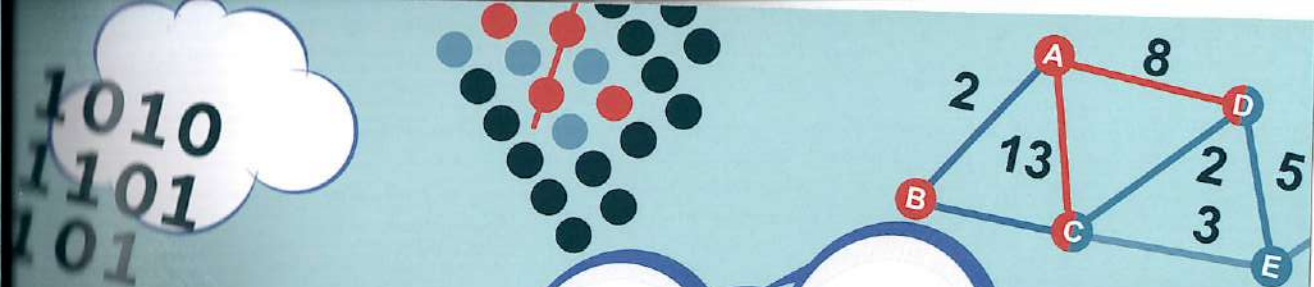




## 2021 國際運算思維挑戰賽

編輯團隊：李忠謀、柯佳伶、林于立、鄭琬馨  
花建民、蔡兆琛、陳佳宜、劉冠伶  
楊舒娟、葉宜珊、梅宜琇



# Bebras 國際運算思維挑戰賽

International Challenge on Informatics and  
Computational Thinking



## 2021題目解析

# 目錄

|                   |    |
|-------------------|----|
| 挑戰賽介紹             | 3  |
| 題本說明              | 4  |
| 題組介紹              | 6  |
| 1. 火山爆發           | 7  |
| 2.1 購物任務 I - 題組一  | 10 |
| 2.2 購物任務 II - 題組二 | 11 |
| 3. 最佳路徑           | 15 |
| 4. 汪達的新被子         | 17 |
| 5.1 一起午睡 - 題組一    | 21 |
| 5.2 一起午睡 - 題組二    | 22 |
| 6. 醫療中心           | 27 |
| 7.1 杯子蛋糕 I - 題組一  | 31 |
| 7.2 杯子蛋糕 II - 題組二 | 32 |
| 8. 液體密度           | 35 |
| 9.1 最長的順序 - 題組一   | 38 |
| 9.2 最長的順序 - 題組二   | 38 |
| 9.3 最長的順序 - 題組三   | 39 |
| 10. 購物去           | 43 |
| 11. 摔角訓練          | 47 |
| 12. 巴伐亞大地         | 51 |
| 13. 座位排列          | 53 |
| 14. 桌子工廠          | 57 |
| 15. 聖誕樹裝飾         | 61 |
| 16. 積木堆疊          | 63 |
| 17. 資源回收再升級       | 67 |
| 18. 搞笑濾鏡          | 71 |

|                  |     |
|------------------|-----|
| 19. 項鍊密碼         | 73  |
| 20. 七段顯示器        | 77  |
| 21.1 杜鵑找家 - 題組一  | 80  |
| 21.2 杜鵑找家 - 題組二  | 81  |
| 22. 復活節彩蛋        | 83  |
| 23. 影片觀看次數       | 87  |
| 24. 比一比          | 91  |
| 25. 偷畫賊          | 93  |
| 26. 特製冰淇淋機器      | 97  |
| 27. 觀光巴士         | 101 |
| 28. 排隊的魚         | 103 |
| 29. 嗨計程車         | 105 |
| 30. 蛋糕製作         | 109 |
| 31. 保險箱密碼        | 113 |
| 32. 柵欄塗油漆        | 115 |
| 33. 火星人密碼        | 119 |
| 34.1 康妮甜點屋 - 題組一 | 122 |
| 34.2 康妮甜點屋 - 題組二 | 123 |
| 35. 海狸歌唱比賽       | 127 |
| 36. 秘密訊息         | 131 |
| 37. 登山挑戰         | 133 |
| 38. 籌碼遊戲         | 137 |

## 挑戰賽介紹

國際運算思維挑戰賽 (International Challenge on Informatics and Computational Thinking, 簡稱 Bebras Challenge) 自 2004 年開始, 每年於 11 月的國際 Bebras 週 (World-Wide Bebras Week) 在全球各國同步舉行。

Bebras Challenge 採用淺顯易懂且生活化的情境式題型, 參與學生需運用抽象化、演算法設計、問題拆解、模式辨識、樣式一般化、自動化等運算思維 (Computational Thinking) 來解決問題挑戰。

Bebras Challenge 可以讓任課教師了解學生的運算思維知能, 發掘具備資訊科學性向的學生, 亦希望透過問題思考過程激起學生對資訊科學的學習興趣。



## 挑戰賽目標

### 激發學生對資訊科學之學習興趣

Bebras Challenge 藉由情境式的任務, 在挑戰的問題中融入資訊科學基本概念; 目的是讓學生了解生活中隨處可見資訊科學概念的運用, 認識相關概念具廣泛的應用性, 進而激發學生對資訊科學的學習興趣。

### 提升學生運用運算思維解決問題之能力

Bebras Challenge 的題目以家庭生活、團體合作、工作安排等生活情境, 引導學生思考進而解決問題。解題過程運用的是運算思維及問題解決能力, 學生僅需該年齡層的基本知識即可作答。從題目敘述中推理出問題重點及解題方向, 亦可引發學生高層次思考, 提升運用運算思維解決問題的能力。

### 降低學生對資訊科學之恐懼

Bebras Challenge 將抽象的資訊科學知識具體化, 以日常生活中會遇到的情境或故事呈現, 題組內容有趣生動, 有助於降低學生對資訊科學的學習焦慮。未曾受過資訊科學正式課程的學生亦能運用邏輯、歸納、推理、運算等能力進行解題, 讓學生對資訊科學的學習具有信心。

## 實施地點

Bebras Challenge 為了讓學生在熟悉的學習環境中進行，全球皆由學校教師於課堂中利用一節課的時間（國小挑戰時間 36 分鐘，國高中皆為 45 分鐘）於線上實施。由於需在課堂中進行，故僅能由教師為學生報名。挑戰賽答題需透過瀏覽器登入系統進行，使用電腦或是行動裝置皆可（為了瀏覽題目之視覺舒適度，行動裝置建議使用平板電腦）。

## 參與對象

Bebras Challenge 每年於 11 月舉辦，參與學生並無特定資格限制。只要是在學學生，皆可透過教師整班報名參加對應年級之挑戰賽組別。挑戰賽依學生年級共分 6 個組別，詳細分組情形請見計分方式。我國於 2016 年起每年舉辦 Benjamin、Cadet、Junior、及 Senior 4 個組別，未來將視參與情形向下推廣。

## 計分方式

Bebras Challenge 依年級分組，各組別的題目分成 3 種難度（易、中、難），根據題目難度有不同計分標準：答對給分、答錯扣分，略過未答則不給分亦不扣分，各組別分數計算分式如下表所示。Benjamin 組別的題數為 12 題，每個難度各 4 題；其他組別的題數為 15 題，每個難度各 5 題。為了避免總分有負分，各組別的預設總分為 60 分，各題皆答錯扣分得到最低 0 分，全部皆答對得到最高分 300 分。


|             |         | 難度   |    |    |    |    |    | 題數 |
|-------------|---------|------|----|----|----|----|----|----|
|             |         | 易    |    | 中  |    | 難  |    |    |
|             |         | 正確   | 錯誤 | 正確 | 錯誤 | 正確 | 錯誤 |    |
| Pre-Primary | 一、二年級   | 尚未開放 |    |    |    |    |    |    |
| Primary     | 三、四年級   | 尚未開放 |    |    |    |    |    |    |
| Benjamin    | 五、六年級   | 16   | -4 | 20 | -5 | 24 | -6 | 12 |
| Cadet       | 七、八年級   | 12   | -3 | 16 | -4 | 20 | -5 | 15 |
| Junior      | 九、十年級   | 12   | -3 | 16 | -4 | 20 | -5 | 15 |
| Senior      | 十一、十二年級 | 12   | -3 | 16 | -4 | 20 | -5 | 15 |

## 本書說明


運算思維挑戰賽題組介紹頁 (P.6) 列出本年度本國在各個組別中不同難度的選用題目。

| Benjamin 組       |                 |                 |
|------------------|-----------------|-----------------|
| 易                | 中               | 難               |
| 購物任務 a ..... 10  | 項鍊密碼 ..... 71   | 最佳路徑 ..... 15   |
| 杯子蛋糕 II ..... 30 | 七段顯示器 ..... 75  | 摔角訓練 ..... 45   |
| 搞笑濾鏡 ..... 69    | 保險箱密碼 ..... 109 | 杜鵑找家 ..... 79   |
| 觀光巴士 ..... 99    | 柵欄塗油漆 ..... 113 | 影片觀看次數 ..... 85 |

本書中的每一題，除了題目背景敘述及問題，會提供命題國家及題目對不同組別學生的難度、參考詳解、此題運用的資訊科學概念、及進階學習關鍵字等說明，如以下四個圖所標示。

 命題國家及題目難度



 參考詳解




正確答案是：

 此題運用之資訊科學概念



資訊科學上的意義

 進階學習的關鍵字



關鍵字



## Benjamin 組

| 易                | 中               | 難               |
|------------------|-----------------|-----------------|
| 購物任務 I ..... 10  | 項練密碼 ..... 73   | 最佳路徑 ..... 15   |
| 杯子蛋糕 II ..... 32 | 七段顯示器 ..... 77  | 摔角訓練 ..... 47   |
| 搞笑濾鏡 ..... 71    | 保險箱密碼 ..... 113 | 杜鵑找家 ..... 80   |
| 觀光巴士 ..... 101   | 柵欄塗油漆 ..... 115 | 影片觀看次數 ..... 87 |

## Cadet 組

| 易               | 中                | 難                |
|-----------------|------------------|------------------|
| 火山爆發 ..... 7    | 最長的順序 I ..... 38 | 汪達的新被子 ..... 17  |
| 杯子蛋糕 I ..... 31 | 復活節彩蛋 ..... 83   | 購物去 ..... 43     |
| 液體密度 ..... 35   | 影片觀看次數 ..... 87  | 摔角訓練 ..... 47    |
| 杜鵑找家 ..... 80   | 比一比 ..... 91     | 海狸歌唱比賽 ..... 127 |
| 觀光巴士 ..... 101  | 嗨計程車 ..... 105   | 登山挑戰 ..... 133   |

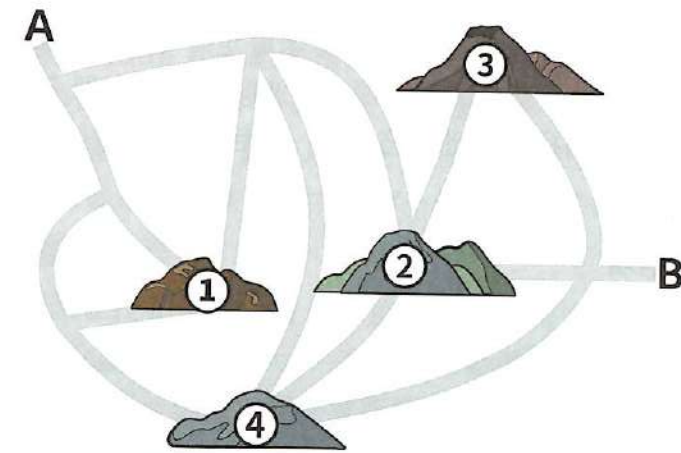
## Junior 組

| 易                 | 中                | 難               |
|-------------------|------------------|-----------------|
| 最長的順序 II ..... 38 | 購物任務 II ..... 11 | 汪達的新被子 ..... 17 |
| 復活節彩蛋 ..... 83    | 比一比 ..... 91     | 一起午睡 I ..... 21 |
| 排隊的魚 ..... 103    | 偷畫賊 ..... 93     | 醫療中心 ..... 27   |
| 火星人密碼 ..... 119   | 特製冰淇淋機器 ..... 97 | 座位排列 ..... 53   |
| 康妮甜點屋 I ..... 122 | 海狸歌唱比賽 ..... 127 | 登山挑戰 ..... 133  |

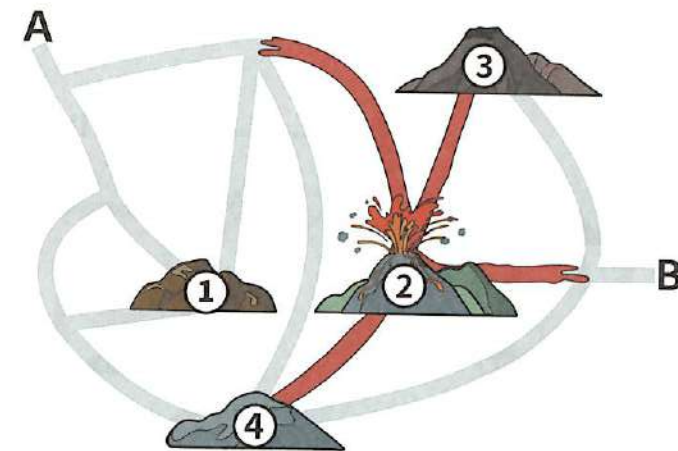
## Senior 組

| 易                | 中                | 難                  |
|------------------|------------------|--------------------|
| 巴伐亞大地 ..... 51   | 一起午睡 I ..... 21  | 醫療中心 ..... 27      |
| 積木堆疊 ..... 63    | 座位排列 ..... 53    | 最長的順序 III ..... 39 |
| 偷畫賊 ..... 93     | 資源回收再升級 ..... 67 | 桌子工廠 ..... 57      |
| 特製冰淇淋機器 ..... 97 | 蛋糕製作 ..... 109   | 聖誕樹裝飾 ..... 61     |
| 排隊的魚 ..... 103   | 秘密訊息 ..... 131   | 籌碼遊戲 ..... 137     |

## 1. 火山爆發



海狸島有四座火山；為了安全起見，若某座火山爆發，與該火山相鄰的路段會封閉，直到下一個分叉路口才能通行。以下圖為例，當火山 2 爆發時，紅色道路無法通行。



海狸迪諾正要從 A 點出發前往 B 點。請問哪兩座火山同時爆發，會讓迪諾無法到達 B 點？

A.



C.



B.



D.

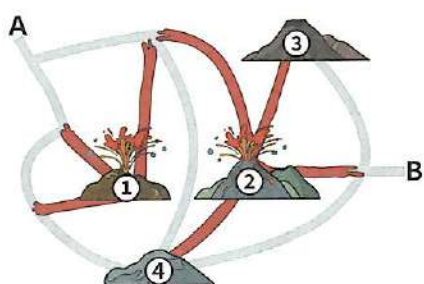




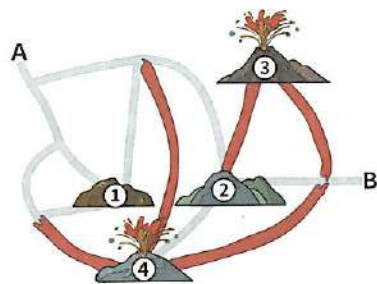
## 正確答案是：C

從 A 走到 B 點的路徑有很多條，然而不管怎麼走，可看出至少都要經過 2 或 4 這兩座火山，因此如果這兩座火山同時爆發，迪諾就無法走到 B 點，如下圖表示；至於其他選項，則都能夠找到其他的方式從 A 走到 B。

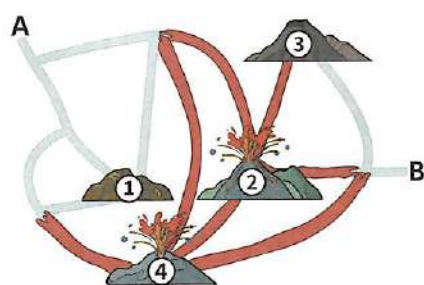
A 選項



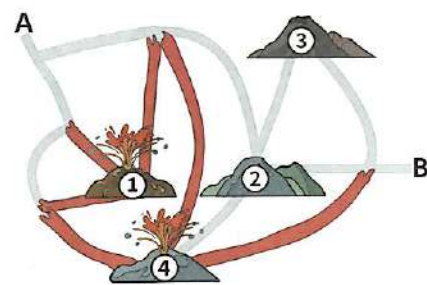
B 選項



C 選項



D 選項



## 關鍵字

圖、連通子圖

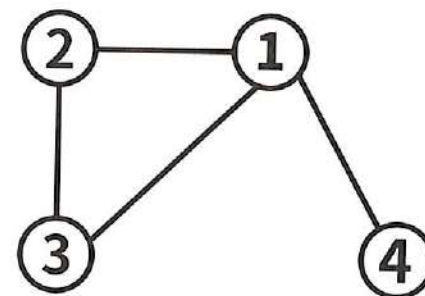


## 資訊科學上的意義

圖 graph 由一組節點 node 和一組連接這些節點的邊 edge 組成。在此題中火山即是「節點」，道路即是「邊」；如果將這個完整的圖，透過刪除選定的節點（火山爆發的位置）和其連結的邊，就可以產生一個比較小的圖，或稱之為子圖，因此這個問題即是設法從最初完整的圖中找到一個子圖，子圖中的任意兩點，最後無法經過一連串的節點而連接。

如果將這個圖中的某個節點刪除，會使得這個圖形成無法連通的圖，那麼這個點就稱之為關節點 articulation point。在這個問題中，每一個火山形成的節點都不是關節點。所以刪除任一個火山是不會斷開圖的。

上述概念可用於檢查網路系統的容錯能力。比方說以下的節點是在表示某個網路系統之間的連結關係，可以很明白的觀察出來，節點 1 是這個系統當中非常重要的節點，如果它產生錯誤甚至中斷，就會讓整個網路系統被斷開成兩個部分，因此就必須對這個節點進行補強或是增加備援的工作。





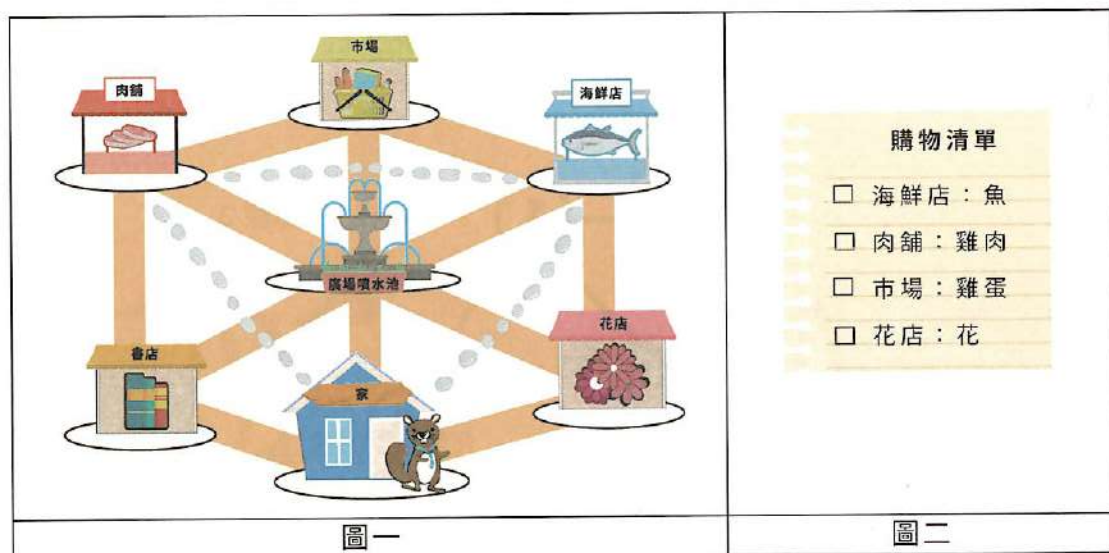
## 2.1 購物任務 I- 題組一

圖一是海狸家族居住的村莊地圖。

在每個地點間有泥巴路 (■) 或是石頭路 (●) 可供選擇。小海狸走泥巴路要花 5 分鐘，走石頭路要花 8 分鐘。

舉例來說，小海狸從廣場噴水池走到花店，或是從廣場噴水池走到書店，最快要花 5 分鐘；從海鮮店走回家，或是從肉舖走回家，最快要花 8 分鐘。

現在，媽媽要小海狸去幫她購物，購物清單如圖二，小海狸必須從家裡出發，買完東西後也要回到家。



小海狸可以自行安排購物順序，請問小海狸所需的最短步行時間是幾分鐘？

- A. 18
- B. 20
- C. 28
- D. 30



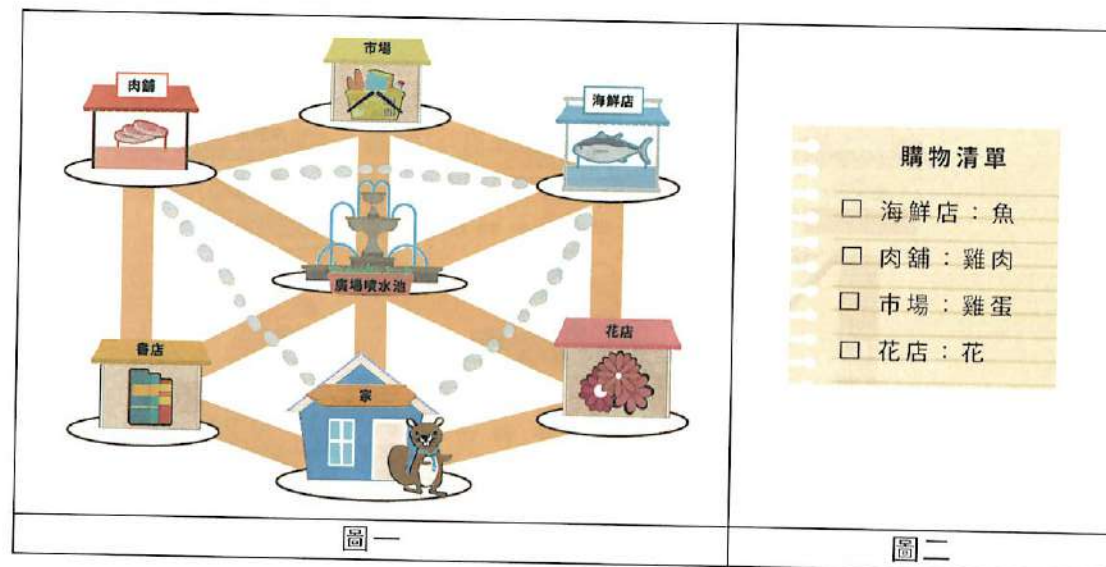
## 2.2 購物任務 II- 題組二

圖一是海狸家族居住的村莊地圖。

在每個地點間有泥巴路 (■) 或是石頭路 (●) 可供選擇。小海狸走泥巴路要花 5 分鐘，走石頭路要花 8 分鐘。

舉例來說，小海狸從廣場噴水池走到花店，或是從廣場噴水池走到書店，最快要花 5 分鐘；從海鮮店走回家，或是從肉舖走回家，最快要花 8 分鐘。

現在，媽媽要小海狸去幫她購物，購物清單如圖二，小海狸必須從家裡出發，買完東西後也要回到家。



小海狸可以自行安排購物順序，但為了要保持魚的新鮮，小海狸回家之前才會去海鮮店買魚。請問小海狸所需的最短步行時間是幾分鐘？（範圍 [5~99] 的一個整數。）

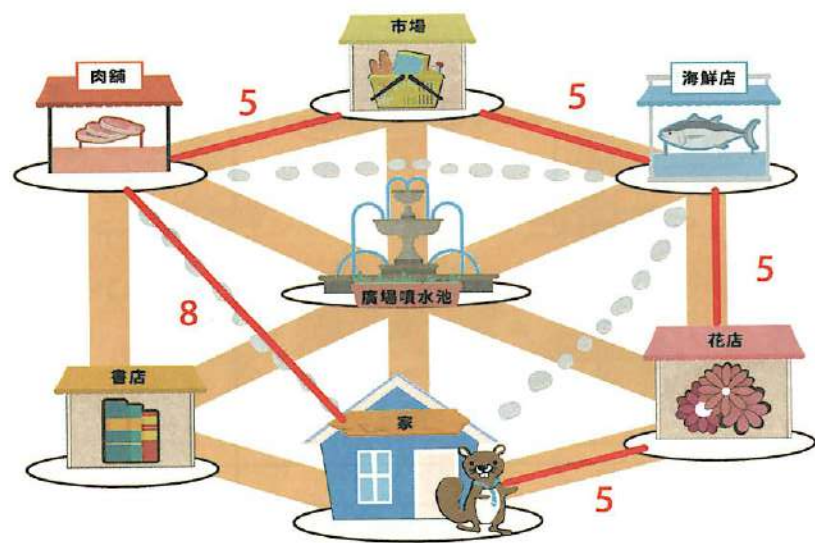


### 題組一的正確答案是：C. 28

小海狸從家裡出發，需經過 4 家店買完東西再回家，因此至少要走過 5 段小路，才能完成任務，而每條小路假設都走較快的泥巴路，也需要 5 分鐘，因此至少需要  $5 * 5 = 25$  分鐘完成任務，故選項 A 和 B 是錯誤的。

去肉舖或海鮮店，可選擇走兩段泥巴路 10 分鐘（分別通過書店或花店），或是直接走石頭路 8 分鐘，但因採買還需要去市場和花店，因此，去海鮮店最有效率的方式就是走泥巴路。同樣地，既然不必去書店，那從家裡走石頭路去肉舖，更有效率。因此，肉舖應該是購物任務的第一站或最後一站。

從商店步行到商店都走泥巴路，共需要  $5 * 4 = 20$  分鐘，而家裡到肉舖走石頭路需要 8 分鐘，因此總步行時間至少為  $8 + 20 = 28$  分鐘。小海狸選擇的路徑如右圖所示。



### 題組二的正確答案是：33

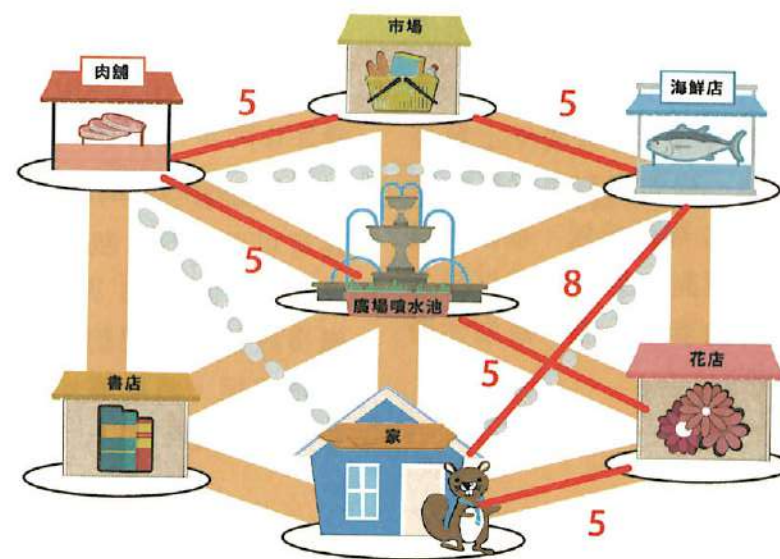
假設你是小海狸，為了魚的新鮮，海鮮店應該是你回家前的最後一站。

因此，原始任務可以分解為兩個有順序的子任務：(1) 在三個商店完成購物，(2) 買魚回家。

所需的最短步行時間，可拆成兩個子任務，每個子任務如下表所示：

| (1) 在三個商店完成購物  |        | (2) 買魚回家    |        | 時間總和 |
|----------------|--------|-------------|--------|------|
| 路線             | 最短步行時間 | 路線          | 最短步行時間 |      |
| 家 - 肉 - 花 - 市場 | 28     | 市場 - 海鮮 - 家 | 13     | 41   |
| 家 - 肉 - 市場 - 花 | 23     | 花 - 海鮮 - 家  | 13     | 36   |
| 家 - 花 - 市場 - 肉 | 20     | 肉 - 海鮮 - 家  | 16     | 36   |
| 家 - 花 - 肉 - 市場 | 20     | 市場 - 海鮮 - 家 | 13     | 33   |
| 家 - 市場 - 肉 - 花 | 25     | 花 - 海鮮 - 家  | 13     | 38   |
| 家 - 市場 - 花 - 肉 | 30     | 肉 - 海鮮 - 家  | 16     | 46   |

透過任務分解，我們可以探索每一個可能的路線來完成子任務，並計算完成每個子任務所需的最短時間，最後，我們可以反過來計算得到完成整個任務所需的最短時間。根據上表，小海狸完成購物任務，所需的最短步行時間路線如下圖所示：





## 資訊科學上的意義

圖 **graph** 是資訊科學中非常重要的資料結構，通常由頂點 **point** 或節點 **node**（通常以點表示）和一組連接這些節點的邊 **edge**（通常以線段表示）所組成，可以用來表示物件與物件之間的關係，邊也可以加上權重值 **weight**，用來表示兩個節點間的「距離」或是運送「成本」；在日常生活中，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，都可以使用圖來表示。

在題組一的任務中，給定每一個邊上的權重值（兩商店間的步行時間），目標是找到圖中兩節點間最短步行時間的路線；相當於是將每一個邊的權重值，設成兩地點間直接相通的距離，從圖尋找兩個節點間 **最短路徑 shortest path problem** 的問題。在圖論中，最短路徑問題的目標是找到兩節點之間距離最短的邊；有時邊有不同的權重，所以節點之間的距離必須乘以權重；在這種情況下，最短路徑問題是透過找到最小加權距離來解決的。

然而在現實生活中，當有成千上萬條路可以選擇時，我們可利用電腦技術輔助，例如 GPS 導航系統，系統內具備良好的路徑規劃演算法，用以找到最短路徑，將能事半功倍，為生活帶來莫大的便利。

在題組二的任務中，為了要保持魚的新鮮，小海狸回家之前才會去海鮮店買魚，就是本題的條件限制。然而在資訊科學中，這類問題被稱為 **滿足條件限制問題 Constraint Satisfaction Problems CSP**。為解決 CSP 問題，必須滿足若干條件限制，才能達成目標，而這種問題可能有很多種解法。CSP 問題經常表現出高複雜性，並且需要在合理的時間內解決問題。

CSP 著名的問題有八皇后問題 8-queens problem、數獨 sudoku 及地圖著色 map coloring 等資源配置最佳化問題。

然而在現實生活中，當有成千上萬條路可以選擇時，我們可利用電腦技術輔助，例如 GPS 導航系統，系統內具備良好的路徑規劃演算法，用以找到最短路徑，將能事半功倍，為生活帶來莫大的便利。



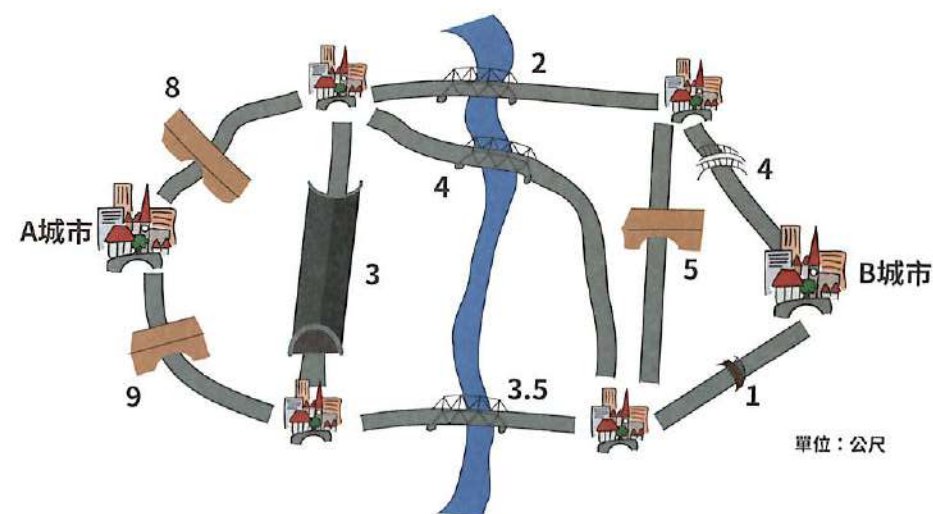
## 關鍵字

題組一：圖、最短路徑

題組二：圖、最短路徑、滿足條件限制問題

## 3. 最佳路徑

海狸國的公路都會經過橋或隧道，為了避免車子載貨後太高無法通過，下面這張地圖標示了每條公路通行的高度限制。

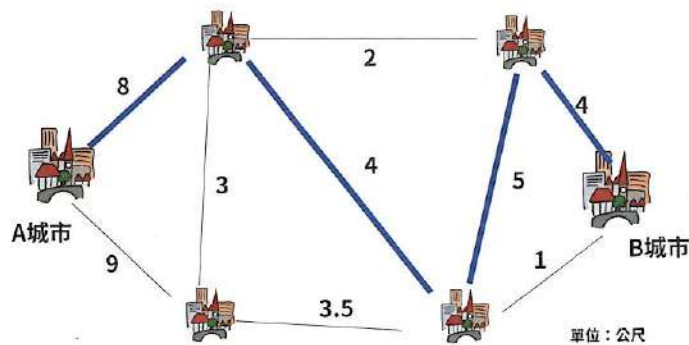


車子要從 A 城市到 B 城市，請問載貨後的高度限制是幾公尺？（請填入數字，可含小數點符號。）



### 正確答案是：4

從 A 城市到 B 城市，卡車一定要過河。而其中只有三條橋能夠過河，這三條橋的高度限制分別是 2、4 及 3.5 公尺。這告訴我們任何高度大於 4 的卡車都無法從 A 城市到達 B 城市。那高度 4 的卡車有機會達成任務嗎？有的！如果卡車用下圖的路線，沿路所經過的高度限制會是 8、4、5 及 4 公尺。最小的高度限制是 4 公尺，所以高度 4 公尺的卡車可以從 A 城市到達 B 城市。另一個不會漏掉路徑的方式是有系統地列出所有從 A 城市到 B 城市（或 B 城市到 A 城市）的路徑，並取路徑中所經公路最低的高度以找出高度限制。



### 資訊科學上的意義

圖 graph 由一組節點 node 和一組連接這些節點的邊 edge（通常以線段表示）組成。在此題中城市即是「點」，公路即是「邊」。將本題城市間的連結畫成一個圖，用深度優先搜尋 DFS 或是廣度優先搜尋 BFS，可以有系統地列出所有從起點到終點的走訪路徑，再從中檢查高度限制。

這題的靈感來自電腦網路（有線網路、wifi）的資料上傳速度限制，資料在網路上的傳送是以「資料封包」為單位，經過「路由器」一再轉送，送達目的地；通常由路由器及路由演算法來指揮傳送路徑。傳輸資料時的一個重要的考慮因素在於網路頻寬（高度限制），頻寬決定了在一定時間內資料封包能夠以多快的速度被送到另一端，這與題目中限制卡車高度的橋樑有點相像。生活中也可將交通網及運輸量（例如客運的載客數）畫成圖，在規劃團體旅遊時，從中尋找某一條交通路徑，能夠將最多的乘客同時從起點經過轉運到達終點。



### 關鍵字

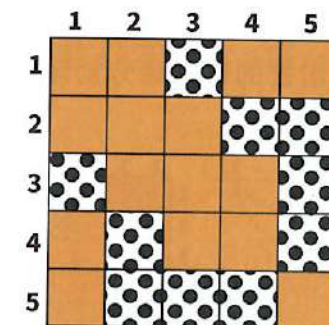
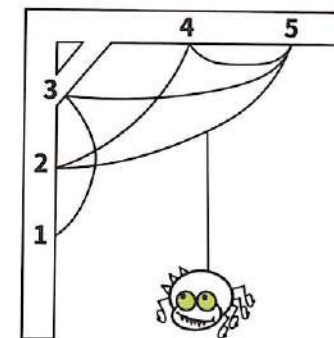
圖、深度優先搜尋、路由器、路由演算法

## 4. 汪達的新被子

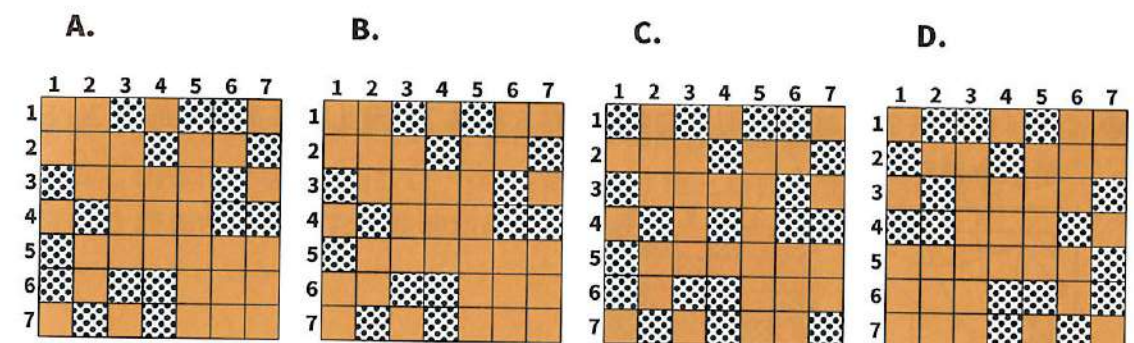
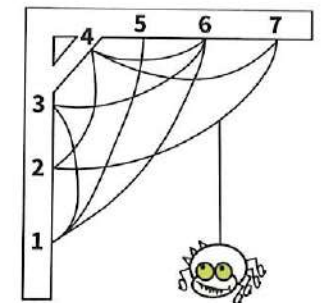
蜘蛛汪達是被子設計師，她會先用蜘蛛網畫設計圖，再依以下規則織成被子：

- 若蜘蛛網和牆角有  $N$  個連接點，就要織出  $N$  格長乘  $N$  格寬的被子。
- 若蜘蛛網中不同的兩個點（點  $X$  和點  $Y$ ）之間有蜘蛛絲相連，則被子上的  $(X, Y)$  和  $(Y, X)$  這兩格就要織成點狀圖案，其餘的格子要織成實心圖案。

下圖就是汪達畫的設計圖，以及織出相對應的被子。



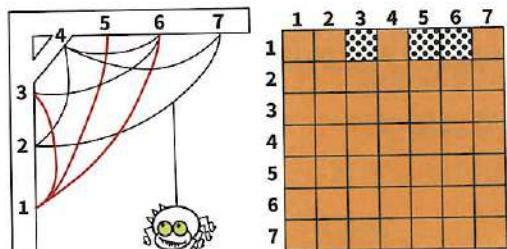
依據汪達新畫的蜘蛛網設計圖，汪達將織出哪一條被子？



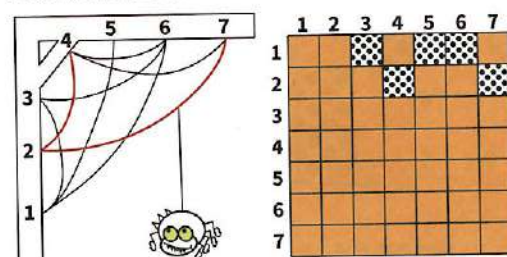


## 正確答案是：A

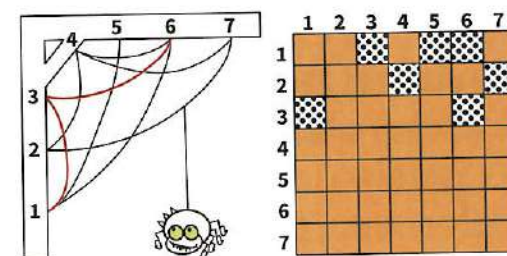
- 從點 1 連出去的蜘蛛絲分別連向點 3、點 5 跟點 6，因此被子第一列的第 3、5、6 欄都會是點狀圖案。



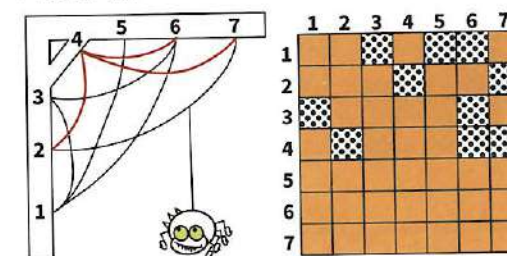
- 從點 2 連出去的蜘蛛絲分別連向點 4 跟點 7，因此被子第二列的第 4、7 欄都會是點狀圖案。



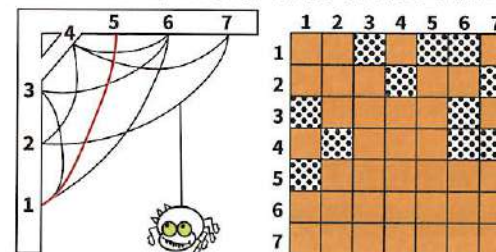
- 從點 3 連出去的蜘蛛絲分別連向點 1 跟點 6，因此被子第三列的第 1、6 欄都會是點狀圖案。



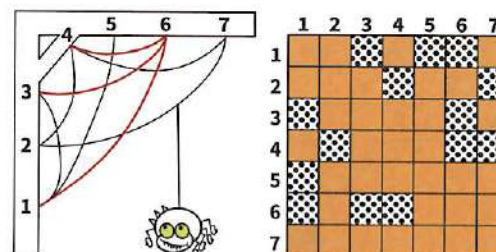
- 從點 4 連出去的蜘蛛絲分別連向點 2、點 6 跟點 7，因此被子第四列的第 2、6、7 欄都會是點狀圖案。



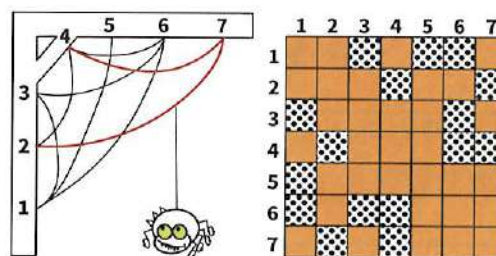
- 從點 5 連出去的蜘蛛絲只有連向點 1，因此被子第五列的第 1 欄是點狀圖案。



- 從點 6 連出去的蜘蛛絲分別連向點 1、點 3 跟點 4，因此被子第六列的第 1、3、4 欄都會是點狀圖案。



- 從點 7 連出去的蜘蛛絲分別連向點 2 跟點 4，因此被子第七列的第 2、4 欄都會是點狀圖案。



選項 B 錯誤的原因在於 (1,6) 跟 (6,1) 這兩格應該是點狀圖案。

選項 C 錯誤的原因在於 (1,1)、(4,4) 跟 (7,7) 這三格應該是無點狀圖案。

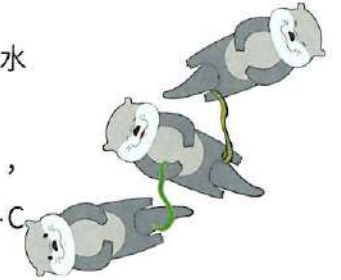
選項 D 錯誤的原因是它被整個旋轉了 90 度。



## 5.1 一起午睡 - 題組一

奧特村的水獺在午睡時會用海藻將彼此綁住，以防被海水沖走。

然而為了避免打結，如果任兩隻水獺已經直接或間接相連，就不會再綁其他的海藻。例如：若水獺按照 A-B, A-C, B-C 的順序相遇，則水獺捆綁海藻的方式如下表所示：



| 1                       | 2                       | 3                                    |
|-------------------------|-------------------------|--------------------------------------|
| A-B 相遇                  | A-C 相遇                  | B-C 相遇                               |
|                         |                         |                                      |
| 水獺 A 和水獺 B 用一條海藻互相綁住彼此。 | 水獺 A 和水獺 C 用一條海藻互相綁住彼此。 | 水獺 B 和水獺 C 已透過水獺 A 相連，故不會再用其他海藻綁住彼此。 |

今天午睡前，水獺按照以下順序相遇：A - B, A - C, B - C, D - E, A - E, D - F, A - F。

請問今天午睡時，水獺 A 身上有幾條海藻與其他水獺相連？



## 資訊科學上的意義

這個蜘蛛網可以被想成是一個圖 graph，一個圖包含了點（蜘蛛網的點）和邊（連接兩個點之間的蜘蛛絲），它可以用來呈現個體與個體之間的關係。例如社交媒體中誰和誰是朋友關係，或者國家與國家之間的航線等。

在本題中，汪達的被子展現了其中一種圖的表現方式，叫做 **相鄰矩陣 adjacency matrix**。相鄰矩陣是將圖論與矩陣理論連結在一起的一個重要媒介。它提供一種有效率的方式來表達圖的結構，讓我們可以很快就看出某個邊是否存在，以及總共有幾個邊跟某個特定點相連等。

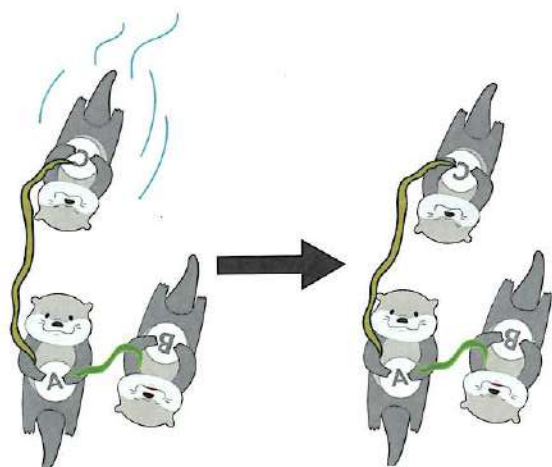


## 關鍵字

相鄰矩陣



## 5.2 一起午睡 - 題組二



奧特村的水獺在午睡時會用海藻將彼此綁住，以防被海水沖走。然而這個村莊正逢海藻短缺，七隻水獺必須共用六條海藻。幸運的是，不相鄰的水獺可以經由其他水獺相連，如上圖水獺 B 和水獺 C 可以透過水獺 A 相連。

今天午睡前，水獺 A~G 依下列順序相遇：A-C, B-C, D-E, E-F, A-G, F-G, A-B, F-A, D-A, E-A。

哪一對水獺不需要用海藻連接？

- A. E-F
- B. A-G
- C. F-G
- D. A-B



## 題組一正確答案是：3

依照相遇的順序，水獺連接的方式如下：

|  |  |
|--|--|
|  | 當 A 遇到 B，他們用海藻互相綁住彼此，A 這點有一條海藻綁住。      |
|  | 當 A 遇到 C，他們用海藻互相綁住彼此，A 這點有兩條海藻綁住。      |
|  | 當 B 遇到 C，因為他們已經經由 A 連接，他們不會在彼此身上綁任何海藻。 |
|  | 當 D 遇到 E，他們用海藻互相綁住彼此。                  |
|  | 當 A 遇到 E，他們用海藻互相綁住彼此，A 這點有三條海藻綁住。      |
|  | 當 D 遇到 F，他們用海藻互相綁住彼此。                  |
|  | 當 A 遇到 F，因為他們已經經由 E 連接，他們不會在彼此身上綁任何海藻。 |

因此，只有三條海藻綁在水獺 A 身上。



## 題組二正確答案是：D. A-B

有七隻水獺但只有六條海藻，為了讓所有的水獺保持連接避免分開，我們必須謹慎地使用海藻，只有在兩隻水獺沒有連接時，才需要使用海藻。

|  |                                    |
|--|------------------------------------|
|  | 當 A 遇到 C，他們用海藻互相綁住彼此。              |
|  | 當 B 遇到 C，他們用海藻互相綁住彼此。              |
|  | 當 D 遇到 E，他們用海藻互相綁住彼此。              |
|  | 當 E 遇到 F，他們用海藻互相綁住彼此。              |
|  | 當 A 遇到 G，他們用海藻互相綁住彼此。              |
|  | 當 F 遇到 G，他們用海藻互相綁住彼此。              |
|  | 當 A 遇到 B，因為他們已經連接了，他們不會在彼此身上綁任何海藻。 |
|  | 當 F 遇到 A，因為他們已經連接了，他們不會在彼此身上綁任何海藻。 |
|  | 當 D 遇到 A，因為他們已經連接了，他們不會在彼此身上綁任何海藻。 |
|  | 當 E 遇到 A，因為他們已經連接了，他們不會在彼此身上綁任何海藻。 |

選項 A：E 和 F 必須用海藻綁住，否則 F 會被海水沖走。

選項 B：A 和 G 必須用海藻綁住，否則 G 會被海水沖走。

選項 C：F 和 G 必須用海藻綁住，否則 ACB 和 DEF 會被海水沖走。

選項 D：當 A 遇到 B 時，因為他們已經連接了，不需要在 A 和 B 彼此綁任何海藻。



## 資訊科學上的意義

在資訊科學中，圖 graph 是由點和點之間的邊組成的資料結構，樹是無環狀且無方向的圖。一個圖的生成樹 spanning tree 包含每個頂點和屬於它的每條邊的樹。換句話說，生成樹為不包含循環且包含所有邊的最大集合的圖，也是連接圖中所有頂點的所有邊的集合。

在此任務中，我們可以將所有水獺視為在圖中的頂點，兩隻水獺互相綁住海藻可以看作是圖中的一邊。因此，我們試圖在此任務中找到圖的生成樹。

在這個任務中，我們以在兩隻水獺之間綁海藻來合併兩組集合。不相交集 disjoint sets 是實現 kruskal 演算法在圖形中找到最小生成樹 Minimal Spanning Tree MST 的重要資料結構。最小生成樹簡單理解就是給定一個帶有權值的連通圖，從眾多的生成樹中篩選出權值總和最小的生成樹，即為該圖的最小生成樹。以最少邊的組合連接所有頂點，找到最小生成樹的經典方法為 kruskal 演算法。不相交集用來防止選定邊中的循環，這正是我們在此任務中避免打結的方法。

最小生成樹可以應用於網路規劃，我們可以將電腦視為圖中的一個頂點，當我們想要以最小成本將所有電腦連接到網路，我們可以找出圖中的最小生成樹以達成我們所需的目的。



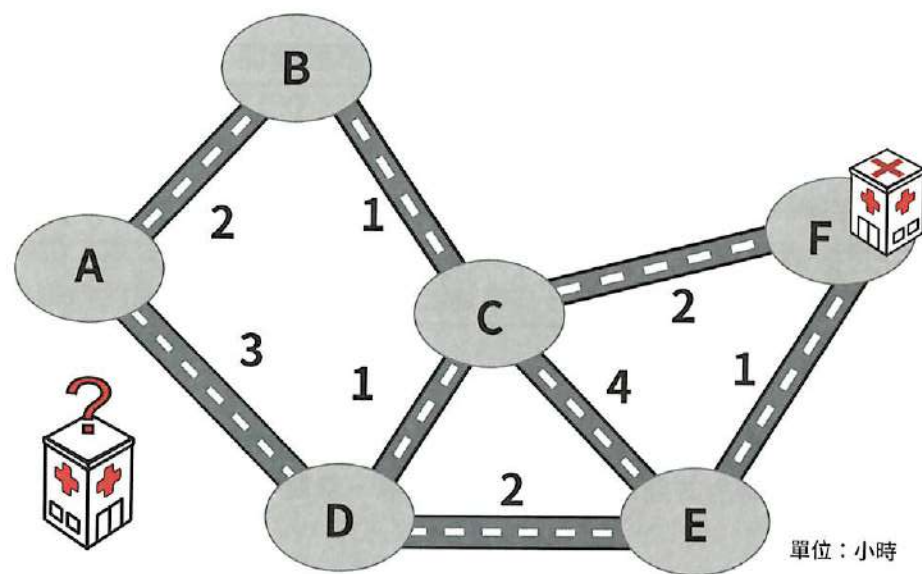
## 關鍵字

生成樹、圖



## 6. 醫療中心

一個地區有六個主要城市，下圖為該地區的地圖，圖中數字代表從一個城市到另一個城市所需的時間。有時你必須穿過一個城市才能到達另一個城市。其中只有兩個城市可以開設醫院。當人們需要緊急醫療時，他們必須盡快抵達醫院就醫。我們的目標是要讓每個城市的居民都可以在最短時間內到達其中一家醫院。



現在城市 F 有一家醫院，請問另一家醫院必須開設在哪個城市，才能達到目標？



## 正確答案是：B. 城市 B

由於第一家醫院開設在城市 F，因此有五種可能來開設第二家。對於每種可能性，我們檢查從每個城市到其中一家醫院所需的最短時間。在下表中，各欄顯示了兩家醫院的可能位置，以及每一列顯示各城市到達其中一家醫院所需的最短時數。

| 出發城市<br>到其中一家醫院的最短<br>時數 | 建置醫院的兩個城市 (一家醫院已指定在城市 F) |       |       |       |       |
|--------------------------|--------------------------|-------|-------|-------|-------|
|                          | A 和 F                    | B 和 F | C 和 F | D 和 F | E 和 F |
| A                        | 0                        | 2     | 3     | 3     | 5     |
| B                        | 2                        | 0     | 1     | 2     | 3     |
| C                        | 2                        | 1     | 0     | 1     | 2     |
| D                        | 3                        | 2     | 1     | 0     | 2     |
| E                        | 1                        | 1     | 1     | 1     | 0     |
| F                        | 0                        | 0     | 0     | 0     | 0     |
| 到達其中一家醫院的最<br>長時數        | 3                        | 2     | 3     | 3     | 5     |

如最後一列所示，將第二家醫院開設在城市 B 是最佳解決方案。每個城市的人都可以在不超過 2 小時的時間內到達 B 與 F 其中一家醫院。



## 資訊科學上的意義

圖 **graph** 用於表示物件與物件之間的關係，是圖論的基本研究對象。圖由頂點或節點 **node** (通常以點表示) 和一組連接這些節點的邊 **edge** (通常以線段表示) 組成。在此題中城市即是「點」，城市間連接的道路即是「邊」。

許多現實生活中的問題以圖表示後，就能透過電腦程式解決。常用圖來解決尋找適當的設施位置的問題，設置設施時，希望盡可能降低最差情況下的設施距離。例如，將醫院設置在適當的點，以達到讓每個城市的居民都能在最短時間內抵達的目標。

在圖論中，計算 **k-中心問題** 是研究組合最佳化的問題，舉例來說，給定  $n$  個城市及其距離的连接圖，我們必須從中選取  $k$  個城市來設置倉庫，在最差情況下，使得各城市與離它最近倉庫的距離達到最小，意即任何城市到達倉庫，都不能比這個最差情況的最小距離還要來得差，而本題尋找醫院設置的適當地點，也屬於 **k-中心問題**，最後選定建置醫院的 B 和 F 城市，從上表中發現，任何城市到達其中一家醫院的最長時數為 2，而這個數字 2，是最長時數中最小的值，因此這個選項是最佳方案。



## 關鍵字

k 中心問題、圖

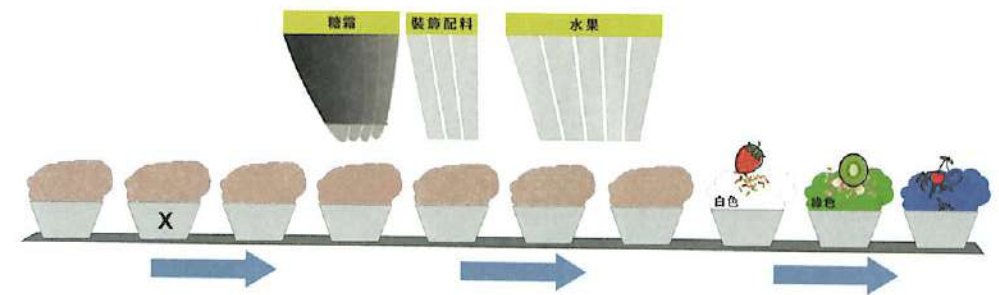


## 7-1. 杯子蛋糕 I- 題組一

海狸麵包店正在製作杯子蛋糕，每個杯子蛋糕上都添加三種材料：第一層擠糖霜、第二層撒裝飾配料、第三層加水果，每一層材料都會依照一定的順序變化。

- 糖霜變化的順序為：綠色→白色→紅色→藍色→(回到綠色，依此類推)
- 裝飾配料變化的順序為：七彩粉(🌈)→巧克力脆片(🍫)→烤堅果(🌰)→(回到七彩粉，依此類推)
- 水果變化的順序為：櫻桃(🍒)→奇異果(🥝)→草莓(🍓)→橘子(🍊)→藍莓(🫐)→(回到櫻桃，依此類推)

下圖是正在運作中的杯子蛋糕生產線，杯子蛋糕由左往右移動。



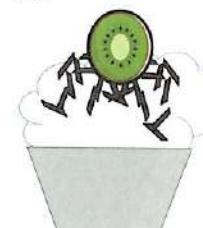
標示 X 的杯子蛋糕成品會是什麼樣子？

A.



紅色糖霜、七彩粉、橘子

B.



白色糖霜、巧克力脆片、奇異果

C.



藍色糖霜、烤堅果、草莓

D.



藍色糖霜、七彩粉、橘子

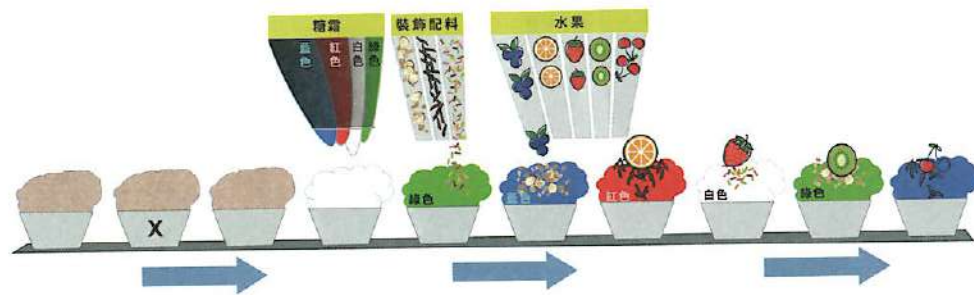


## 7-2. 杯子蛋糕 II- 題組二

海狸麵包店正在製作杯子蛋糕，每個杯子蛋糕上都會添加三種材料：第一層擠糖霜、第二層撒裝飾配料、第三層加水果，每一層材料都會依照一定的順序變化。

- 糖霜變化的順序為：綠色→白色→紅色→藍色→(回到綠色，依此類推)
- 裝飾配料變化的順序為：七彩粉(🌈)→巧克力脆片(🍫)→烤堅果(🌰)→(回到七彩粉，依此類推)
- 水果變化的順序為：櫻桃(🍒)→奇異果(🥝)→草莓(🍓)→橘子(🍊)→藍莓(🫐)→(回到櫻桃，依此類推)

下圖是正在運作中的杯子蛋糕生產線，杯子蛋糕由左往右移動。



標示 X 的杯子蛋糕成品會是什麼樣子？

- A. 紅色糖霜、七彩粉、橘子
- B. 白色糖霜、巧克力脆片、奇異果
- C. 藍色糖霜、烤堅果、草莓
- D. 藍色糖霜、七彩粉、橘子



## 正確答案

題組一答案：D

題組二答案：D

正確答案是 D，藍色糖霜 + 七彩粉 + 橘子。

因為杯子蛋糕 X 不會輪到紅色、白色糖霜，所以選項 A、B 皆不正確，選項 C 的烤堅果和草莓正好輪到 X 的前一個杯子蛋糕，因此 C 也不正確。

杯子蛋糕 X 添加材料的說明如下：

1. 擠完白色糖霜後，接著是紅色→藍色，X 輪到藍色糖霜。
2. 撒完七彩粉後，接著是巧克力脆片→烤堅果→七彩粉，X 輪到七彩粉。
3. 在藍莓之後接著是櫻桃→奇異果→草莓→橘子，X 輪到橘子。





## 資訊科學上的意義

本任務說明程式設計中的 **指令 instruction**、**演算法 algorithm** 和變數的儲存狀態 remembering state。

在本任務中「擠藍色糖霜」、「撒七彩粉」、「加奇異果」等就像是程式中的指令，告訴電腦要做什麼，而添加材料的順序，如：綠色→白色→紅色→藍色，或七彩粉→巧克力脆片→烤堅果等就是演算法。

演算法是指程式執行的步驟，按照順序執行指令是電腦中非常重要的概念——我們告訴電腦該做什麼，而它就依照指令一步一步執行。在程式語言中指令的順序也非常重要，不同的順序會影響程式執行的結果，就如同本任務中每一層每一種材料添加的順序影響了最終蛋糕的結果。

儲存狀態是程式語言中必要的一部分。電腦程式會將資訊儲存起來，並透過改變它的狀態來影響程式的行為，其中最常見的儲存方式是將特定的狀態或數值儲存在 **變數 variable** 中。

電腦經過特定步驟處理資料或解決問題的過程，即是演算法的基礎。演算法的指令可以改變變數的內容，來記錄執行某個動作後的結果，每個杯子蛋糕就像變數一樣用來儲存資料，而材料就像放置在這些變數裡的資料。

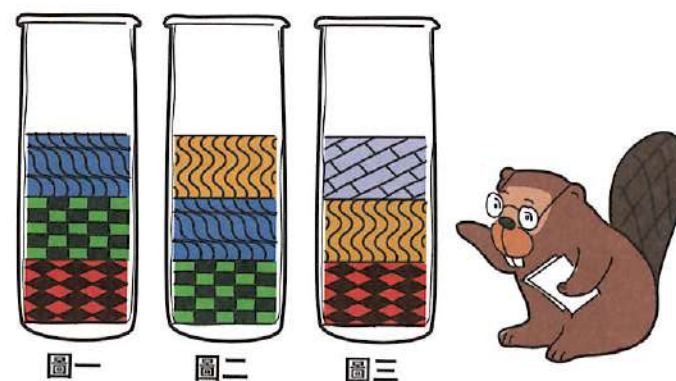


## 關鍵字

指令、變數、演算法

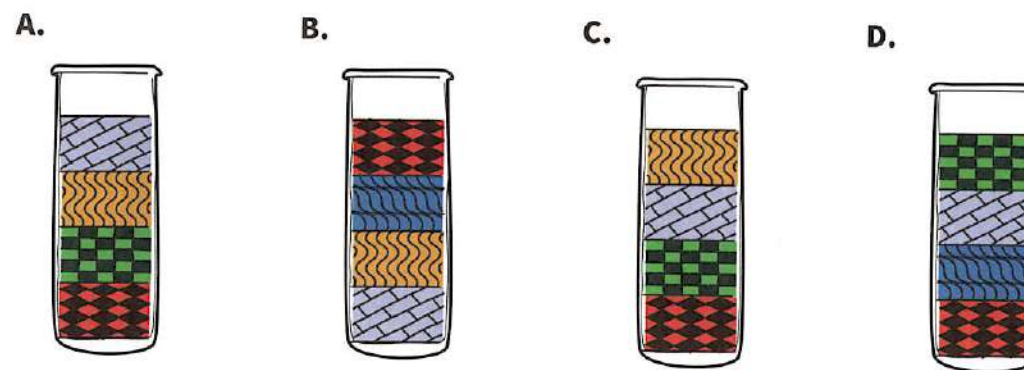
## 8. 液體密度

海狸馬克記得課堂中，科學老師將幾種不同的液體分別倒入以下三個試管中，來示範「在同一個容器中，密度較高的液體會沉到密度較低的液體之下」的現象：



馬克的朋友明天生日，他想要在容器中倒入四種不同的液體，作為朋友的生日禮物。

根據科學老師的示範，下列哪個容器中液體的上下順序是正確的？





## 正確答案是：A

由科學老師示範的三個試管，我們可以推導出下方的表格：

|      | 由圖一可知 | 由圖二可知 | 由圖三可知 | 所有液體 |
|------|-------|-------|-------|------|
| 密度較低 |       |       |       |      |
|      |       |       |       |      |
|      |       |       |       |      |
|      |       |       |       |      |
| 密度較高 |       |       |       |      |

因此，只有 (A) 選項試管中液體的上下順序才是正確的。此外，其他選項中的液體正確的上下順序分別為：

選項 B 的正確順序



選項 C 的正確順序



選項 D 的正確順序



## 資訊科學上的意義

在此挑戰題中，我們必須找出各試管之間相同的模式，從已知的三個試管中推導出五種液體的密度大小關係，這個過程就是 **模式識別 pattern recognition**。在此題任務中，由觀察得知液體的密度為需識別的**模式**，我們可以從已知的三個容器中知道五種液體的密度關係，再由這些關係找出適合的答案。**約束 constraint** 是必須滿足問題條件的解決方法。

在此任務中，液體密度為液體排序的主要約束條件。模式識別是指對表徵事物或現象各種形式的信息進行處理和分析，以對事物或現象進行描述、辨認、分類和解釋的過程，模式識別只需尋找問題中的模式並使用適合的方法來解決問題。

在現實生活中，我們必須查看不同種類的模式，並提取適用於所有模式的通用規則，辨識出模式後，把問題解決過程中反覆執行的操作模組化，在許多問題中模式識別常常是解決問題或是拆解問題的關鍵能力。

模式識別可用於文字和語音識別、人臉、指紋識別、遙感和醫學診斷等方面，在我們日常生活中十分常見。當我們手持電子護照快速通關時，通關處需要掃描指紋和面相來確認遊客的身份這就是指紋識別和人臉識別。當你拿起蘋果手機跟語音助手 siri 對話的時候，它能準確地辨別你的話語從而做出回應，這就是語音識別；Youtube、Netflix 等會根據您的興趣來推薦您可能會有興趣的影片與節目同時也是模式識別的應用。



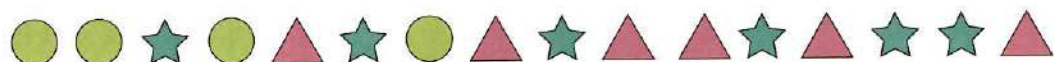
## 關鍵字

模式辨別、約束、排序

## 9-1. 最長的順序 - 題組一

### 題組一

下面是由三種圖形所組成的圖形順序，你可以在這個順序中挑選兩個圖形，替換成其他任意圖形。



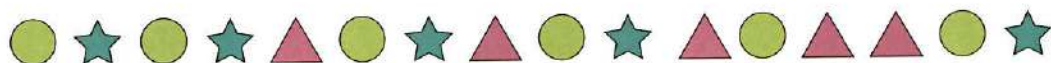
替換後，最多有幾個一樣的圖形會連續排在一起？

- |           |           |
|-----------|-----------|
| A. 連續 4 個 | C. 連續 6 個 |
| B. 連續 5 個 | D. 連續 7 個 |

## 9-2. 最長的順序 - 題組二

### 題組二

下面是由三種圖形所組成的圖形順序，你可以在這個順序中挑選三個圖形，替換成其他任意圖形。



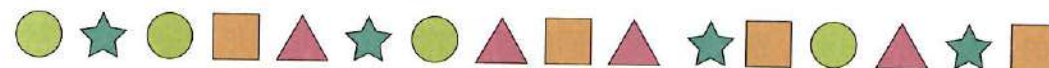
替換後，最多有幾個一樣的圖形會連續排在一起？

- |           |           |
|-----------|-----------|
| A. 連續 6 個 | C. 連續 8 個 |
| B. 連續 7 個 | D. 連續 9 個 |

## 9-3. 最長的順序 - 題組三

### 題組三

下面是由四種圖形所組成的圖形序列，你可以在這個順序中挑選三個圖形，替換成其他任意圖形。



替換後，最多有幾個一樣的圖形會連續排在一起？

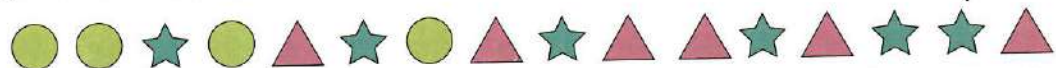
(請填入範圍 [1~16] 的半形整數數字)



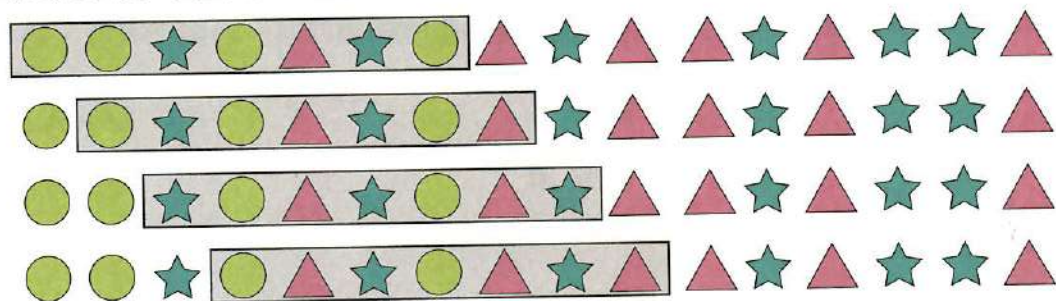
### 題組一的正確答案是：C. 連續 6 個

為了說明原因，我們必須證明兩件事情：(1) 長度為 6 的連續圖形是可能的。  
(2) 長度為 7 以上的連續圖形是不可能的。

● 第一件事情很容易證明。下圖就是一個可能的例子：



● 要證明第二件事情的話，我們可以先檢查在這個圖形順序中的每個「長度為 7 的片段」。如果某個片段可以變成長度為 7 的連續圖形，那麼扣除我們可以改變的 2 個之外，它至少要含有 5 個相同的圖形才行。在這個圖形順序中，總共可以切出 10 個「長度為 7 的片段」，下面列出其中幾個，以此類推。



結果我們可以發現：沒有任何一個片段裡面含有 5 個以上的相同圖形。也就是說：這個圖形順序就算在改變兩個圖形後，也沒辦法找到長度為 7 的連續圖形。而既然要找到長度為 7 的連續圖形已經不可能，那麼 8 以上就更不可能了。

綜上所述，我們可以得知：這個圖形順序在經過我們替換兩個圖形之後，所能得到的連續圖形，其最大長度就是 6。



### 題組二的正確答案是：B. 連續 7 個

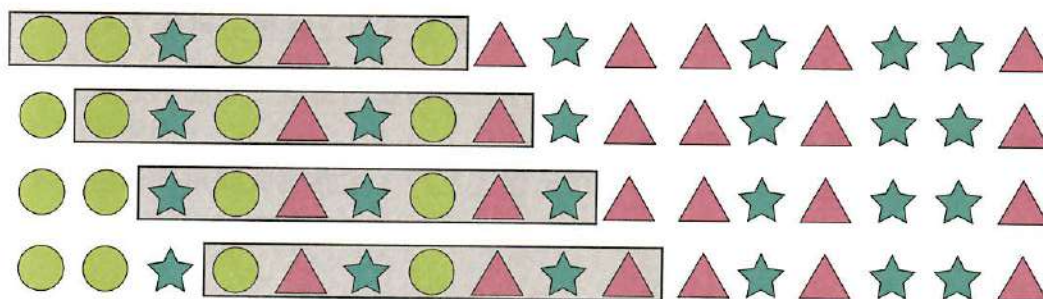
為了說明原因，我們必須證明兩件事情：(1) 長度為 7 的連續圖形是可能的。  
(2) 長度為 8 以上的連續圖形是不可能的。

● 第一件事情很容易證明。下圖就是一個可能的例子：



● 要證明第二件事情的話，我們可以先檢查在這個圖形順序中的每個「長度為 8 的片段」。如果某個片段可以變成長度為 8 的連續圖形，那麼扣除我們可以改變

的 3 個之外，它至少要含有 5 個相同的圖形才行。在這個圖形順序中，總共可以切出 9 個「長度為 8 的片段」，下面列出其中幾個，以此類推。



結果我們可以發現：沒有任何一個片段裡面含有 5 個以上的相同圖形。也就是說：這個圖形順序就算在改變三個圖形後，也沒辦法找到長度為 8 的連續圖形。而既然要找到長度為 8 的連續圖形已經不可能，那麼 9 以上就更不可能了。

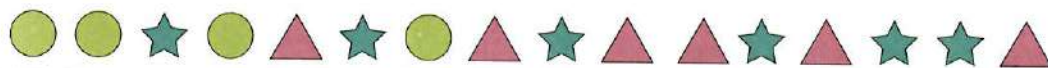
綜上所述，我們可以得知：這個圖形順序在經過我們替換三個圖形之後，所能得到的連續圖形，其最大長度就是 7。



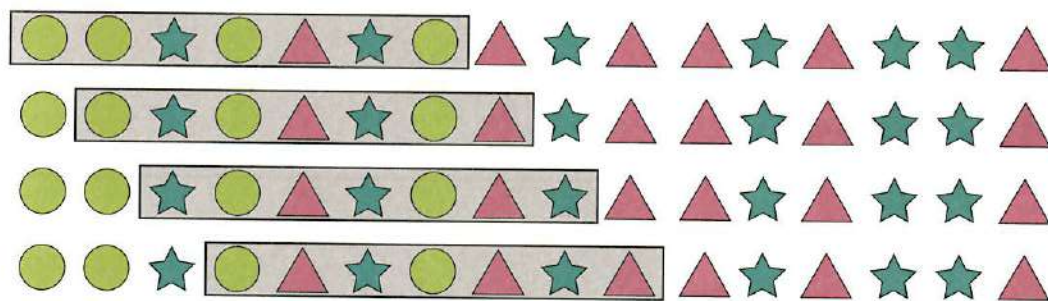
### 題組三的正確答案是：連續 6 個

這題的正確答案是 6。為了說明原因，我們必須證明兩件事情：(1) 長度為 6 的連續圖形是可能的。(2) 長度為 7 以上的連續圖形是不可能的。

● 第一件事情很容易證明。下圖就是一個可能的例子：



● 要證明第二件事情的話，我們可以先檢查在這個圖形順序中的每個「長度為 7 的片段」。如果某個片段可以變成長度為 7 的連續圖形，那麼扣除我們可以改變的 2 個之外，它至少要含有 5 個相同的圖形才行。在這個圖形順序中，總共可以切出 10 個「長度為 7 的片段」，下面列出其中幾個，以此類推。





結果我們可以發現：沒有任何一個片段裡面含有 4 個以上的相同圖形。也就是說：這個圖形順序就算在改變三個圖形後，也沒辦法找到長度為 7 的連續圖形。而既然要找到長度為 7 的連續圖形已經不可能，那麼 8 以上就更不可能了。

綜上所述，我們可以得知：這個圖形順序在經過我們替換三個圖形之後，所能得到的連續圖形，其最大長度就是 6。



## 資訊科學上的意義

本題的任務是希望替換掉少數圖形就能找到最長的連續圖形片段，如同在資訊科學中，需要在符合某些特定規則的情況下尋找出最長（影像或字串序列）子字串；「替換掉少數圖形」就是符合特定規則，「連續圖形片段」就是最長子字串。在尋找最長序列時很常用的方法，包括 **雙指標方法 two pointer method** 和 **滑動視窗 sliding window** 演算法。

一般在搜尋物件時，只會用到一個指標來記錄位置，而雙指標方法就是使用兩個相同方向（快慢指標）或相反方向（對撞指標）的指標來進行掃描，進而達到想要的目的。

滑動視窗演算法是指在一個字串或陣列上進行操作時，只在某個特定大小的範圍內進行操作，而不是在整個字串和陣列上操作，這樣一來就可以降低問題的複雜度，也可以降低迴圈的巢狀深度。在本題中，我們將圖形順序切成固定長度的片段來檢查，其意義也是一樣的。

尋找最長子字串對很多情況來說都是有用的，特別是尋找兩個字串裡面的「共同子字串」。尋找最長共同子字串可以用來幫忙偵測文件是否有抄襲的情況，也可以用來幫忙壓縮資料，找到之後可以把多餘或重複的資料刪除。



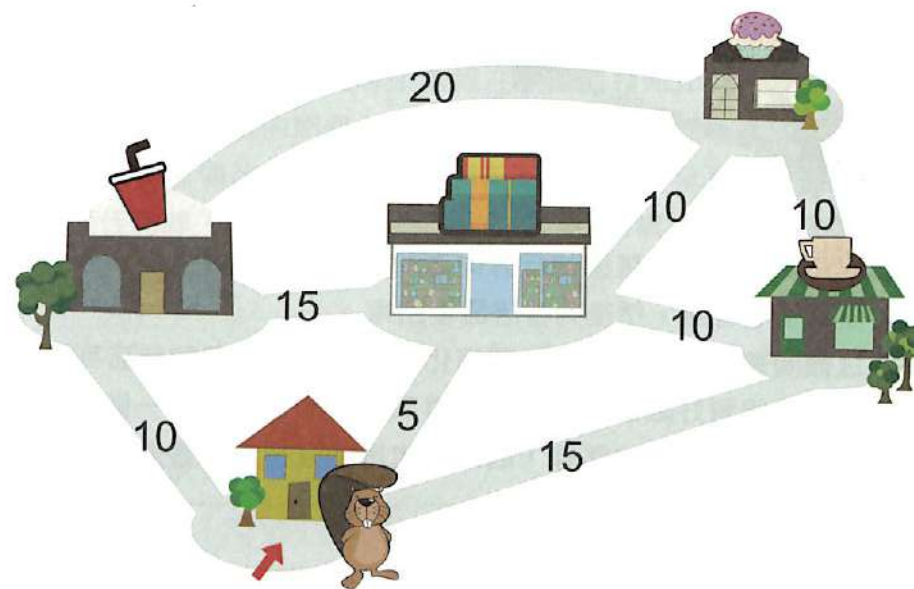
## 關鍵字

雙指標方法、滑動視窗

## 10. 購物去

海狸想去買東西。

地圖中紅色箭頭指向海狸的家，道路上標示的數字，表示海狸從道路的一端移動到另一端所需要的時間。

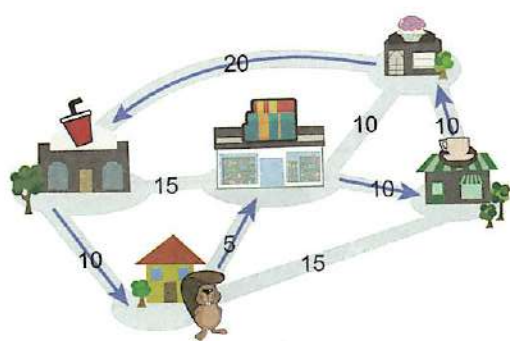


海狸從家裡出發，逛過四家店後回到家裡，所需的最短時間為何？（請輸入範圍 [0~100] 的整數。）



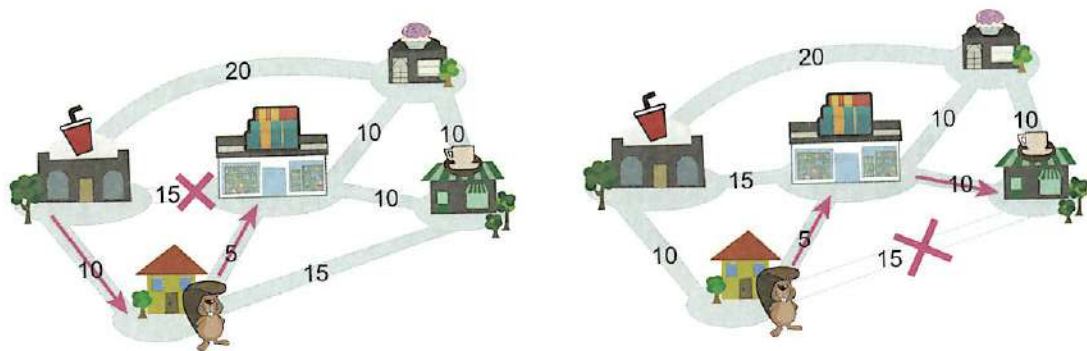
## 正確答案是：55

正確答案為 55，最短時間的走法有兩種，其中一種方式如下圖，從家裡先走到書店，然後走到咖啡店，接著到蛋糕店，然後飲料店最後回家。



另一種走法則是朝著上面這張圖的反方向走，也就是從家裡先走到飲料店，再走到蛋糕店，接著是咖啡店，然後到書店最後回家。

要快速解決這個問題，我們可以將地圖簡化，海狸可以忽略飲料店到書店之間的路線，因為這條路線剛好跟飲料店到家裡再到書店的路線時間一樣，同樣的道理也可以忽略從家裡到咖啡店的路線，因為它也跟從家裡走到書店再到咖啡店的路線時間一樣。



當地圖簡化之後，剩下的路線可以怎麼走就很單純了。



## 資訊科學上的意義

這個題目當中海狸要逛過四家店之後回到家裡，就像旅行推銷員問題當中要拜訪所有城市的概念一樣。**Traveling Salesman Problem 旅行推銷員問題 (TSP)** 是指給定一系列城市和兩城市之間的距離，嘗試找出走過每一座城市一次並回到起始城市的最短迴路。這個思維在作業研究和理論電腦科學中非常重要。這個題目當中，海狸要逛過四家店之後回到家裡，就像 TSP 問題當中要拜訪所有城市的概念一樣。

此問題也很廣泛的運用在實際的情形當中，例如快遞機器人嘗試找出貨物配送到不同目的地的路線，或是車上的導航軟體在規劃路線時如何避開塞車路段，旅行推銷員問題其實是個困難的問題，也許上述題目看起來並不複雜，但如果我們將想要遍訪的城市從數個改為上百個，用目前已知的演算法可能就會耗費大量的時間去推算出最佳路線。因此實務上我們只會嘗試找出接近最佳的路線走法。



## 關鍵字

旅行推銷員問題、計算複雜性理論



## 11. 摔角訓練

海狸強森想要成為一位職業摔角選手。摔角比賽時可以做的動作有下列六種：



- 躺下 (P1)
- 站立 (P2)
- 奔跑 (P3)
- 靠著擂台邊界繩 (P4)
- 站在擂台角落 (P5)
- 高空重擊 (P6)

他的教練想教他連續技，每個連續技都是由一連串的动作組成。在有限時間內，為了讓自己熟練，強森希望能專精學會越少連續技越好，但同時，他也希望每個動作都能被應用在連續技當中。以下是教練能教他的各種連續技：

| 連續技    | 動作                               |
|--------|----------------------------------|
| 飛身撲    | 奔跑、高空重擊 (P3、P6)                  |
| 過肩摔    | 站立、高空重擊 (P2、P6)                  |
| 金勾臂    | 站立、奔跑、高空重擊、靠著擂台邊界繩 (P2、P3、P6、P4) |
| 反身肘擊   | 站立、站在擂台角落、靠著擂台邊界繩 (P2、P5、P4)     |
| 肘節固定   | 站立、躺下 (P2、P1)                    |
| 助跑式飛身壓 | 奔跑、躺下 (P3、P1)                    |

請問強森最少需要學幾個連續技才能讓六種動作都被用到？

- A. 2 個
- B. 3 個
- C. 4 個
- D. 5 個



## 正確答案是：B. 3 個

有多種方法可以只用 3 個連續技就達到目標。若運用貪心演算法的策略，選擇能新增最多未使用過的動作的連續技，則可達到選出最少的連續技，用到所有動作。這題中，「金勾臂」包含了 4 個動作—站立、奔跑、高空重擊、靠著擂台邊界繩 (P2、P3、P6、P4)，因此先選擇「金勾臂」這個連續技。

接下來只剩兩個動作需要被包含，也就是一 躺下 (P1) 和 站在擂台角落 (P5)，剩下的連續技都無法一次包含這兩個動作，所以我們選出有包含 躺下 (P1) 的連續技「肘節固定」，以及唯一有包含站在擂台角落 (P5) 的「反身肘擊」(選出「助跑式飛身壓」和「反身肘擊」也可以)。最後共使用這三個連續技，即可包含所有 6 個動作。

若要證明可包含所有動作的最少連續技數是三個連續技，只需要顯示採用兩個連續技必不可能達到目標。可一一檢查任意兩個連續技的組合都無法完全包含 6 個動作，尤其是總動作數加起來小於 6 的組合更是明顯不符合 (例如「飛身撲」和「過肩摔」的組合最多只有 4 個動作，不可能涵蓋到 6 種動作)，其餘組合的檢查也非常簡單。



## 資訊科學上的意義

這個題目相當於解決一個集合覆蓋問題 set covering problem。想像有六個集合，每個集合代表一種連續技，集合中的元素包含了組成該連續技的各式動作。本任務是要找出最少量的集合 (連續技)，這些集合的聯集包含所有的元素 (動作)。生活中全覆蓋問題應用的例子，例如：(1) 每個人都有不同的專長，要找最少的人，涵蓋完成某項任務所需要的專長。(2) 每個工具都有不同的輔助功能，要選最少的工具，涵蓋完成某項任務所需要的輔助功能。

詳解中找出答案所使用的方法叫做貪心演算法 greedy algorithm，也就是在每一步驟中都選擇當下能看到最有利的選項，希望最後的結果就是最好的；選擇連續技時，演算法會優先選擇包含最多動作的連續技。值得注意的是，雖然貪心演算法能快速解題，但無法保證可找出任何問題的最佳解。舉例來說，若連續技「飛身撲」多包含一個躺下 (P1) 的動作，則「飛身撲」和「反身肘擊」兩個連續技即可包含所有的動作。生活中找零錢的問題 (假設零錢足夠，如何用最少的硬幣數找錢)，採用貪心演算法解決則可以找出最佳解。



## 關鍵字

貪心演算法、集合覆蓋問題



## 12. 巴伐亞大地

在遙遠的巴伐亞大地上有四大部落，各部落的代表旗幟分別是數位部落 、手機部落 、勾勾部落 和筆電部落 ；每個部落擁有數量不等的村莊，各村莊都會插上所屬部落的旗幟以供識別，而村莊分佈圖如下：



有一天，四大部落決定要合併了。不過，為了不讓巴伐亞大地陷入混亂，他們共同決定一次只能合併兩個部落。合併時間以月為單位，合併兩個部落所需的時間等於這兩個部落村莊數量的總和。即日起開始將兩個部落合併成為一個共同部落，而合併的過程不斷重複，直到最後統一成為一個巴伐亞大部落為止。

請問完成所有部落合併的時間，最少需要幾個月？

- A. 23
- B. 24
- C. 25
- D. 26
- E. 27



## 正確答案是：B. 24

由於巴伐亞大地有 4 大部落，因此會經過 3 次合併才能統一所有部落；由於合併時間為村莊數量的總和，巴伐亞大地共有 12 個村莊，所以最後一次合併一定需要 12 個月的時間；因此我們只需要考慮前面 2 次合併所需的時間最少，也就是合併所需的次數最少就可以了。以下為這 4 個部落合併的各種可能，因此可以找到完成所有部落合併最少需要 24 個月。

- 2+3=5, 5+3=8, 8+4=12；總共需要 5+8+12=25 個月
- 2+3=5, 5+4=9, 9+3=12；總共需要 5+8+12=26 個月
- 2+3=5, 3+4=7, 7+5=12；總共需要 5+7+12=24 個月
- 3+3=6, 6+2=8, 8+4=12；總共需要 6+8+12=26 個月
- 3+3=6, 6+4=10, 10+2=12；總共需要 6+6+12=28 個月
- 3+3=6, 4+2=6, 6+6=12；總共需要 6+6+12=24 個月
- 4+2=6, 6+3=9, 9+3=12；總共需要 6+9+12=27 個月
- 4+2=6, 3+3=6, 6+6=12；總共需要 6+6+12=24 個月
- 4+3=7, 2+3=5, 7+5=12；總共需要 7+5+12=27 個月
- 4+3=7, 7+2=9, 9+3=12；總共需要 7+9+12=28 個月
- 4+3=7, 7+3=10, 7+10=17；總共需要 7+10+17=34 個月



## 資訊科學上的意義

貪心演算法 **greedy algorithm** 簡單來說是在每一步選擇中都採取在當前狀態下最有利的選擇，從而希望導致的最後結果就是最好的。但在很多問題中，貪心演算法不一定會產出整體的最佳解，卻能在合理時間內得到一組近似最佳解的答案。以本題而言，所採取的最有利策略是採最小化合併，每次合併前都先找出當下村莊數量最少的兩個部落進行合併，直到全部合併完成。

我們也可以舉旅遊例子來說，先找出要旅遊的國家後，在所有要旅遊的目標國家中先挑選從臺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，直到完成所有國家的旅遊，這就是採用貪心演算法的策略。但要注意的是，若在挑選從臺灣出發費用最低的國家後，再由此國家出發到其他國家的交通費都是超級貴的話，那可能一開始先去其他國家會有更划算的解法。



## 關鍵字

貪心演算法

## 13. 座位排列



海狸班上共有 7 隻小海狸，每隻小海狸都有自己座號的旗幟。教室中的座位排成一列，如上圖所示，目前每隻小海狸隨意的挑選了一個座位。

老師希望小海狸可以由左到右按照座號由小到大坐好，為了避免混亂，每次只能有兩隻小海狸交換座位。例如：當 3 號和 1 號交換座位，表示 3 號要坐到 1 號的位子上，1 號要坐到 3 號的位子上。

只要利用有限的交換次數，就可以讓小海狸按照座號由小到大排好。

最少要交換幾次座位，才能讓所有小海狸由左到右按照座號由小到大坐好？

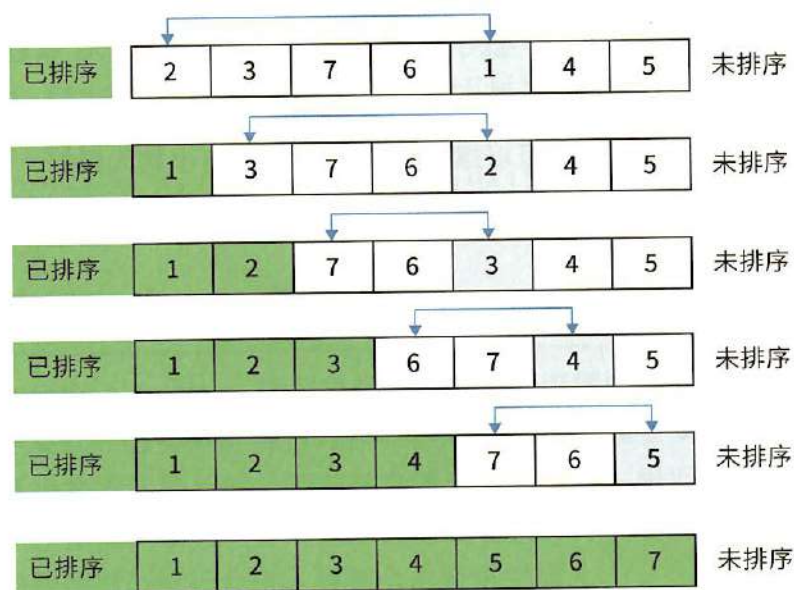
- A. 3 次
- B. 4 次
- C. 5 次
- D. 6 次



## 正確答案是：C. 5 次

正確答案是 5 次。每次兩隻小海狸交換座位，5 次後即可達成由小到大坐好，如圖一所示。

在此使用的是選擇排序 selection sort 演算法，這種演算法會將原始序列切割為兩個子序列：置於整個序列前端的已排序子序列；在整個序列尾端的則是未排序子序列。一開始，已排序子序列是空的，而未排序子序列則包含整個原始序列。每次從未排序的子序列中找出最小的元素（若要由大到小排列，則挑選最大的元素），將此元素與未排序的子序列中第一個元素交換，交換完畢後成為前端已排序的子序列中最後一個元素。



在此題中，我們先將所有小海狸視為未排序的數列，接下來從所有小海狸中挑選編號最小的 1 號，1 號必須與最左邊的 2 號交換座位，交換完畢後，1 號就坐在正確的位置上（排序完成）。接下來只要用同樣的想法，從尚未排序完成的小海狸中挑選編號最小的 2 號，將 2 號與第一個尚未排序的 3 號交換，就有兩隻小海狸坐在正確的位置上，以此類推，總共需要 5 次交換。



## 資訊科學上的意義

**排序演算法** 是可以將資料根據指定的規則進行排序的演算法，在資訊科學中，從最基礎的資料排序處理到更複雜的處理程式中，排序演算法都扮演相當重要的角色。因此，大量的資訊科學家投入相關研究，發展出不同的排序演算法，每種排序演算法都有各自的特色、優點和缺點。

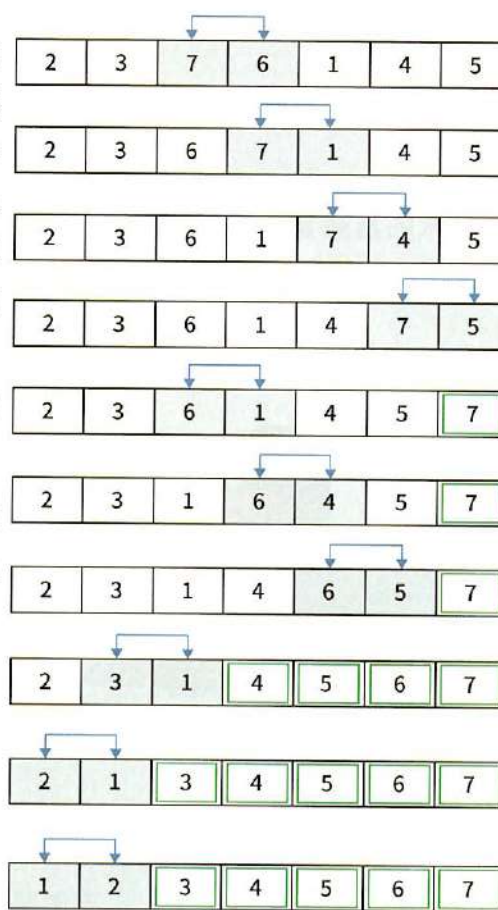
**選擇排序演算法 selection sort** 是從氣泡排序演算法 bubble sort 改進而來。

在選擇排序中：每一輪從未排序的元素中挑選最小的，與第一個未排序的元素進行一次交換，即可讓一個元素排在正確的位置上。

而氣泡排序則是在每一輪中將元素兩兩進行比較，如果順序錯誤就進行交換，經過多次的交換可將一個元素排在正確的位置上。如果總共有  $n$  個元素要進行排序，兩種排序都需要做  $\frac{n \times (n-1)}{2}$  次的比較，但是選擇排序最

多只要進行  $(n-1)$  次交換即可完成排序，氣泡排序最多可能需要進行  $\frac{n \times (n-1)}{2}$  次交換才能完成排序，因此程式在電腦實際運作時，選擇排序通常會比氣泡排序快一點。

在此題中，可以參考圖二使用氣泡排序結果，將比使用選擇排序速度慢一點。



## 關鍵字

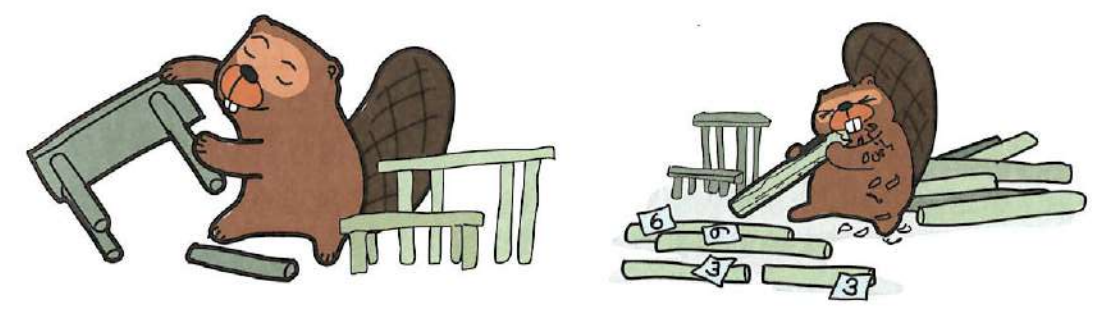
選擇排序、排序演算法



## 14. 桌子工廠

阿狸是位桌子工匠，他製作的桌子由桌面及 4 支相同長度的桌腳組成。為了符合顧客的需求，他會製作各種不同高度的桌子讓客人挑選。

阿狸總是到處尋找適合做桌腳的木頭。雖然他無法每次都取得長度適當的木頭，但他可以把較長的木頭啃短成他需要的長度，當然被啃掉的部份就不能使用了。



現在阿狸倉庫有 32 支木頭，其長度及數量表列如下：

|    |    |   |   |   |   |   |   |   |   |
|----|----|---|---|---|---|---|---|---|---|
| 長度 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| 數量 | 3  | 6 | 3 | 3 | 5 | 3 | 3 | 2 | 4 |

阿狸需要製作 8 張桌子。由於啃木頭很辛苦，請幫忙阿狸計算，他最少要啃短幾支木頭才能完工？（範圍 [0~32] 的整數。）



## 正確答案是：6

要求得最佳解，我們先歸納幾個規則與事實，好進行接下來的計算：

- 規則 1：每張桌子有 4 支腳，如果同一種長度的木頭庫存量  $\geq 4$ ，就直接取出 4 支或 4 的倍數支木頭來製作桌子，直到這款長度的木頭剩下的數量不足 4。

依照這個規則，阿狸不需要啃短任何木頭就可以製作一個高度 2、一個高度 6 和一個高度 9 的桌子。剩下的數量如下表一：

|      |    |   |   |   |   |   |   |   |
|------|----|---|---|---|---|---|---|---|
| 長度   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 木頭數量 | 3  | 2 | 3 | 3 | 1 | 3 | 3 | 2 |

- 規則 2：阿狸要用 32 根木頭做出 8 張桌子，代表所有的木頭都必須使用，而且木頭只能啃短不能加長。

依照這個規則，我們來檢查最長的木頭：長度 10 的木頭只有 3 支，無法做成一張桌子，代表這 3 支木頭都必須被啃短，其他每一種長度的木頭則各自有需要補足的數量，我們可以列成如下表二：

|      |    |   |   |   |   |   |   |   |
|------|----|---|---|---|---|---|---|---|
| 長度   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 木頭數量 | 3  | 2 | 3 | 3 | 1 | 3 | 3 | 2 |
| 缺少數量 | 0  | 2 | 1 | 1 | 3 | 1 | 1 | 2 |

為了讓啃短的木頭數量最少，我們可以優先製作高度 4、5、7、8 的桌子，因為這幾種高度的桌子都只缺 1 支腳而已。另外因為所有庫存都必須使用，代表長度 3 的這 2 支木頭也必須製作成一張桌子，所以 3、4、5、7、8 這幾種高度的桌子優先製作，它們總共缺少  $2+1+1+1+1=6$  支桌腳。剩下長度 6、9、10 的木頭目前不會拿來做桌子，而這 3 種長度木頭的庫存量是  $1+2+3=6$ ，剛好可以補齊上述所缺的 6 支桌腳。到這裡我們可以得出答案，最少需要啃短 6 支木頭才能製作出 8 張桌子。



## 資訊科學上的意義

**最佳化 optimization** 是尋求一個問題最佳解答的過程。在固定條件限制下，將問題以數學方式表示，並透過運算或搜尋的方法求出最佳解 optimal solution，就是最佳化的步驟。以桌子工廠這一題來說，桌腳的長度與數量是固定的，而每張桌子有 4 支腳，且必須做出 8 張桌子的條件也是固定的，想要找到的最佳解就是「需要變短的桌腳數量的最小值」。

在真實世界中，每一種需要最佳解的問題性質都不同，例如最短的路徑、最少的花費等，因此不存在單一個最佳化方法來對應所有問題。在科學領域也發展出許多 **演算法 algorithm** 來找最佳解，其中有些演算法雖然無法保證找到最佳解，但可以幫助我們找到接近最佳解的結果。

最佳化在科學、工程、數學界都是非常重要的研究課題，近年來發展快速的人工智慧也十分重視這個領域。例如利用人工智慧尋找最符合特徵的人臉，或判斷投資的最佳時機等。



## 關鍵字

最佳化、演算法



## 15. 聖誕樹裝飾

蘿拉和麥麥用 18 種不同的裝飾品裝飾他們的聖誕樹。

這些飾品分為三種形狀（星形、圓形、鐘形）、三種尺寸（大、中、小），或是有無花紋組成。

麥麥有一件最喜歡的裝飾品，希望蘿拉透過問問題，來猜出麥麥最喜歡的是哪一件，但只能問尺寸、形狀和花紋的問題。而且蘿拉只能問「是非題」，麥麥也只能回答是或否，例如「你最喜歡的裝飾品有花紋嗎？」。



在運氣很差的情況下，請問蘿拉至少要問幾題，一定能確定哪個是麥麥最喜歡的裝飾品？（範圍 [3~8] 的一個整數）

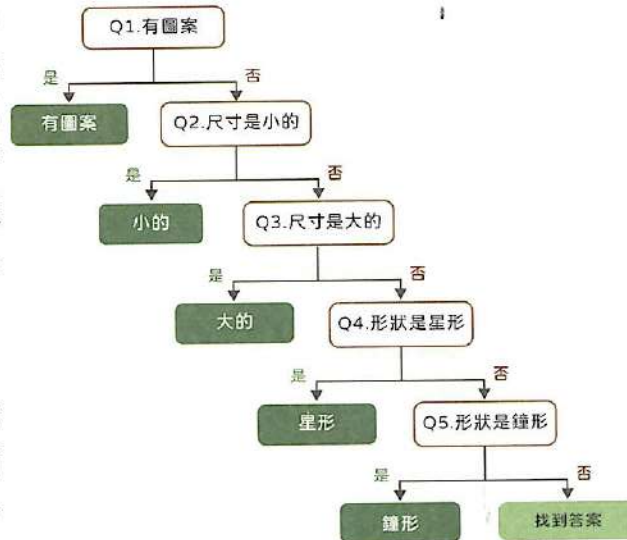


## 正確答案是：5

為了得知麥麥最喜歡的裝飾品是否有圖案，蘿拉第1個問題就問「是否有圖案」，依據麥麥的回答：是或否，蘿拉就能立刻知道是否有圖案。

接著蘿拉問裝飾品「尺寸是否是小的」，如果答案是否定的，蘿拉就排除小這個答案，仍然有中和大這兩種尺寸，接著蘿拉問「尺寸是否是大的」，如果答案是否定的，蘿拉就能得知尺寸為中，所以確認尺寸需要2個問題來確定答案。

而形狀跟尺寸一樣，有三種選擇，所以蘿拉也是需要2個問題來確定形狀，因此，蘿拉總共需要5個問題，才能猜出麥麥最喜歡的裝飾品。



## 資訊科學上的意義

分類 classification 是資訊科學中非常重要的主題，在分類任務中，我們基於物品特性，嘗試將每個物品分類到某個類別中。在本題任務中，我們依據尺寸、形狀或圖案，對裝飾品進行分類。分類任務類似遊戲--猜猜我是誰 (guess who?)，最多只能詢問對方20個問題就要找出答案，這個遊戲主要是透過問問題，可以讓我們將一群可能的答案分成兩組，其過程類似二分搜尋法。

在資訊科學中，可以透過 **決策樹 decision tree** 演算法來將物品分類，決策樹常用於分類和迴歸問題。決策樹主要是使用樹的階層概念，用一個倒著的樹狀結構，從最上層的根節點開始，在每個節點挑一個特徵，經由分支做出決策將資料分類，再繼續往下一層前進。本題任務的決策樹如上圖，我們觀察，在最差情況 worst case 下，如果答案有三個選擇 (如形狀)，至少要問兩題，才能確認答案，因此，本題任務至少要問5題，才能確定裝飾品答案為何。現實生活中常見的分類問題有醫學影像分類判讀、人臉分類辨識和垃圾郵件檢測。



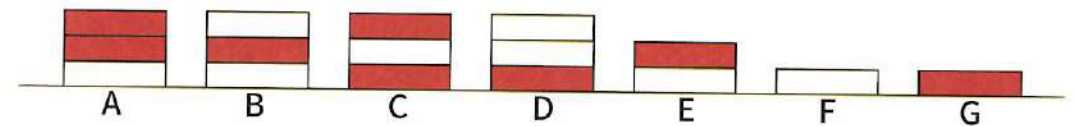
## 關鍵字

決策樹、分類



## 16. 積木堆疊

小狸有16個大小相同的積木，紅色及白色各8個；小狸將這些積木堆疊成7堆並加上編號A~G，放在桌子上，如下圖所示：



小狸想在不變更各堆積木數量及顏色排列順序的前提下，將編號A~G的7堆積木重新組合成2堆，這2堆的高度相同 (各有8個小積木)，而且由下往上的顏色排列順序也需相同。

小狸會用編號紀錄組合方式，例如 (x, y, z) 表示 x 放最下面，由下而上依序堆放 y 及 z。請問哪種組合方式無法達到小狸的要求？

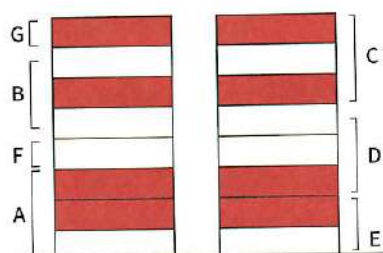
- A. (A, F, B, G) 和 (E, D, C)
- B. (B, A, E) 和 (F, D, G, C)
- C. (B, E, A) 和 (F, D, G, C)
- D. (B, E, A) 和 (F, D, C, G)



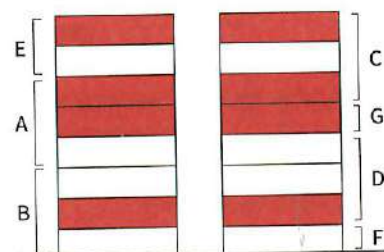
## 正確答案是：C. (B, E, A) 和 (F, D, G, C)

依照各選項的編號紀錄，重新組合積木後如下圖所示；只有選項 C 的顏色排列順序不相同，無法達到题目的要求，其他答案皆可達到题目要求。

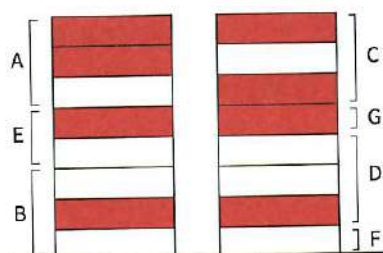
選項 A：(A, F, B, G) 和 (E, D, C)



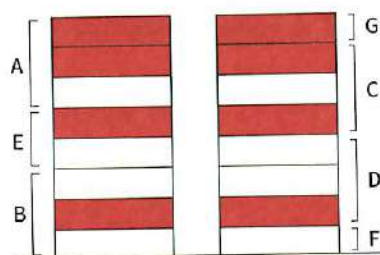
選項 B：(B, A, E) 和 (F, D, G, C)



選項 C：(B, E, A) 和 (F, D, G, C)



選項 D：(B, E, A) 和 (F, D, C, G)



## 資訊科學上的意義

程式設計師通常透過撰寫程式準確地告訴電腦該做什麼，讓電腦順利執行任務。有時程式設計師會像在此任務中一樣，簡單地定義一組電腦必須遵循的規則 rule，以及規則套用的先後順序，這樣就不用對符合規則條件的資料一一指定所要做的相同處理。資訊科學不僅涉及如何編寫讓電腦遵循的規則，有些人還研究使用哪種規則更有效率，根據規則運行時，哪些結果是可達成的，哪些是無法達成的。而證明某個解法找出來的已是最佳解，有時是非常困難的事情。通常，為了確認不存在更好的解法，唯一的方法是列出所有的解法，這稱為窮舉搜尋法 exhaustive search。

以本題而言，將編號 A~G 的積木堆疊，先找出所有可以堆疊出數量為 8 個且不重覆的子集合；例如以編號 A 開始找符合數量總和為 8 的子集合中有 {A, B, E}, {A, B, F, G}, {A, C, E}、{A、C、F、G}、{A、D、E}、{A、D、F、G}，再以編號 B 開始找符合數量總和為 8 的且不重覆子集合…以此類推，即為窮舉搜尋。

在絕大多數的情況下，人工進行窮舉搜尋是不可行的，但是，以電腦系統化進行窮舉搜尋卻是相當直觀且快速。以遊戲為例，一個遊戲所有可能發生的情況有時候可以用樹狀結構來表示，這種樹稱為「遊戲樹」。一棵遊戲樹有一個起始節點，代表遊戲開始的初始狀態，接下來向下展出一層一層的節點，每一個節點代表從上一個節點之後可能發生的情況。依這樣的方式延續下去，就可以把遊戲進行時所有可能發生的情況窮舉出來。



## 關鍵字

規則、窮舉搜尋



## 17. 資源回收再升級

海狸們是不浪費資源的動物，他們總是盡量利用舊材料來製作成新的工具，以達到資源回收再升級的目標。下面的圖示就是一個例子，舊車胎等可以被重新製成輪子、二輪腳踏車、手推車與三輪腳踏車。



小狸很環保，他喜歡動手做回收升級，並且會出售他做好的工具。以下是每一種工具的售價：



小狸目前收集到下面這些舊材料：6 個舊車胎、6 根鐵管、2 根木棍。請問靠這些材料他最多可以賺到多少錢？（範圍 1~99 的一個整數。）
















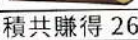




## 正確答案是：30







因為每個物品都要用到舊車胎，每個舊車胎最多可以賺到 5 元，所以小狸收集到的舊車胎有 6 個，所以最多可以賺 30 元。

想要賺到最多的錢，通常會想到從製作最高價的物品開始，本題中最高價是三輪車，我們先計算做出三輪車的狀況：

| 步驟   | 製作   | 剩下材料  |
|--|--|---|
| 1 1 台三輪車用掉 3 個車胎、4 個鐵管，賺得 15 元。              | <br>三輪腳踏車: \$15            |  X 3<br> X 2<br> X 2       |
| 2 因為第二高價的二輪車至少需要 3 個鐵管，所以無法做出二輪車。            | <del></del><br>二輪腳踏車: \$10 |  X 3<br> X 2<br> X 2      |
| 3 第三高價是手推車，使用 2 個車胎與 2 個木棍做出 2 台手推車，賺得 10 元。 | <br>手推車: \$5             |  X 1<br> X 2<br> X 0 |
| 4 最後用 1 個車胎與 1 個鐵管做出輪子，賺得 1 元。               | <br>輪子: \$1             |  X 0<br> X 1<br> X 0 |

以上累積共賺得 26 元。

我們再試算一下，如果從製作第二高價的物品開始，有沒有可能賺更多呢？

| 步驟   | 製作   | 剩下材料   |
|--|--|--|
| 1 1 台二輪車用掉 2 個車胎、3 個鐵管。小狸可以做 2 輛二輪車，賺得 20 元。 | <br>二輪腳踏車: \$10 |  X 2<br> X 2 |
| 2 使用 2 個車胎與 2 個木棍做出 2 台手推車，再賺得 10 元。         | <br>手推車: \$5    |  X 0<br> X 0 |

以上累積共賺得 30 元。

如果從製作第三貴的物品來計算，還有可能賺更多嗎？第三貴的是手推車，由於我們只有 2 個木棍，最多只能做出 2 台手推車，計算結果會和第 2 種狀況相同，也沒有賺更多錢的組合，所以答案是最多賺得 30 元。



## 資訊科學上的意義

**最佳化 optimization** 是尋求一個問題最佳解答的過程。在固定條件限制下，將問題以數學方式表示，並透過運算或搜尋的方法求出最佳解 optimal solution，就是最佳化的步驟。這一題小狸擁有的材料數量是固定的，而每項工具的單價也是固定的，在這些條件下我們想要賺得最多的錢，就是一個尋找最佳解的過程。

以這一題來說，一開始會想到優先製作最貴的物品，以賺得最多的錢，這種解題思考稱為 **貪心演算法 greedy algorithm**，也就是在每一步驟中都選擇當下能看到最有利的選項，希望最後的結果就是最好的。然而貪心演算法不一定會產出整體的最佳解，像這一題從最貴的三輪車製作，就只能得到 26 元的解答，但 26 元卻不是最佳解。在很多狀況下，貪心演算法不一定會產出整體的最佳解，但能在合理的時間內得到一組接近最佳解的答案。

最佳化在科學、工程、數學界都是非常重要的研究課題，近年來發展快速的人工智慧也十分重視這個領域。例如利用人工智慧尋找最符合特徵的人臉，或判斷投資的最佳時機等。



## 關鍵字

最佳化、貪心演算法



## 18. 搞笑濾鏡





有一款手機拍照軟體提供四種搞笑濾鏡，每一種濾鏡的效果如下圖所示：



下圖是小海狸用這款軟體拍照後，將照片套用「增加腮紅」和「改變臉型」兩種濾鏡後的照片。



請問小海狸原本拍的照片長什麼樣子？

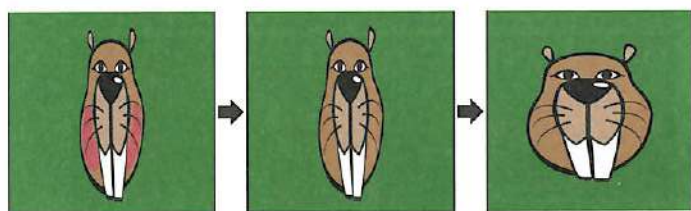
- A.  B.  C.  D. 



## 正確答案是：D

因為小海狸沒有套用「大牙效果」濾鏡，所以我們可以推論原始照片的牙齒和套用濾鏡後照片的牙齒是一樣大的，而選項 A 和選項 C 的牙齒比套用濾鏡後的照片小，因此不是正確答案。另一方面，套用濾鏡後的照片有鬍鬚，因此原始的照片也有鬍鬚，選項 A 和 B 沒有鬍鬚也不正確。

我們也可以從套用濾鏡後的照片往回推，如下圖。我們將濾鏡一一移除，先將腮紅消除，再取消臉型的改變，就能得到原始的照片。



## 資訊科學上的意義

在此任務中，我們將使用一組 **屬性 property** 來描述海狸的臉。海狸的每個屬性都可以有不同的值：

- 牙齒：大 / 小
- 鬍鬚：存在 / 不存在
- 臉型：圓 / 瘦
- 腮紅：存在 / 不存在

每一種濾鏡都可能改變這些屬性的值。我們在改變屬性值的過程中，如果有將變動的歷程記錄下來，即「記錄檔 log」，就可以將 **資料還原** 到之前的樣子，以本任務為例，我們可以將濾鏡的效果一一移除，就能得到原始的照片。



## 關鍵字

資料屬性、資料還原



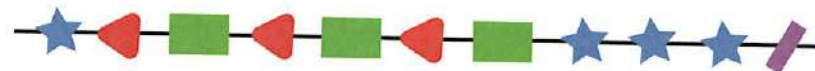
## 19. 項鍊密碼

小狸喜歡用各種形狀的珠子設計項鍊，並用精簡的表示法與朋友分享設計。小狸用 4 個不同字母代表不同形狀的珠子（S 代表星形 ，T 代表三角形 ，R 代表矩形 ，L 代表長條形 ）。而且小狸想使用以下規則縮短表示法：

- 如果有幾個相同形狀的珠子連在一起，只需在代表珠子形狀的字母前寫下珠子連續的數量。
- 如果有重複的珠子排列順序，只需在括號中寫出代表珠子形狀的字母排列，再將這個排列重複的次數寫在括號前。

如果不符合以上兩種規則，就直接寫代表珠子形狀的字母。

例如，以下的項鍊設計：



可以表示為  $ST2(RT)R3SL$ ，一共用了 11 個符號（符號包含數字、字母或括號）；也可以表示為  $S3(TR)3SL$ ，一共用了 9 個符號。

以下項鍊設計的最短表示法要用幾個符號？



- A. 12                      B. 13                      C. 14                      D. 15

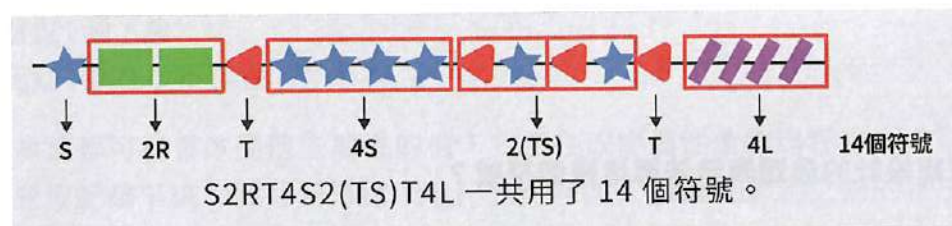
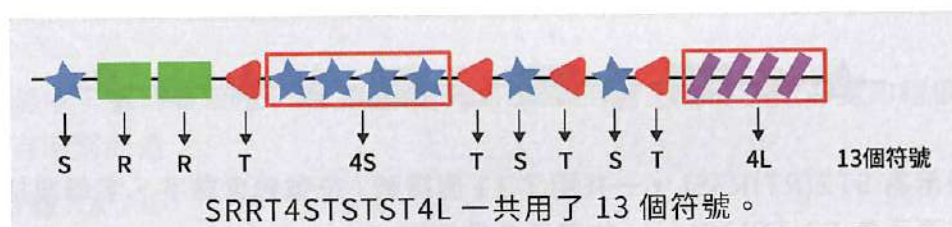
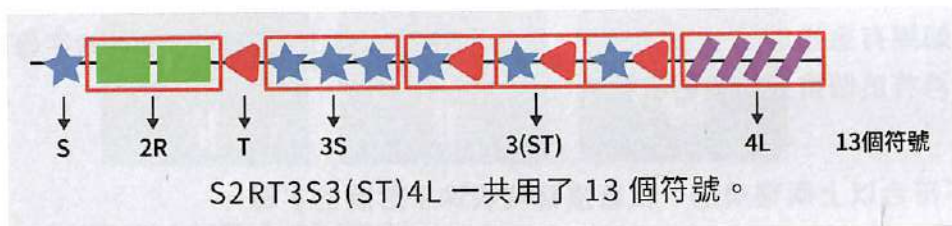


正確答案是：B.13



這條項鍊設計有多種可能的表示法，取決於選擇「星形 + 三角形」還是「三角形 + 星形」作為重複的模式。

以下表示了不同的表示法，而前兩個表示法是最短的：



上述兩種表示法，採用 TSTS 或是 2(TS) 是關鍵點，可顯示符號序列有重複時使用壓縮表示法未必更短：例如 TSTS 不採用壓縮表示只需用 4 個符號，若壓縮表示為 2(TS)，反而需要用到 5 個符號。



## 資訊科學上的意義

此題目運用的資訊科學概念為 資料壓縮 **data compression**，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，本題的編碼方式在編號後可以完整地解碼出原始資料，屬於壓縮方式中的無損壓縮方法。壓縮技術是根據資料序列中的出現樣式，希望用較少的資料量對它們進行編碼表示。

此問題中採用的這種壓縮技術，必須考慮不同的編碼方式才能得到最佳解。當項鍊珠子出現的相同序列重複更多次的話，更能顯出可以簡潔地記下訊息。

此題目的目標是希望用最少的符號表示出項鍊的設計，此種類型的題目稱為 **最佳化問題**。如果沒有注意到如何有效的運用壓縮表示來減少資料量，將無法較快速發現最佳的解法，而需列舉出所有可能的解法（窮舉法）。



## 關鍵字

資料壓縮、最佳化問題



## 20. 七段顯示器

湯姆士發現他數位時鐘上顯示的數字怪怪的，於是他找了一個可以正常顯示的時鐘來對照。他嘗試將兩個時鐘的數字設為 3，然後再設為 9；實驗後湯姆士找出他的時鐘有燈條故障了。

以下是湯姆士的實驗結果：

| 數字 | 正常的顯示 | 怪怪的結果 |
|----|-------|-------|
| 3  |       |       |
| 9  |       |       |

當將數字設為 1 時，時鐘怪怪的結果會像什麼樣子？

| 數字 | 正常的顯示 | 怪怪的結果 |
|----|-------|-------|
| 1  |       | ?     |

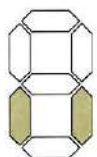
- A.
- B.
- C.
- D.



## 正確答案是：C

正確答案是 C。

透過湯瑪士的實驗，可以發現有兩段燈條故障了。



燈條故障的情況有很多種：(1) 兩段燈條顯示的位置互換了，(2) 燈條該亮的時候不亮，不該亮的時候亮，(3) 無論何種情況就是不亮，(4) 無論何種情況就是亮。

看到怪怪結果，可能是發生以下幾種故障狀況：

| 左邊燈條     | 右邊燈條     |
|----------|----------|
| 故障情況 (1) | 故障情況 (1) |
| 故障情況 (2) | 故障情況 (2) |
| 故障情況 (4) | 故障情況 (3) |
| 故障情況 (4) | 故障情況 (2) |
| 故障情況 (2) | 故障情況 (3) |

而這些情況下，當數字為 1 時，皆會是以下顯示的畫面，所以 C 選項就是答案。

| 數字 | 正常的顯示 | 怪怪的結果 |
|----|-------|-------|
| 3  |       |       |
| 9  |       |       |
| 1  |       |       |



## 資訊科學上的意義

**邏輯推理** 是資訊科學領域中重要的能力。在資訊科學中，**布林值** 是一種資料型態，具有兩個值（通常標記為真和假），是用來表示邏輯和布林代數的值。

**除錯** 是程式設計師經常做的工作。當面對像這樣的問題時，程式設計師會將問題拆成較小的問題（一次看一個燈條）以檢查不正常的地方。他們會用邏輯檢視有問題的輸出之間是否有關聯，而在這題中，下方輸出的二個燈條左右對調了。在這題除錯的過程中會需要用到歸納推理以及圖形識別的技巧。

在日常生活中，可將解決某項問題的檢查項目整理成清單，再依清單中的每個項目逐一進行檢查。例如：當汽車無法啟動時，會逐一檢查是電池沒電了還是燃料不足，或是還有某些磨損或損壞的零件影響汽車運作，找出問題後加以故障排除，讓功能恢復正常，這就是除錯的過程。



## 關鍵字

邏輯推論、布林、除錯



## 21.1 杜鵑找家 - 題組一

斑點杜鵑自己不築巢，牠們會從樹的底部開始以下列步驟尋找空的鳥巢，直到找到鳥巢住進去。

步驟 1：往上尋找，直到遇到第一個鳥巢。

步驟 2：如果那個鳥巢是空的，牠就直接住進去。

步驟 3：那個鳥巢已被其他斑點杜鵑佔用，牠會數一數巢中杜鵑頭上的斑點數量：

- 如果斑點數量比自己的多，就往樹幹的左方，再從步驟 1 開始繼續尋找。
- 如果斑點數量跟自己的一樣多或比較少，就往樹幹的右方，再從步驟 1 開始繼續尋找。

這棵樹上有五個空鳥巢，樹下也剛好有五隻斑點杜鵑正由左到右排隊入住，如下圖順序。



哪一隻鳥會住進最高的鳥巢呢？

A.



B.



C.



D.



## 21.2 杜鵑找家 - 題組二

斑點杜鵑自己不築巢，牠們會從樹的底部開始以下列步驟尋找空的鳥巢，直到找到鳥巢住進去。

步驟 1：往上尋找，直到遇到第一個鳥巢。

步驟 2：如果那個鳥巢是空的，牠就直接住進去。

步驟 3：那個鳥巢已被其他斑點杜鵑佔用，牠會數一數巢中杜鵑頭上的斑點數量：

- 如果斑點數量比自己的多，就往樹幹的左方，再從步驟 1 開始繼續尋找。
- 如果斑點數量跟自己的一樣多或比較少，就往樹幹的右方，再從步驟 1 開始繼續尋找。

這棵樹上有五個空鳥巢，樹下也剛好有五隻斑點杜鵑正由左到右排隊入住，如下圖順序。



這五隻鳥分別會住進哪些鳥巢呢？

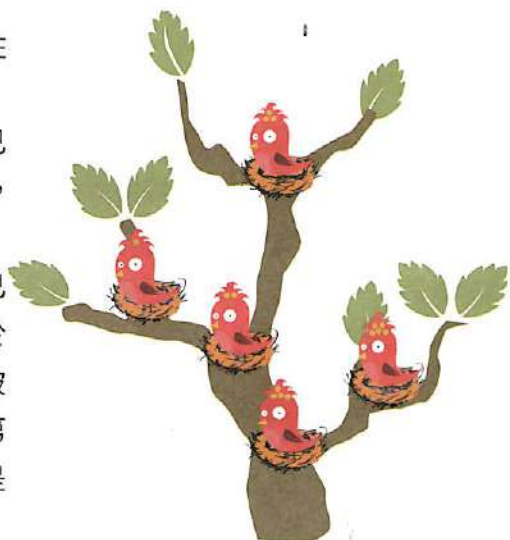


## 正確答案

題組一答案：D. 有 3 個斑點的杜鵑。

題組二答案：如下圖。

- 第一隻鳥，也就是有 4 個斑點的杜鵑會住進最低的鳥巢。
- 第二隻杜鵑有 2 個斑點，而最低的鳥巢已被有 4 個斑點的杜鵑佔據，又因為 4 大於 2，第二隻杜鵑移至左邊的空鳥巢住進去。
- 第三隻杜鵑有 3 個斑點，而最低的鳥巢已被有 4 個斑點的杜鵑佔據，又因為 4 大於 3，第 3 隻杜鵑向左移動，下一個鳥巢已被有 2 個斑點的杜鵑佔據。3 大於 2，所以第三隻杜鵑向右移至下一個鳥巢——也就是最高的鳥巢。



圖一



## 資訊科學上的意義

以題目的方法來分配鳥兒的住處能帶來許多有趣的優勢，讓尋找特定杜鵑的工作變得更快速。如果你在尋找的杜鵑具有斑點數小於目前所觀察到的鳥兒，那就往樹的左邊搜尋；反之，就往樹的右邊搜尋。靠著持續把樹分成兩個區塊範圍，就能快速的找到你要的杜鵑。

這樣整理儲存數據的資料結構稱為 **二元搜尋樹 binary search tree**，它經常被應用在可動態新增或刪除資料，且需要快速找出特定資料儲存位置的情況。



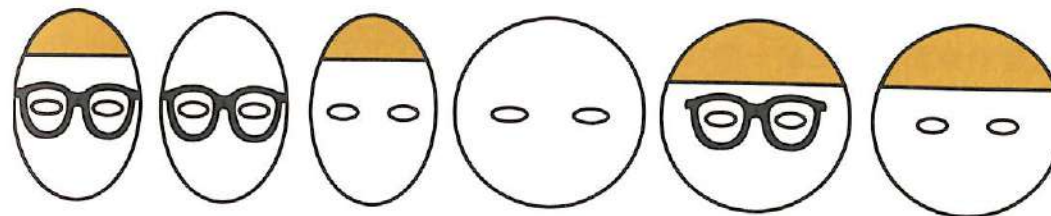
## 關鍵字

二元搜尋樹



## 22. 復活節彩蛋

瑪莉畫了下面六個不同造型的復活節彩蛋。西蒙選了其中一個，要讓瑪莉猜猜他選了哪一個，但瑪莉只能問兩個問題。



瑪莉要問哪兩個問題才能確定西蒙選的彩蛋？

A.

| 彩蛋是橢圓還是正圓？ | 彩蛋是否有眼鏡？ |
|------------|----------|
|            |          |

C.

| 彩蛋是橢圓還是正圓？ | 彩蛋頭頂是否有顏色？ |
|------------|------------|
|            |            |

B.

| 彩蛋頭頂是否有顏色？ | 彩蛋是否有眼鏡？ |
|------------|----------|
|            |          |

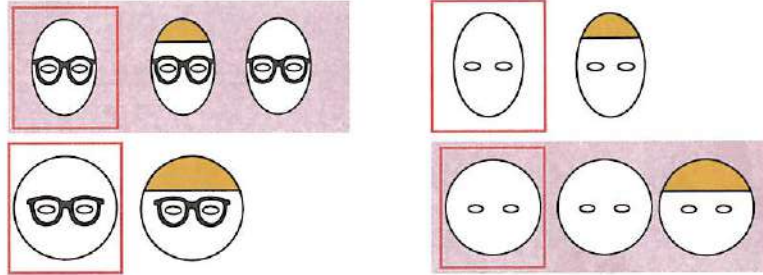
D.

沒有辦法只問兩個問題確定是哪一顆彩蛋。

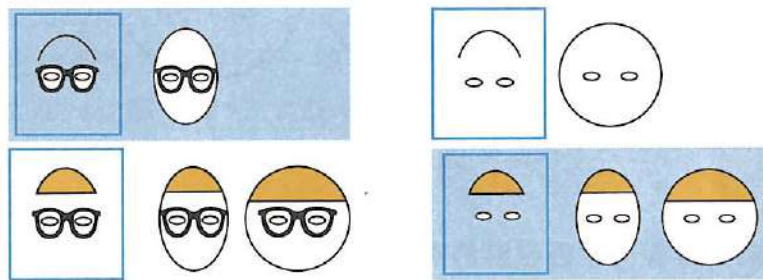


## 正確答案是：D

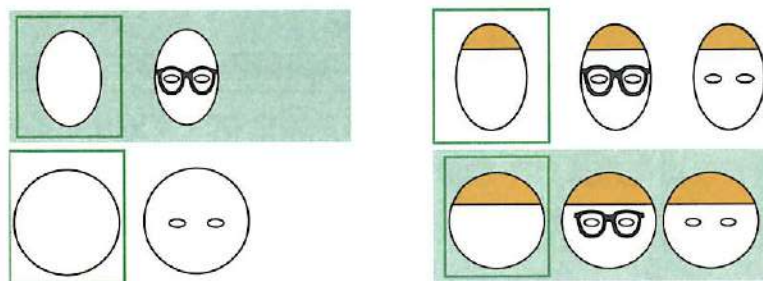
問完選項 A 之後的分類狀況：



問完選項 B 之後的分類狀況：



問完選項 C 之後的分類狀況：



我們可以從上面這個分類表看出：只問兩個問題的話，就只能分出四種不同的類別，可是我們卻有六顆彩蛋要放進這四種類別之中。因此不管我們怎麼放，都無法將這六個彩蛋分別放到不同的類別裡面。也就是說，一定會有兩顆以上的彩蛋被放在同一個類別裡面。

因此，不管瑪莉問的問題是什麼，在只問兩個問題的前提下，瑪莉都有可能遇到無法確定彩蛋的狀況。



## 資訊科學上的意義

在資訊科學裡，**二進制屬性 binary attribute** 是很常出現的。目前的電腦系統大都採取二進制系統，這是因為電子元件的特點為有電和沒電兩種狀態，有電就可以表示為 1，沒電就可以表示為 0，每個 1 或 0 稱為一個位元 bit。在本題中，瑪莉問完一個問題後，彩蛋就會依照「符合」及「不符合」這兩個狀況而分成兩堆，這跟二進制屬性的道理是一樣的。

在電腦的世界中，不論是文字、圖片、影片或音樂都可以透過編碼轉換成二進制的形式。例如：文字的數位化方法是將每一個文字對應一個二進位碼來作識別，當我們從鍵盤輸入 "A" 這個字母時，電腦便會將我們輸入的 "A" 轉換成對應的二進位碼（其中一種編碼為：01000001）。

當每個屬性都只有兩種可能的值，而我們又有  $n$  個屬性可以判別時，我們就可以將事物區分成  $2^n$  種狀況。在存放二進制屬性值的時候，我們可以使用位元，或是使用 **布林值 boolean**（一種邏輯變數，存放兩種值：true 跟 false）。當我們在分析搜尋演算法時，了解二進制屬性的概念就是很重要的基礎。



## 關鍵字

二進制數字、布林值



## 23. 影片觀看次數

海狸波波和貝貝經常在影音平台上發布影片，平台每個月都會通知他們影片被觀看的次數。波波和貝貝不希望觀看次數在傳送過程被別人知道，因此平台將這個數字用一串黑色圓圈 ● 和黃色圓圈 ● 符號表示後傳給他們，波波和貝貝可用下方的對照表將符號轉換為數字。

|             |             |             |             |             |
|-------------|-------------|-------------|-------------|-------------|
| 0 : ● ● ● ● | 1 : ● ● ●   | 2 : ● ● ● ● | 3 : ● ● ● ● | 4 : ● ● ● ● |
| 5 : ● ● ● ● | 6 : ● ● ● ● | 7 : ● ● ● ● | 8 : ● ● ● ● | 9 : ● ● ● ● |

下圖是這個月波波和貝貝收到的影片觀看次數符號：



請問這個月波波和貝貝的影片觀看次數是幾次？

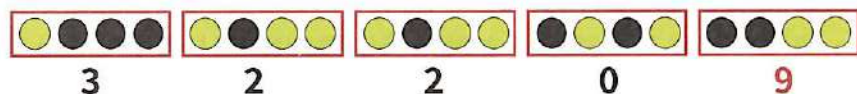
- A. 32209
- B. 32208
- C. 417011
- D. 417511



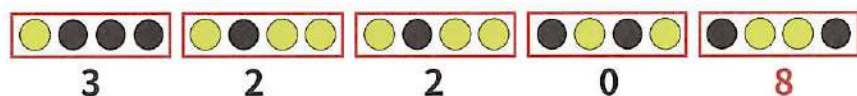
## 正確答案是：D. 417511

我們將選項的數字對應到符號，並標示出錯誤的數字。

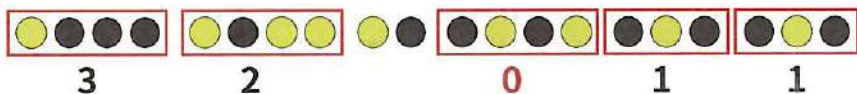
選項 A



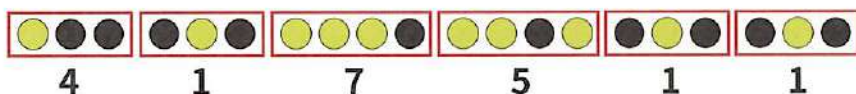
選項 B



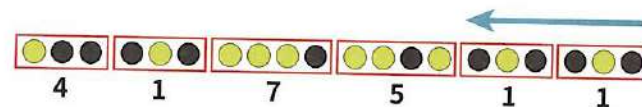
選項 C



選項 D



所以正確答案是 D 選項 417511。觀察符號時可以由右往左或左往右，由右往左檢查整串符號比較容易找到對應的數字。由右往左的檢查過程中每一段符號都只會對應到唯一一個數字，這是因為在本任務中的數字符號是「無後綴」(suffix-free) 的，即任一組代表數字的符號都不是其他數字符號的後綴。



從符號的左邊開始檢查也可以找到答案，但可能會遇到一些「錯誤的開始」，因為這些數字符號並非「無前綴」。

舉例來說：一開始可能會把符號 解碼成 4 ，但後續的 就找不到對應的數字了。



## 資訊科學上的意義

在電腦的世界中，不論是文字、圖片、影片、音樂都可以轉換成 **二元碼 binary code** (0 或 1) 的形式，相當於本任務中的黑色圓圈 或黃色圓圈 。在本任務中，每個數字用到的符號數量不固定，有的數字由 4 個符號組成、有的由 3 個符號組成。這樣的編碼稱為 **可變長度碼 variable-length code**，可變長度碼在生活中有很廣泛的應用。包含壓縮演算法、常見的 UTF-8 編碼等。

在詳解中提到本任務中的數字符號是「無後綴 suffix-free」的，這表示由右往左解碼是比較容易的。在實務中可變長度碼通常是由左往右 (或從頭至尾) 檢查，所以會設計成「無前綴 prefix-free」的，稱作「前置碼 prefix code」。



## 關鍵字

二元編碼、可變長度碼

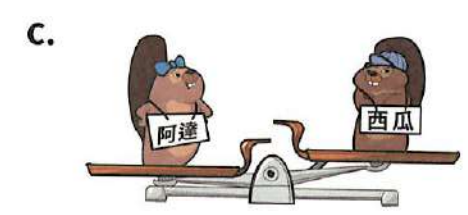
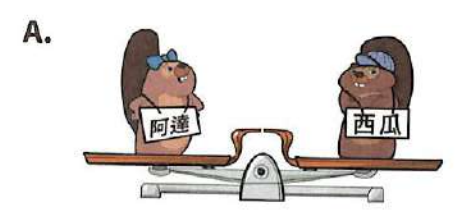


## 24. 比一比

某天四隻海狸發現一個天秤，就調皮的站上去拍照，以下是其中三張照片：



從上面三張照片推斷，下列哪一張照片也是在同一天拍的？





## 正確答案是：C

可以透過推導解這個任務。

- 如果 (A) 這張照片是正確的，意味著阿達跟西瓜一樣重，從中間的照片可知此時貝塔跟丹丹一樣重，但是從最右邊的照片觀察，天平理論上也應該維持平衡（因為阿達加上丹丹的體重也應該與西瓜加上貝塔的體重相同），由此可知 (A) 這張照片是不正確的。
- 如果 (B) 這張照片是正確的，意味著貝塔跟丹丹一樣重，上述的推論一樣成立，也就是阿達跟西瓜一樣重，但最右邊的照片還是推翻了這個假設，所以 (B) 這張照片還是不正確。
- 如果 (D) 這張照片是正確的，意味著貝塔比丹丹重，同樣的以中間的照片觀察，為了維持平衡，西瓜就一定會比阿達來得重，因此當貝塔跟西瓜在同一邊的時候，就肯定會比阿達跟丹丹還要重，所以 (D) 這張照片也是不正確的。
- 最後，如果 (C) 這張照片是正確的，表示西瓜比阿達要輕，若以這個條件去檢驗這三張照片，會發現都是合理的。



## 資訊科學上的意義

**邏輯推理** 是從一系列事證中推斷出新的資訊。把這些條件的描述判斷（也就是邏輯）進行推理，如果結論是正確的，則視為正確（或是視為 1），否則視為不正確（或是視為 0），以此來研判之後的條件是否正確或錯誤，這個題目裡學生必須根據 3 張照片的資訊條件，比較推斷出新的資訊，確認結論是否正確。在培養資訊科學基本能力上，是要能提出符合已知條件的假設，並且能夠從已知的條件當中獲得更多的訊息，然後在這個問題當中去找出解決的方法，在推理過程中，「比較」則是一種常見的方法，比方說排序一份清單，你就需要藉由「比較」來決定此時程式應該執行哪一個區段。

邏輯推理是電腦科學的基礎，在 LISP 或 prolog 程式語言當中，邏輯推理尤其重要，它被運用在符號操作與邏輯推理上，本質上來說，邏輯推理可以說是程式語言在解決問題的過程，此外我們也經常使用它來檢驗程式的複雜度與正確性。



## 關鍵字

邏輯推理、約束、比較

## 25. 偷畫賊

「文藝物流」是一家專門運送名畫的快遞公司。要運送的畫作會在公司仔細包裝後，依先後順序由下到上堆疊好，再由快遞員直接送到目的地。每一位快遞員要送貨時，會從堆疊的最上面拿取一幅畫作後出發。為保障物品的安全，公司會詳細記錄每幅畫作完成包裝與準備運送的相關資訊。

| 畫作包裝記錄 |           |
|--------|-----------|
| 時間     | 畫作名稱      |
| 11:40  | 〈草地上的海狸〉  |
| 12:15  | 〈歡喜狸〉     |
| 12:55  | 〈日月爭輝〉    |
| 13:30  | 〈魔法森林〉    |
| 14:18  | 〈橡樹與樺樹〉   |
| 15:10  | 〈沼澤中的羅曼史〉 |



| 畫作運送記錄 |       |
|--------|-------|
| 時間     | 快遞員代碼 |
| 12:25  | A     |
| 13:35  | C     |
| 14:35  | A     |
| 14:40  | B     |
| 15:20  | C     |
| 15:35  | D     |

這天物流公司接到緊急通知，名畫〈日月爭輝〉並沒有如期送達目的地的美術館，推測是快遞員把畫偷走了！

請問是哪位快遞員偷走了〈日月爭輝〉這幅畫？

- A.
- B.
- C.
- D.



## 正確答案是：B

正確答案是 B。要找出這題答案，很重要的關鍵是「晚到的畫作會放在最上層，快遞員也總是取走最上層的畫作」。接下來透過畫作的包裝記錄與出貨記錄，可以整理出如下表各個時間點的畫作狀況；就能得出答案：〈日月爭輝〉是在 B 快遞員的手上。

|       | 包裝記錄       | 出貨記錄                  | 完成後畫作堆疊順序                     |
|-------|------------|-----------------------|-------------------------------|
| 11:40 | 放入〈草地上的海狸〉 |                       | 〈草地上的海狸〉                      |
| 12:15 | 放入〈歡喜狸〉    |                       | 〈歡喜狸〉<br>〈草地上的海狸〉             |
| 12:25 |            | A 快遞員取走最上層<br>〈歡喜狸〉   | 〈草地上的海狸〉                      |
| 12:55 | 放入〈日月爭輝〉   |                       | 〈日月爭輝〉<br>〈草地上的海狸〉            |
| 13:30 | 放入〈魔法森林〉   |                       | 〈魔法森林〉<br>〈日月爭輝〉<br>〈草地上的海狸〉  |
| 13:35 |            | C 快遞員取走最上層<br>〈魔法森林〉  | 〈日月爭輝〉<br>〈草地上的海狸〉            |
| 14:18 | 放入〈橡樹與樺樹〉  |                       | 〈橡樹與樺樹〉<br>〈日月爭輝〉<br>〈草地上的海狸〉 |
| 14:35 |            | A 快遞員取走最上層<br>〈橡樹與樺樹〉 | 〈日月爭輝〉<br>〈草地上的海狸〉            |
| 14:40 |            | B 快遞員取走最上層<br>〈日月爭輝〉  | 〈草地上的海狸〉                      |



## 資訊科學上的意義

堆疊 **stack** 是一種抽象的資料型態。它記錄線性資料進出的方式，資料只能從一個端點 top 加入與刪除，進出的原則是後進先出 Last In First Out, LIFO。堆疊通常搭配陣列或 **串列** 實作，另外一種相似的資料結構是佇列 queue。

本題中畫作待運送的順序以及快遞員取畫的順序都是採用堆疊來進行。只要把握住堆疊進出的原則，並且搭配時間記錄，就可以清楚掌握每一幅畫作是由哪一位快遞員運送的。日常生活中類似堆疊的例子很多，例如光碟桶存放光碟的順序等。在資訊科學中，堆疊後進先出的特性經常使用在儲存資料和處理資料上，例如：瀏覽器的返回鍵，就是應用堆疊的方法；當按返回鍵（←）時，最先看到的頁面就是之前最後被瀏覽過的網頁。另外遞迴、深度優先搜尋、走迷宮等問題也都使用堆疊的概念。



## 關鍵字

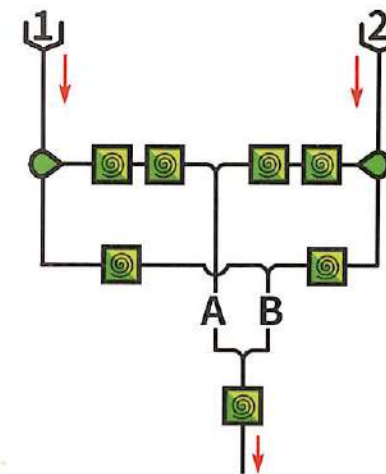
堆疊、串列



## 26. 特製冰淇淋機器

小狸愛吃堅果口味與香草口味的冰淇淋，他有一台特製的冰淇淋機器。

機器有 1 跟 2 兩個管線可放入配料，配料會順著管線從上向下流動，最後做出堅果或香草口味的冰淇淋。下圖是機器管線圖，綠色代表堅果，黃色代表香草，流動路線請參考圖片箭頭方向。



這台機器有幾個裝置，功能介紹如下：

| 圖示 | 裝置    | 功能   |
|----|-------|--|
|    | 配料轉換器 | 把堅果轉成香草，或者把香草轉成堅果。                               |
|    | 配料轉換器 | 如果堅果配料經過這個裝置，會轉往尖端的方向流動；如果香草配料經過這個裝置不受影響，繼續向下流動。 |
|    | 管線交疊  | 代表兩支管線獨立行走並在此交疊，內容物不會混合。                         |

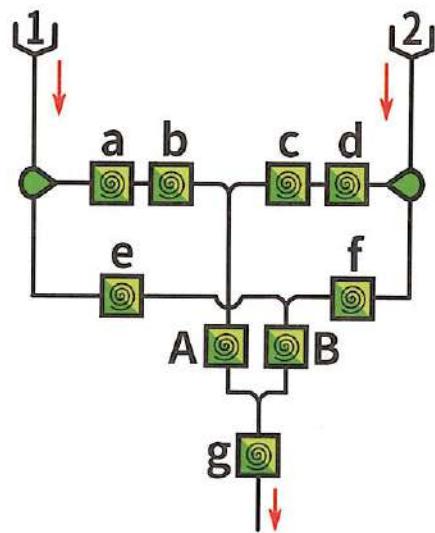
今天小狸想吃堅果冰淇淋，若 1、2 管線只能放入同一種配料，現在有 A、B 兩處，請問要加裝配料轉換器在哪處，才能讓機器最後一定會做出堅果冰淇淋？

- A. 放 A 位置
- B. 放 B 位置
- C. A 和 B 位置都要放
- D. 放 A 或 B 位置都不對



## 正確答案是：C. A 和 B 位置都要放

我們先看正確答案 C 選項「A 和 B 位置都放轉換器」的執行狀況，如下圖：



放入香草配料時，從入口到出口會經過配料轉換器【e+B+g】或是【f+B+g】，最後輸出堅果口味冰淇淋；放入堅果配料時，從入口到出口會經過配料轉換器【a+b+A+g】或是【d+c+A+g】，最後做出堅果口味冰淇淋，符合題目要求。

接下來再看其他選項的狀況

### 1. 只放在 (A) 位置：

放入香草配料時，從入口到出口會經過配料轉換器【e+g】或是【f+g】，最後做出香草口味冰淇淋，不符合題目要求。

### 2. 只放在 (B) 位置：

放入堅果配料時，從入口到出口會經過配料轉換器【a+b+g】或是【d+c+g】，最後做出香草口味冰淇淋，不符合題目要求。

### 3. 都不放配料轉換器：

那麼上面第 1 和第 2 點的狀況同樣會發生，最後做出香草口味冰淇淋，不符合題目要求。



## 資訊科學上的意義

這一題主要的科學概念是 **三進制邏輯 ternary logic**。我們平時學習的二進制邏輯值，只有真與假兩種結果，但是三進制邏輯具有三種狀態，分別是真、假，與不確定的第三值。

這個题目的三進制值分別是香草配料、堅果配料以及沒有配料（空的管線）。

☞ 配料轉換器相當於傳統二進制轉換，可以讓香草與堅果互換，但是本題另外設計了 🍷 堅果控制器，讓某些管線可能沒有配料流入，因此空的管線在經過

☞ 配料轉換器之後，它的內容仍然是空，形成了第三種邏輯值。

除了電路設計、**邏輯閘** 會運用三進制邏輯之外，資料庫語言 SQL 也是三進制邏輯的實際應用。除了真、假之外，SQL 語言設計了 NULL 值，代表目前資料是缺漏的狀態。NULL 值與 0 值的意義不同，前者是真實狀態應該有資料，但是資料庫的欄位內容缺漏，而 0 值是代表欄位中儲存確切的數值為 0。另外，市面上也確實有電腦是以三進制邏輯概念設計出來的，並且和傳統二進制電腦運作得一樣棒。



## 關鍵字

三進制邏輯、邏輯閘



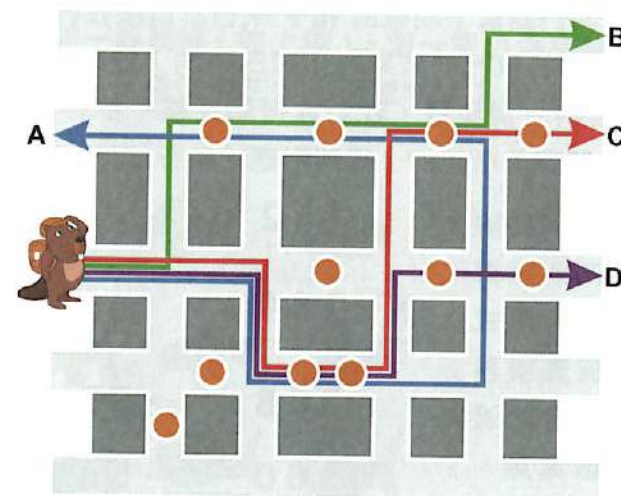
## 27. 觀光巴士

小海狸在旅行時從圖中的位置搭上觀光巴士，並沿途記錄了一些訊息。

他記錄了每一次轉彎，但沒有記錄是左轉還是右轉，他還記錄了經過第幾個路口或公車站 ● 的數目。以下是他記錄的內容：

- 出發
- 轉彎
- 轉彎
- 經過 3 個公車站後轉彎
- 直走到第 1 個路口轉彎

請問小海狸所搭的觀光巴士路線是哪一條？



- A. 路線
- B. 路線
- C. 路線
- D. 路線



### 正確答案是：B. 路線

正確答案是 B 選項綠色路線，觀光巴士從起點出發，在 2 次轉彎（左轉和右轉）後，行經 3 個公車站，接著左轉行駛到第 1 個路口再右轉抵達 B 點。

- A 選項的藍色路線不正確，因為觀光巴士在 2 次轉彎（右轉和左轉）後，只經過 2 個公車站（而不是小海狸記錄的 3 個）。接著巴士左轉兩次，行經 3 個公車站後抵達 A 點。
- C 選項的紅色路線不正確，因為觀光巴士在 2 次轉彎（右轉和左轉）後，也只經過 2 個公車站就左轉。接著行駛到第 2 個路口才轉彎（而不是小海狸記錄的第 1 個路口）。
- D 選項的紫色路線不正確，因為觀光巴士在 2 次轉彎（右轉和左轉）後，也只經過 2 個公車站就左轉。轉彎後行駛到第 1 個路口再右轉抵達 D 點。



### 資訊科學上的意義

本任務是關於如何精準的追蹤小海狸的記錄，並驗證哪些路線不符合小海狸記錄的內容。因為沒有記錄是左轉還是右轉，所以每一條記錄單獨來看都不足以確認是哪一條路線，但是我們可以用刪去法將不符合記錄的路線排除。

在撰寫程式的時候，我們要寫下明確的指令來告訴電腦我們希望它完成的事情，但是我們常常會發現程式執行的結果跟預想的不同，這時候除錯 debugging 就是寫程式時非常重要的一個步驟，我們會從頭完整一步一步執行程式碼，然後一一驗證它是否會依照我們所想的，得到對應的執行結果。

我們在生活中也常常用到除錯的概念，例如當四則運算計算錯誤時，可以利用直式一步一步驗算；在檢查機器故障時，也會按照一定的流程檢查，例如先開啟電源、再調整參數等等。

$$\begin{array}{r}
 52 \\
 \times 284 \\
 \hline
 208 \xrightarrow{A} 52 \times 4 = 208 \\
 4160 \xrightarrow{B} 52 \times 80 = 4160 \\
 20400 \xrightarrow{C} 52 \times 400 = 20400 \\
 \hline
 24768
 \end{array}$$



### 關鍵字

指令、除錯、程式設計



## 28. 排隊的魚

小海狸們在河中看到有一群魚 (20 條) 在水裏排成一條直線，位置排列編號為 1~20，如圖一所示。



圖一

當小海狸們喊出二個數字 A 和 B (A<B)，則位置 A 左側所有的魚都會游走，並且位置 B 右側所有的魚也都會游走。

舉例來說，當小海狸們喊出 2 和 17 二個數字後，某些魚兒就會游走，只剩 16 條魚留在排列的直線，如圖二所示。



圖二

而魚群的位置編號就會從左側的第 1 隻魚開始重新編號。



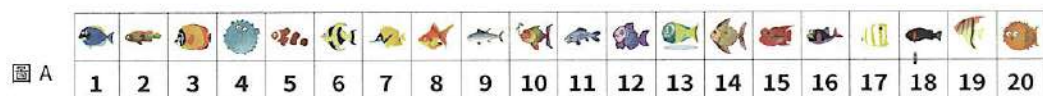
一開始有 20 條魚排成一條直線 (如上圖)，第一隻小海狸喊出 4 和 18，接著第二隻小海狸喊出 6 和 12，最後第三隻小海狸喊出 2 和 5。請問最後哪些魚還留在直線上？

- A.
- B.
- C.
- D.
- E.



### 正確答案是：D

一開始 20 條魚的魚群排列位置及編號如圖 A。



第一隻小海狸喊出 4 和 18 後，圖 A 中編號 1~3 及 19~20 的魚會游走，新的魚群排列位置及編號如圖 B。



第二隻小海狸喊出 6 和 12 後，圖 B 中編號 1~5 及 13~15 的魚會游走，新的魚群排列位置及編號如圖 C。



最後第三隻小海狸喊出 2 和 5 後，圖 C 中編號 1 及 6~7 的魚會游走，最後的魚群排列位置如圖 D。



### 資訊科學上的意義

序列 sequences 是將元素排成一列，每個元素不是在其他元素之前，就是在其他元素之後，而元素之間的順序非常重要；其存放方式可使用陣列 array 儲存。本題可用序列呈現魚兒排列在直線的前後關係如圖 A 表示。依海狸們下達指令，刪除指令數字前後的子序列後，其序列編號可依序重新排列。許多現實生活中的問題可以序列方式安排其優先順序。舉例，當序列為指令序列時，指令的先後順序不同將會影響執行的結果；當序列是一個字母或其他符號的排列時，通常會被稱為一個字串 string，通常應用於分割子字串 substring、片段 slice 或類似的敘述。



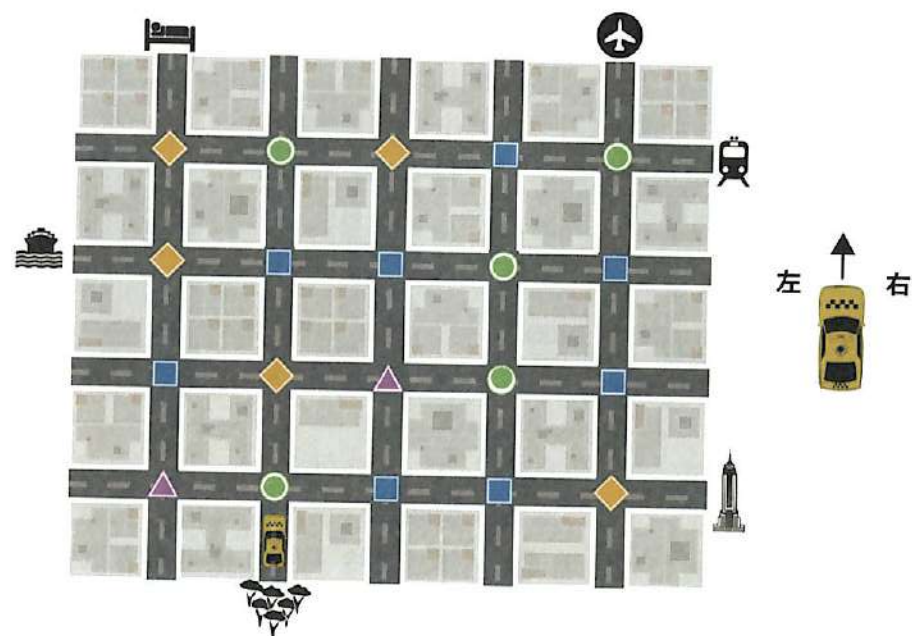
### 關鍵字

序列、字串、陣列

## 29. 嗨計程車！

在海狸城市中有四種交通號誌 ，用來控制自動駕駛計程車「向前直行」、「向左轉」、「向右轉」或「向後轉」。小海狸不知道每種號誌各代表什麼指令，他只知道自動駕駛計程車會依每個經過路口設定的號誌執行指令，前往下個路口。

下圖中的自動駕駛計程車依號誌指示，會從公園 出發，最後抵達機場 。



請幫小海狸找出每個交通號誌各代表什麼指令？


- A. 向右轉 向左轉 向前 向後轉
- B. 向前 向左轉 向右轉 向後轉
- C. 向前 向右轉 向左轉 向後轉
- D. 向左轉 向右轉 向前 向後轉



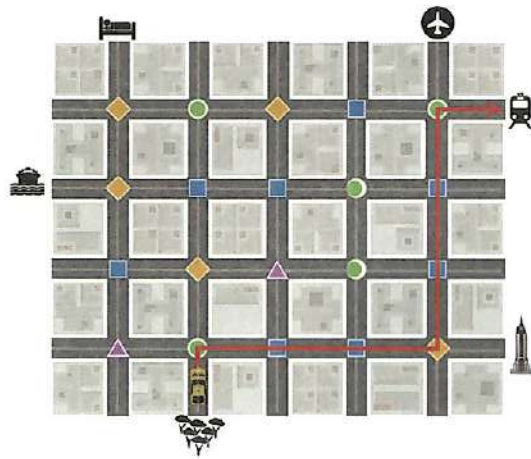
## 正確答案是：C

以下顯示依各選項指令，從起點到終點的路徑：


選項 A

錯誤指令，依 A 指令將會抵達車站 

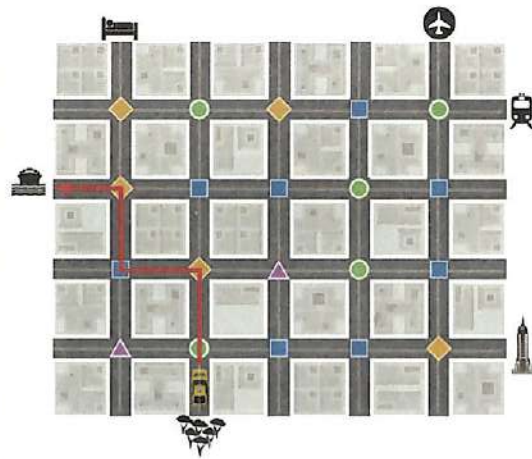
● 向右轉    ◆ 向左轉    ■ 向前    ▲ 向後轉




選項 B

錯誤指令，依 B 指令將會抵達港口 

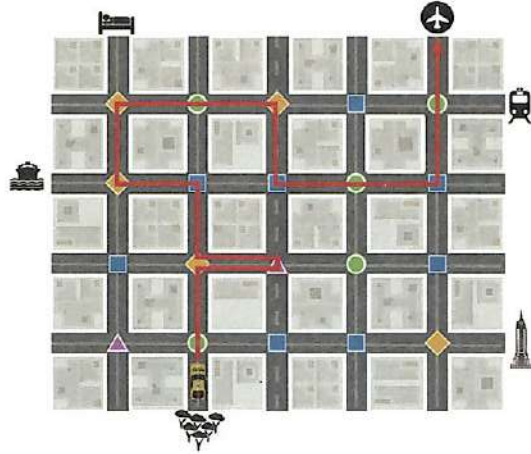
● 向前    ◆ 向左轉    ■ 向右轉    ▲ 向後轉




選項 C

正確指令，依 C 指令將會抵達機場 

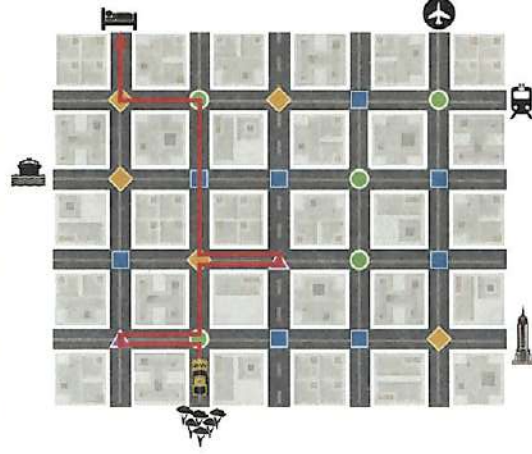
● 向前    ◆ 向右轉    ■ 向左轉    ▲ 向後轉



選項 D

錯誤指令，依 D 指令將會抵達醫院 

● 向左轉    ◆ 向右轉    ■ 向前    ▲ 向後轉



## 資訊科學上的意義

每個 **指令 instruction** 會改變電腦中記錄資料的狀態 state(計程車位置及移動方向)，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。當程式設計師向電腦發出指令時，必須準確了解每個指令執行的效果，否則執行出來的結果可能不是他們所要的。

在本題中自動車遇到燈號就會進行對應的指令，當指令的意義不同時，自動車的行走路徑便會隨之改變。在現代生活中，我們要去不熟悉的地方常會使用導航程式，程式會找出一系列可到達終點的路徑指令，只要我們依照指令行走，便可到達我們要去的地方。導航程式用來找出起點到終點的方法稱為路徑搜尋演算法，可能採用最短路徑、最快路徑，或經由其他地點等限制來搜尋路徑。



## 關鍵字

指令、程式設計

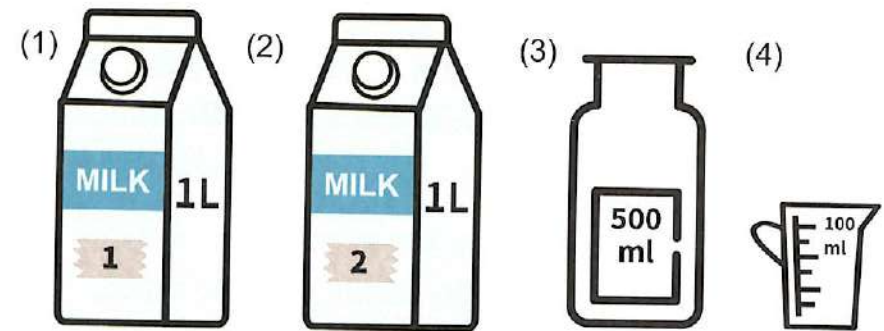




### 30. 蛋糕製作

小娜有兩瓶已經開過的 1 公升牛奶，每瓶目前都只剩一半多一點。她想製作蛋糕，需要準確測量出 700 毫升的牛奶。她除了兩瓶牛奶之外，還有一個 500 毫升的空瓶子和一個 100 毫升的空杯子。

小娜只可以使用以下 4 個容器測量，直到某一容器準確盛裝 700 毫升牛奶為止。



小娜用 (1) → (2) 代表將容器 1 的牛奶倒入容器 2，直到容器 2 已滿，或是容器 1 已空。以下哪個是實現小娜目標的最短順序？

- A. (1) → (4); (1) → (3); (1) → (2); (3) → (1); (4) → (1); (2) → (1).
- B. (1) → (2); (2) → (3); (3) → (4); (4) → (2); (3) → (4); (4) → (2).
- C. (2) → (3); (2) → (1); (3) → (2); (1) → (4); (4) → (2); (1) → (2).
- D. (2) → (3); (2) → (1); (1) → (4); (4) → (2); (1) → (4); (4) → (2).



## 正確答案是：B

這題的正確順序是 B：(1) → (2)；(2) → (3)；(3) → (4)；(4) → (2)；(3) → (4)；(4) → (2)。首先，小娜將第 2 個瓶子裝滿 1000 毫升，然後她將第 2 個瓶子的牛奶，倒入第 3 個瓶子中裝滿 500 毫升，因此第 2 個瓶子剩 500 毫升，接著，她將第 3 個瓶子中的牛奶，倒入第 4 個杯子中裝滿 100 毫升後，再倒入第 2 個瓶子中，重覆 2 次，前後總共倒了 6 次，完成任務。

其他選項為什麼不正確，說明如後：

選項 A：

(1) → (4)；(1) → (3)；(1) → (2)；(3) → (1)；(4) → (1)；(2) → (1)。

第 1 個瓶子倒入第 4 個的杯子，裝滿 100 毫升後，接著第 1 個瓶子剩的牛奶再倒入第 3 個瓶子，此時，我們無法確定第 3 個瓶子是否會裝滿 500 毫升的牛奶。

選項 C：

(2) → (3)；(2) → (1)；(3) → (2)；(1) → (4)；(4) → (2)；(1) → (2)。

前五個步驟都能確定倒入的量，但最後一個步驟：第 1 個瓶子倒入第 2 個瓶子，我們無法確定倒入的量是多少。

選項 D：

(2) → (3)；(2) → (1)；(1) → (4)；(4) → (2)；(1) → (4)；(4) → (2)。

第 2 個瓶子倒入第 3 個瓶子中裝滿 500 毫升，第 2 個瓶子把剩下的牛奶倒入第 1 個瓶子，第 2 個瓶子現已全空，現在第 1 個瓶子倒滿第 4 個瓶子 100 毫升後，再倒入第 2 個瓶子，所以第 2 個瓶子現有 100 毫升牛奶，再重覆 1 倒 4，4 倒 2，故第 2 個瓶子目前只有 200 毫升牛奶，並未完成盛裝 700 毫升牛奶任務。



## 關鍵字

演算法、指令順序



## 資訊科學上的意義

在日常生活中，我們要解決一個問題，必須思考解決問題的步驟，而這一系列的流程與步驟，在資訊科學中我們稱之為**演算法 algorithm**。著名的電腦科學家 Donald Ervin Knuth，在他的著作《電腦程式設計藝術》(The Art of Computer Programming) 中，提出演算法有五個特性：

1. 輸入 input：每個問題都有零個或以上的輸入
2. 輸出 output：每個問題應有一個或以上輸出，輸出是演算法計算的結果
3. 有限性 finiteness：應在有限的步驟或時間內完成
4. 明確性 definiteness：每一個步驟應可以明確執行，並且執行結果也是明確的
5. 有效性 effectiveness：每一個步驟必須與用紙筆推導出來的結果相符，還必須具有可行性

為了完成此次的任務，盛裝 700 毫升牛奶，我們必須依據指令一步一步執行，就如同電腦執行程式時，為了解決問題，電腦會一個步驟一個步驟依指令順序完成任務，包含電腦系統或應用程式的運作，都是依據演算法來執行。



### 31. 保險箱密碼

海狸銀行保險箱的密碼是由下列 8 個符號中取 3 個排列而成。



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|   |   |   |   |   |   |   |   |

密碼每天都會自動更改，更改的規則是每天向右推移一個符號。也就是說， 會換成它右邊的符號 ，而最右邊的符號 會推移到左邊第一個符號 。

例如：如果週日的密碼是 ，週一的密碼將是 。

上週日銀行經理將保險箱密碼設定為 ，並將接下來一整週的密碼記下來。

請問下列哪個會是記下來的密碼？





正確答案是：D

一整週的密碼記錄如下表：

|     |             |
|-----|-------------|
| 上週日 | # ▶ ♥ (524) |
| 週一  | ● * # (635) |
| 週二  | △ ♥ ● (857) |
| 週三  | ☾ # △ (857) |
| 週四  | ⊗ ● △ (168) |
| 週五  | ▶ △ ⊗ (271) |
| 週六  | * ☾ ▶ (382) |



資訊科學上的意義

加密和解密是資訊科學部分中稱為密碼學的基本概念。在此任務中使用的是凱撒加密法，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通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)。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。在本任務中，如果有人知道符號表，並觀察密碼的變化，很容易就能發現規律並破解密碼。



關鍵字

加密、凱薩密碼

## 32. 柵欄塗油漆

小狸想幫他家中花園的12根柵欄塗上油漆。他想把柵欄塗上紅色、橘色、黃色、綠色、藍色和紫色，而且每種顏色各要塗滿2根柵欄。



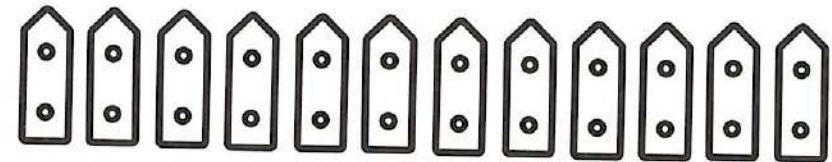
他有3個分別裝滿紅色、黃色和藍色油漆的桶子，以及3個用來混合顏色的空桶(空桶子與油漆桶容量一樣)。用1整桶油漆剛好可以塗滿4根柵欄。



為了得到橘色、綠色和紫色，他必須按照以下規則混合油漆：



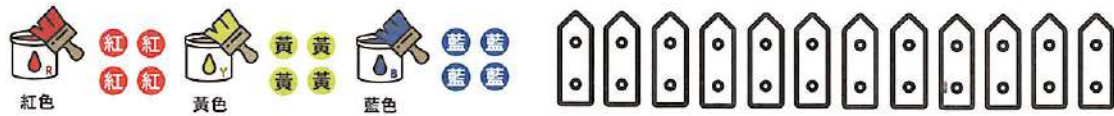
小狸最多能將幾根柵欄塗滿他希望的顏色？



- 他可以塗滿8根柵欄
- 他可以塗滿10根柵欄
- 他可以塗滿所有12根柵欄
- 他可以塗滿6根柵欄



正確答案是：C. 他可以塗滿所有 12 根柵欄



為了完成任務，首先，1 桶油漆可分成 4 份，小狸用 2 份紅色油漆將 2 根柵欄塗成紅色；用 2 份黃色油漆將 2 根柵欄塗成黃色；再用 2 份藍色油漆將 2 根柵欄塗成藍色（如下圖）。



現在，紅色、黃色及藍色油漆，各剩下 2 份。要將 2 根柵欄塗成橘色，他先在第 1 個空桶中混合 1 份的紅色油漆和 1 份的黃色油漆，得到 2 份的橘色油漆。這個步驟後，他剩下 1 份紅色油漆，1 份黃色油漆和 2 份藍色油漆。接下來，他在第 2 個空桶中混合 1 份的黃色油漆和 1 份的藍色油漆，得到 2 份的綠色油漆，將 2 個柵欄塗成綠色。最後，他將剩下的 1 份紅色油漆與最後 1 份藍色油漆在第 3 個空桶中混合，得到 2 份的紫色油漆，將最後兩個柵欄塗滿。



關鍵字

工作指定問題、任務分配問題



資訊科學上的意義

此問題採用有限的油漆在柵欄塗上油漆，得到特定顏色油漆需要不同的油漆混合，且限定每根柵欄只塗上一種顏色，且每種顏色油漆只能分配到二根柵欄。目標是怎樣分配油漆，讓塗上油漆的柵欄數最多，類似的問題屬於**任務分配問題**。

以本題為例，有 3 桶不同顏色的油漆，每種顏色皆可塗 4 根柵欄，共有 12 份油漆。但是否能塗滿 12 根柵欄，則必須看分配方式是否符合顏色的要求限制。紅、黃、藍各先使用了 2 份，剩餘份數正好可配出橘色、綠色與紫色各 2 份，如圖所示。

|    |    |    |    |       |    |       |    |       |
|----|----|----|----|-------|----|-------|----|-------|
|    |    |    |    |       |    |       |    |       |
| 紅色 | 黃色 | 藍色 | 紅色 | 橘色    | 黃色 | 綠色    | 藍色 | 紫色    |
| 紅  | 黃  | 藍  | 紅  |       | 黃  |       | 藍  |       |
| 紅  | 黃  | 藍  |    | 紅 + 黃 |    | 黃 + 藍 |    | 紅 + 藍 |
| 紅  | 黃  | 藍  |    | 橘     |    | 綠     |    | 紫     |

在生活中也經常遇到類似的問題，例如工廠裡的機器用來製造產品零件，每台機器每天能製做的產品零件數是有限的，有些產品只需要單一零件，有些則需要組合不同零件；所以要規劃機器生產零件的種類及時間分配，看如何能在固定機器數及限定時間內完成需要的產品數量。另一個應用範例是學生在考試前，在時間有限情況下，要考慮什麼時間適合讀什麼科目，以及如何分配各科的讀書時間，才能複習完各考科以提高成績，皆類似一個任務分配問題。





### 33. 火星人的密碼

火星人設計了一種文字加密方式，加密過程分成兩個部分。

我們用 "MARS" 這個單字來舉例：

| A | B | M | N  | O  | R   | S   | T   | U   |
|---|---|---|----|----|-----|-----|-----|-----|
| 1 | 2 | 4 | 10 | 50 | 180 | 300 | 650 | 960 |

- 第一部分：依照上面對照表，將單字中每個字母的代表數相加。  
 $M=4, A=1, R=180, S=300$ ，相加後得到  $4+1+180+300=485$ 。
- 第二部分：將單字轉換成每個字母的順序編號。  
 "MARS" 依照字母順序排列會是 "A, M, R, S"，按順序編號為： $A=1, M=2, R=3, S=4$ ；因此 "MARS" 的順序編號是：2134。

把以上兩個部分的結果合在一起，就完成加密；因此 "MARS" 加密後為 485；2134。

用火星人的方式加密 "SATURN"，會得到下列哪組數字？

- A. 1440 ; 415632
- B. 1440 ; 718964
- C. 2101 ; 415632
- D. 2101 ; 718964





## 正確答案是：C. 2101 ; 415632

本題其實可以不用計算就能找到正確答案。

首先，這個密碼的第一個部份是由每個英文字母的代表數來加總所產生的，在選項中只有兩種可能的答案：1440 跟 2101。而在 SATURN 這個字當中，T 跟 U 的代表數在表格的最右邊，都是很大的數字，分別為 650 跟 960。顯而易見的，整個字加總後會超過 1440，因此第一部份的正確答案是 2101 才對。

這個密碼的第二個部份也只有兩種可能：415632 跟 718964。這個部份的數字代表的是每個字母的順序編號。然而，SATURN 這個字裡面只有 6 個英文字母，因此不應該出現 7 以上的數字。也就是說，718964 是錯的，415632 才是正確答案。

選項 A 是錯的，因為密碼的第一部份是錯的。

選項 D 是錯的，因為密碼的第二部份是錯的。

選項 B 是錯的，因為密碼的兩個部份都是錯的。當然，我們也可以證明選項 C 的密碼是對的，步驟如下：第一個部份：S=300, A=1, T=650, U=960, R=180, N=10。總和 = 300+1+650+960+180+10= 2101 第二個部份：單字裡的字母順序為 S=4, A=1, T=5, U=6, R=3, N=2。合起來就是 415632。因此，完整的密碼就是 2101 ; 415632。



## 資訊科學上的意義

本題用到了兩個資訊科學的概念：檢查總和 **check sums** 與 換位密碼法 **transposition cipher**。

檢查總和通常用在編碼 **encoding** 而不是加密，它可以透過某些錯誤檢測的方法，對傳送資料的完整性進行檢查。檢查總和通常都會置放在數位資料後面來驗證資料的正確性，常應用於檢查網路傳輸或是光碟、硬碟裡的資料是否發生錯誤。在本題中，我們透過對照表，把每個字母的代表數字相加，得到了一個總和，成為密碼的第一個部份。

換位密碼法通常是透過字元之間的互相替代或換位規則，把原始文字轉換為加密後的文字。而在本題中，我們是把每個字母置換為它在原始單字中的順序編號，從而得到密碼的第二個部份。

**密碼學 cryptography** 是研究如何使訊息安全地被傳遞與接收的學問，通常利用數學方法來對資料加密和解密。密碼演算法領域中有很多種方法，此題後半段運用的是換位密碼法。

密碼學讓我們能夠儲存資訊，或透過並不安全的網路（例如網際網路）來傳遞訊息，而任何不該看到這個資訊的人都將無法讀取它。加密時會使用數學方法把「明文資料」與一個「加密金鑰」結合在一起，以產生「加密資料」也就是所謂的「密文」，之後便必須利用「解密金鑰」才能將「密文」還原回「明文資料」。



## 關鍵字

密碼學、換位密碼法、檢查總和

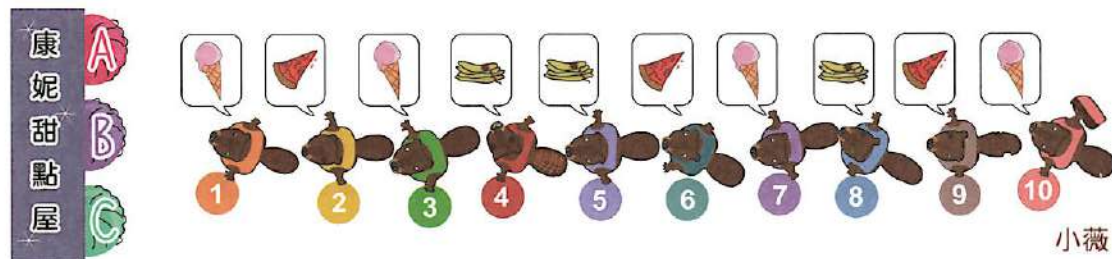


### 34.1 康妮甜點屋 - 題組一

海狸鎮上有間康妮甜點屋，甜點屋外的黑板會寫上今天準備的甜點及其製作時間，如下圖：



為了減少大家等候的時間，康妮開了三個櫃台接受大家點餐，每個櫃台一次服務一位客人，海狸點完餐後，會繼續在該櫃台等待直到取完餐點。如果有兩個以上櫃台同時空出，海狸就依櫃台 ABC 的順序去櫃台點餐。下面這張圖把前面十位海狸各自想要買的甜點都標示出來了，最後一位是小薇。



按照以上排隊的狀況，請推測小薇應該會到哪一個櫃台點餐？

- A. 櫃台
- B. 櫃台
- C. 櫃台

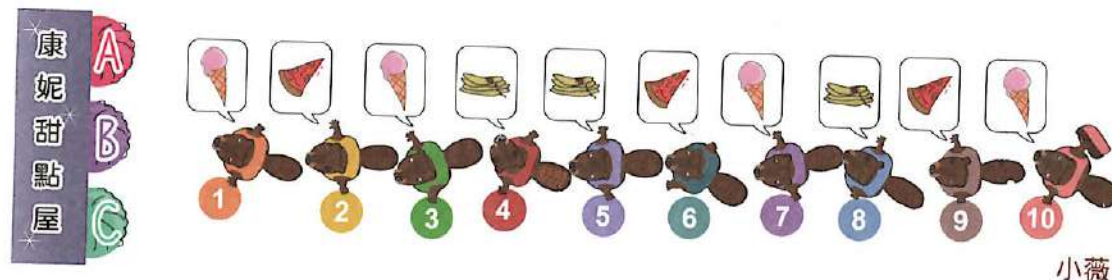


### 34.2 康妮甜點屋 - 題組二

海狸鎮上有間康妮甜點屋，甜點屋外的黑板會寫上今天準備的甜點及其製作時間，如下圖：



為了減少大家等候的時間，康妮開了三個櫃台接受大家點餐，每個櫃台一次服務一位客人，海狸點完餐後，會繼續在該櫃台等待直到取完餐點。如果有兩個以上櫃台同時空出，海狸就依櫃台 ABC 的順序去櫃台點餐。下面這張圖把前面十位海狸各自想要買的甜點都標示出來了，最後一位是小薇。



讓寫出三個櫃台分別會服務到哪幾個編號的海狸？



## 正確答案

題組一答案：B. 櫃台

題組二答案：A 櫃台會服務到編號 1、4、6 的海狸

B 櫃台會服務到編號 2、8、10 的海狸

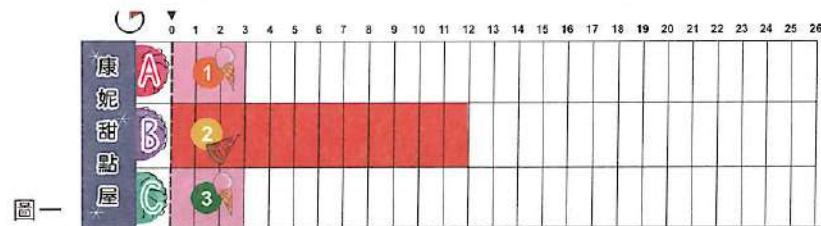
C 櫃台會服務到編號 3、5、7、9 的海狸

詳細圖解如圖六。

為了解釋這個問題，以下將會採用時間軸的圖解方式來協助說明，此外圖上也會標示目前正在處理的海狸編號以及目前已經經歷的時間過程。

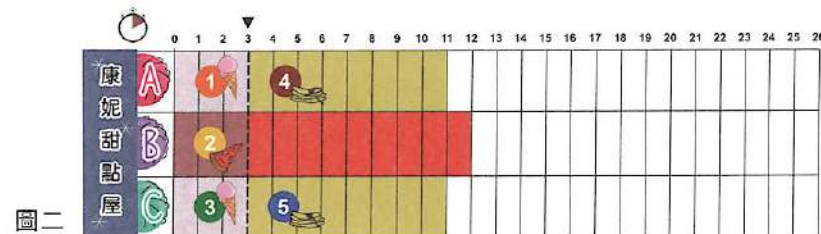
剛開始每一個櫃台都沒有處理餐點，因此最前面的三隻海狸可以馬上進行點餐：

- 1 號海狸在 A 櫃台點冰淇淋，因此排入 A 櫃台處理的時間表上 3 分鐘
- 2 號海狸在 B 櫃台點比薩，因此排至 B 櫃台處理的時間表上 12 分鐘
- 3 號海狸在 C 櫃台點冰淇淋，因此排至 C 櫃台處理的時間表上 3 分鐘



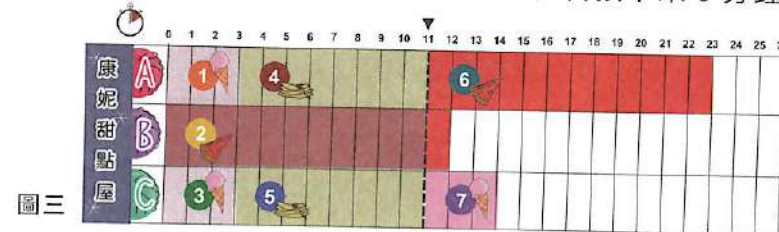
接下來就是要注意當某個櫃台正在準備餐點的時候，可以把這個櫃台當成不受理或是忙碌的狀態，此時這個櫃台就無法接受下一隻海狸的點餐，也就說接下來的 3 分鐘，A 櫃台和 C 櫃台都無法受理下一筆點餐，而 B 櫃台則有 12 分鐘的時間是無法受理的。3 分鐘過後，A 和 C 櫃台會結束冰淇淋的點餐作業，也就是完成 1 號海狸和 3 號海狸的部分，此時 A 櫃台和 C 櫃台就能再次受理點餐，因此接下來的兩隻海狸就能馬上到櫃台點餐：

- 4 號海狸到 A 櫃台點鬆餅，因此排入 A 櫃台接下來 8 分鐘的時間表
- 5 號海狸到 C 櫃台點鬆餅，因此排至 C 櫃台接下來 8 分鐘的時間表

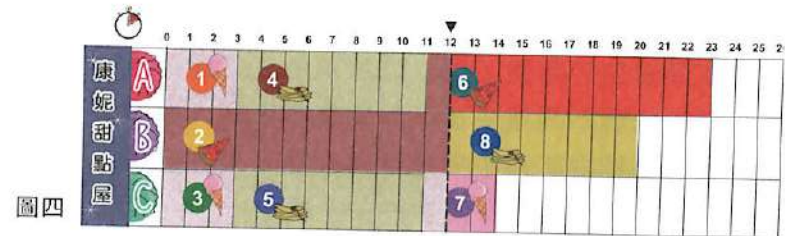


到第 11 分鐘時，A 櫃台與 C 櫃台再次完成點餐作業，因此 4 號及 5 號海狸離開，兩個櫃台再次恢復成受理點餐的狀態，因此接下來的兩隻海狸可以到櫃台點餐：

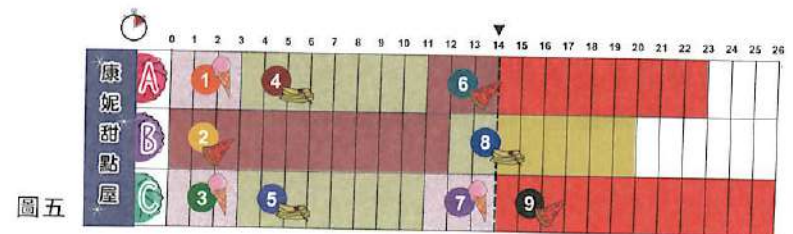
- 6 號海狸到 A 櫃台點比薩，排至 A 櫃台接下來 12 分鐘的時間表
- 7 號海狸到 C 櫃台點冰淇淋，排至 C 櫃台接下來 3 分鐘的時間表



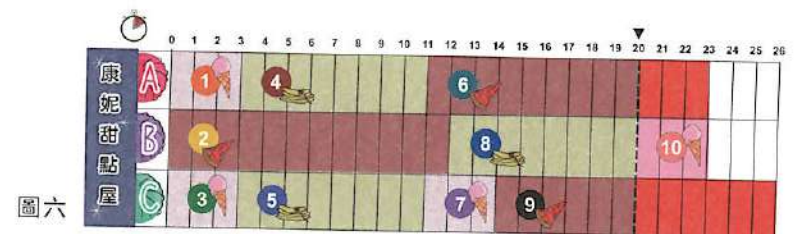
到第 12 分鐘時，B 櫃台終於完成最初的比薩點餐，因此重新恢復成受理點餐狀態，所以 8 號海狸可以到 B 櫃台點鬆餅，排上 B 櫃台接下來 8 分鐘的時間表。



到第 14 分鐘時 C 櫃台完成點餐 (7 號海狸) 因此 9 號海狸到 C 櫃台點比薩，排上 C 櫃台 12 分鐘。



最後在第 20 分鐘時 B 櫃台完成點餐 (8 號海狸) 恢復為受理狀態，因此 10 號海狸 (也就是小薇) 會到 B 櫃台進行點餐。





## 資訊科學上的意義

這個問題討論到作業系統的排程觀念。**排程 scheduling** 是指當多個程序在進行的時候，要如何有效分享一個或多個處理器，使他們盡可能不要閒置下來，而能夠儘快將這些程序做完，換句話說就是讓處理器盡可能越忙碌越好。

在這個問題中，每個櫃台就像是一個處理器，而海狸們的點餐就像是程序，櫃台的處理順序就像是排程演算法的參照表。作業系統的排程演算法讓中央處理器 CPU 可以多工處理，調配得當就能讓處理器在同一時間內盡可能的執行程序，並且降低之後程序的等待時間。

電腦通常都會有好幾個中央處理器或 **處理器核心**，它們可以進行基本運算（如加法、乘法），大部分是獨立運作的。在這個問題中，我們用三個櫃台（A、B 和 C）都是獨立處理訂單的概念，來比喻為電腦的中央處理器。

因此，當某個櫃台開始準備餐點的時候，可以理解為中央處理器正在執行運算，而這些運算因為程式給定的任務不同產生了不同的指令序列。通常數百個這樣的序列（這裡稱為程序）就必須排隊等待中央處理器空閒下來才能送給它執行，而這樣的行程調度或安排，便是由電腦的作業系統來完成。

為了解決這個任務，我們在這裡扮演的角色就是作業系統：也就是向每個櫃台下達指令準備餐點（也就是排程）。

類似的例子在我們的生活中很常見，舉例來說，在超市買東西排隊等待結帳，另一個例子是，當只有一間會議室，而有多個小組都想使用這間會議室的時候，我們如果希望找出最有效率的方式來使用會議室；或是在有限時間內，我們想要儘可能完成更多的會議，便是要為這些小組會議與會議室進行排程處理。



## 關鍵字

處理、排程、處理核心



## 35. 海狸歌唱比賽



海狸村舉辦海狸歌唱比賽，共有六位歌手進入最後決賽。決賽邀請四位評審老師，每位評審根據自己的標準評分，分數越高表示歌唱得越好。得分表如下：

| 歌手 | 評審  |     |     |     |
|----|-----|-----|-----|-----|
|    | 南老師 | 麥老師 | 瑞老師 | 詹老師 |
| 小安 | 80  | 8   | 60  | 0   |
| 小貝 | 90  | 10  | 80  | 50  |
| 小凱 | 85  | 7   | 90  | 100 |
| 丹丹 | 100 | 9   | 100 | 30  |
| 小依 | 95  | 6   | 70  | 10  |
| 小藍 | 75  | 5   | 50  | 20  |

為了找到冠軍，主辦單將每位評審給的分數轉換成名次，例如南老師心中第一名是丹丹，麥老師心中第一名是小貝；再將每位歌手得到的四個名次加總，總和最小者就是冠軍。

請問下列哪一位歌手是冠軍？

- A. 小安
- B. 小貝
- C. 小凱
- D. 丹丹
- E. 小依



## 正確答案是：D. 丹丹

首先我們必須對每位評審老師給的分數進行名次排序，會得到下表，得分最高的歌手為第 1 名、得分最低的歌手為第 6 名，接下來將每位歌手的名次相加，名次總和最小的丹丹就是冠軍。

| 評審<br>歌手 | 南老師 |    | 麥老師 |    | 瑞老師 |    | 詹老師 |    | 名次總和 |
|----------|-----|----|-----|----|-----|----|-----|----|------|
|          | 分數  | 名次 | 分數  | 名次 | 分數  | 名次 | 分數  | 名次 |      |
| 小安       | 80  | 5  | 8   | 3  | 60  | 5  | 0   | 6  | 19   |
| 小貝       | 90  | 3  | 10  | 1  | 80  | 3  | 50  | 2  | 9    |
| 小凱       | 85  | 4  | 7   | 4  | 90  | 2  | 100 | 1  | 11   |
| 丹丹       | 100 | 1  | 9   | 22 | 100 | 1  | 30  | 3  | 7    |
| 小依       | 95  | 2  | 6   | 5  | 70  | 4  | 10  | 5  | 16   |
| 小藍       | 75  | 6  | 5   | 6  | 50  | 6  | 20  | 4  | 22   |



## 關鍵字

正規化、公平評估、穩健性



## 資訊科學上的意義

在此任務中，因為每位評審的標準不一，若直接以總分或平均來決定名次，反而不是多數評審的首選，因此需要先將各個評審給的分數轉換成名次，讓評斷的單位相同，才能做後續比較及判斷。在資訊科學的領域中我們經常需要找出解決問題的最佳方案，代表我們必須先訂定一種共同評估標準以找出最佳解，這就是 **公平評估 fair evaluation**。

**正規化 normalization** 是資料庫設計的一系列原理和技術，其目的是為了降低資料的「重複性」與避免「更新異常」的情況發生。透過將資料正規化後再執行資料處理，可以讓演算法的執行結果符合預期，不容易出現錯誤。一旦輸入資料被正規化，數據就可以按照在此任務中完成的方式進行排序，而這種排序後的值在列表中都有其唯一的評級。

**穩健性 robustness** 是指一個電腦系統在執行過程中處理錯誤，以及演算法在遭遇輸入、運算等異常時繼續正常執行的能力。在資訊科學中，資料的正規化可運用在機器學習演算法中，機器學習通常會使用大量具有各種特徵的數據來訓練機器，若事先做好資料前處理，可以增快機器學習訓練的成效和速度。例如假設使用具有特徵的數據來訓練機器學習模型，以房子的面積、房間的數量、到學校的距離、房子的屋齡與到商店的距離等來預測房子的價格，每個特徵的單位如地區、年齡、距離不同，價格區間也會有較大差異。



## 36. 秘密訊息

大狸和小狸習慣將彼此的秘密訊息轉換為數個 0 或 1 組成的數字來傳遞，以防別人知道他們的秘密，但是這個方法被老狸識破了！

他們決定將訊息做一點轉換，第一步是將原來的數字兩兩一組轉換為對應的英文字母 A, B, C, D：

|    |    |    |    |
|----|----|----|----|
| 00 | 01 | 10 | 11 |
| A  | B  | C  | D  |

第二步則是將第一步產生的英文字母，再次兩兩一組轉換為下表中對應的文字，即可產生新的訊息：

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| AA | AB | AC | AD | BA | BB | BC | BD | CA | CB | CC | CD | DA | DB | DC | DD |
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A  | B  | C  | D  | E  | F  |



如果最後訊息是 C13，原始的數字應該是什麼？

- A. 110000010011
- B. 110010101111
- C. 101010111011
- D. 101011001001



## 正確答案是：A. 110000010011

從最終訊息 C13 往回推，根據題目中第二張表格：C 會變成 DA、1 變成 AB、3 變成 AD，因此訊息變成 DAABAD。

接著根據題目中第一張表格：D 變成 11、A 變成 00、B 變成 01，因此得到原始的數字是 110000010011。



## 資訊科學上的意義

密碼學 cryptology 主要研究資訊保密、資訊完整性的驗證、資訊發布後不可否認性等資訊安全問題。當訊息透過公開管道傳遞時，任何人都有機會像題中的老狸一樣竊聽訊息。為了保護訊息內容，通常會進行加密 encrypt，讓人即使攔截到訊息也無法讀懂訊息。此題中將訊息透過表格內對應字母進行替換，就是一種傳統且簡單的訊息加密方式。

20 世紀以後的現代密碼學則更加複雜，大致上可分為使用同一把鑰匙 key 進行加密和解密的對稱式加密 symmetric key encryption，以及使用公鑰 public key 加密並用私鑰 private key 解密的非對稱式加密 asymmetric encryption。

平時瀏覽的網頁網址最前端常見的 https，其實就是一種運用到加密技術的網路通訊協定；常聽到加密數位貨幣（比特幣、以太幣等）也是運用密碼學和區塊鏈技術的產物；勒索病毒也是運用非對稱式加密製成。



## 關鍵字

密碼學、編碼



## 37. 登山挑戰

四隻海狸出發去登山，他們每天只會吃掉一整份口糧或者完全不吃，但我們無法知道海狸在哪幾天吃口糧。只知道安安每八天吃掉四份口糧，阿貝每七天吃掉四份口糧，喳喳每四天吃掉二份口糧，阿寶每九天吃掉六份口糧。

| 天  | 1           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |  |
|----|-------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|--|
| 安安 | [4 packets] |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |  |
| 阿貝 | [4 packets] |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |  |
| 喳喳 | [2 packets] |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |  |
| 阿寶 | [6 packets] |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |  |

為期三週的登山之旅過後，他們四個最多共吃掉多少包口糧？（範圍 [16~99] 的整數。）



## 正確答案是：50

下表顯示了每一隻海狸吃掉口糧的最大可能數量，這個任務的解題關鍵在於是否能夠理解海狸可以盡可能地在他們固定的週期上就先把口糧吃掉，而不是每隔幾天才吃掉一份，比方說安安，前八天他吃了四份，接下來的八天也吃掉四份，雖然在登山結束前只剩五天，安安還是可以在結束前這五天內把接下來的四份口糧吃掉，因此這段時間，他可以吃掉 12 份口糧。

| 天  | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  | 20  | 21  |     |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 安安 | [糧] |     |     |     | [糧] |     |     |     | [糧] |     |     |     | [糧] |     |     |     |     |     |     |     |     |     |
| 阿貝 | [糧] |     |     |     | [糧] |     |     |     | [糧] |     |     |     | [糧] |     |     |     |     |     |     |     |     |     |
| 喳喳 | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] |
| 阿寶 | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] | [糧] |

依照上述的推導，阿貝可以吃掉 12 份，喳喳前 20 天可以穩定吃掉 10 份口糧，而最後一天它可以吃掉一份，因此最後他可以吃掉 11 份口糧；阿寶前 18 天可以吃掉 12 份口糧，最後三天則每一天都可以吃掉一份，因此這段時間，它可以吃掉 15 份口糧。

因此在 21 天內四隻海狸最多可以吃掉 50 份口糧。



## 資訊科學上的意義

在資訊科學中，這類型的問題被稱為 **限制滿足問題 Constraint Satisfaction Problems, CSP**。為了解決這類的問題，我們會選定一個具有最多條件限制的變數（稱為最大限制變數），然後再陸續找出其他限制的變數。而這個問題的答案就是必須滿足這些限制的變數。

所謂的限制指的是必須遵守的規則，而滿足指的是依照規則執行，在這個任務當中，提到每幾天吃到的口糧數量就是所提到的「限制」，而你必須確認在這些時間點內如何找出口糧的最大使用量，即為所謂的「滿足」。我們雖然使用圖表的排列方式來直接觀察解決，但限制滿足問題通常需要進行多次的猜想跟改變，這個過程稱之為 **回溯**。

限制滿足是商業、經濟、公共服務和醫療領域解決問題時常用的方法。我們雖然會嘗試用電腦程式來輔助我們解決問題，但電腦科學家也已經證明許多限制滿足問題，即使是使用程式也很難解決。著名的八皇后問題，圖著色問題，乃至於很常見的填字遊戲、數獨及其他一些邏輯益智遊戲，都與限制滿足問題有關連。



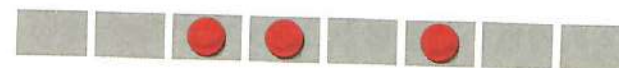
## 關鍵字

回溯、限制滿足問題















### 38. 籌碼遊戲

海狸喜歡玩一種自創的籌碼遊戲。








海狸會由左至右依序移動到下一個格子中，並根據海狸是否戴上帽子和格子中是否有籌碼而有不同的動作，以下圖示說明海狸對應的動作：

|   |   |   |
|---|---|---|
|   |    |    |
|    |    |    |
|   |    |    |
|  |   |   |
|   |  |  |



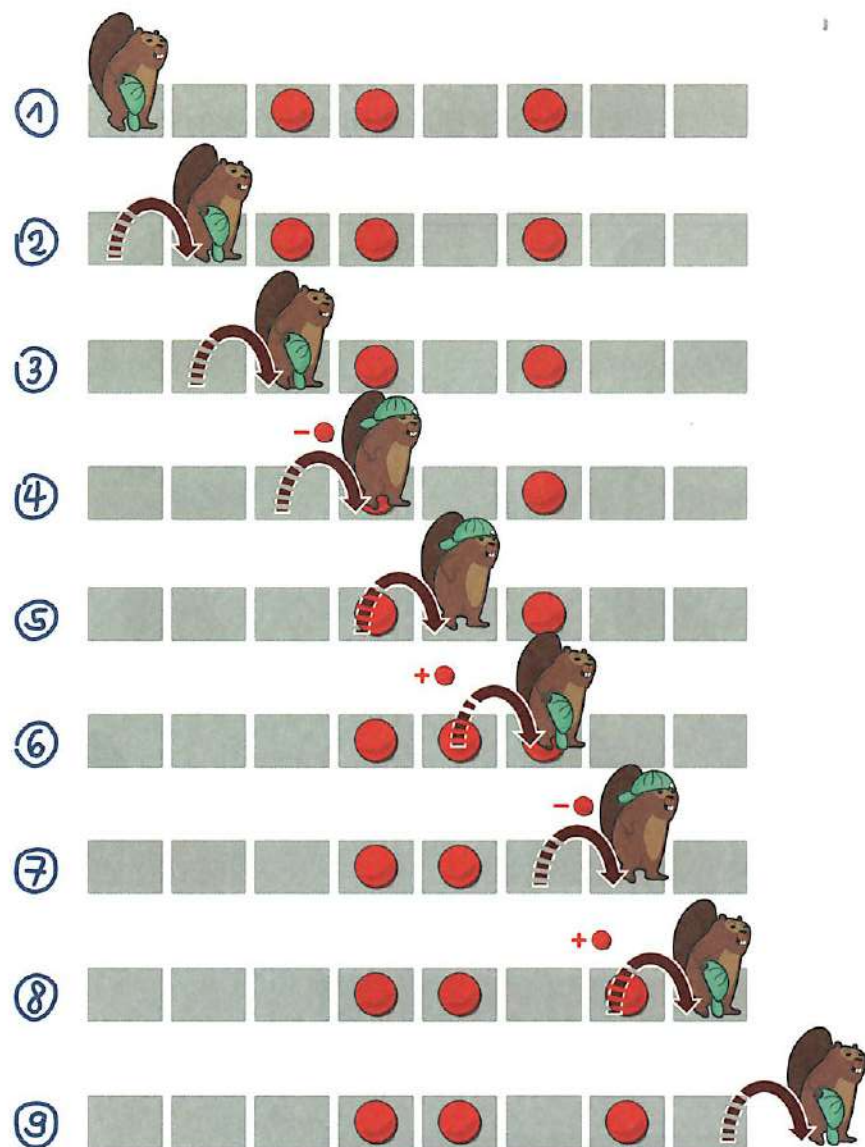
遊戲一開始，海狸並沒有戴上帽子，當海狸移動到最右邊時，哪些格子內會有籌碼呢？

- A. 
- B. 
- C. 
- D. 
- E. 



## 正確答案是：C

此題只要依照規則一步步執行，就可以得到答案，步驟可參考下圖：



## 資訊科學上的意義

**圖靈機 turing machine** 是英國數學家、資訊科學家艾倫·圖靈 Alan Turing 在 1936 年提出的抽象化運算模型，圖靈機可完成任何人類所能進行的計算，圖靈機由以下幾個部分組成：

- 紙帶：長度為無限長，分成一小格一小格，每一格可以印一個符號（題目中的格子）。
- 符號：由有限的字母組成（題目中的籌碼）。
- 讀寫頭：可以在紙帶上左右移動，讀取目前格子上的符號，並根據規則決定改寫其他符號或是變成空格（題目中的海狸）。
- 規則：定義機器如何根據狀態和讀寫頭讀取到的符號有不同的動作（題目中海狸共有戴上帽子和不戴上帽子兩種狀態，根據狀態和格子中有無籌碼而有不同的動作）。

目前我們所使用以二進制進行運算的電腦，都是採用馮·紐曼架構 Von Neumann architecture，而馮·紐曼正是以圖靈機為概念，進而設計出第一部通用型計算機。



## 關鍵字

圖靈機