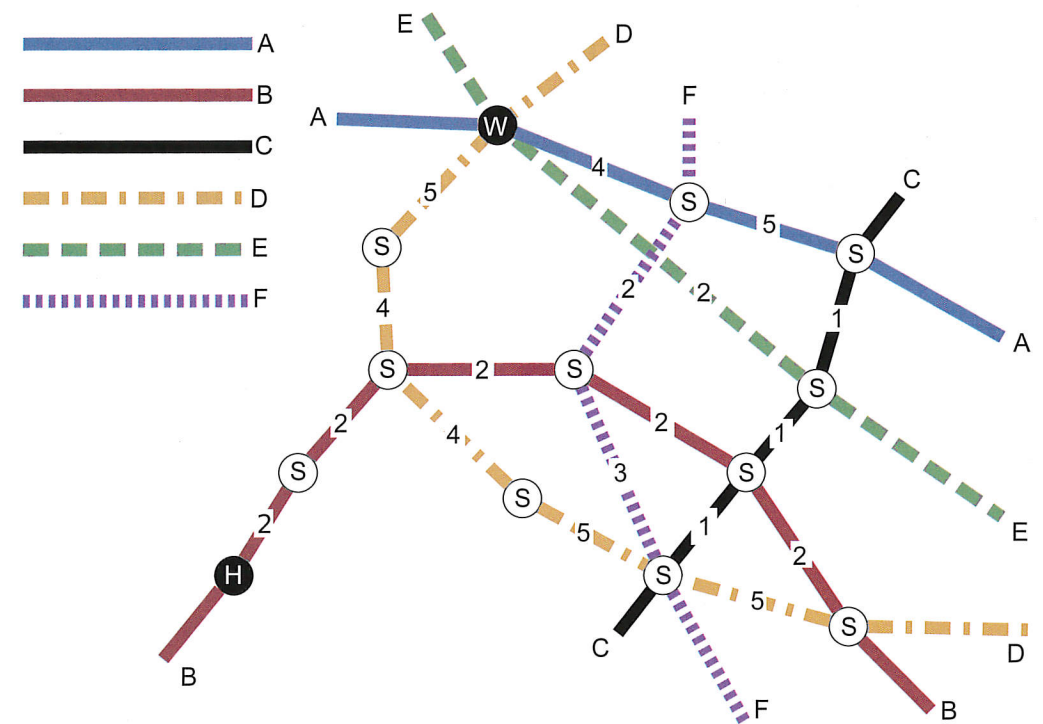
這樣的向左向右編碼就會成為我們所知的二元碼 (binary code)。有趣的是這些碼不需要用逗點及空格來分隔，當你在解碼的時候，比對完某個字母完整的二元碼，就可解出一個字，不會有多不同字的可能，因此無需多餘的區隔符號，這樣的編碼方式叫做字首編碼 (prefix code)。舉例來說，有一串編碼 011001，要解碼時從根節點依序往 0 1 0 1 0 1 的分支，可解出第一個字母為 E。剩下的 001，從根節點依序往 0 0 0 的分支，可解出第二個字母為 A。剩下的 1，從根節點依序往 1 的分支，可解出第三個字母為 A。

# 4. 通勤上班

海狸小娜每天都搭捷運去上班，她家到公司沒有直達的捷運線，必須轉換好幾條路線。

下列地圖顯示捷運的六條路線，其中：

- ▶ 1. 小娜家所在的捷運站標識為“H”
- ▶ 2. 公司所在的捷運站標識為“W”
- ▶ 3. 可以轉搭其他路線的捷運站標識為“S”
- ▶ 4. 兩站間的「數字」代表站與站之間的行車時間。



假設不同路線間轉搭的時間可以忽略，哪一條路線可以讓小娜從她家最快到達公司？

- (A)
- (B)
- (C)
- (D)

## ✔ 正確答案：C

每個選項路線對應的時間如下

- (A)  $2 + 2 + 4 + 5 = 13$
- (B)  $2 + 2 + 2 + 2 + 4 = 12$
- (C)  $2 + 2 + 2 + 2 + 1 + 2 = 11$
- (D)  $2 + 2 + 2 + 2 + 1 + 1 + 5 + 4 = 19$

因此路線 (C) 可以讓小娜從她家最快到達公司。

## i 在資訊科學上的意義

圖型結構是資訊科學中非常重要的資料結構，因為它可以用來表示日常生活中，資料間不同關係所形成的網絡，例如人與人之間互動所形成的社交網絡，或是不同地點間的交通網絡。在這個題目中，給定每一個連接邊上的權重值（兩站間的行車時間），目標是找到圖中兩節點間的最少行車時間路線。這個問題相當於是將每一個連接邊的權重值，設成兩地點間直接相通的距離，從圖型中尋找兩個節點間最短路徑的問題，可以使用 Dijkstra 演算法來解決此問題。

## 💡 關鍵字

圖型走訪、最短路徑

# 5. 串聯

若在一串英文字母裡有二個接連而且相同的字母片段，我們稱之為「串聯片段」，而「串聯長度」則是指這個串聯片段的字母個數。

例如：字串 AABABA 中，有三種不同的串聯片段，分別為 AA（長度為 2，由兩個 A 組合而成）、ABAB（長度為 4，由兩個 AB 組合而成）、BABA（長度為 4，由兩個 BA 組合而成）。

下列英文片段中，最長的串聯長度為多少？

TCTACTAACCTACTAACAC

- (A) 4
- (B) 6
- (C) 8
- (D) 10 以上



## ✔ 正確答案：D

以本題為例，我們可以依下列步驟來計算串聯長度：

- 1. 從原英文片段的第一個字母 T 開始列出，至下一個相同字母 T 之前，列出可能產生串聯的字串，以本題共可列出四個可能字串，如下所示：

TCTACTAACCTACTAACAC TCTACTAACCTACTAACAC

TCTACTAACCTACTAACAC TCTACTAACCTACTAACAC

- 2. 接下來比對每個選項後方緊接的字串，看看是否有重複出現

TCTACTAACCTACTAACAC TCTACTAACCTACTAACAC

TCTACTAACCTACTAACAC TCTACTAACCTACTAACAC

由上列可知，這四個字串都無法延伸串聯，所以無法產生串聯片段。

- 3. 以此類推，再從原英文片段的第二個字母 C 開始列出可能的字串，並與後方接續字串比對，如下所列：

TCTACTAACCTACTAACAC → 串聯長度為 6

TCTACTAACCTACTAACAC → 無法串聯

TCTACTAACCTACTAACAC → 串聯長度為 16

- 4. 用同樣方式反覆幫每一個字母檢查後，我們可知最長的串聯長度為 16。

## i 在資訊科學上的意義

在字串中尋找串聯或者比對兩字串中相同子字串的技術，時常運用在生物資訊與分析自動系統間的狀態變化。在生物資訊領域，如果發現一個未知的動物基因序列，研究者一般會在人類基因序列資料庫中搜尋比對是否包含類似的

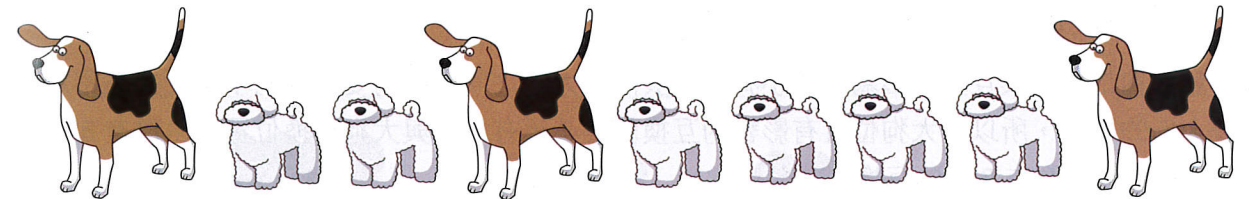
基因，要在基因序列這麼長的字串中尋找相同或類似的子字串，對人類而言是一件困難的工作，但是電腦卻可透過簡單的反覆處理，快速且有效率地完成任務。

## 💡 關鍵字

字串、串聯、重複、反覆

# 6. 狗狗互換

有兩種不同種類的狗按照下面的位置排排站。



當相鄰的兩隻狗互相交換位置，叫做一次互換。

現在想要用多次互換，把三隻大狗排在三個連續的位置上。

請問最少用幾次互換可以做到？

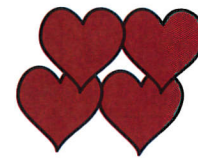
- (A) 5 次
- (B) 6 次
- (C) 7 次
- (D) 8 次



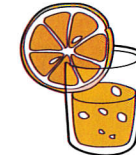
## 7. 飲料小舖

4 個好朋友一起去旅行，他們決定在路邊的飲料小舖停下來喝點東西。

可惜的是，這間小舖提供的 4 種飲品都只剩下最後一杯的存量，4 個朋友一人喝一杯之後就賣光了。以下是 4 個朋友對每一種飲品的喜好程度，愛心越多代表越喜歡。



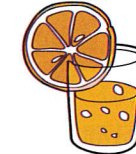
Anna



Bernard



Christine



Daniel



例如 Anna 最喜歡可樂，再來是果汁，再來是咖啡 ... 以此類推。

現在我們要幫 4 個朋友分配飲品，盡量滿足每個人的喜好，也就是讓愛心的總數越多越好。請計算一下，我們最多可以獲得幾個愛心（請僅填入數字，作答範例：10）？

### ✔ 正確答案：B

三隻大狗可以按照以下的方法排到連續的位置：

- 首先將第一隻大狗向右互換兩次
- 然後將最後一隻大狗向左互換四次

每隻小狗都會參與互換，因為每隻小狗都位於兩隻大狗之間。把兩隻相鄰的小狗互換並不會改變大狗的位置，所以對大狗位置有影響的互換就是必須有小狗與大狗交換位置。由於有 6 隻小狗，所以至少會有 6 次的互換。

注意：若不動左邊那隻大狗，試著把右邊兩隻大狗都向左移動；或是不動右邊那隻大狗，把左邊兩隻大狗都向右移動，都需要六次以上的互換，才能將三隻大狗排到連續的位置。

### i 在資訊科學上的意義




如果我們把狗當作是儲存電腦記憶體中的資料，「互換」可看作是交換兩份資料的位置。在電腦中由於儲存資料的記憶體空間是有限的，「互換」這個運算可以只用額外一筆資料的記憶體空間，對某個資料結構中任兩筆資料值進行儲存位置的交換。許多排序演算法，透過有系統的對一串資料進行大小比較及互換，將資料組織成遞增或遞減的順序儲存。在這個題目中限定只能相鄰兩隻狗彼此互換，氣泡排序法也採取類似的概念：在每一輪的排序步





驟，透過比較相鄰資料是否沒按順序排好，決定是否交換這兩筆相鄰資料。為了盡可能快速地執行，進行互換的次數必須越少越好。另外類似的概念是當一筆數量較大的資料被切成幾個小片段，分散儲存在記憶體的不同位置沒有連續存放（就像三隻大狗排在不連續的位置），則每次存取要找到不同位置而降低效率。若進行資料的位置互換，將資料存放到連續的位置，則可增進資料讀取效能。

### 💡 關鍵字








資料互換、排序演算法、氣泡排序法

## ✔ 正確答案：14

因為有 3 個人最喜歡 ，但只有 1 人能喝到  獲得 4 顆愛心，剩下兩人則只能選擇次要喜歡的飲料，獲得 3 顆愛心；Daniel 可以喝到最喜歡的飲料 ，獲得 4 顆愛心，總計獲得  $4+3+3+4=14$  顆愛心。我們可以透過以下的分配方式獲得這個數字：

|   |   |   |   |
|---|---|---|---|
|  |  |  |  |
| Anna  | Bernard   | Christine   | Daniel  |

讓我們用另外一種方法解題：

因為最後一個人最喜歡 ，而其他人都最不愛它，所以最後一個人一定要選擇 ，才有可能獲得最多愛心。接下來，前 3 個人最喜歡的飲料相同，不過最前面的 Anna 第二喜歡的飲料是 ，而其他 2 人都把  排第三名，所以前面的 Anna 應該選擇 ，其他 2 人就分別選擇  和 。

## i 在資訊科學上的意義

最佳化 (Optimization) 是資訊科學中重要的觀念。在這個例子中，最佳化任務就是要想辦法讓 4 個人獲得最多的愛心。最佳化問題的應用相當廣泛，像是匹配問題 (matching problem)，就是盡量讓每個人配到最喜歡的飲料，以求得最滿意的答案，這在真實世界中格

外重要。舉例來說，世界上有很多病人排隊等待器官移植以活命，但是為了能夠成功完成手術，器官移植受到種種限制，例如：器官捐贈者與病患必須血型相符等。這些條件限制讓真實世界中的匹配問題變得更複雜，而難以讓所有人都滿意。

## 💡 關鍵字

最佳化

## 8. 紅燈藍燈

海狸小貝使用 7 個燈發送消息給她的朋友，每個燈可以是紅色或藍色。

小貝使用前 5 個燈來表示要發送的信件。為了檢查該消息傳送是否有錯誤，小貝按照下列方法使用最後 2 個燈：

- ▶ 在前 5 個燈中，如果藍燈的數量是偶數，則第 6 個燈設為藍色；否則（藍燈數量是奇數時），第 6 個燈設為紅色。
- ▶ 在前 6 個燈中，如果紅燈的數量是偶數，則第 7 個燈設為紅色；否則（紅燈數量是奇數時），第 7 個燈設為藍色。
- ▶ 注意，0 要當成偶數。





例如，如果小貝將前 5 個燈設置如下：



那麼第 6 個燈應該是藍色的（因為前 5 個燈中有 2 個藍燈），第 7 個燈應該是藍色的（因為前 6 個燈中有 3 個紅燈）。所以，她傳送消息的正確燈號應該是：



以下哪個燈號代表消息傳送是正確的？

- (A) 
- (B) 
- (C) 
- (D) 

## ✔ 正確答案：C

對於選項 (A)，因為前 6 個燈中有 1 個紅燈，第七個燈應為藍色。

對於選項 (B)，因為前 5 個燈中有 0 個藍色燈，而 0 是偶數，所以第六個燈應該是藍色。

對於選項 (C)，前 5 個燈中有 1 個藍燈，所以第 6 個燈是紅色的。在前 6 個燈中有 5 個紅燈，所以第 7 個燈是藍色的，完全正確。

對於選項 (D)，因為前 6 個燈中有 3 個紅燈，第 7 個燈應為藍色。

## i 在資訊科學上的意義

資料及訊息傳遞過程有可能會發生錯誤，例如有人在唸出他們的電話號碼、信用卡號碼等，有可能其中某個數字錯誤或順序顛倒了。我們可以使用一些檢查碼技術來發現有錯誤甚至進行更正，便可以減少辨識信息時的語意錯亂。在這個題目中，第 6 和第 7 個燈是用來作為檢查燈，以確保所有的燈號都已正確傳送：如果第 6 或第 7 個燈的顏色不符合設定規則，就可知道收到的燈號有錯誤。

信用卡號碼、ISBN 號碼和 UPC 代碼（商品上的條碼）都具有這些檢查碼，如果收到的訊息有任何錯誤，則可以很快被檢查出來，不至於造成繼續傳遞錯誤資訊。在網路的封包傳遞過程中，很需要這樣檢查碼的機制，因為網路在傳遞的過程中時常會有封包遺失，若接收端不知道其收到的資料不完整，則會辨識出錯誤的訊息。但若知道接收的資料不完整，則可以請求對方重新發送封包。

## 💡 關鍵字

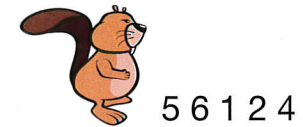
檢查碼

## 9. 海狸編碼

海狸小文上了一堂課，課堂中要觀察一串海狸和樹圖片的排列順序記錄下來，如下圖：

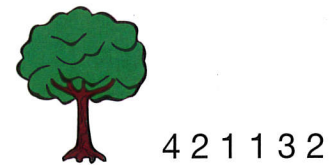


小文想到一個好辦法，用較短的方法來表示這串資料。例如上面的資料用下列方式記錄：



也就是先記下一開始起始的圖片（圖片可能是海狸或樹），寫出該圖片連續出現的次數；接著寫出下一張不同圖片連續出現的次數；接著再寫出下一次圖片改變後的圖片連續出現次數…以此類推。

現在小文的筆記上有一筆資料記錄著：



請問原來課堂上海狸和樹的排列順序是哪一串呢？

- (A)
- (B)
- (C)
- (D)

### ✔ 正確答案：B

起始圖片是樹，後面是數字 4，所以我們知道前面先是 4 棵樹。接下來數字 2 就表示接著 2 隻海狸、接著數字 1 就是接著 1 棵樹。以此推類，接下來再交替出現 1 隻海狸, 3 棵樹, 2 隻海狸。

### i 在資訊科學上的意義

資料壓縮可以節省時間（透過網路發送訊息的時間）、空間（在磁碟或 RAM 裡儲存訊息的空間）、和金錢（因為時間和空間需要錢）。資訊科學家設計了各種不同的資料壓縮技術，根據資料序列中的出現樣式，使用較少的訊息對它們進行編碼。這個問題中採用的這種壓縮技術稱為游程編碼 (run length encoding)，如果海狸及樹出現的序列更長的話，更能顯出可以簡潔地記下訊息，並且這個編碼可以很容易地解碼出原始資料。在資訊科學中，資料表示成 1 和 0 組成的符號序列，而不是海狸和樹的

圖片序列，但可以運用相同的方式進行壓縮編碼。許多類型的資料，例如視訊檔案 (.mp4) 或音訊檔案 (.mp3)，圖片檔案 (.png) 甚至文字訊息 (.txt)，都可以透過各種演算法，採用較省空間的編碼方式進行壓縮，並可解壓縮回來。壓縮方法依資料經過壓縮後，資訊是否不受損失，能完全取消復原到壓縮前的原樣，分成無失真資料壓縮 (Lossless Compression) 及破壞性資料壓縮 (Lossy Compression)，游程編碼是屬於無失真資料壓縮方法。

### 💡 關鍵字

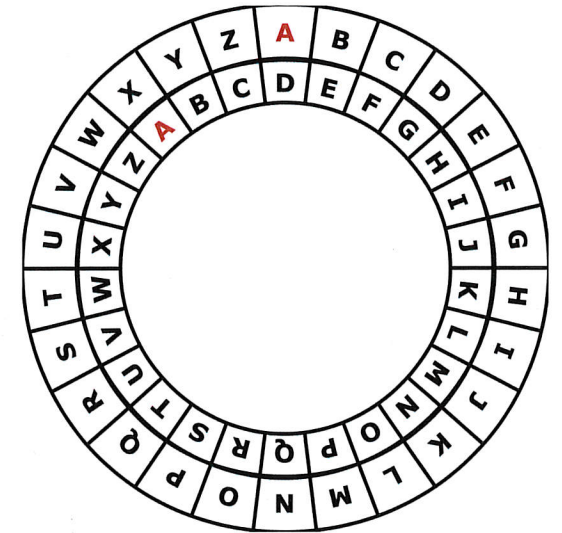
資料壓縮、游程編碼

## 10. 午餐吃什麼？

Bibo 和 Biba 喜歡利用加密的方式來互傳訊息，為了幫訊息加密，他們使用了一個叫做「加密碟」的裝置，加密碟的設計包含一個外碟和內碟（如下圖）。

每天 Bibo 會送一個加密訊息給 Biba，內容講的是今天中午要吃什麼？該訊息是這樣加密的：

1. Bibo 寫下要想吃的餐廳名稱，如 PIZZA
2. 在每一個字母的下方他會標註一個 1 到 9 的數字來表示內碟要從原來字母對齊的位置往左（逆時針）轉多少格。例如在右圖中內碟往左轉了三格。
3. 接著 Bibo 把旋轉之後對應到的內碟字母寫下來。右圖中外碟 P 對應到的內碟字母是 S。
4. Biba 把加過密的訊息送給 Biba，然後 Biba 則要解碼之後才知道午餐的內容。



範例：Bibo 要吃 PIZZA 所以把訊息加密成

|        |   |   |   |   |   |
|--------|---|---|---|---|---|
| 原始訊息   | P | I | Z | Z | A |
| 逆時針旋轉  | 3 | 1 | 4 | 3 | 1 |
| 加密後的訊息 | S | J | D | C | B |

今天 Biba 收到的訊息如下，請問 Bibo 今天午餐想要吃什麼呢？（請填入英文大寫；作答範例：APPLE）

|        |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|
| 原始訊息   | ? | ? | ? | ? | ? | ? | ? |
| 逆時針旋轉  | 3 | 5 | 1 | 7 | 2 | 4 | 8 |
| 加密後的訊息 | O | F | T | H | I | R | I |

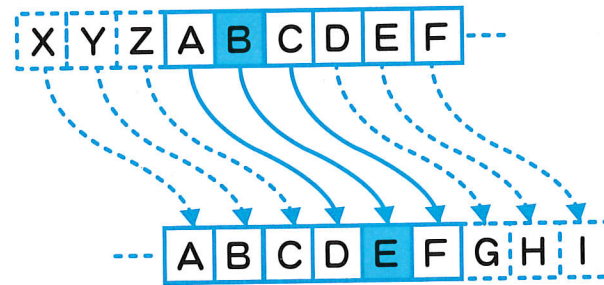
### ✔ 正確答案：LASAGNA

要回推原來的訊息必須先將內外碟轉回起始位置，也就是兩個 A 要對在一起。接著，只要按照訊息上的格數將外碟往右（順時針）轉。然後找出字母出現在內碟與其對應外碟的字母就可以得到答案。重複每一個步驟就講全部訊息解出來。

### i 在資訊科學上的意義

「凱撒加密法」為現存最早的加密法之一。加密是為了讓訊息或資訊只讓獲得允許的人看得懂的編碼過程。就技術上而言加密流程常需要一個密鑰來幫忙產生，而解密則是一個相反的流程。簡言之就是將明文轉換為密文之後再轉換為明文的過程。所以加密演算法包含了一組加密與解密的方法。

在「凱撒加密法」中，加密人先挑選一個數字當作密鑰然後把明文中的字母做相對應的位移。例如：如果密鑰數字是 3，則將 B（英文字母順序中的第 2）轉換為 E（英文字母順序中的第 5）。而凱撒加密碟盤則是最簡單明瞭的轉換工具。



然而這個加密法的缺點就是僅用一個數字當作密鑰，所以攻擊者只要測試幾個數字之後就能輕易破解。但是如果將密鑰改變成兩個或是更多數字的話（如本題），破解的難度就會大幅提升（維吉尼亞加密法）。

### 💡 關鍵字

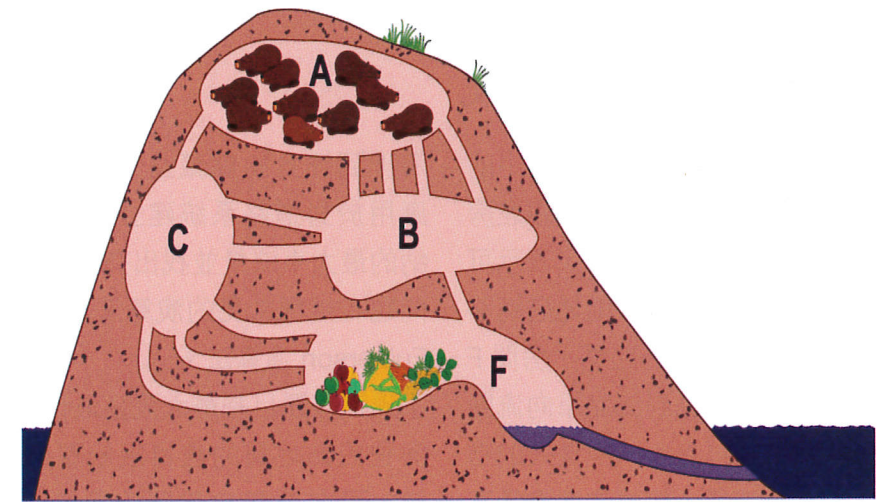
密碼學、多字母加密法、凱撒加密法、維吉尼亞加密法

## 11. 水壩地道 (簡易)

「家園水壩」建造了可連接四個房間 (A, B, C, F) 的地道。前三個房間 (A、B 和 C) 是客廳，第四個房間 (F) 是食物儲藏室（見下圖）。

有 10 隻海狸正待在 A 房間，而牠們想到 F 房間吃東西。因為所有海狸都很餓，牠們全都想要盡快到達食物儲藏室。

通過一條地道到另一個房間需要 1 分鐘，但每條地道同一時間只能有一隻海狸通行。



房間之間有特定數量的地道連接：

- ▶ A 和 B 之間：4 條地道。
- ▶ A 和 C 之間：1 條地道。
- ▶ B 和 C 之間：2 條地道。
- ▶ B 和 F 之間：1 條地道。
- ▶ C 和 F 之間：3 條地道。

但每個房間都沒有容量限制，也就是每個房間都可以容納全部的海狸。

在 2 分鐘內，最多有幾隻海狸可以到達食物儲藏室？（請填入一個 0 到 10 之間的整數；作答範例：9）

## ✔ 正確答案：2

最好的情況是 2 分鐘後，會有 2 隻海狸到達食物儲藏室。

圖中有兩條最短的路線從 A 到 F：“A 到 B 到 F”、“A 到 C 到 F”。這兩條路線各自可容納 1 隻海狸，都需 2 分鐘能到達食物儲藏室。有一條路線“A 到 B 到 C 到 F”可容納 2 個海狸，但它需要 3 分鐘才能到達食物儲藏室。

因此，只有 2 隻海狸可以在 2 分鐘內到達食物儲藏室，分別使用從房間 A 到 B 到 F，以及從房間 A 到 C 到 F 的路線。

## i 在資訊科學上的意義

我們可以把這個地道網路表示成圖形理論中的流程網路，也就是把房間表示成節點，相連房間之間的地道數為連接邊的容量限制，表示可同時通過到另一房間的海狸數量。則地道網路可對應到一個以房間 A 的節點為起點，房間 F 的節點為終點，只允許從起點到終點單向流動的有向圖。每個邊同時可接收一些進入的物件（海狸）數目稱為其流量，但要注意流量不能超過每個邊的容量限制。

這個题目的目標是要讓經過地道網路的最大流量最佳化，使得在指定的時間內，能有最多的

海狸到達食物儲藏室。這樣的網路表示法可應用於對道路系統建立交通模型，評估可容納的最大車流量並找出交通瓶頸。要解決相關問題有幾種常見的演算法，其中之一是 Ford-Fulkerson 演算法。

不過這個题目是最大流量最佳化的特例版，當海狸因為地道的數目限制暫時無法繼續前進時，可在 B 和 C 房間中等待。但在傳統的流程網路問題中，物件是不能停留在某個節點的，必須立即通過節點從其他邊（通道）流出。

## 💡 關鍵字

圖型結構、最大流量

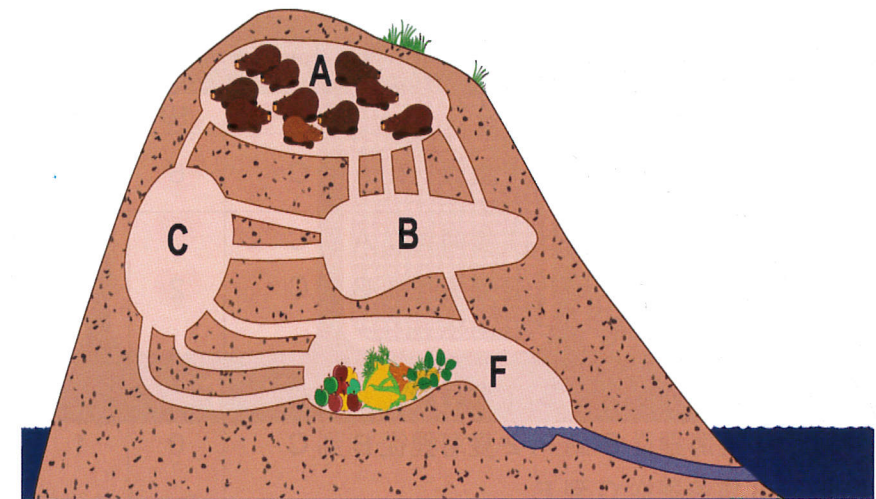


# 12. 水壩地道 (進階)

「家園水壩」建造了可連接四個房間 (A, B, C, F) 的地道。前三個房間 (A、B 和 C) 是客廳，第四個房間 (F) 是食物儲藏室 (見下圖)。

有 10 隻海狸正待在 A 房間，而牠們想到 F 房間吃東西。因為所有海狸都很餓，牠們全都想要盡快到達食物儲藏室。

通過一條地道到另一個房間需要 1 分鐘，但每條地道同一時間只能有一隻海狸通行。



房間之間有特定數量的地道連接：

- ▶ A 和 B 之間：4 條地道。
- ▶ A 和 C 之間：1 條地道。
- ▶ B 和 C 之間：2 條地道。
- ▶ B 和 F 之間：1 條地道。
- ▶ C 和 F 之間：3 條地道。

但每個房間都沒有容量限制，也就是每個房間都可以容納全部的海狸。

要讓所有海狸都到達食物儲藏室，最短需要幾分鐘？（請填入一個 0 到 10 之間的整數；作答範例：9）

## ✔ 正確答案：4

為了要讓全部海狸在最短時間到達食物儲藏室，必須盡量使用房間 A 到 B 之間的地道，才可很快地減少起點處的海狸數量。最佳解決方案是在第一分鐘有 3 隻海狸出發，在第二分鐘另外 3 隻海狸再出發。因為要離開房間 B 只有 3 條地道，如果海狸們以最高容量（4 隻海狸）使用從房間 A 到 B 的地道，則房間 B 到

C 和 B 到 F 的連接地道會成為瓶頸，每個時間會多一隻海狸必須在房間 B 中等待。

下表說明海狸們在每一分鐘的移動結果，直到所有海狸都到達食物儲藏室。這個任務的最佳解決方案是 4 分鐘，但有好幾種方法可以讓所有海狸全部到達食物儲藏室，以下我們列舉出其中一種海狸不必在房間 B 等待的移動方式。

| 時間     | 移動方式   | 移動後房間中的海狸數量 |   |   |    |
|--------|--|-------------|---|---|----|
|        |  | A           | B | C | F  |
| 一開始    |  | 10          | 0 | 0 | 0  |
| 第 1 分鐘 | 房間 A: 3 隻海狸從 A 到 B, 1 隻海狸從 A 到 C   | 6           | 3 | 1 | 0  |
| 第 2 分鐘 | 房間 A: 3 隻海狸從 A 到 B, 1 隻海狸從 A 到 C<br>房間 B: 1 隻海狸從 B 到 F, 2 隻海狸從 B 到 C<br>房間 C: 1 隻海狸從 C 到 F | 2           | 3 | 1 | 0  |
| 第 3 分鐘 | 房間 A: 1 隻海狸從 A 到 B, 1 隻海狸從 A 到 C<br>房間 B: 1 隻海狸從 B 到 F, 2 隻海狸從 B 到 C<br>房間 C: 3 隻海狸從 C 到 F | 0           | 1 | 3 | 6  |
| 第 4 分鐘 | 房間 B: 1 隻海狸從 B 到 F<br>房間 C: 3 隻海狸從 C 到 F   | 0           | 0 | 0 | 10 |

## i 在資訊科學上的意義

我們可以把這個地道網路表示成圖形理論中的流程網路，也就是把房間表示成節點，相連房間之間的地道數為連接邊的容量限制，表示可同時通過到另一房間的海狸數量。則地道網路可對應到一個以房間 A 的節點為起點，房間 F 的節點為終點，只允許從起點到終點單向流動的有向圖。每個邊同時間可接收一些進入的物件（海狸）數目稱為其流量，但要注意流量不能超過每個邊的容量限制。

這個题目的目標是要讓經過地道網路的最大流量最佳化，使得在指定的時間內，能有最多的

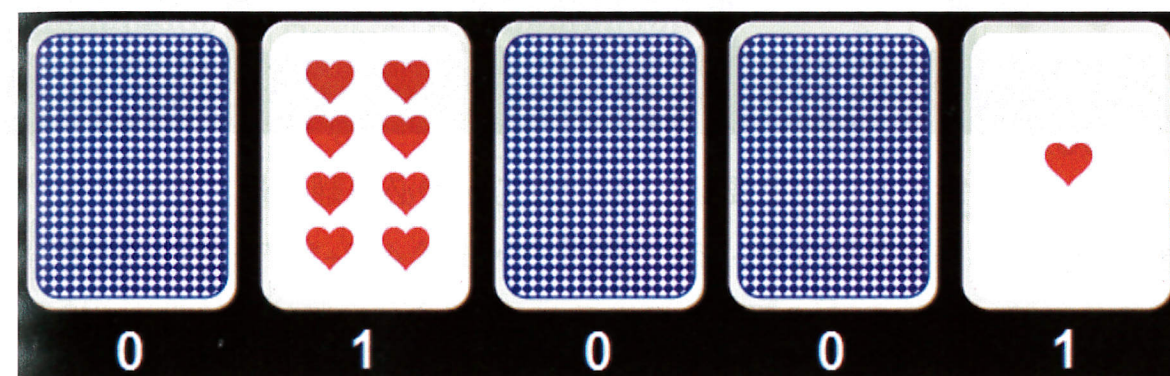
海狸到達食物儲藏室。這樣的網路表示法可應用於對道路系統建立交通模型，評估可容納的最大車流量並找出交通瓶頸。要解決相關問題有幾種常見的演算法，其中之一是 Ford-Fulkerson 演算法。

不過這個题目是最大流量最佳化的特例版，當海狸因為地道的數目限制暫時無法繼續前進時，可在 B 和 C 房間中等待。但在傳統的流程網路問題中，物件是不能停留在某個節點的，必須立即通過節點從其他邊（通道）流出。

## 13. 卡牌密碼

五張卡牌向下覆蓋在桌面上，由左而右分別有 16、8、4、2、1 顆愛心，每張卡牌下面有 0 或 1 的卡牌密碼，如果數字是 1 就可以把卡牌翻開顯示卡牌上的愛心，如果數字是 0 則必須保持覆蓋，最後計算桌面上所有愛心數量，就是該組卡牌密碼代表的數字。

舉例來說，下圖的卡牌密碼是 01001，總共有九個愛心，所以這組卡牌密碼代表的數字是 9



請問 26 的卡牌密碼是多少？

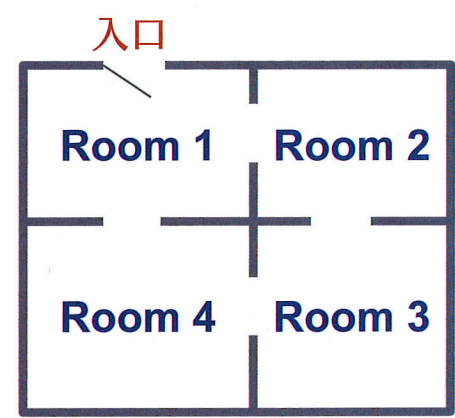
- (A) 01111
- (B) 10010
- (C) 11010
- (D) 11011

# 14. 入侵偵測

在海狸博物館裡有一套能偵測入侵者的智能安全系統，所謂入侵者是指未經由入口進入博物館的人。當有人進出任一個房間時，系統會準確地偵測每個房間的人數，並記錄在表格裡。(註：同一時間可能有多人進出房間)

下表為智能安全系統每個時間點的人數紀錄，下圖為博物館的房間格局，博物館共有四個房間，編號分別為 Room1 到 Room4，而入口處就在圖中 Room1 的上方。

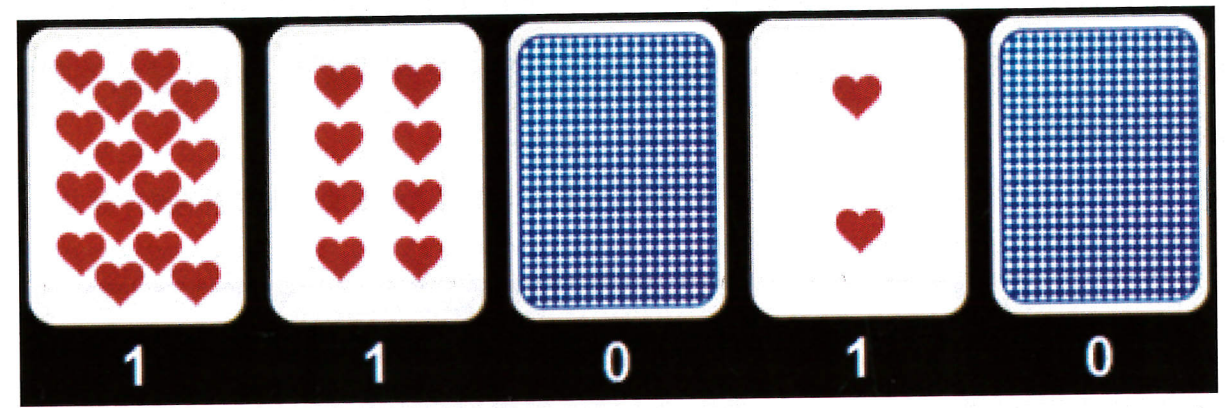
| Time  | Room1 | Room2 | Room3 | Room4 |
|-------|-------|-------|-------|-------|
| 10:00 | 2     | 0     | 0     | 0     |
| 10:07 | 3     | 0     | 0     | 0     |
| 10:08 | 2     | 1     | 0     | 0     |
| 10:12 | 4     | 1     | 1     | 0     |
| 10:13 | 2     | 2     | 3     | 0     |
| 10:17 | 5     | 2     | 2     | 1     |
| 10:20 | 4     | 1     | 2     | 2     |



系統會在 10 點幾分首次偵測到入侵者？(請填入一個 0 到 59 之間的整數；作答範例：0)

## ✔ 正確答案：C

由上圖可知數字 26 應該用卡牌密碼 11010 代表，而且我們也可以證明，這將是唯一一組能代表 26 的卡牌密碼：



由於  $8+4+2+1=15$ ，所以如果要顯示 26 顆愛心，勢必得翻開最左邊的卡片；接下來要用右邊四張卡牌顯示  $26-16=10$  顆愛心，由於  $4+2+1=7$ ，所以必須翻開 8 顆愛心的卡牌；剩下  $10-8=2$  顆愛心要顯示，4 顆愛心的卡牌如果翻開就超過了，1 顆愛心的卡牌翻開仍不足，因此只剩下唯一的選擇翻開 2 顆愛心的卡牌。

## i 在資訊科學上的意義

由右而左看卡牌時，卡牌上分別有 1, 2, 4, 8, 16 顆愛心，每一張卡牌的愛心數量都是前一張的兩倍，我們可以透過翻開或覆蓋這些卡牌來表達 0~31 之間任何一個數字。這些愛心其實是根據二進制數字系統安排的，

二進制系統中只能使用 0 和 1，這套系統剛好也是電腦儲存資料的方法，在電腦的世界中，不論是文字、圖片、影片、音樂都可以透過編碼轉換成二進制的形式。

## 💡 關鍵字

二進制系統

## ✔ 正確答案：13

| Time  | Room1 | Room2 | Room3 | Room4 | 備註   |
|-------|-------|-------|-------|-------|--|
| 10:00 | 2     | 0     | 0     | 0     | 有 2 人從入口進入 Room1   |
| 10:07 | 3     | 0     | 0     | 0     | 又有 1 人從入口進入 Room1  |
| 10:08 | 2     | 1     | 0     | 0     | 有 1 人從 Room1 進入 Room2  |
| 10:12 | 4     | 1     | 1     | 0     | 有 1 人從 Room2 進入 Room3<br>有 1 人從 Room1 進入 Room2<br>有 3 人從入口進入 Room1 |
| 10:13 | 2     | 2     | 3     | 0     | 偵測到入侵者   |
| 10:17 | 5     | 2     | 2     | 1     |  |
| 10:20 | 4     | 1     | 2     | 2     |  |

我們將每個時間點可能發生的事件寫在備註欄中，由此可發現，系統偵測到入侵者的時間為 10:13，因為這個時間點有二人進入 3 號房間，但前一個紀錄的時間點 (10:12) 與 3 號房間相鄰的 2 號房間和 4 號房間只有一人，所以有一個入侵者進入 3 號房間。

## i 在資訊科學上的意義

美國的星巴克咖啡導入 Density 系統供消費者透過 Density app 就能輕鬆得到某家分店人數的資訊，並輕易避開擁擠時段。Density 的外觀是一個小巧的黑色立方體，店家只要把這個設備放置在門的邊緣，它便能透過紅外線距離感測器 (Infrared Distance Sensors) 偵測人的進出。Density 能夠計算每日進出建築物的人數，以及某個時刻建築物裡的人數，店家也不

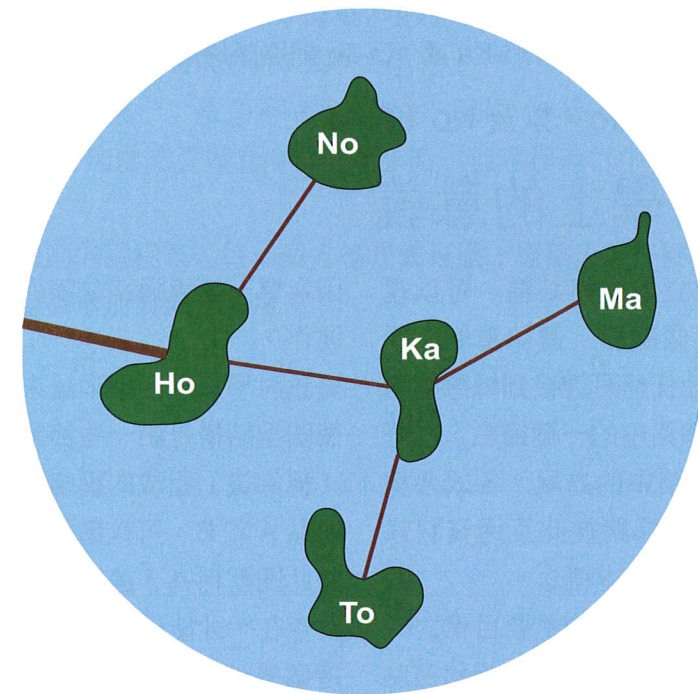
用擔心是否會侵犯消費者隱私，因為 Density 不會取得個人身份資訊。另外像是機場這類重要的場所，安全系統會持續紀錄各區域人數。電腦程式透過攝影機所拍攝的即時畫面去偵測人數，這些程式使用人工智慧技術去辨識物體是否為人類，並透過簡單的規則去偵測非法入侵者。

## 💡 關鍵字

智能安全系統、規則、狀態轉變

# 15. 不斷線的網路

HoNoMaKaTo 群島由五個美麗的島嶼組成：Ho 島、No 島、Ma 島、Ka 島及 To 島。其中只有大島 Ho 島透過大頻寬電纜連接到網際網路。此外，其他的小島間由小電纜所連接。例如 Ho 島和 No 島相連、Ho 島和 Ka 島相連、Ka 島和 Ma 島相連、以及 Ka 島和 To 島相連。透過這些小電纜，所有的島都可經由 Ho 島的大電纜連接到網際網路。如下圖：



這些島上的居民希望不論是哪一條小電纜中斷，每個島嶼仍可以連接到網際網路。如果再鋪設兩段小電纜，請問下列哪一種連接方式能達成網路不斷線的目標？

- (A) 連接 Ho 和 To，以及 No 和 Ma.
- (B) 連接 Ho 和 To，以及 Ma 和 To
- (C) 連接 Ka 和 No，以及 No 和 Ma
- (D) 即使再連接兩條小電纜也無法做到

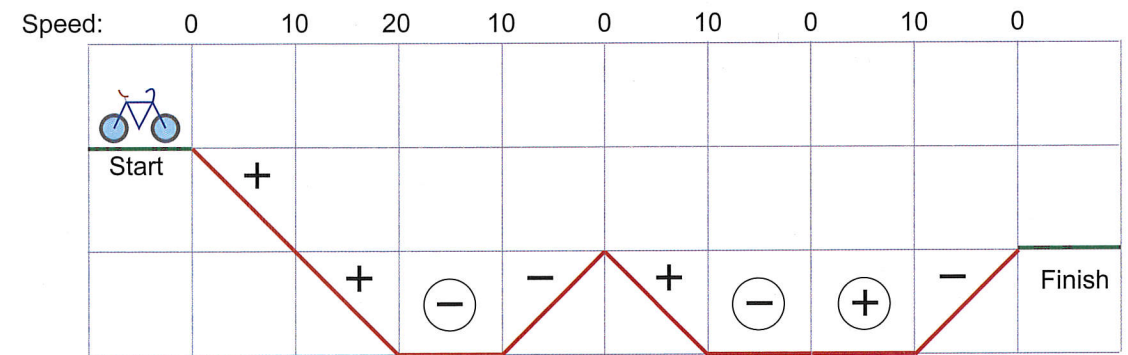
# 16. 飛輪趣

「飛輪趣」是最近很熱門的健身訓練課程。進行訓練時，課程會提供「上坡路段」、「下坡路段」與「平地路段」的騎乘組合，以帶領你達到健身效果。

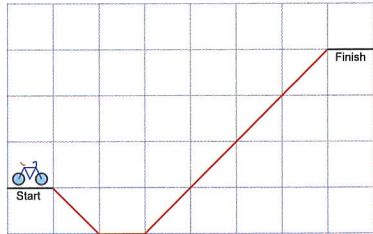
以下是飛輪趣訓練的基本規則：

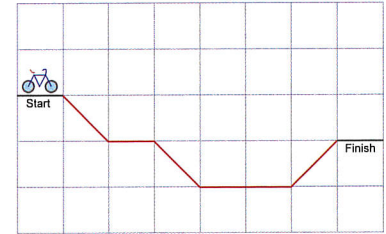
- ▶ 課程開始時，車子時速為 0 公里 / 小時。
- ▶ 經過一個下坡路段，車子時速會增加 10 公里 / 小時。
- ▶ 經過一個上坡路段，車子時速會減少 10 公里 / 小時。
- ▶ 平地路段時，騎士可以二選一：選擇將時速增加 10 公里 / 小時，或選擇將時速減少 10 公里。
- ▶ 課程結束時，車子的時速應該正好為 0 公里 / 小時。但課程進行中，騎士不可以讓飛輪靜止下來。

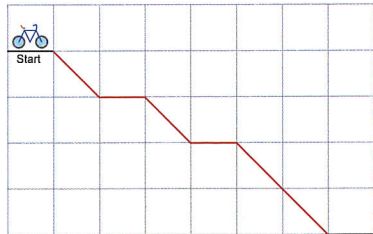
下圖是一個符合訓練規則的課程表，每一個方格代表經過一種路段。仔細看一下圖片中的圓圈，代表平地路段時騎士選擇增加或減少時速。

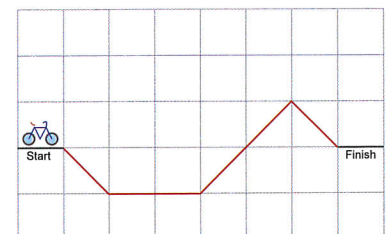


以下有 4 張飛輪趣的課程表，可是只有一張有辦法符合訓練規則。請問是哪一張？

(A) 

(C) 

(B) 

(D) 

## ✔ 正確答案：A

當連接一條 Ho-To 網路：若 Ho-Ka 或 Ka-To 網路斷線，To 仍有和 Ho 連線。若 Ho-Ka 斷線，Ka 仍可以透過 To 連線。

當連接一條 No-Ma 網路：若 Ho-Ka 或 Ka-Ma 網路斷線，Ma 仍可以透過 No 連線。若 Ho-No 斷線，No 仍可以透過 Ma 和 Ka 連線。若 Ho-Ka 斷線，Ka 仍可以透過 Ma 和 No 連線。

若連接一條 Ma-To 網路：當 Ho-No 斷線時，No 就沒有網路。

若連接一條 Ka-No 網路：當 Ho-Ka 或 Ka-To 網路斷線時，To 就沒有網路。

因此答案是連接 Ho 和 To，以及 No 和 Ma。

## i 在資訊科學上的意義

HoNoMaKaTo 群島建立的網路結構，可以視為全球網際網路的一個縮影。例如電腦、手機、電視、冰箱和其他任何可連接到網際網路的設備，都可以視為網路中的一個節點，就好比是 HoNoMaKaTo 網路中的島嶼。各式各樣的東西設備都可以連上網路彼此互通資料訊息，這就是「物聯網」的基本概念。

在 1960 年代，網際網路發明的主要目標之一是建立一個可靠的網路，尤其是某些網路節點間斷線不應導致整個網路的癱瘓。對於其他類型的網路，例如交通網路、供應鍊等，更不能

因為單一連線通路故障而導致整個系統都無法運作。

資訊科學家以圖型理論來推理網路的特性，一個圖型結構是由一些節點，以及節點間的連接（稱為邊）組成的網路。若一個圖型中任兩個節點 A 和 B，可以透過一個或多個邊連通，則將此圖型稱為「連通圖」。如果有一個邊一定要存在，才能保持圖型中所有節點的連通，則這個邊被稱為一個「橋樑 (bridge)」。在可靠的網路中，必須避免有「橋樑」存在。

## 💡 關鍵字

網際網路、物聯網、連通圖、橋樑 (bridge)

## ✔ 正確答案：C

在 C 課程中，3 個平地路段可以使用【十（加速）、－（減速）、－（減速）】，這樣就可以在結束時讓速度剛好歸零。或者你也可以使用【－、十、－】或【－、－、十】，都符合訓練的規定。其他選項的課程都無法符合課程規則，說明如下：

- 課程 A：即使在唯一的平地路段使用加速，速度也會在第 4 個路段時提早歸零靜止。
- 課程 B：即使在 2 個平地路段都使用減速，結束時的速度一定超過 0，無法符合最後停止的規則。
- 課程 D：請注意最後一個路段是加速。由於我們不可能在課程中讓飛輪的速度是負數，所以我們也不可能在結束時讓時速剛好變成 0。

## i 在資訊科學上的意義

括號是一種常見的表示符號。例如一組數學運算式： $[n(n-1)]/2$  或  $(a+b)(a-b)$ ，括號決定了運算的優先順序。我們都知道括號一定是成對出現的，在運算式中，左括號和右括號位置和數量一定要匹配且正確，這樣才能得出正確的結果。

同樣的，在各種程式語言中，也有許多成對括號的概念。例如網頁使用的 HTML 語言，一個段落會以 `<p>` 作為開頭，以 `</p>` 作為結尾，而且每個標籤也都使用“`<`”和“`>`”夾起來。為什麼括號在程式語言中這麼常見呢？

## 💡 關鍵字

戴克語言 (Dyck language)

原因很簡單：成對的括號很容易找出開頭和結尾，所以電腦容易處理。

在飛輪趣這個題目中，每個路段可以視為一個括弧符號：如果下坡加速是左括號，上坡減速就是右括號。平地路段可當做一個空格，視需要填入左括號或右括號。一個課程要符合飛輪趣的規則，那麼左右括號就必須平衡才行。我們可以把課程 C 的路段寫成  $( ( ( ) ) )$ ，最後可以發展出： $( ( ( ) ) )$ ，或  $( ) ( ( ) )$ ，或  $( ) ( ) ( )$ ，這些式子中左右括號都是平衡的。

# 17. 新聞編輯

有 10 位學生擔任校刊編輯的工作，每個星期五他們會進行寫作或編修自己的文章。

在以下的工作時間表，紅色區塊為每位學生使用電腦進行編輯文章的時間。每台電腦的功能都是一樣的，且任何一個小時內，一次只能有一個學生使用某台電腦進行文章編輯工作。

|          |    | 時間   |      |       |       |       |       |       |
|----------|----|------|------|-------|-------|-------|-------|-------|
|          |    | 8:00 | 9:00 | 10:00 | 11:00 | 12:00 | 13:00 | 14:00 |
| 學生<br>編號 | 1  |      |      |       |       |       |       |       |
|          | 2  |      |      |       |       |       |       |       |
|          | 3  |      |      |       |       |       |       |       |
|          | 4  |      |      |       |       |       |       |       |
|          | 5  |      |      |       |       |       |       |       |
|          | 6  |      |      |       |       |       |       |       |
|          | 7  |      |      |       |       |       |       |       |
|          | 8  |      |      |       |       |       |       |       |
|          | 9  |      |      |       |       |       |       |       |
|          | 10 |      |      |       |       |       |       |       |

要讓這 10 位學生能根據上面的工作時間表使用電腦，請問最少需要幾台電腦？

- (A) 4  
(B) 5  
(C) 6  
(D) 10

## ✔ 正確答案：B

在 09:00 和 10:00 之間，5 名學生皆需要使用電腦，因此我們至少需要 5 台電腦才能解決問題。如果我們正確地安排，如下表所示，5 台電腦就足夠了。

|          |    | 時間   |      |       |       |       |       |       |
|----------|----|------|------|-------|-------|-------|-------|-------|
|          |    | 8:00 | 9:00 | 10:00 | 11:00 | 12:00 | 13:00 | 14:00 |
| 學生<br>編號 | 1  |      | PC3  | PC3   |       |       |       |       |
|          | 2  |      |      | PC1   | PC1   | PC1   | PC1   |       |
|          | 3  | PC1  | PC1  |       |       |       |       |       |
|          | 4  |      |      |       |       | PC3   | PC3   | PC3   |
|          | 5  |      | PC4  | PC4   |       |       |       |       |
|          | 6  |      |      |       | PC2   | PC2   |       |       |
|          | 7  |      |      | PC3   | PC3   | PC3   | PC3   | PC3   |
|          | 8  |      | PC3  |       |       |       |       |       |
|          | 9  | PC2  | PC2  | PC2   |       |       |       |       |
|          | 10 |      |      |       |       |       | PC2   | PC2   |

安排方式如下：每個時間有學生到達時，依學生編號從空著可使用的電腦中，安排一個編號最小的電腦給他，當學生完成工作後立刻空出電腦。

## i 在資訊科學上的意義

資源的排程及優化是現實生活中經常要處理的一個問題，例如旅館的房間預訂系統中，同一房間對應到兩筆訂房的分配時間不可重疊，而且房間的分配使用希望達到最佳利用率是非常重要的。此外，如何對多場演講安排會議室，也需要一個能節省會議室使用間數的最佳排程方法。

對這個問題，可以用一個圖型結構來幫忙分析。首先建立對各個學生以一個節點來表示，如果兩個學生需要使用電腦的時間有重疊，對應節點就建立一個編連結起來，稱為時間區間圖。要找出最少的電腦數，相當於是在這個圖的著色問題：對節點著色，但有相連邊的節點不能塗同一個顏色，最少需要多少種顏色。

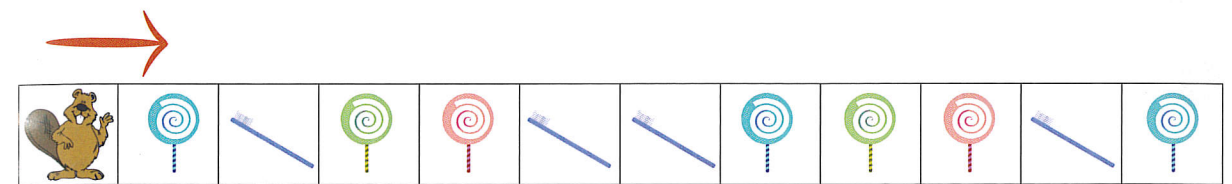
## 💡 關鍵字

排程問題、時間區間圖、圖著色問題

# 18. 棒棒糖和牙刷

小海狸到一條糖果通道，每隔一步的位置都放著一個棒棒糖或一支牙刷。這個通道只能往前不能後退，每走一步小海狸都必須根據那個位置放的東西，決定要吃糖果，刷牙或不做任何事繼續往前走，但不能把棒棒糖或牙刷帶著走。

小海狸想開心的沿著通道盡量吃糖果，但又想遵守牙齒健康守則：最多吃兩支棒棒糖後必須先刷牙，才能再吃下一支棒棒糖。



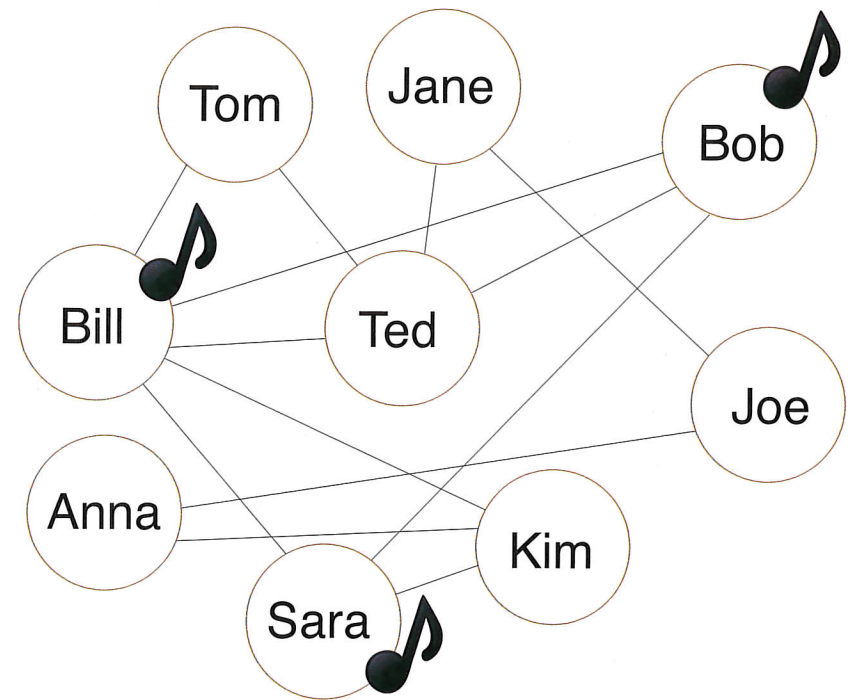
為了遵守牙齒健康守則，請問小海狸最多可以吃幾支棒棒糖？

- (A) 3 支
- (B) 5 支
- (C) 6 支
- (D) 7 支

# 19. 購買專輯

下圖表示班級中的朋友關係，若兩人間有連線，表示有朋友關係，若無連線則沒有朋友關係。

星期一有一個當紅的歌星發表了最新專輯，有些同學在首發當天就買了這張專輯，在下圖中這些同學的名字旁會有音符標示。



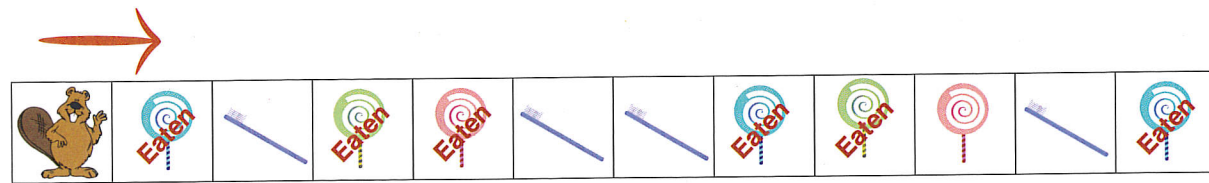
首發日後的每一天，每位尚未購買專輯的同學，若他的朋友中有一半或一半以上的人在前一天或之前已購買專輯，則該同學會受到鼓勵而在同一天也購買專輯。

請問最快到哪一天全班同學都會擁有這張專輯呢？

- (A) 星期六
- (B) 星期四
- (C) 星期三
- (D) 星期天

## ✔ 正確答案：C

小海狸最多吃兩支棒棒糖就要找到一支牙刷。若小海狸在遇到第一支牙刷時沒有刷牙，則在遇到第三支棒棒糖時就不能吃。為了吃最多支棒棒糖，遇到牙刷時除非前一部剛剛刷過，否則應該都要刷牙，接下來才能吃到較多的棒棒糖。所以他往前走每一步做的動作依次如下，在連續有 3 支棒棒糖的情況，第 3 支棒棒糖就不能吃，直到刷完牙，才能再吃最後一支棒棒糖，所以最多可吃 6 支棒棒糖。



## i 在資訊科學上的意義

在這個題目，海狸必須盡可能多吃棒棒糖，如果沒有限制，那就是一遇到棒棒糖就吃。但題目中加入了限制條件，所以要如何在符合限制條件下吃到最多棒棒糖，這就不是一個非常簡單的問題了。「限制條件」是一種規則，它會強迫你去做某些事或不做某些事。例如，當你玩遊戲時，你需要遵守與遊戲相關的所有規則。每種遊戲都有自己的規則，有些規則很簡單，但有時也很會複雜。

在資訊科學中，限制條件是在解決問題時一個非常重要的部分，它們可能會讓問題難以解

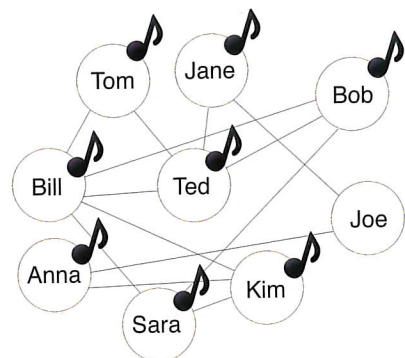
決。所以為了解決問題，弄清楚限制條件是非常重要的。而資訊科學要解決的問題中，有很多問題是要找出目標值能達到最大或最小的答案，例如這裡是要吃最多的棒棒糖，有的目標值是要做最多的工作，或是花最少的成本等等，稱為找最佳解。找答案的方法可透過列出所有的可能性，一一計算目標值找出最佳解，但這樣要花很多計算時間。如果能分析出限制條件跟目標值的關係，就能訂出一個策略，較快速的找到最佳解。

## 💡 關鍵字

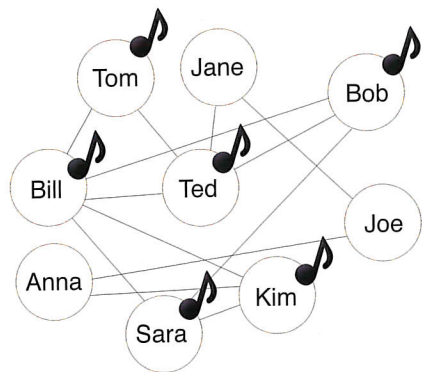
限制條件、最佳解

## ✔ 正確答案：B

因為在星期一，Tom 的 2 個朋友有 1 個朋友 (Bill) 已買專輯，Ted 的 4 個朋友有 2 個朋友 (Bill 跟 Bob) 已買專輯，Kim 的 3 個朋友有 2 個朋友 (Bill 跟 Sara) 已買專輯，所以在星期二 Tom，Ted 和 Kim 會買下專輯，已買專輯的人標示如下圖。



因為在星期二，Ana 的 2 個朋友有 1 個朋友 (Kim) 已買專輯，Jane 的 2 個朋友有 1 個朋友 (Ted) 已買專輯，所以在星期三 Ana 和 Jane 會買下專輯，已買專輯的人標示如上圖。



因為在星期三，Joe 的 2 個朋友 (Anna 跟 Jane) 都已買專輯，所以在星期四 Joe 會買下專輯，這時候全班同學都擁有這張專輯。

## i 在資訊科學上的意義

現在有許多社群網站，這些網站上使用者的互動關係形成社交網絡。這個題目是透過社交網絡傳播影響的概念，這裡採用群體行為的門檻模型來表示傳播擴散影響，也就是說如果一個人的朋友中達到一部分比例有某個想法或意見，這個人很可能也會隨著這些朋友的影響而改變主意。這種社交網絡可以用一個雙向的圖型結構來表示，在這問題中可以學到傳播擴散的模擬。資訊或影響的傳播是漸進的，經常用

### 💡 關鍵字

圖型結構、社交網絡、網絡傳播

來建立市場網絡中使用者行為（例如購買某個商品）的擴散模型。有的模型還會考慮朋友之間有不同的影響力，就像好朋友的影響通常比泛泛之交的影響大，所以對連接邊設定不同的比重值。一個社交網絡中，也可以分析網絡中的成員是不是彼此緊密相連，用來找出互動密切的社群 (community)，或是可以找出最快影響到最多人的關鍵人物。

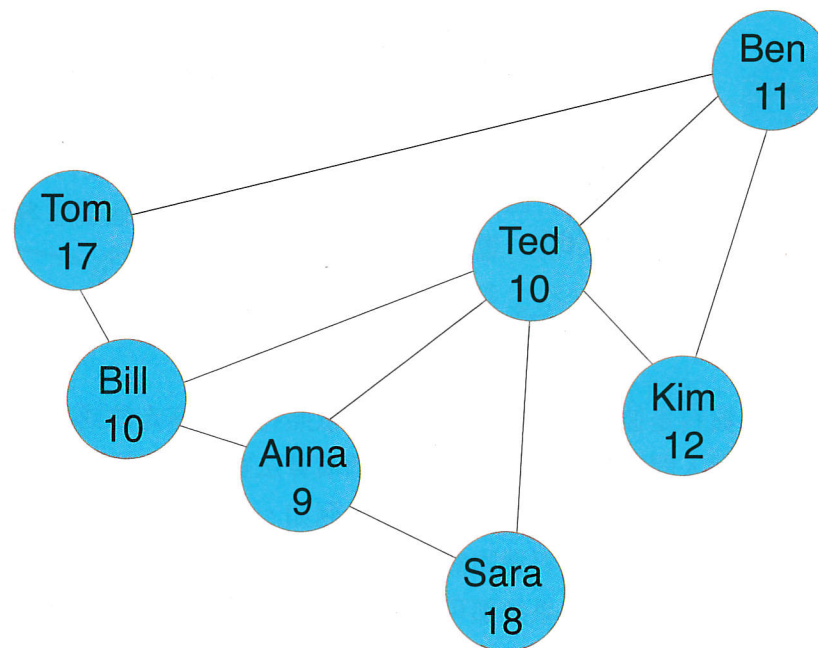
## 20. 書籍分享俱樂部

一群朋友參加了書籍分享俱樂部。俱樂部對成員有一些規定：

當你看完一本未讀過的書，或是收到別人交還你傳給他讀的書時，請依下面規則處理：

- ▶ 把它傳給你未讀過那本書的朋友中，年齡最小的那一位，請他看完。
- ▶ 若你所有的朋友都看過這本書了，就把它交還給當初給你書的那位朋友。

下圖顯示俱樂部中 7 名學生的英文名字及年齡。他們之間的朋友關係以連線表示。



現在 Ben 看了一本沒有其他人讀過的新書，想和他的朋友分享。請問誰將是最後一個讀這本書的人？

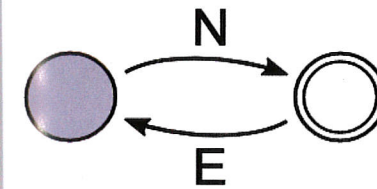
- (A) Tom
- (B) Sara
- (C) Bill
- (D) Kim

## 21. 機器人行走路徑

最新款的海狸機器人被設計成可接受一連串指令之後移動，指令由 N 和 E 組成而成，N 代表機器人往北移動一步，而 E 代表機器人往東移動一步。

舉例來說，指令 NNEEE 會讓機器人向北移動兩步後再向東移動三步。要產生一組合法的指令必須使用一個圖形化的指令產生器，指令產生器的說明如下：

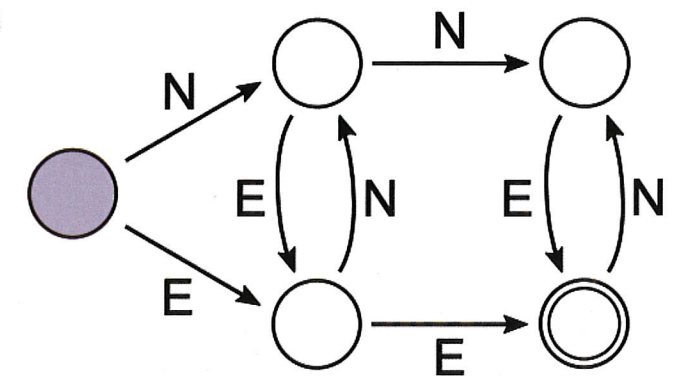
- ▶ 深色圓形代表產生指令的起始位置。
- ▶ 雙圈圓形代表指令產生完畢。
- ▶ 箭號表示由某一個位置移動到另一個位置，並產生箭號上的指令。
- ▶ 如果機器人走到了雙圈圓形的位置，但仍有箭號可繼續產生指令時，則可以繼續產生指令。



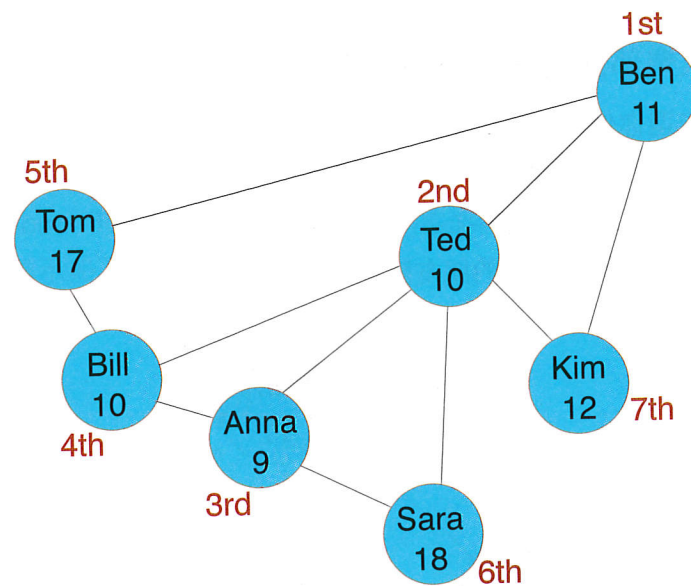
左圖的指令產生器為例，一開始從深色圓形開始，依向右箭號移動到雙圈圓形位置（此時產生指令 N），接著可以選擇依向左箭號移動（此時產生指令 E），或是停止在雙圈圓形位置。因此，N、NEN 或是 NENENENENENEN 都是合法的指令，但是像 NENNE、NENE 或 ENE 則否。

海狸機器人現在使用了一組新的指令產生器（如右圖），請問下列敘述何者是正確的？

- (A) 有些指令以 N 收尾  
 (B) ENNE 是不合法的指令  
 (C) ENEENE 是不合法的指令  
 (D) NEENENNE 是不合法的指令



✔ 正確答案：D



根據規則：Ben 讀完傳給 Ted，Ted 讀完傳給 Anna，Anna 讀完傳給 Bill，Bill 讀完傳給 Tom。Tom 讀完後他的朋友都看過這本書了，所以還給 Bill，Bill 又還給 Anna。Anna 傳給沒看過的 Sara，Sara 讀完後他的朋友都看過這本書了，所以還給 Anna，Anna 還給 Ted，Ted 傳給沒看過的 Kim。Kim 讀完後他的朋友都看過這本書了，所以還給 Ted，Ted 最後再還給 Ben。因此，Kim 是最後一個讀這本書的人。

### i 在資訊科學上的意義

很多資訊科學家處理的資料是彼此有關聯的，例如社交網絡可視為人與人間有友誼關係的關聯，道路網可視為多個城市間有道路連接的關聯，食物鏈可視為生物之間有捕食者與獵物關係的關聯。因此，現實世界中的很多系統都可以想作一種網絡。為了呈現網絡架構，資訊科學家用圖型的資料結構來表示資料間的連結，並提出多個演算法來分析圖型結構中的各種特性。其中一個著名的演算法為深度優先搜索，

用來依照特定順序走訪或搜尋一個圖型結構。這個演算法對於了解一個圖型結構的特性很有用，例如：一個圖型結構包含有多少個相連通子圖？（一個相連通子圖裡的任兩個節點可以找到一連串連結關係讓彼此相連）。這個題目根據會員的朋友網絡架構傳送書本，書本流通的傳送順序規則相當於深度優先搜索的順序，最後每個人都有讀過書，表示者這個朋友網絡架構是相連通的。

### 💡 關鍵字

社交網絡、圖型結構、深度優先搜尋

## 22. 文字編輯 app

海狸 Ada 發現有一款新的聊天軟體，可以透過特殊符號傳送有特色的文字，特殊符號的用法如下所示：

- ▶ \* 表示粗體  
例如：傳送 `*beaver*` 會顯示 **beaver**
- ▶ / 表示斜體  
例子：傳送 `/beaver/` 會顯示 *beaver*
- ▶ \_ 表示底線  
例子：傳送 `_beaver_` 會顯示 beaver
- ▶ \$ 表示強調  
例子：傳送 `$beaver$` 會顯示 **beaver**

當然，也可以使用組合的文字功能，例如：

- ▶ 傳送 `_beaver/` 會顯示 *beaver*
- ▶ 傳送 `/_beaver/` 會顯示 *beaver*
- ▶ 傳送 `$_beaver_$` 會顯示 **beaver**

當然，也可以使用組合的文字功能，例如：

- ▶ 傳送 `:birthday:` 會顯示 🎂
- ▶ 傳送 `:island:` 會顯示 🌴
- ▶ 傳送 `:house:` 會顯示 🏠

請問 Ada 要如何利用這個聊天軟體發送下面的訊息？

You are invited to a super 🎂 party at the 🏠 on the 🌴 with **great music and food**. See you there!

- (A) \*You\* are invited to a `_super_` birthday party at `*/the/*` :house: on the :island: with `$/great music and food/$`. See you `/there!`
- (B) \*You\* are invited to a `_super_` :birthday: party at `*_the_*` :house: on the :island: with `$/great music and food/$`. See you `_there!`
- (C) \*You\* are invited to a `_super_` :birthday: party at `*/the/*` :house: on the island with `*$great music and food*$`. See you `_there!`
- (D) \*You\* are invited to a `/super/` :birthday: party at `*_the_*` house on the :island: with `$great music and food$`. See you `/there!`

### ✔ 正確答案：D

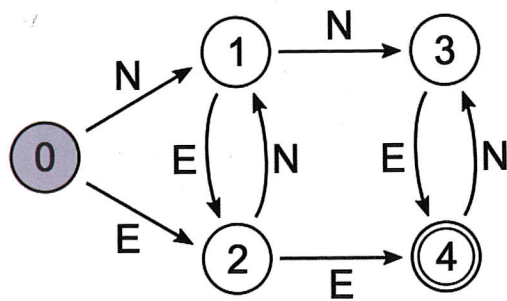
- 因所有合法指令都會以 E 結尾，所以沒有指令會以 N 結尾。
- 只要從起點開始，沿著下、上、右、下，就可以產生合法指令 ENNE。
- 若從起點沿著下、上、下、右、上、

- 下，就可以產生合法指令 ENEENE。
- 從起點開始，沿著上、下、右來到雙圈圓形位置後，我們可以發現接下來的指令必定是 N、E 交錯出現，所以出現 NN 的確是不合法的指令。

### i 在資訊科學上的意義

此題為有限狀態機 (finite-state machine 或是 finite-state automata)，在資訊領域中常用來定義某個系統，可以由有限的狀態所組成、以及如何在這些狀態之間的轉移和動作等行為的模型。通常此系統都會明確的定義初始狀態以及終止狀態。狀態機由不同的狀態所構成，通常都以圓形與標號來示意不同的狀態，例如起始狀態以填滿的圓形表示，終止狀態則用雙圈來區別，箭號則說明了狀態之間的轉移。例如下方的狀態機，不同的狀態以數字說明，移動路徑則有 N 與 E 兩種方向。

總結以上說明，此狀態機有以下 5 種狀態。



### 💡 關鍵字

移動路徑、合法路徑、路徑方向

## ✔ 正確答案：B

我們可以把題目的特色文字和對應的特殊符合整理如下：

**You** 要使用 \*You\*

**Super** 要使用 \_super\_

 要使用 :birthday:

**the** 要使用 \*\_the\_\* 或 \*\_the\*\_

 要使用 :house:

 要使用 :island:

**great music and food** 要使用 `$/great music and food/$` 或 `/$ great music and food$`

**there!** 要使用 \_there!\_

## i 在資訊科學上的意義

海狸 Ada 所使用的文字可視為一種標記式語言 (markup language)，以純文字及標註式的標籤所構成。常見的文字編輯器即是使用標記式的方式去改變文字型態，例如，在欲標示的文字開頭與文字結尾用特殊符號去框架它。目前被大量使用的標記式語言有網路上的 HTML，它構成了網頁的基本格式。

## 💡 關鍵字

特色文字：經由標註過後的文字

特殊符號：用來框架文字的標記符號

# 23. 還書整理

小傑和小珊在學校圖書館工作，館內的藏書都放在一個長度為一公尺的書架上。

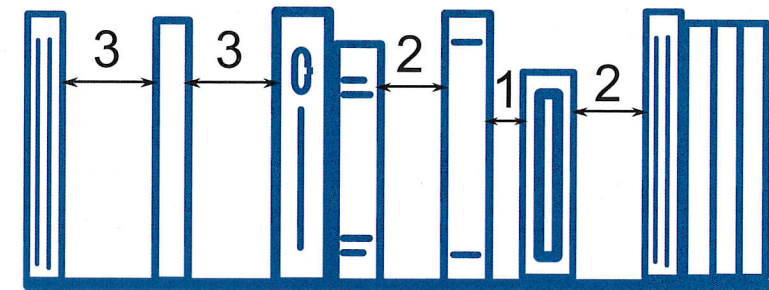
當一本書被借走時，書架上會空出這本書厚度的位子。

每天，所有的還書都會先堆放在桌子上。在圖書館閉館前，小傑和小珊會從還書堆中最上面，逐一把書直立放回書架的空位上。但是他們兩人選擇空位的方法不同：




▶ 小傑先拿還書堆中最上面的一本，從左到右檢查書架上的空位，把書本放進第一個足夠大的空位。每一次他都拿最上面的書做一樣的事情，直到他擺放完所有的還書或遇到一本他找不到空位可放的書才停止。

▶ 小珊將每本書都先找當時最大的空位靠左對齊擺放，如果最大空位有兩個時，就選擇左邊的空位擺放。直到所有的還書被歸位或遇到一本她無法放置的書籍時，她才會停止。

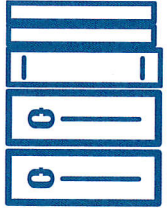
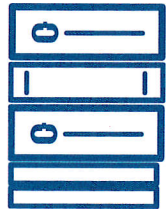
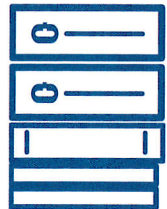
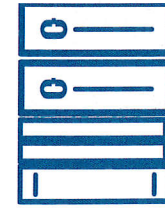
有一天閉館前，書架的擺設狀況如下圖，空位的寬度是以公分表示。



歸還的書有下面三種不同的大小：

| 厚度 (公分) | 1   | 2   | 3   |
|---------|---|---|---|
| 書籍種類    |  |  |  |

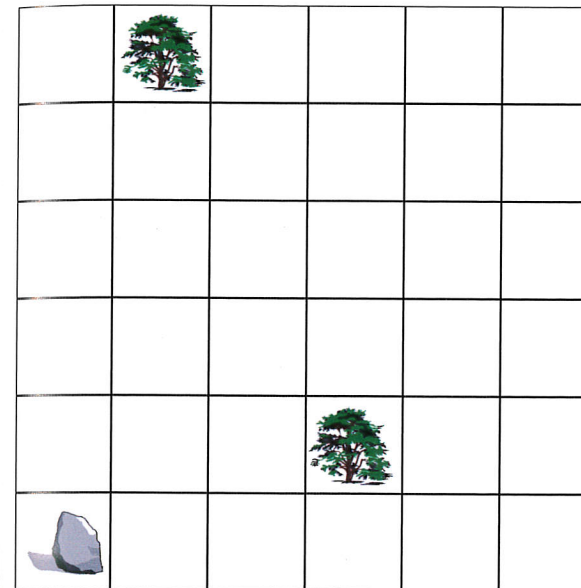
如果歸還的書堆有下圖幾種不同的堆放順序，哪個書堆用小傑的方法比小珊的方法能把更多的書歸位？

| (A)   | (B)   | (C)   | (D)   |
|---|---|---|---|
|  |  |  |  |

# 24. 迷路的小海狸

一隻小海狸在森林裡和媽媽走失了。

把小海狸走失的區域用以下 6 乘 6 的方格圖表示，這個區域內有兩棵樹和一塊石頭，對應位置如下圖所示



- ▶ 小海狸能走到上下左右的相鄰方格，但是不能直接走到斜對角（對角線）的方格，兩個方格之間的距離，是指從一個方格走到另一方格中間經過的最少方格數。例如：石頭到距它較近那棵樹的距離是 4。
- ▶ 小海狸位於其中一個方格內，可以看到距離 4 以內方格中的物體。

海狸媽媽透過手機問小海狸：「你可以看到多少石頭和樹呢？」

小海狸答：「我只看到兩棵樹，沒看到石頭。」

海狸媽媽問：「你距離兩棵樹分別有多遠呢？」

小海狸答：「我和其中一棵樹的距離為 2，和另一棵樹的距離為 4。」

請問迷路的小海狸在方格圖中的可能位置有幾個？

- (A) 3
- (B) 4
- (C) 5
- (D) 6

## ✔ 正確答案：D

下表分別顯示選項的四疊書，小傑和小珊可擺放的狀況。選項 D 中小傑可以放置 5 本書，小珊只能放 4 本書。

|     | 小傑 | 小珊 |
|-----|----|----|
| (A) |    |    |
| (B) |    |    |
| (C) |    |    |
| (D) |    |    |

## i 在資訊科學上的意義

記憶體是電腦的重要組成部分，雖然記憶體容量不斷提高，但記憶體仍是寶貴資源，如何提高記憶體利用率，並對記憶體內資料實現有效保護是記憶體管理的主要任務。電腦能夠將多個程式載入在記憶體中，以便同時執行它們。需要執行的程式會需要不同的記憶體空間，電腦的作業系統會選取一個程式將其載入到可使用的連續記憶體空間中，正類似於將歸還書區

中的書放到書架上。小傑和小珊採用的方法相當於兩種記憶體管理策略，分別稱為「最先符合法」(first-fit) 和「最差符合法」(worst-fit)。

值得注意的是，當應用於記憶體請求時，要執行的程式相當於「歸還的書堆」，不是提前能知道的，因此決定哪種策略較好並不容易。

## 💡 關鍵字

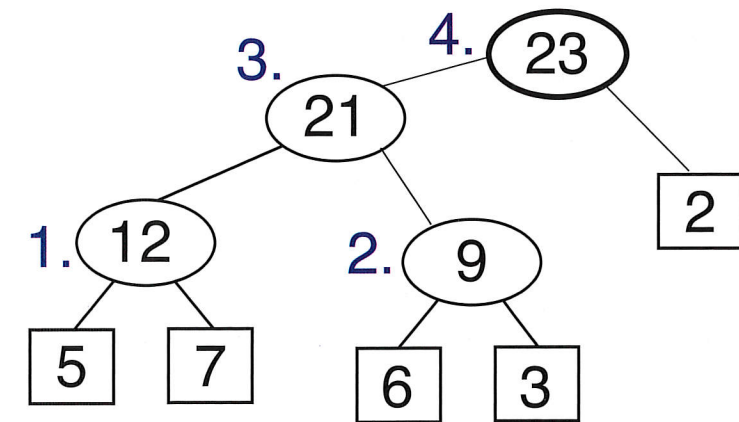
最先符合法、最差符合法

# 25. 熔合金砂

海狸村莊的河裡蘊含著大量的金砂，海狸們會在河裡淘金，並且請工匠將金砂熔合成一個更大的金砂。

工匠每次只能熔合兩顆金砂，而且會依據熔合後金砂的重量收取費用，收費標準是1公克1塊錢。舉例來說，假設手上有5顆金砂重量分別為5公克、7公克、6公克、3公克和2公克。如果依照以下的步驟熔合金砂：

- ▶ 1. 將5公克和7公克的金砂熔合成12公克的金砂，需花費12元。
- ▶ 2. 將6公克和3公克的金砂熔合成9公克的金砂，需花費9元。
- ▶ 3. 將12公克（步驟1的結果）和9公克（步驟2的結果）的金砂熔合成21公克的金砂，需花費21元。
- ▶ 4. 將21公克（步驟3的結果）和2公克的金砂熔合成23公克的金砂，需花費23元。

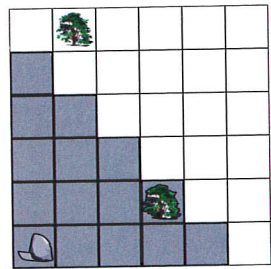


總共花費  $12 + 9 + 21 + 23 = 65$  元。但同樣的這5顆金砂如果按照另一種順序，（5公克 + 7公克 = 12公克，3公克 + 2公克 = 5公克，12公克 + 5公克 = 17公克，17公克 + 6公克 = 23公克）就只要  $12 + 5 + 17 + 23 = 57$  元。

假設手上目前有8顆金砂，重量分別是7公克、1公克、3公克、2公克、6公克、2公克、5公克、4公克。請問要將這8顆金砂熔合成1顆金砂，所需要的最少花費是多少？

- (A) 80元
- (B) 85元
- (C) 90元
- (D) 95元

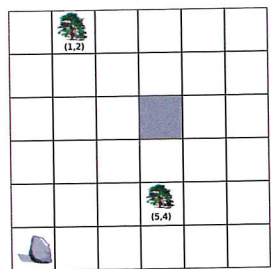
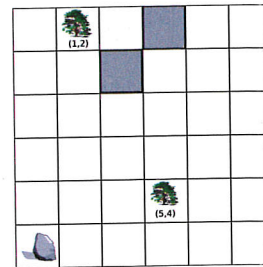
## ✔ 正確答案：A



小海狸告訴海狸媽媽他看不到任何石頭，這表示小海狸和石頭之間的距離超過4。因此，小海狸不可能位於灰色塊中，如下圖所示。

小海狸告訴海狸媽媽他和一棵樹之間的距離是4，而另一棵樹的距離是2，所以我們應該考慮與兩棵樹（座標位置(1,2)及(4,4)）的距離分別是4及2，或是2及4的狀況。

下圖中的2個灰色方塊皆與座標位置(1,2)的樹距離4，與座標位置(4,4)的樹距離2，且與石頭間的距離超過4以上。



下圖中的灰色方塊與座標位置(1,2)的樹距離2，與座標位置(4,4)的樹距離4，且與石頭間的距離超過4以上。

因此小海狸可能的位置有3個。

## i 在資訊科學上的意義

在資訊科學中，資料的位置經常用網格來表示，而本題中所定義的距離稱為曼哈頓距離，也就是兩個網格的水平差距及垂直差距相加得到的距離。曼哈頓距離的命名是從規劃為方型建築區塊的城市（如曼哈頓）中，路口間最短的行程路徑而來。

為了找出小海狸的位置，我們必須找到符合所

有條件的可能位置。由小海狸的給定條件（與每個樹之間的距離、看不見石頭的位置），可以用來縮小可能位置的範圍。資訊科學要解決的問題，經常會從所有的答案中（稱為搜尋空間），根據所給予的條件來篩選需要考慮的可能答案，以減少找到最後答案需要進行的計算。

## 💡 關鍵字

曼哈頓距離、搜尋空間

## 26. 文字加密遊戲

海狸小姐的興趣是玩文字遊戲，她從報紙上得到了啟發，發明了一種字母表示式如下所示

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| A | B | C | J | K | L | S | T | U |
| D | E | F | M | N | O | V | W | X |
| G | H | I | P | Q | R | Y | Z |   |

例如，單字 BEAVER 可以被編碼成



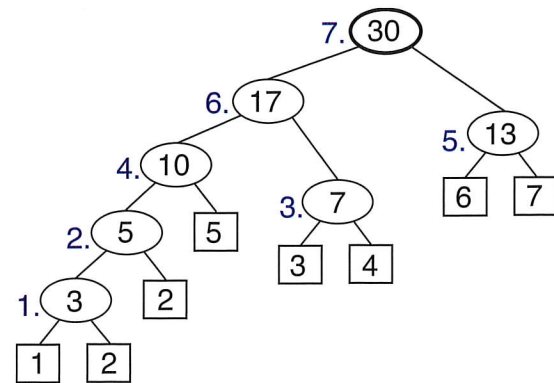
請觀察單字和編碼的規則，回答以下一串符號是哪一個單字加密而成的？



### ✔ 正確答案：B

如果按照下圖藍色數字順序熔合金砂，我們可以只用 85 元就將所有的金砂熔合在一起。下圖中的方塊代表一開始的金砂，方塊裡的數字代表金砂的重量；圖中橢圓的部分代表熔合過程中產生的金砂，橢圓中的數字代表熔合後的重量（也相當於該次熔合的費用）。

那麼，這樣的順序是怎麼找出來的呢？首先，要熔合八顆金砂一定只需要七次，而在這七次中，先熔合的金砂它的重量（相當於費用），會累積到後面的熔合費用中。根據以上的觀察，我們可以知道最佳的順序是：在每一次熔合中，挑選目前所有金砂中重量最小的兩個金砂熔合。



### i 在資訊科學上的意義

在日常生活中，我們常常遇到需要尋找一個問題的最佳解，例如要規畫一個五個國家的旅遊時，每個國家彼此間有不同的交通費，我們就會想要找出一個最佳順序，使得交通費總和花費最少。要解決這樣的問題有很多方法，其中一個方法稱為貪心演算法 (Greedy algorithm)，這個方法是說：在解決問題的過程中，每次都挑選當下最有利的選項，很多問題可以透過貪心演算法得到最佳解，像是本題

與霍夫曼編碼。以旅遊的例子來說，就是先挑選從台灣出發花費最少的國家，然後再從該國家出發，挑選下一個花費最少而且還沒去的國家。不過要注意的是，對有些問題來說貪心演算法得到的結果並不一定是最佳解，在上述的例子中，如果第一個挑選的國家從台灣出發費用最低，但從該國家出發到其他四個國家的交通費都是超級貴的話，那很有可能一開始去別的国家會有更划算的順序。

### 💡 關鍵字

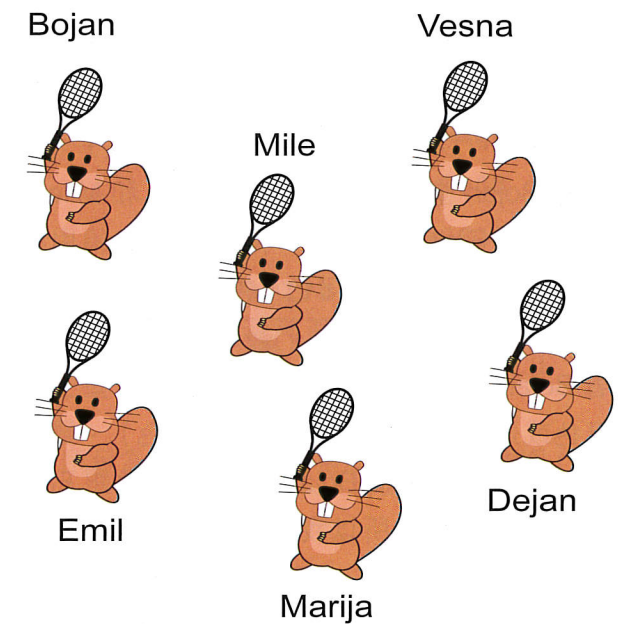
貪心演算法、最佳解、霍夫曼編碼

## 27. 網球賽

有六隻海狸 (Bojan, Vesna, Mile, Emil, Dejan, 和 Marija) 參加了海狸杯網球賽，在賽程安排完後，主辦單位遇到了一個棘手的狀況：

雖然主辦單位會提供選手球拍，但每位選手在自己的比賽中都想用固定的球拍，可是在經費有限的情況下，無法支付每位選手一隻球拍。幸好比賽場地只有一個，不會有兩場比賽同時進行，所以只要妥善分配球拍，是用少於六隻球拍來完成所有比賽的。

|        |    |       |
|--------|----|-------|
| Bojan  | vs | Emil  |
| Emil   | vs | Mile  |
| Mile   | vs | Vesna |
| Vesna  | vs | Bojan |
| Dejan  | vs | Vesna |
| Dejan  | vs | Mile  |
| Marija | vs | Emil  |
| Bojan  | vs | Mile  |



在比賽期間主辦單位最少需要準備幾支球拍提供給六位參賽者呢？

- (A) 2  
(B) 3  
(C) 4  
(D) 5

### ✔ 正確答案：CREATIVE

仔細觀察 BEAVER 和編碼的結果可以發現，此符號表對應的方式，即是缺口方向與各個英文字母的之間的關係。黑點則是各自對應到不同的表格。

### i 在資訊科學上的意義

在本題中所使用的加密方式為替換式加密法 (substitution cipher) 或是取代式加密法，簡單來說，就是將原文中的字母或符號，用另一種對應的方式取代。沒有對應的密碼對照 (cipher key) 其實很難將加密過後的資料還原。在資訊領域中，另一個常使用的加密方法是「公開金鑰加密」(Public-key cryptography)，這個方法需要兩個金鑰，運作的原理是使用其中一把金鑰加密的資料，只能用另一個金鑰才能解密，因此其中一個金鑰可以公開，稱為公開金鑰，而另一個金鑰就不公開，留做解開加密的文件，稱為私有金鑰。

### 💡 關鍵字

加密 (encryption)、編碼 (encoding)、字碼 (code)

## 28. 國旗

一個只有黑白兩色的圖形在電腦上可以用一個填滿數字的長方形表格來表示，最簡單的數字表示方法為 1 代表白色、0 代表黑色

像圖 1 就是某個只有黑白兩色國旗的表格。

|        |     |        |        |
|--------|-----|--------|--------|
| 101111 |     | 101010 | 111111 |
| 010110 |     | 010101 | 111110 |
| 111111 |     | 101010 | 111100 |
| 010110 |     | 010101 | 111000 |
| 101111 |     | 101010 | 110000 |
| 圖 1    | 圖 2 | 圖 3    | 圖 4    |

在表格中，每個 0 或 1 的位置可以用坐標表示，如圖 2 所示為 6 行 6 列的表格和對應的坐標圖（X、Y 坐標皆由 0 開始）。

假設現在有兩個國旗的表格：棋盤狀國旗如圖 3、三角形國旗如圖 4。

小狸從這兩個國旗選一個出來，告訴你在這個國旗的表格中，坐標 (1, 4) 上的數字為 1，請問他選的是哪種國旗？

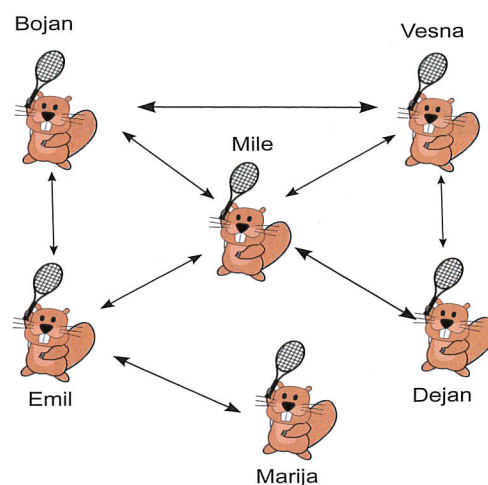
- (A) 棋盤狀國旗
- (B) 三角形國旗
- (C) 兩者皆有可能
- (D) 兩者皆非

### ✔ 正確答案：B

三支球拍。整個賽程可以畫成下面的圖。每一條連接線表示某場比賽會遭遇的兩位選手。圖中清楚顯示會進行比賽中的兩位選手無法共用一把球拍。

要解決這個問題我們可以把相互之間沒有連接線的海狸選手塗上相同的顏色，也就是說兩隻海狸在不同比賽中可以共用球拍。所以用最少的顏色來完成著色也代表我們可以用最少的球拍來完成全部賽程。

首先我們可以任意選擇第一個顏色和選手，比方把 Bojan 塗上紅色。因為 Bojan 和 Mile、Vesna 和 Emil 有連接線，所以我們必須用紅色以外的其他顏色來著色。接者，我們可以把 Mile 塗上藍色。這時我們發現 Emil 同時連接著 Bojan 和 Mile，所以勢必要把 Emil 塗上非紅色或是藍色的另一種顏色（如綠色）。雖然 Vesna 也同時連接 Bojan 和 Mile 但他跟 Emil 之間沒有連接，所以我們可以把 Vesna 塗上跟 Emil 一樣的綠色。再回頭看看 Dejan，他僅與 Mile 和 Vesna 有相連，所以應該塗上非藍色或綠色，也就是跟 Bojan 一樣的紅色。最後 Marija 只有跟 Emil 有相連，所以只要選擇非



綠色的顏色（紅色或藍色）即可。

既然我們只用了三個顏色完成著色，也就是說我們可以只用三支球拍來完成所有比賽，其中 Bojan 和 Dejan 共用一支（紅色），Emil 和 Vesna 共用一支（綠色），Mile 用一支（藍色），而 Marija 可選擇和 Bojan 或 Dejan 共用或是和 Mile 共用一支。那我們可能只用兩支球拍完成賽程嗎？答案是不可能，因為有其中一個狀況是 Bojan、Mile 和 Emil 的賽程，他們三位選手必須用三支不同的球拍才行，所以三支球拍已經是最少的選擇。

### i 在資訊科學上的意義

許多現實生活中的問題都可以用「圖」這個資料結構來表示。一個「圖」包含了一組節點（表示成一個小圈圈）和一組連接線（表示成直線或是曲線），連接線的兩端一定是節點。再者，許多現實生活問題的方式則可以利用在「圖」中著上不同的顏色來找出解答。此題就是一個很好的範例。在資訊領域中，類似這樣的問題通稱為「著色問題」。

### 💡 關鍵字

圖、著色問題

## ✔ 正確答案：B

如果我們將棋盤狀國旗和三角形國旗在坐標 (1,4) 的元素圈起來，可以發現棋盤狀國旗坐標 (1,4) 的元素為 0，三角形國旗坐標 (1,4) 的元素為 1，所以答案為 (B) 三角形國旗。

| 棋盤狀國旗表示法 |     |   |   |   |   |   | 三角形國旗表示法 |     |   |   |   |   |   |
|----------|-----|---|---|---|---|---|----------|-----|---|---|---|---|---|
|          | X → |   |   |   |   |   |          | X → |   |   |   |   |   |
| Y ↓      | 0   | 1 | 2 | 3 | 4 | 5 | Y ↓      | 0   | 1 | 2 | 3 | 4 | 5 |
| 0        | 1   | 0 | 1 | 0 | 1 | 0 | 0        | 1   | 1 | 1 | 1 | 1 | 1 |
| 1        | 0   | 1 | 0 | 1 | 0 | 1 | 1        | 1   | 1 | 1 | 1 | 1 | 0 |
| 2        | 1   | 0 | 1 | 0 | 1 | 0 | 2        | 1   | 1 | 1 | 1 | 0 | 0 |
| 3        | 0   | 1 | 0 | 1 | 0 | 1 | 3        | 1   | 1 | 1 | 0 | 0 | 0 |
| 4        | 1   | 0 | 1 | 0 | 1 | 0 | 4        | 1   | 1 | 0 | 0 | 0 | 0 |
| 5        | 0   | 1 | 0 | 1 | 0 | 1 | 5        | 1   | 0 | 0 | 0 | 0 | 0 |

坐標(1,4)的元素為 0

坐標(1,4)的元素為 1

## i 在資訊科學上的意義

資料表示法在資訊科學上是很基礎的議題，它用來表示儲存在電腦上的各類資料，如：數值、文字、圖片、聲音、影片等。不同類型的資料採用不同的資料表示法，如數值資料（整數）採用 2 的補數表示法、文字資料（英數、符號）採用 ASCII 碼。

而圖形可以用點陣式儲存在電腦上，其方法為直接將整個圖形分割成為二維的格點（我們稱之為“pixel”像素），再去儲存每一個點的資

訊。單色圖形每個像素只要一個位元便可表示（如本題所述 1 為白、0 為黑），灰階圖形每個像素則需 8 個位元表示（依明暗度分成 256 個層），彩色圖形的每個像素則由紅色、綠色、藍色等三原色 (RGB) 所構成，每個顏色各 8 位元。

圖形表示法通常用矩陣來呈現，矩陣中的每個元素代表一個像素，元素的位置以坐標 (x, y) 來表示，左上角的坐標為 (0, 0)。

## 💡 關鍵字

圖形表示法、矩陣、矩陣元素

# 29. 完美洗牌

法拉有一副海狸牌共八張，一開始法拉先將海狸牌由大到小排列後，再依照顏色分類（如下圖的起始狀態）。

接著法拉正在練習一種特殊的洗牌方式稱之為完美洗牌，完美洗牌包含下圖的兩個步驟：

|             |  |                       |                |
|-------------|--|-----------------------|----------------|
|             |  |                       |                |
| 起始狀態<br>洗牌前 | 步驟一<br>在不改變牌的順序下，將牌等分成兩份。上面四張牌為第一組，下面四張牌為第二組 | 步驟二<br>將第一組和第二組的牌交錯混合 | 最終結果<br>完成一輪洗牌 |

經過幾輪的洗牌，法拉發現牌竟然回到一開始的順序了，請問從一開始起始狀態算起，法拉總共洗了幾次牌？

- (A) 2 次完美洗牌
- (B) 3 次完美洗牌
- (C) 4 次完美洗牌
- (D) 5 次完美洗牌

## 30. 帽子與腳踏車

海狸班的同學列隊成一排（如下圖），在隊伍中還放了四個標記：“Hat on”、“Hat off”、“Get on”、“Get off”。

只有在“Hat on”和“Hat off”標記中間的海狸可以借到帽子，另外只有在“Get on”和“Get off”標記中間的海狸可以借到腳踏車。



有一些海狸只能借到帽子或腳踏車其中一樣東西而已。請你算算看，這樣的海狸一共有幾隻？（請僅填入數字。作答範例：1）

### ✔ 正確答案：B

答案是 (B) 經過三輪的完美洗牌，牌堆就會回到起始狀態

第一輪洗牌後的結果，會變成題目中所繪的最終結果

第二輪洗牌後，由上到下依序會是：

紅 5，紅 3，黑 5，黑 3，紅 4，紅 2，黑 4，黑 2

第三輪洗牌後，由上到下依序會是：

紅 5，紅 4，紅 3，紅 2，黑 5，黑 4，黑 3，黑 2

### i 在資訊科學上的意義

完美洗牌又稱為法羅洗牌 (Faro shuffle)，是一種魔術師和撲克牌玩家都經常使用的技法。事實上，在 1860 年出版的一本書「揭露博弈中的科學 (A Grand Expose of the Science of Gambling)」中，就提到如何使用完美洗牌在賭局中作弊。

此題詳細的描述了完美洗牌的步驟，每個步驟

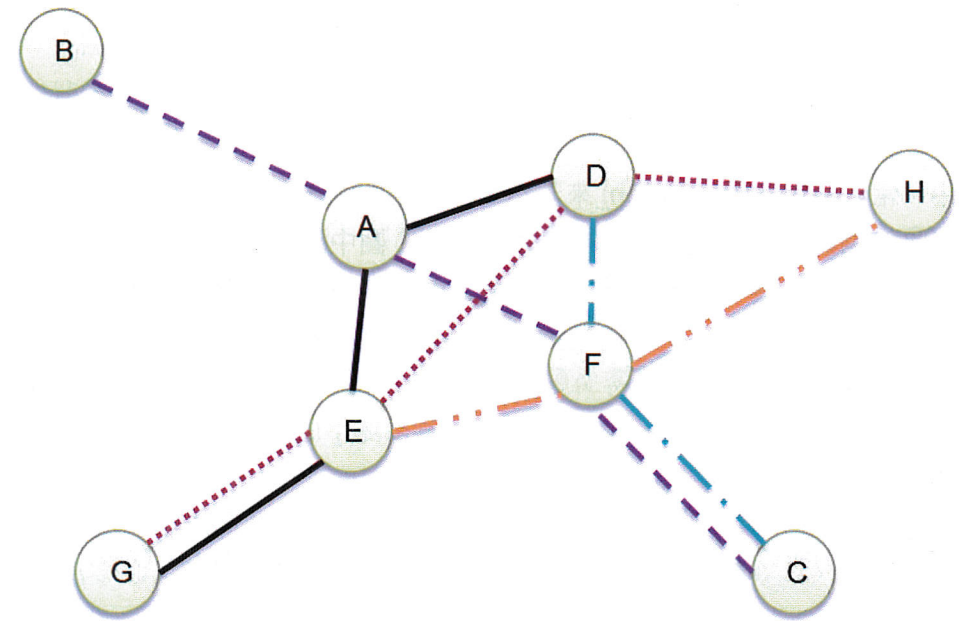
都必須正確的執行，才能得到最終的結果。在資訊科學中，演算法 (algorithm) 是用以解決特定問題的有限步驟和敘述，如果將這些演算法轉換成對應的程式，並在電腦上執行，我們會發現每次執行結果都會一樣，無論是網路通訊、密碼學或平行運算等各種領域中，演算法都擁有相當重要的地位。

### 💡 關鍵字

演算法 (algorithm)

# 31. 降低鐵路營運成本

海狸島上的鐵道系統，包含了 8 個火車站 (A、B、C、D、E、F、G、H) 以及 5 條線路 (紅線、紫線、黑線、橘線、藍線)，如下圖所示。



這個系統的最大特色，即是從任意一個車站，到任意其他車站，最多只需轉乘一次。舉例來說，若要從 B 車站出發前往 H 車站，這時只需搭乘至 F 車站轉乘，就能到達目的地。

由於鐵路公司想降低營運成本，因此要停駛部分路線，但前提是仍要滿足任兩站最多只需轉乘一次。請問鐵路公司最多能夠停駛幾條路線？

- (A) 1 條
- (B) 2 條
- (C) 3 條
- (D) 4 條

## ✔ 正確答案：4

我們只要計算

1. 從原英文片段的第一個字母 1. 位於 [Hat on] 和 [Get on] 之間的海狸數量。
2. 位於 [Hat off] 和 [Get off] 之間的海狸數量。

最後加總就是答案。

要注意這一排海狸中，位於 [Get on] 和 [Hat off] 的這 3 隻同時有帽子和腳踏車，所以不能算在答案裡面。

## i 在資訊科學上的意義

在邏輯的世界中，一個邏輯敘述的結果就會是「真(true)」或「假(false)」，例如：「0 是偶數」為真，「 $1 + 1 = 3$ 」為假。在電腦中，「真」和「假」常常分別用 0 與 1 表示。對於較複雜的邏輯敘述，可以透過邏輯運算連結起來，常見的像是「且 (and)」、「或 (or)」與「互斥或 (exclusive-or)」。「互斥或」的意義是只有

當兩個敘述其中一個為真時，運算後的結果為真，其餘都為假。以本題的題目為例，假設我們將海狸由左向右依序編號，並且將可以借到帽子與可以借到腳踏車的海狸在對應的格子中填入 1，其餘填 0，最後將每隻海狸的「借到帽子」與「借到腳踏車」進行「互斥或」運算，運算結果為 1 的就是符合題目要求的海狸。

|       |   |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|---|
| 海狸編號  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 借到帽子  | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 借到腳踏車 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 互斥或   | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

## 💡 關鍵字

邏輯、邏輯閘、數位邏輯



## 32. 海狸飯店

海狸飯店的客房有 5 層樓，每層有 8 間房。清掃機器人會遵循以下的指令：

- ▶ 指令 C：搜尋這一層樓任何一間尚未清潔過的房間，並把它們清掃乾淨。
- ▶ 指令 U：往上一層樓。
- ▶ 指令 D：往下一層樓。
- ▶ 整數  $n$  (指令)：表示重覆做括弧內的指令  $n$  次，括弧內可以有一個以上的指令。

舉例來說，如果你想讓機器人在同一樓層樓清掃兩個未清潔過的房間，請輸入 2(C)。如果你希望機器人清掃完這兩個房間後，接下來往下一層樓，就要輸入 2(C) D。

為了確保機器人會清完飯店的所有房間，機器人必須從客房第一層樓開始，結束也回到第一層樓，未清掃完同一樓層所有房間前不能到其他樓層。

請注意機器人在遇到無法執行指令的情況，就會自動停住不再執行：包括這一層樓的房間都已清理過卻收到指令 C，在第五層樓收到指令 U，或在第一層樓收到指令 D。

請問下列哪個選項中的指令能正確地讓機器人完成這項任務？

- (A) 4(8(C) U) 8(C) 4(D)
- (B) 4(8(C) U) 8(C) D
- (C) 5(8(C) U) 4(D)
- (D) 5(C) U 4(D)

### ✔ 正確答案：B

在移除掉紅線 (GEDH) 與藍線 (DFC) 後，這個系統仍能保持起迄至多一次轉乘的特色。那有可能停駛三條線仍保持正常營運嗎？這個問題表示只使用兩條路線即可保持原有的特色。由於 B 站與 A 站只有紫線 (BAFC) 能到達，所以紫線必定不能停駛。剩下的四個站剛好可以被紅線 (GEDH) 所經過，但是紫線與紅線卻完全沒有相交的車站可以做轉乘，因此無法停駛三條路線。

### i 在資訊科學上的意義

在資訊領域中，常常利用「圖」(graph) 來表示一個複雜的系統或問題，一個「圖」當中最重要兩個元素就是節點 (node) 和邊 (edge)。以本題為例，圖中的節點代表車站，而邊代表車站的連結。有了這樣的表示法後，我們就可以藉由搜尋與計算去回答許多問題，像是任意兩個車站是否有連通？任意兩個車站的最少花費是多少？或是如何使用最少的邊（或是成本最低的邊）使得各車站都能保持連通？

### 💡 關鍵字

鐵路網路 (railway network)、轉乘 (train transfer)、停駛 (shut down railway line)

## ✔ 正確答案：A

機器人要清理同一層樓的八個房間並上一層樓，我們用 8(C) U 的指令。這個指令需重複四次，所以用指令 4(8(C) U)，做完後機器人將停在五樓。我們不能指示機器人重複 5 次 8(C) U 指令，因為這樣最後一個 U 會指示它去到屋頂。接下來機器人在五樓需要清理八個房間，所以用 8(C) 指令。在所有五樓的房間清理完畢後，機器人需返回到一樓，所以最後使用 4(D) 指令。

答案 (B) 不正確，因為做完 4(8(C) U)，機器人會只清理 5 樓某一間房間，並且下一層樓重複清理房間，並下到地下室。

答案 (C) 不正確，因為機器人會上到屋頂上，下 4 層後是停在第 2 層樓。

答案 (D) 不正確，因為機器人只會清理一樓的五個房間，並進入地下室後停止。

## i 在資訊科學上的意義

大部分的程式語言都會提供迴圈結構的語法，像“for”和“while”都是傳統的迴圈結構。此題目提供相當簡單的程式語言語法：其中不同的指令間以空白區隔，表示要循序執行的指令序列。並提供一種簡單的循環結構，只提供類似“for”迴圈來控制，將某一段相同的指令序列重複執行多次。若能加上“if”的條件控制語法，這個程式語法將能寫出更多的指令執行控制功能。要判斷程式的執行是否正確，要能夠了解循序執行指令與循環結構執行的意義，然後模擬追蹤每次指令做完的狀態改變。例如在這個問題中是哪些房間會被打掃，以及機器人會停在哪個樓層，才能確保程式執行完是否正確完成任務。

## 💡 關鍵字

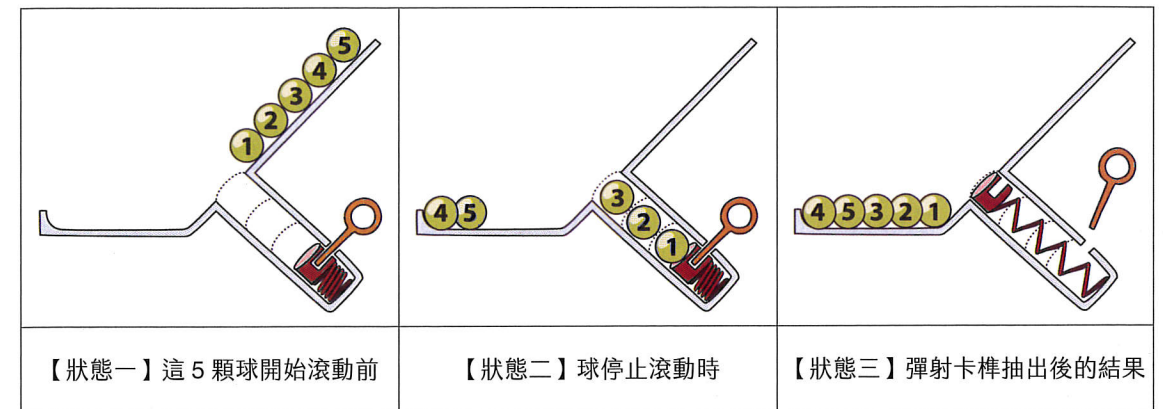
程式語言、循環結構、迴圈結構

# 33. 滾球裝置

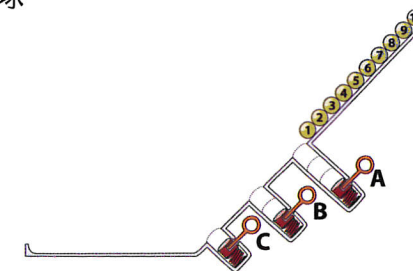
以下這個滾球裝置會將編好號碼的球沿著斜坡滾下，滾下的球依序滾入洞中直到洞被填滿，接著後面的球則會越過洞口，繼續往前滾。

每個洞底都有一個彈射裝置，當裝置的卡榫被抽出，則洞內的球將被彈出洞外往前滾。這些球掉進洞中再彈出來，球原先的順序將改變。





下圖為滾動 5 顆球的不同時間狀態：



現在有 10 顆球沿著斜坡滾下 (如下圖所示)，坡下方有 A、B、C 三個洞，A 洞可容納 3 顆球、B 洞可容納 2 顆球，C 洞可容納 1 顆球。等 10 顆球全數停止滾動時，依照 A、B、C 的順序依序抽出彈射卡榫彈射出洞中的球。



請問下列何者為這 10 顆球最後的順序？

- (A) 
- (B) 
- (C) 
- (D) 

## 34. 雪克機器

在資訊科學課堂上，學生們正在撰寫一段小函數來處理一串數字。這一串數字包含一個有順序性的五個數字並表示成  $[a, b, c, d, e]$ 。而每一個函數個功能就是把原來的數字順序經過函數轉換後成為另一種順序。

Anna 寫了一個函數  $anna([a, b, c, d, e])$  功能是把數字順序轉換成  $[e, b, c, d, a]$ 。

Bob 也寫了一個函數  $bob([a, b, c, d, e])$  功能是把數字順序轉換成  $[e, d, c, b, a]$ 。

老師檢查他們的函數都可以正確轉換後也寫了一個新函數  $Shaker([a, b, c, d, e])$

這個函數的功能是組合了 Anna 與 Bob 的函數成為  $anna(bob(anna([a, b, c, d, e])))$ 。

老師也展示了這個函數的運作方式：

- ▶  $anna(bob(anna([a, b, c, d, e])))$  首先是括號內最內層的  $anna$  函數結果轉換出來
- ▶  $anna(bob([e, b, c, d, a]))$  接者把剛剛的結果再進行內層  $bob$  函數的轉換
- ▶  $anna([a, d, c, b, e])$  結果再次進行  $anna$  函數的轉換
- ▶  $[e, d, c, b, a]$  最後的結果

現在老師又寫了一個新的函數  $test([a \square b \square c \square d \square e])$ ，其運作的方式如下：

- ▶  $bob(shaker(anna([a, b, c, d, e])))$

請問經過這個函數的轉換後，原來的數字順序  $[a \square b \square c \square d \square e]$  會變成如何呢？

- (A)  $[a, b, c, d, e]$
- (B)  $[e, b, c, d, a]$
- (C)  $[e, d, c, b, a]$
- (D)  $[a, d, c, b, e]$

### ✔ 正確答案：D

- A 洞有容納 3 顆球的空間，所以球 1、2、3 依次滾入 A 洞，球 4 到 10 會依次滾過。
- B 洞具有容納 2 個球的空間，所以球 4 跟 5 依次滾入 B 洞，球 6 到 10 依次滾過。
- C 洞則有容納 1 顆球的空間，所以球 6 滾入 C 洞，球 7 到 10 依次滾過。

停止滾動時位於底部由左而右的球順序為 7、8、9、10。然後拉出 A 洞中的彈射卡榫，並按 3、2、1 的順序彈出球，滾動到底部。因此，位於底部的球依序是 7、8、9、10、3、2、1。然後拉出 B 洞中的彈射卡榫，並按照 5、4 的順序彈出球。此時，位於底部的球依序為 7、8、9、10、3、2、1、5、4。最後，將 C 洞的彈射卡榫釘拉出，球 6 滾到底部，因此正確答案是 7、8、9、10、3、2、1、5、4、6。

### i 在資訊科學上的意義

這些洞的滾入彈出方式就像資料結構中的堆疊。堆疊是一種記錄組織資料的方式，它是基於後進先出原則 (LIFO)。例如，第一個落入洞中的球，是從洞中最後彈出一顆球。儘管是一個非常簡單的想法，它在許多不同的情況下都是很實用的。例如，可以使用堆疊 (stack) 來檢查  $(1 + 2) * 3$  中的左右括號是一樣多，但  $((4 + 5) * (6 - 7))$  中的括號不是。想法是將每次

讀到一個左括號就存入一個堆疊 (stack) 中 (這個操作稱為 push)，當讀到一個右括號時，就從堆疊的頂部移除一個匹配的左括號 (這個操作稱為 pop)。如果 push 跟 pop 都可順利執行，且最後堆疊是空的，就知道左右括號是一樣多。用電腦解迷宮路徑問題，通常也會用堆疊來記錄走過的路徑，這樣才能在遇到死路時知道如何一步一步退回去找其他路徑。

### 💡 關鍵字

堆疊 (stack)、後進先出 (LIFO)

## ✔ 正確答案：B

- shaker 函數的轉換結果其實和 bob 函數轉換結果是一樣的。
- 所以 shaker(bob()) 會互相抵銷轉換的結果。
- 因此 test 函數其實就跟單獨進行 anna 函數轉換是一樣的。

## i 在資訊科學上的意義

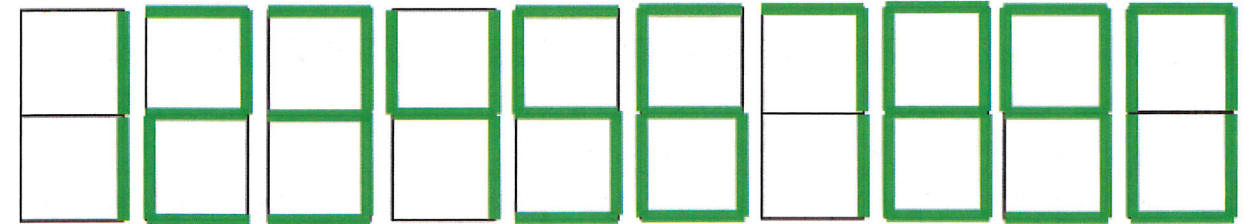
「函數」是表示了一種對應關係，以本題來說，anna 函數表示頭尾兩個元素交換的關係，而 bob 是將所有元素反序的對應關係。而當我們把 anna 函數放到 bob 函數中的時候 bob(anna([a, b, c, d, e])), 我們就得到一個合成函數 (composition function)。藉由合成函數，我們就可以使用一些基本函數組合產生複雜功能的合成函數，十分方便。除此之外，如果我們將函數視為事件的因果關係，則合成函數即可表示一連串具有因果事件的組合。

## 💡 關鍵字

演算法分析、陣列、資料結構、後進先出

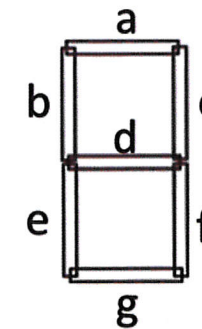
# 35. 數字辨認系統

有一個簡易的數字辨認系統，它能認得下列用線段組合而成的數字：



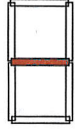
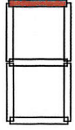
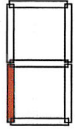
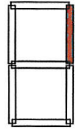
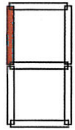
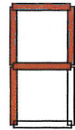

雖然每一個數字都是由最多七個線段組成，但其實要能辨認數字其實並不需要知道所有的線段。

問題是若要能明確辨認出所有數字 (0...9)，哪些線段是必要的？



- (A) abcde
- (B) bcdef
- (C) dcefg
- (D) adefg

## ✔ 正確答案：A

|   |   |  |   |   |   |   |
|---|---|--|---|---|---|---|
|  |                    |                                     |  |                  |  |  |
| 要知道是否僅需要四個線條讓我們看看下面的理由。因為數字 0 和 8 只有一個線段的差異，所以這個線條必須被保留才能分辨兩者。                    | 因此，所有數字可以被分成兩組。一組是不包含此線段的 {0, 1, 7} 和另一組包含此線段的 {2, 3, 4, 5, 6, 8, 9}。在第一組中 1 和 7 只有相差一個線段，所以他必須被保留。 | 接著根據這兩個線段所有數字可以又可進一步分成四組：{0, 7}、{1}、{4}、{2, 3, 5, 6, 8, 9}。此時根據兩個線段已能分辨 1 或 4。現在讓我們再找找還有哪些數字僅相差一個線段。而 8 和 9 只差異這個線段。 | 依據是否包含了這個三個線段，所有數字可以分成 {0}、{7}、{1}、{4}、{2, 6, 8}、{3, 5, 9}。同樣地，6 與 8 也只有一个線段的差異。  | 再依據這四個線段的出現與否，數字可分成 {0}、{7}、{1}、{4}、{2, 8}、{6}、{3, 9}、{5}。所以四個線段仍不足。我們又發現 3 與 9 只差一個線段。所以也把他加入必要線段。 | 最後根據這五個線段的出現情形，所有數字都可以明確被辨認出來了。   |   |

## 在資訊科學上的意義

資訊科學中有一門稱為「模式識別」(pattern recognition) 的領域，探討的是如何讓電腦對於真實世界的物體或現象進行辨別的技术。常見的像是人臉辨識、手寫辨識、語音辨識等等。然而對電腦而言，所有的資料都是 0 與 1 組成的，那要怎麼辨識呢？有一個方法是利用事先設計好的「特徵」(feature) 來進行判別。舉一

個簡單的例子來說，假設給定一群 10 歲兒童的身高、體重、血型與膚色，要來判別性別，以身高和體重當作辨識的特徵就會比使用血型和膚色來的有效。而對於本題數字辨識來說，這七個線段每一個都可以視為一個特徵，挑選能有效識別資料的特徵來進行辨識，就是本題要傳遞的概念。

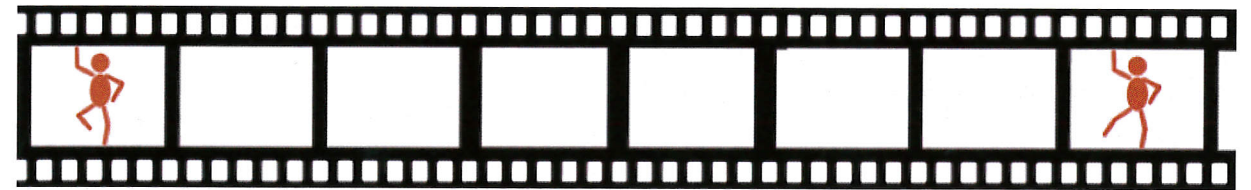
### 💡 關鍵字



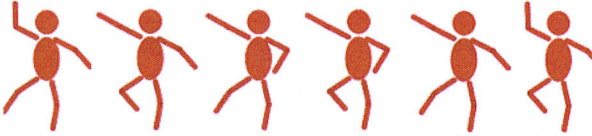

辨識、最小化、位數

## 36. 舞者

小狸製作了一個舞者的動畫，這個舞者在同一時間只能做以下三個動作的其中之一：移動左手、移動右手、移動右腳。

給你這個動畫的第一個畫面和最後一個畫面，請問下列哪一個選項的畫面放到中間六格時，符合相鄰兩個窗格的舞者僅能有一處不同動作的規定。



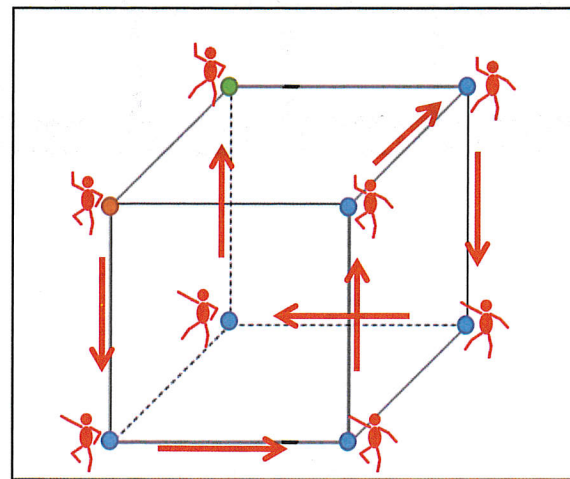
- (A) 
- (B) 
- (C) 
- (D) 

## ✓ 正確答案：A

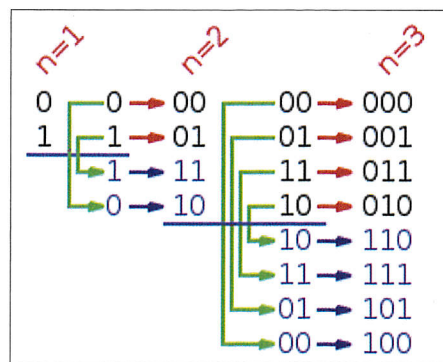
本題只需檢視各選項的畫面序是否符合只做一個動作的原則，僅A選項符合。  
若是要自行產生符合規定的畫面序，可從題目給定的第一個畫面開始，尋找符合只做一個動作原則的畫面新增為第二個畫面，依序建立，若期間找不到符合原則的下一個畫面，那就「退回」上一個畫面，並尋找另一個符合原則的畫面新增，直到所有畫面皆已置入畫面序。

## i 在資訊科學上的意義

本題舞者只能做三個動作，所以可以將此八個不同動作的畫面置於立方體的端點，每個端點有三個移動方向—「上下」代表移動右手、「左右」代表移動左手、「前後」代表移動右腳，如右圖所示。若你可找到一條經過所有的端點（僅一次）的路徑，則可得到正確的畫面序。另一個可適用於三個動作以上的解法為格雷碼 (Gray codes)。格雷碼由貝爾實驗室於1940年提出，格雷碼是一個數列集合，相鄰兩數間只有一個位元改變，且格雷碼的順序不是唯一的，其主要目的在避免訊號傳送錯誤



以本題為例，我們可以把三個動作編碼為三位元的二進位數字，例如001代表彎曲右腳、彎曲右手、伸直左手，只要產生一個三位元的格雷碼即可還原為畫面序。而快速產生格雷碼可採用鏡射排列，n位元的格雷碼可以從n-1位元的格雷碼以上下鏡射後加上新位元的方式（上半段新位元為0、下半段新位元為1）快速的得到，如右圖所示。

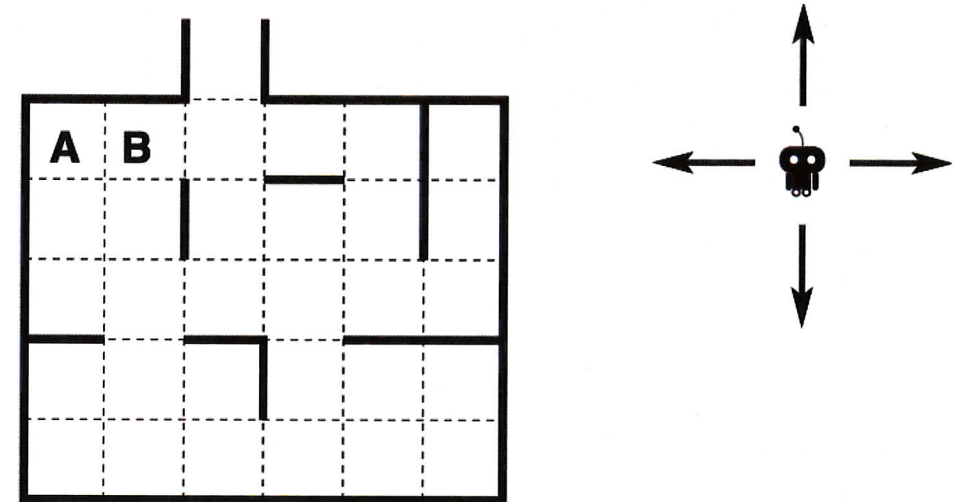


## 💡 關鍵字

動畫、順序、窗格、動作、端點、退回

# 37. 走出迷宮

讚哥製造了一個機器人，他可以前後左右移動。讚哥將它放在一個有圍牆的小迷宮中如以下地圖，圖中的實線代表機器人不能通過的圍牆，虛線所圍的格子則代表機器人可以移動的區域。



讚哥會用一連串的指令，讓機器人接受「↓」、「↑」、「←」、「→」四種指令依箭頭方向移動。執行指令的規則如下：

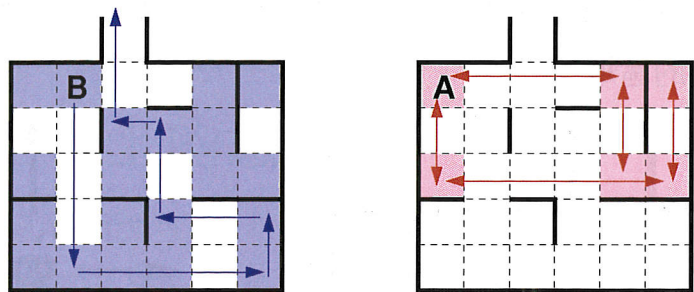
- ▶ 機器人接受一個指令後，將一直沿著那個指令的方向移動，直到撞上牆壁或離開出口才會停止。
- ▶ 機器人一次執行一個指令，等到停下後（也就是撞上牆壁時），機器人才會執行下一個指令。舉例來說，如果機器人一開始在A的位置，「↓」的指令可讓機器人向↓方向移動兩格後停下來。如果是「→」的指令，機器人會向→方向移動四格後停下來。沒有指令可讓機器人由A位置開始，而停在B位置。

請問從A和B哪一個起始位置可以讓機器人走出迷宮？

- (A) A和B皆可
- (B) 只有A可以，B不可以
- (C) 只有B可以，A不可以
- (D) A和B皆不可以

## ✔ 正確答案：C

它可以從位置 B 走出迷宮，但不能從 A。左圖中的藍色箭頭顯示從位置 B 開始移動的方式。右圖中的紅色箭頭顯示從位置 A 開始移動的方式。藍色或紅色方塊代表機器人可能停下來的位置。



在此題目中，從 B 開始若接受「←」指令會停在 A，接下來的移動的方式跟右圖中相同，無法走出迷宮。但 B 若接受「↓」指令，則可如左圖中的藍色箭頭顯示如何離開迷宮的移動路徑。

## i 在資訊科學上的意義

機器人程式是一個很好的範例，你可以清楚看到每一指令執行控制機器人的移動結果。其他類型程式也是一樣的原則：每個「指令」會改變電腦中記錄資料的「狀態」（機器人墜及移動方向），可以透過將計算結果輸出到螢幕上，了解指令執行如何影響資料狀態，以及最後是否達到目標狀態。

機器人接下來是否能離開迷宮，取決於目前位於哪個位置，而與先前怎麼走到這裡的移動順序無關。所以你可以從起點位置，根據不同指令，標記接下來可能走到的位置，再從這些已

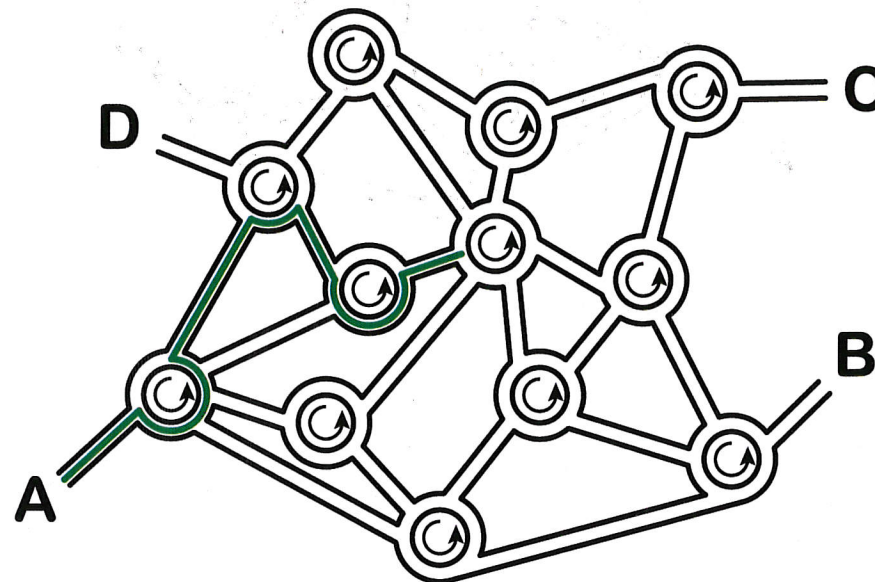
標記位置繼續標出下個位置，以此推類直到走出迷宮，或無法標出新的位置為止。在演算法中，找出所有可能的位置移動路徑叫做「解答」的「搜尋」，有各種搜尋解答的演算法。廣度優先搜尋法是重複對已標記的位置，一一標記出所有尚未走過的下個位置。深度優先搜尋法則從目前位置下一步可能的停止位置先選一個標記，接著繼續往下標記再下一步可能的一個位置，繼續下去若遇到死路或走回標記過的位置，則要退回上一停止位置（稱為回溯），嘗試另一個方向的移動指令

## 38. 環狀城市

在環狀城市中，如果你想要由 A 點出發走地圖中綠色的路線，必須這樣走：

- ▶ 1. 首先，進到第一個圓環時，依圓環中的逆時針方向走，然後從第 4 個分岔路離開圓環。
- ▶ 2. 接著進到下一個圓環，依逆時針方向，走第 1 個分岔路離開圓環。
- ▶ 3. 再進到下一個圓環，依逆時針方向，走第 2 個分岔路離開圓環。

為了簡略上述冗長的文字，導航系統會用 "4 1 2" 來表示上述走法。

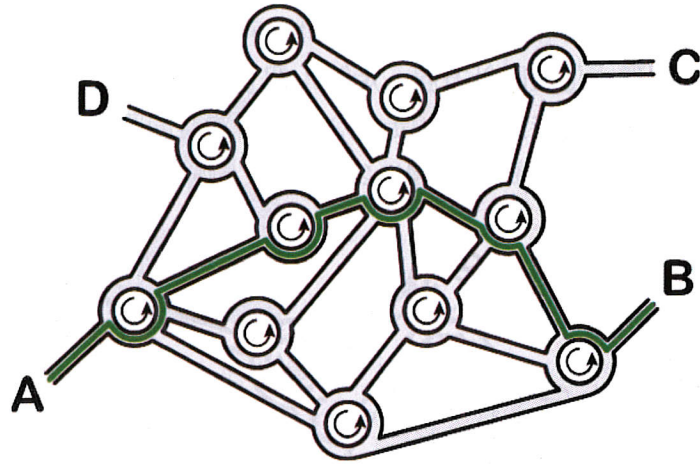


如果我們從 A 點出發，依照導航系統所指示的 "3 1 3 2 3" 走法，會走到哪裡呢？

- (A) A
- (B) B
- (C) C
- (D) D

## ✔ 正確答案：B

依照導航系統所指示的 "3 1 3 2 3" 走法，所走的路徑如下圖所示會走到 B。



## i 在資訊科學上的意義

這個題目用到程式語言的一個重要元素：指令的循序組合，表示按順序依次執行指令。我們給汽車的一連串指令，相當於電腦程式中的循序指令，調換指令順序組合可能走到不同的目的地，或無法達到目的地。例如 "3 1 3 3 2" 走法不會到達 B，也無法到達 C。但要到達某一個目的地，也有可能有一個以上的指令循序組合可達到，例如 "3 1 2 2 2" 也可以到達 B。若有多個程式可達到同樣執行效果，我們通常會

選用較省時間或儲存空間的程式。

此外，這個題目中指令對應的數字，相當於是指令的一個「參數」。因為這裡的一個指令可分成更細的步驟：「進到下一個圓環」，「依逆時針方向走直到第 n 個分岔路離開圓環」，只是數字 n 的值不同。把一些常用的步驟組合放在一起，形成一個指令單位，由給定參數改變不同設定（如 n 值），就是基本的模組化程式設計概念。

## 💡 關鍵字

循序指令、模組化程式設計

# 39. 太多個 1 了！

小狸打了一篇很長很長的文章後才發現了一個天大的錯誤，那就是文章中所有的 1 應該是 11，而 11 應該是 1。幸好小狸非常聰明，他有一個文字編輯器可以將文章中的一串文字取代成另一串文字。

例如：將句子 "See what happops in a soptopce in which I replace all occuropces." 中的 op 用 en 取代，原本的句子就變成 "See what happens in a sentence in which I replace all occurences."

小狸該如何使用編輯器修正他的文章呢？

- (A) 先將所有 11 取代成 1，再將所有 1 取代成 11
- (B) 先將所有 1 取代成 11，再將所有 11 取代成 1
- (C) 假設文章中都沒有出現 \$，先將所有 1 取代成 \$，再將所有 \$ 取代成 11，最後將所有 11 取代成 1
- (D) 假設文章中都沒有出現 \$，先將所有 11 取代成 \$，再將所有 1 取代成 11，最後將所有 \$ 取代成 1

## ✔ 正確答案：D

- A 選項：將所有的 11 取代成 1 之後，全部都變成 1，無法區分哪些原來是 1 哪些原來是 11；再將所有的 1 取代成 11，最終就只有 11。
- B 選項：將所有的 1 取代成 11，則原來的 1 變成 11、原來的 11 變成 1111，再將所有的 11 取代成 1，就回復到原始文章內容。
- C 選項：將 1 取代成 \$，再將 \$ 取代成 11，會將 1 變成 11，而 11 維持不變；再將所有的 11 取代成 1，最終結果將是全部都變成 1。
- D 選項：為正確的文字互換方式。

## i 在資訊科學上的意義

對於初次接觸程式設計的人來說，最常遇到的疑惑之一是為什麼「把 1 換成 11，而把 11 換成 1」這件事會這麼複雜？原因就在於程式的執行是一個指令接著一個指令執行，而且資料被取代後就無法得知之前的內容了。以一個類似的問題為例子，如何將程式中兩變數 a, b 的儲存內容值互換？常見錯誤的寫法是：

```
b ← a;
a ← b;
```

## 💡 關鍵字

文字編輯器、指令序、變數內容互換

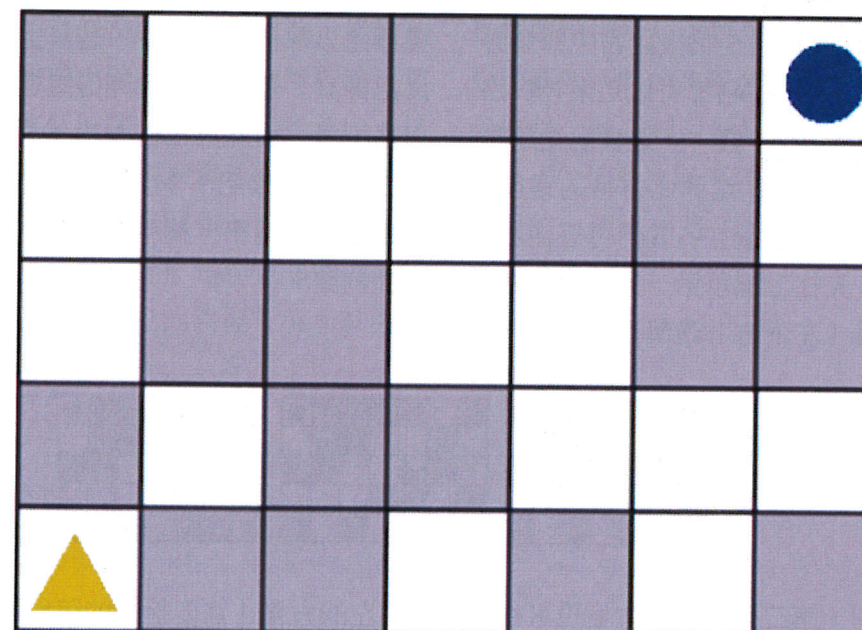
這將使得變數 a, b 的值都變成 a 原先的內容。正確得作法是必須藉由第三個變數 temp 來當暫存區，做法如下：

```
temp ← a;
a ← b;
b ← temp;
```

這概念與本題選項 D 做法相同。

# 40. 拆除障礙牆

有一個迷宮由空地（白色方塊）和障礙牆（灰色方塊）組成，迷宮中的移動規則是：可以從水平或垂直方向從一個空地移動到相鄰的空地（不能對角線移動）。

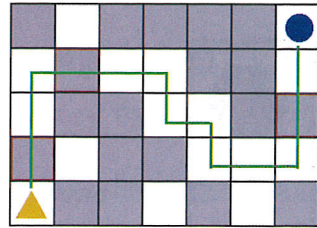


最少要拆除幾個障礙牆把它變成空地，就能夠從左下方三角形的的位置移動到迷宮右上角圓形的位置。

(請填一個 0 到 5 之間的整數。作答範例：0)

## ✔ 正確答案：3

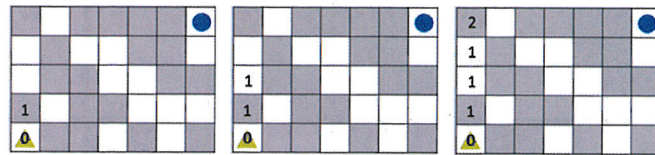
最少需要拆除的障礙牆數量為 3。在下面的圖中，這些牆壁用紅色的框標示出來。綠線顯示拆除障礙牆後可以移動的路徑。解決這個問題的有系統的方法是



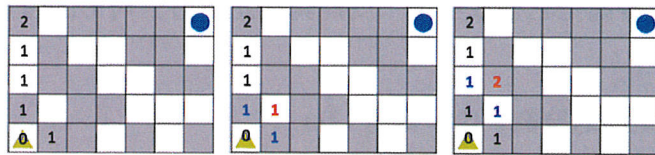
先假設迷宮中每個方格都可以走來尋找路徑，一邊標記從起點走到此方格的路徑上

「需要拆除的障礙牆數量」。我們可以從左下角的方格開始，先往上標示第一列其他方格。由於第一個方格（左下方）是起點，所以我們

可以將其標記為 0。接下來對每個處理的方格，根據其上下左右方格中已知「需要拆除的障礙牆數量」的最小值，如果這個方格是障礙牆，就用該值加 1 後再標記於目前方格，否則就直接標記該數值在方格中。若其上下左右方格還沒記錄其「需要拆除的障礙牆數量」值，就先忽略不管。因此第二個方格（第一個方格上面的方格）應該記下 1，下一個方格記下 1，依此類推，直到第一欄中的所有方格都被標記如下面的圖。



接下來看第二列，第二列的每一方格現可根據它下或左的方格「需要拆除的障礙牆數量」選擇最小的數，如果目前方格是障礙牆，該數字加 1 後再標記於目前方格，否則就直接標記該數字，結果如下圖所示。



注意，在一個方格初次標記數字後，如果它上下左右的方格中標記數字有更改，表示可能檢查到新的路徑，就必須再次檢查該方格，是否可能根據它上下左右方格標記出更低的數值更改原標記。所以重要的是要檢查幾次，直到沒有標記需要進一步更改為止。以下圖片中，後來被更改值的方格以黃色顯示。



最後，當所有的方格標記不會再變動後，可以看到終點的右上方方格被標記為 3，表示要到達右上角最少必須拆除 3 個障礙牆。

## i 在資訊科學上的意義

尋找迷宮中的路徑是一個廣為人知的資訊科學問題，這個問題類似搜尋迷宮的想法，但增加了額外的條件要求拆除盡可能少的障礙牆。要有系統地解答這個問題，需要考慮從起始方格逐一掃描走到其他方格的演算法。這個問題也跟尋找最短路徑的觀念有關係，當要找到一個地點的最短路徑，可以從已知到它鄰居的最短路徑再加上鄰居到這地點的路徑長度，其中再

取最小值決定。在這個問題中，到一個地點的最短路徑相當於從起點走到此方格的路徑上最少「需要拆除的障礙牆數量」。而若此方格為障礙牆，鄰居到這地點的長度相當為 1，否則為 0。在路徑長度不會有負值限制下，經常被採用的一個最短路徑搜尋演算法為 Dijkstra 演算法。

## 💡 關鍵字

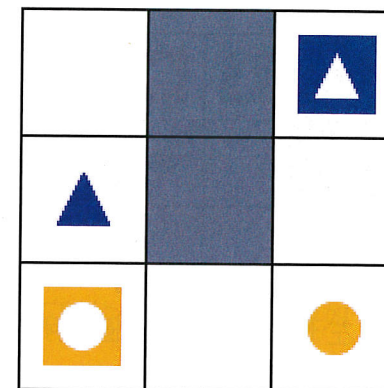
迷宮、最短路徑

# 41. 棋子回家

有個電腦遊戲在一個由白色及灰色方格構成的棋盤上玩，白色方格可放棋子，灰色方格不能放或移入旗子。

白色方格上一開始會放以下兩類棋子：

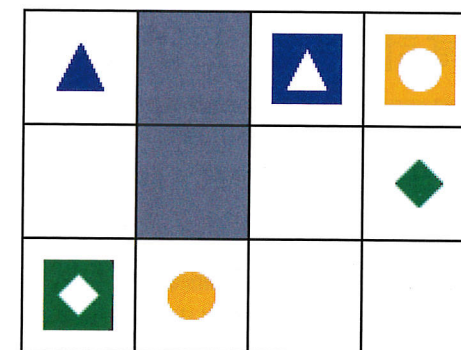
- ▶ 圓形、三角形或菱形的各色形狀棋。
- ▶ 各形狀棋的家 - 中央有挖空形狀棋的方塊。



每個形狀棋可使用：上、下、左、右 四種指令，分別表示向上、向下、向左、向右移動到相鄰方格，但相鄰方格必須是白色且沒有其他任何棋子，或是包含這個形狀棋的家才可移入。如果一個形狀棋移到自己的家，那個形狀棋和它的家接下來會從棋盤上消失。

玩家的任務是讓每個形狀棋都回到它們的家（都從棋盤上消失）。

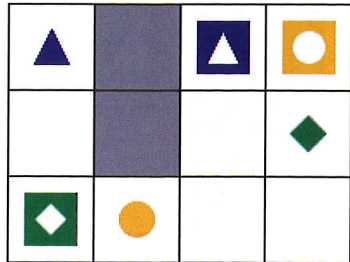
以上圖的棋盤為例：形狀棋回家的正確移動順序指令是：圓形：（左左），三角形：（下右右右上）  
選項中哪一串移動順序指令可以完成右圖棋盤的任務？



- (A) 圓形：（右右），菱形：（左下右右），圓形：（上上），三角形：（下下右右上上）
- (B) 菱形：（左下左左），圓形：（右右上上），三角形：（下下右右上上）
- (C) 圓形：（右右），菱形：（左左下左），圓形：（上上），三角形：（下下右右上上）
- (D) 圓形：（右右），菱形：（左下左左），圓形：（上上），三角形：（下下右右上上）

## ✔ 正確答案：D

(A) 執行圓形：(右右) 後如下圖，因此執行菱形：(左下右右) 會遇到圓形所在位置，因此是錯誤的。

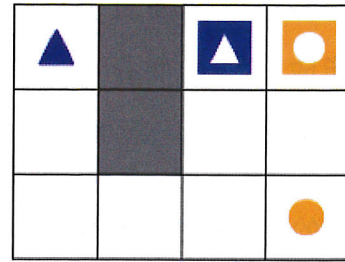


(B) 執行菱形：(左下左左) 會遇到圓形所在位置，因此是錯誤的。

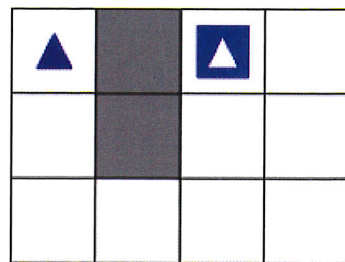
(C) 執行圓形：(右右) 後如 (A) 所示上圖，因此執行菱形：(左左下左) 會遇到灰色方格，因此是錯誤的。

(D) 執行圓形：(右右) 後如 (A) 所示上圖，執行菱形：(左下左左) 後可讓菱形回到

它的家並從棋盤上消失如下圖。



執行圓形：(上上) 後可讓圓形回到它的家並從棋盤上消失如下圖。



最後執行三角形：(下下右右上上) 後可讓三角形也回到它的家並從棋盤上消失。

## i 在資訊科學上的意義

要解決此問題必須執行一連串的執行指令，電腦程式是由一連串指令構成的。指令的前後順序不同會影響執行結果，因此必須分析指令執行先後次序的影響。例如這個問題中，圓形如果不先往右移 2 個方格，菱形與方形會被圓形擋住而無法回到自己的家。接下來菱形不先移開，圓形也無法回到自己的家。這些條件的分析，可以幫助決定如何安排指令的先後執行

順序來解決問題。值得注意的是有些指令的順序調換，並不會影響最後的結果，例如答案 (D) 的指令順序改成

圓形：(右右)、菱形：(左下左左)、三角形：(下下右右上上)、圓形：(上上)

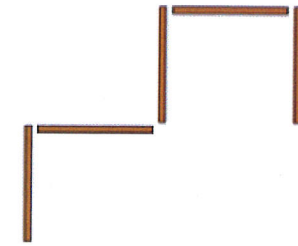
仍會得到相同的結果。所以要解決同一個問題的電腦程式，可以用不同的指令順序，可以有各種寫法。

## 💡 關鍵字

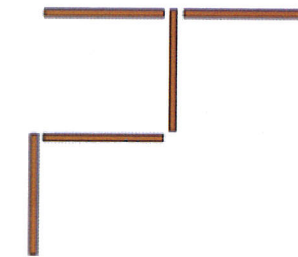
指令序列

# 42. 五枝牙籤

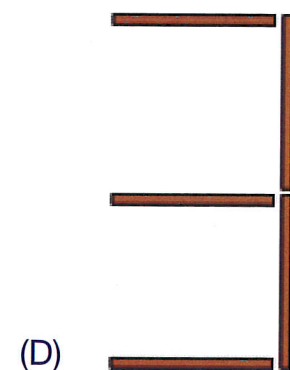
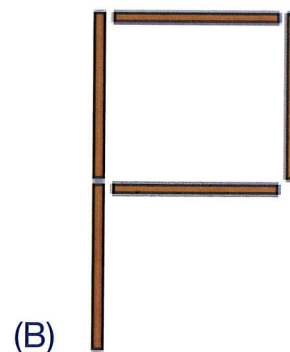
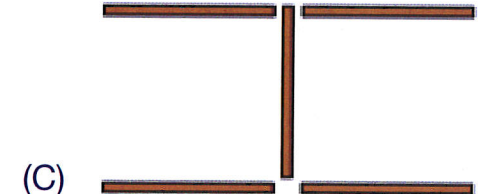
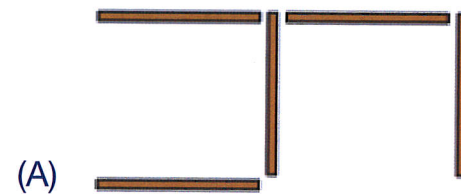
阿亞有 5 枝牙籤，他把牙籤放在桌上，並擺成以下形狀。



小羅走到桌子旁邊，她拿起上圖中的一枝牙籤，放在不同的位置，形狀改變如下圖。

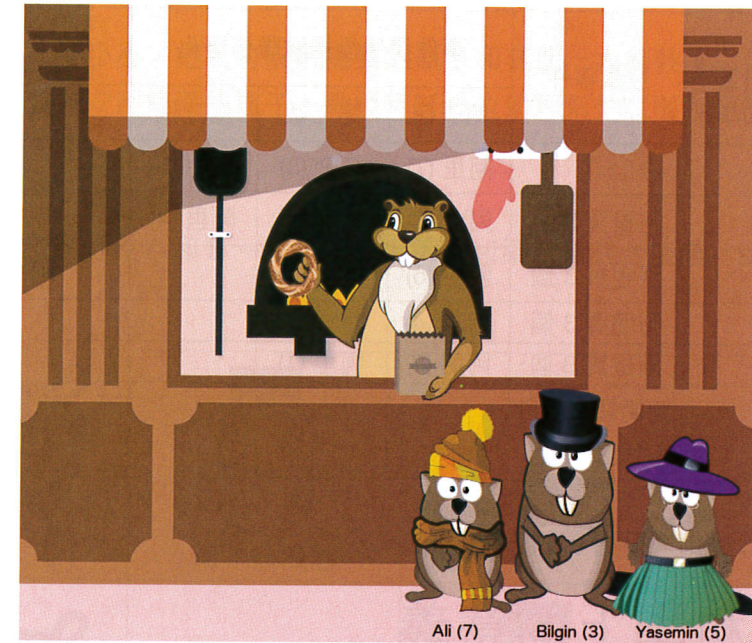


然後小鮑走到桌子旁邊，他也拿起一枝牙籤，放在不同的位置。請問小鮑無法排出下列哪個形狀？



# 43. 麵包店

海狸村的麵包店每 2 分鐘現烤出爐製作出 3 個貝果。



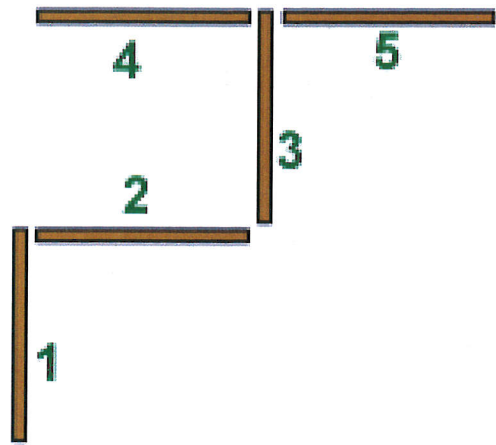
在麵包店前面有一條排隊線，一次只服務一位客人。麵包店早上七點開始開門營業，而且是第一個客人點了以後才開始做。一開門營業已經有三隻海狸在排隊。第一隻海狸阿里要買 7 個貝果，第二隻海狸阿金要買 3 個貝果，第三隻海狸阿明要買 5 個貝果。但是排隊輪到的海狸一次最多只能購買 3 個貝果，如果要買更多的話，必須立刻到隊伍的最後面重新排隊。

請問開始營業幾分鐘後第三隻海狸阿明可以買到他需要的 5 個貝果？

- (A) 6 分鐘
- (B) 8 分鐘
- (C) 10 分鐘
- (D) 12 分鐘

## ✔ 正確答案：D

為了更容易說明解答，我們把牙籤加上編號。



水平放置的牙籤有左右兩個端點，垂直放置的牙籤則有頂端和底端。

## i 在資訊科學上的意義

想使用牙籤排出形狀時，就要了解並掌握一連串基本操作（移動一枝牙籤到另一個地方）所形成的結果。我們要精確小心才能得到正確答案，這很像寫一個電腦程式及測試它是否正確的過程。雖然這個問題看起來像是一個數學的解題任務，不過解這個問題除了空間想像力外，還要思考移動牙籤的各種排法，這是在以程式解決問題時需要的能力。除了必須了解前已經進行的步驟，還必須清楚下一個步驟如何執行。

把牙籤排出來的一個形狀想成一個狀態，每次

- (A) 把牙籤 1 以垂直向下的方式，放在牙籤 5 的右端，就排出選項 (A) 的形狀。
- (B) 把牙籤 5 垂直放在牙籤 4 左端和牙籤 1 頂端之間，就排出選項 (B) 的形狀。
- (C) 把牙籤 1 水平放置在牙籤 2 的右端，就排出選項 (C) 的形狀。
- (D) 移動任何一枝牙籤，都排不出選項 (D) 的形狀。

移動一隻牙籤到另一位置（一個操作指令），所形成的形狀是另一個狀態。資訊科學家可以用圖型的節點來表示不同的狀態，若一個狀態（形狀）透過一個操作指令會到達另一個狀態（形狀），就會有一個有方向性的「邊」將這兩個狀態節點連起來，表示接收到這個操作指令會從前一個狀態「轉換」到另一個狀態；這是有限狀態機的基本觀念。寫程式的過程就是要將資料從一個初始狀態，想出經過哪一串連續的操作指令，可以到達目標狀態。

## 關鍵字

有限狀態機



## ✔ 正確答案：C

下面的表格顯示每 2 分鐘出爐後排隊隊伍的變化情況：對於每隻海狸，括號內的數字表示他還需要買的貝果數量。例如阿里 (7) 意味著阿里正在還需要 7 個貝果。

| 時間 (分鐘) | 發生事件               | 事件發生後排隊隊伍變化 |        |        | 事件發生後已買好的人 |
|---------|--------------------|-------------|--------|--------|------------|
|         |                    | 第一個         | 第二個    | 第三個    |            |
| 0       |                    | 阿里 (7)      | 阿金 (3) | 阿明 (5) |            |
| 2       | 阿里買 3 個            | 阿金 (3)      | 阿明 (5) | 阿里 (4) |            |
| 4       | 阿金買 3 個            | 阿明 (5)      | 阿里 (4) |        | 阿金         |
| 6       | 阿明買 3 個            | 阿里 (4)      |        |        |            |
| 8       | 阿里買 3 個            | 阿明 (2)      | 阿里 (1) |        |            |
| 10      | 阿明買 2 個<br>阿里買 1 個 |             |        |        | 阿明 □ 阿里    |

所以阿明將在第十分鐘出爐時可買到 5 個貝果。

## i 在資訊科學上的意義

在這個問題中，麵包師就像電腦的中央處理器 (CPU)，處理器的速度反應在一定的時間能做多少運算指令。海狸要向麵包店買貝果，類似於處理程序 (或程式) 向處理器提出運算的需求，海狸想要的貝果是處理一個程序需要的運算量。因此每個程序要運算完成，需要占用處理器的一些運算時間。當有多個程序要執行

時，電腦的作業系統要負責安排管理處理器進行這些程序的順序控制，稱為中央處理器排程 (CPU scheduling)。有很多種進行排程的演算法，這個問題採用稱為循環演算法 (round-robin) 的調度方法，將處理器在多個程序間輪流切換處理。

## 💡 關鍵字

作業系統、排程問題、循環演算法

# 44. 安眠魔藥

海狸教授開發了一種新的安眠魔藥，無色無味像白開水一樣，卻有強力的效果：只要喝上一滴就會馬上沉睡 30 分鐘。

可是，糊塗的教授一不小心，把裝有安眠魔藥的瓶子和其他一大箱裝著白開水的瓶子混在一起了。為了盡快找出裝有安眠魔藥的瓶子，教授拜託幾位助理幫忙試喝，也請你幫忙他一起找出正確的藥瓶。

例如：目前有兩位海狸助理 A 和 B。海狸 A 喝下第 1 和第 3 瓶混合液體，海狸 B 喝下第 2 和第 3 瓶混合液體。

|   |  |  |  |
|---|--|--|--|
| <br>海狸 A   | <br>第 1 瓶液體 |  | <br>第 3 瓶液體   |
| <br>海狸 B |  | <br>第 2 瓶液體 | <br>第 3 瓶液體 |

- ▶ 如果只有海狸 A 睡著了，代表第 1 瓶液體是安眠魔藥。
- ▶ 如果只有海狸 B 睡著了，代表第 2 瓶液體是安眠魔藥。
- ▶ 如果海狸 A 和 B 都睡著了，代表第 3 瓶液體是安眠魔藥。
- ▶ 如果海狸 A 和 B 都沒有睡著，代表這 3 瓶液體都是白開水。

如此一來，雖然有一大箱的瓶子要檢測，但 2 位助理就可以確定其中 3 瓶液體的成分。

我們現在有 3 位海狸助理可以幫忙。請問按照上述的方式進行一回合試喝，最多可以確定幾瓶液體的成分？

- (A) 4 (B) 5 (C) 6 (D) 7 (E) 8

## ✔ 正確答案：D

如果有 3 位海狸助理幫忙這個實驗，他們在一回合試喝中最多可以確定 7 瓶 (23-1) 液體的成分。

|      |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|
|      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 海狸 A | ✔ |   | ✔ |   | ✔ |   | ✔ |   |   |
| 海狸 B |   | ✔ | ✔ |   |   | ✔ | ✔ |   |   |
| 海狸 C |   |   |   | ✔ | ✔ | ✔ | ✔ |   |   |

- 如果只有海狸 A 睡著，代表第 1 瓶裝的是安眠魔藥，其他是白開水。  
如果只有海狸 B 睡著，代表第 2 瓶裝的是安眠魔藥，其他是白開水。
- 如果海狸 A 和 B 都睡著，代表第 3 瓶裝的是安眠魔藥，其他是白開水。
- 如果只有海狸 C 睡著，代表編號 4 瓶子裝的是安眠魔藥，其他是白開水。
- 如果海狸 A 和 C 都睡著，代表第 5 瓶裝的是安眠魔藥，其他是白開水。

依據以上說明，3 位海狸在一回合中可以確定 7 瓶 (23-1) 液體的成分。

## i 在資訊科學上的意義

二進制數字 (binary number) 是指以 2 為進位基底的數字系統，它只使用兩種數字符號：通常是 0 和 1。二進制數字除了也可以用來做加減乘除計算外，如果給予 0 和 1 不同的意義，也很容易幫助思考一個問題的可能狀態有幾種。例如我們可以假定在本題 0 代表的是白開水，1 代表的是安眠魔藥，如果以 3 個 0 和 1 來表示 3 位海狸的實驗狀態，我們可以很容易

的得到下表並且得之所以可能情況：

| 二進制表示法 | 二進制表示的意義       | 結果         |
|--------|----------------|------------|
| 001    | 只有第 1 隻海狸睡著    | 第 1 瓶是安眠魔藥 |
| 010    | 只有第 2 隻海狸睡著    | 第 2 瓶是安眠魔藥 |
| 011    | 第 1、2 隻海狸都睡著   | 第 3 瓶是安眠魔藥 |
| 100    | 只有第 3 隻海狸睡著    | 第 4 瓶是安眠魔藥 |
| 101    | 第 1、3 隻海狸都睡著   | 第 5 瓶是安眠魔藥 |
| 110    | 第 2、3 隻海狸都睡著   | 第 6 瓶是安眠魔藥 |
| 111    | 第 1、2、3 隻海狸都睡著 | 第 7 瓶是安眠魔藥 |
| 000    | 沒有海狸睡著         | 這 3 瓶都是白開水 |

## 💡 關鍵字

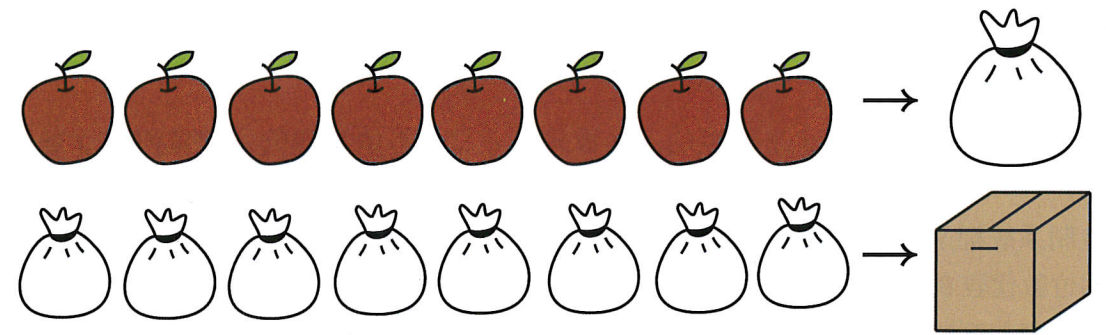
二進制

# 45. 蘋果裝箱

海狸家族有一片蘋果園，每年蘋果盛產時，他們都在思考應該如何將蘋果包裝運送到市場比較好。

今年他們打算根據以下規則包裝：

例如：目前有兩位海狸助理 A 和 B。海狸 A 喝下第 1 和第 3 瓶混合液體，海狸 B 喝下第 2 和第 3 瓶混合液體。



- ▶ 1. 將蘋果放入袋子中，每袋都只能有剛好八顆蘋果，如果最後有不滿八顆的蘋果，就留在袋子外面
- ▶ 2. 將袋子放入箱子中，每個箱子都只能放剛好八袋蘋果，如果最後有不滿八袋的部分，就應該留在箱子外面

今天海狸家族採收了 275 顆蘋果，請問包裝過後會有多少箱？又有多少袋蘋果被留在箱子外？多少顆蘋果變成散裝的？

- (A)
- (B)
- (C)
- (D)

## ✔ 正確答案：B

每個箱子內總共有 64 顆蘋果 (8 袋 × 每袋 8 顆蘋果)，海狸家族採收了 275 顆蘋果，可以裝滿 4 個箱子 ( $275 \div 64 = 4.296875$ )，此時剩下  $275 - 256 = 19$  個蘋果未包裝，可以再裝滿 2 個袋子 ( $19 \div 8 = 2.375$ )，最後剩下  $19 - 2 \times 8 = 3$  個散裝的蘋果。

驗算一下可以發現：

選項 A 總共有  $3 \times 64 + 7 \times 8 + 7 = 255$  顆蘋果

選項 C 總共有  $3 \times 64 + 5 \times 8 + 1 = 233$  顆蘋果

選項 D 總共有  $4 \times 64 + 1 \times 8 + 6 = 270$  顆蘋果

## i 在資訊科學上的意義

電腦內部資料都是用 0 和 1 來儲存的，這種只有 0 和 1 兩種狀態的系統就叫做二進制系統 (Binary)。

而八進制系統則是使用八個不同的符號表示 (0~7)，八進制數字中最右邊的位數代表  $8^0 = 1$ ，第二個位數代表  $8^1 = 8$ ，第三個位數代表  $8^2 = 64$ ，依此類推，每個位數都是右邊位數的八倍。一個二進制的 3 位數，若改採用八進制表示，就只需要使用 1 個位數就能表達同樣的數字，因此在一些系統上仍能看到使用八進制顯示數字。

在題目中，每個箱子內有 82 顆蘋果，每個袋子內會有 81 顆蘋果，散裝的蘋果則每個都是 80，剛好代表十進制的 275 可以用八進制系統表示成 423。

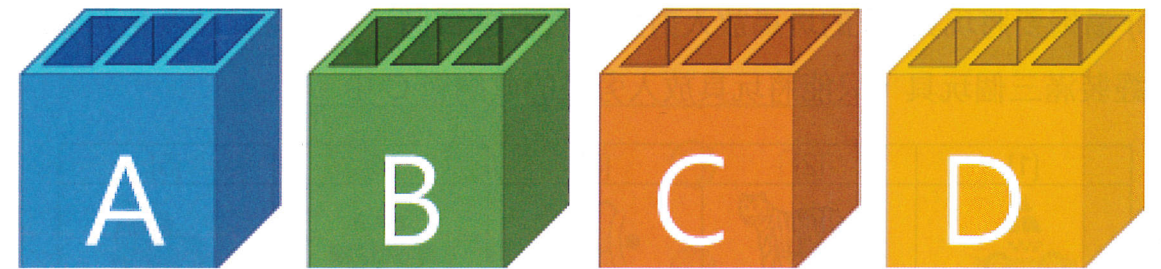
## 💡 關鍵字

八進制系統、二進制系統



# 46. 玩具收納

小海狸湯姆老是把玩具丟的到處都是，媽媽一怒之下準備了三個收納箱，要湯姆立刻收拾乾淨。



湯姆打算按照下面的規則收納玩具：

- ▶ 將交通工具放入收納箱 A
- ▶ 將有條紋的動物放入收納箱 B
- ▶ 將有斑點的動物放入收納箱 C

每個收納箱最多可以放三個玩具，如果某個收納箱滿了，他會把玩具放到旁邊的收納箱。例如：收納箱 A 滿了，玩具就放到收納箱 B；收納箱 B 滿了，玩具就放到收納箱 C；收納箱 C 滿了，玩具就放到收納箱 A。

湯姆總共有九個玩具，他收納的順序如下：

| [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|     |     |     |     |     |     |     |     |     |

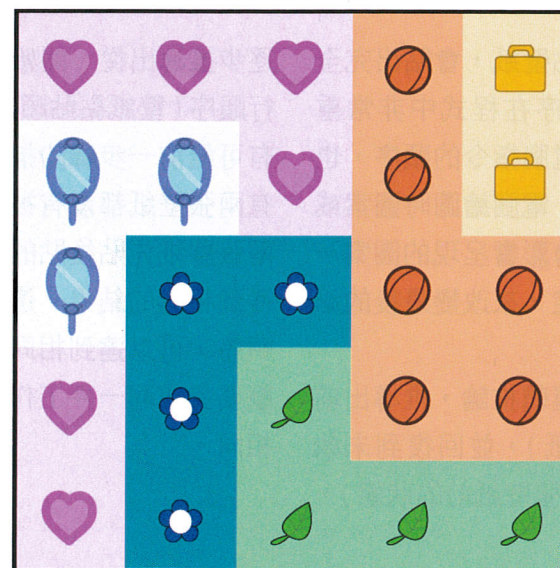
湯姆下次要拿玩具時，會在哪个收納箱中找到斑馬？(請回答大寫字母 A、B 或 C)



# 47. 貼壁紙

海狸小賓使用各種大小的正方形或長方形壁紙來貼牆壁，每種壁紙有不同的顏色及圖案，但每種花色都只有一張

小賓沒有剪刀，不能剪壁紙，但是可以把壁紙貼到先貼好的不同花色壁紙上面，蓋住下面壁紙的一部分。小賓貼出來的結果如下圖。



請問小賓是採用哪個順序依序將壁紙貼上牆壁的？

- (A)
- (B)
- (C)
- (D)

## ✔ 正確答案：C

玩具 [1] 到 [5] 都可以根據規則，直接放到對應的收納箱中，可是當：

- (I) 湯姆要將玩具 [6] 的消防車放入對應的收納箱 A 時，就會發現收納箱 A 已經裝滿三個玩具了，所以只好將玩具放入旁邊的收納箱 B。
- (II) 湯姆要將玩具 [9] 的斑馬放入對應的收納箱 B 時，會發現此時收納箱 B 也已經裝滿三個玩具，只能將玩具放入旁邊的收納箱 C 中。

|          |       |       |         |       |
|----------|-------|-------|---------|-------|
| [1]      | [2]   | [3]   | [4]     | [5]   |
|          |       |       |         |       |
| 收納箱 A    | 收納箱 B | 收納箱 B | 收納箱 A   | 收納箱 A |
| [6]      | [7]   | [8]   | [9]     |       |
|          |       |       |         |       |
| 收納箱 B(I) | 收納箱 C | 收納箱 C | 收納箱 (I) |       |

## i 在資訊科學上的意義

當有大筆資料需要儲存，而且會經常查詢搜索時，如何編排儲存方式將會是個關鍵。其中一個常用的概念就是透過雜湊函數 (hash function)。雜湊函數是一個將資料與一個固定範圍值的對應關係，藉由雜湊函數的計算，我們可以幫每一筆資料對應到雜湊表 (hash table) 中的一個位置 (bucket)。如果雜湊函數設計妥當，可以讓資料平均的存放到雜湊表的不同位置，減少未來搜尋資料所需耗費的時間。

然而，多筆資料可能被對應到相同位置，稱為碰撞 (collision)。如果雜湊表中對應位置的儲存區已經滿了，無法存放新的資料就會出現溢位 (overflow) 問題，此時則必須根據定義的溢位處理方式存放資料。在此題中我們使用的是線性探測 (linear probing)，也就是存放不下的資料，就往後尋找可以存放的位置。字典是生活中常見的雜湊表的例子，利用字母順序當作雜湊函數，讓我們可以快速在字典中找到想要找的單字。

## 💡 關鍵字

雜湊表、雜湊函數、溢位處理、線性探測

## ✔ 正確答案：A

畫公事包的壁紙是唯一的完整長方形壁紙，表示沒有被另一張壁紙覆蓋，因此應該是最後一張。拿掉公事包壁紙，籃球壁紙變成完整長方形，因此是前一張壁紙。依照此方式往前推，再前一張是葉子壁紙，再前一張是花卉壁紙，再前一張是鏡子壁紙，第一張是愛心壁紙。所以貼壁紙的順序依次為：愛心壁紙、鏡子壁紙、花卉壁紙、葉子壁紙、籃球壁紙、公事包壁紙。

## i 在資訊科學上的意義

此題目中不同順序來貼壁紙，會貼出完全不同的牆壁花樣。執行順序在程式中非常重要，你做事情的順序、給電腦指令的順序，也可能影響最後得到的結果。電腦繪圖時圖案或數位圖片重疊的順序，也會影響呈現的圖案。可透過不同圖層的順序設定，來改變最後的結果。

此外，這個問題可以透過邏輯推論，推導出前一步驟的指令（貼某張壁紙），並回復到未執行該指令前的狀態（未貼某張壁紙前的狀態），

逐步推導出從一開始到結束的唯一一串指令執行順序（壁紙黏貼順序）。然而實際的情況下，有可能前一步驟的指令有多種可能，例如最後有兩張壁紙都沒有被其他壁紙覆蓋到，那麼這兩張壁紙先貼後貼的順序就不只一種，但可以得到相同的結果。這表示有可能有不同的指令順序，可以達到相同的執行結果，這也是為什麼要完成同一件工作，給定的程式碼不一定都相同。