



2020 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、蔡兆環、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



2020題目解析

目錄

挑戰賽介紹	3
題本說明	4
題組介紹	6
1. 圓木分類器	7
2. 積木塔	9
3. 圖書館的書	11
4. 灑水器	13
5. 跳躍的袋鼠	15
6. 尋訪泰迪熊	17
7. 參觀展覽	19
8. 八卦傳千里	21
9. 城堡裡的海狸	23
10. 疫情危機	27
11. 危險病毒	31
12. 蛇與梯	33
13. 電動車	37
14. 粉刷房屋	39
15. 城鎮與高速公路	43
16. 海狸 V.S. 袋鼠	45
17. 雜貨商店	49
18. 最重的標誌	51
19. 水瓶	53
20. 遵循規則	55
21. 智慧停車	59
22. 零錢包	63
23. 最後的贏家	67

挑戰賽介紹

24. 圖書編碼	71
25. 宴會密碼	73
26. 派對密碼	75
27. 樹木數獨	79
28. 樹獨數獨	81
29. 園中動物	83
30. 魔法藥水	87
31. 家用設備	91
32. 神奇飲料機	95
33. 密碼檢測	97
34. 戲劇演出	99
35. 星星與月亮	101
36. 物件分類	103
37. 不撞牆	105
38. 跳格子	107
39. 機器人的行走	109
40. 神奇數字機 I	112
41. 神奇數字機 II	113
42. 發光面板	117
43. 警衛雅克	119
44. 海狸美容院	121
45. 可以飛到哪？	123
46. 拿	125

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

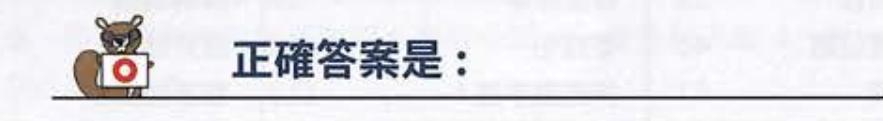
易	中	難			
海狸美容院	7	星星與月亮	15	樹木數獨	27
圓木分類器	9	蛇與梯	17	警衛雅克	29
戲劇演出	11	最重的標誌	21	灑水器	31
圖書編碼	13	圓中動物	23	可以飛到哪？	33

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

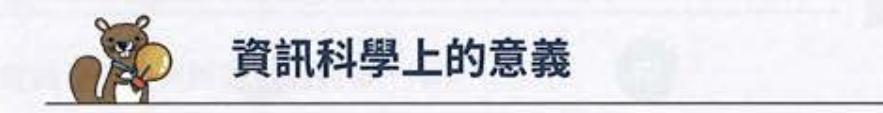


參考詳解



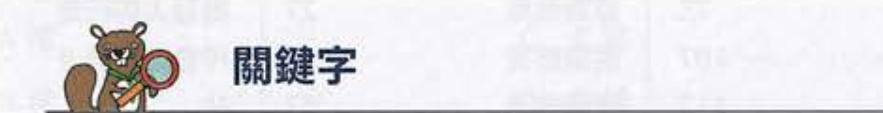
正確答案是：

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

2020 運算思維挑戰賽題組

Benjamin/ 易 Cadet/ 易 Junior/ 易 Senior/ 易

Benjamin 組

易	中	難			
圓木分類器	7	蛇與梯	33	灑水器	13
圖書編碼	71	最重的標誌	51	樹木數獨	79
戲劇演出	99	園中動物	83	警衛雅克	119
海狸美容院	121	星星與月亮	101	可以飛到哪？	123

Cadet 組

易	中	難			
尋訪泰迪熊	17	積木塔	9	海狸 V.S. 袋鼠	45
水瓶	53	圖書館的書	11	零錢包	63
魔法藥水	87	粉刷房屋	39	最後的贏家	67
星星與月亮	101	宴會密碼	73	家用設備	91
物件分類	103	樹木數獨	79	神奇數字機 I	112

Junior 組

易	中	難			
積木塔	9	電動車	37	八卦傳千里	21
城堡裡的海狸	23	智慧停車	59	疫情危機	27
城鎮與高速公路	43	零錢包	63	雜貨商店	49
最後的贏家	67	神奇數字機 I	112	樹獨數獨	81
神奇飲料機	95	發光面板	117	不撞牆	105

Senior 組

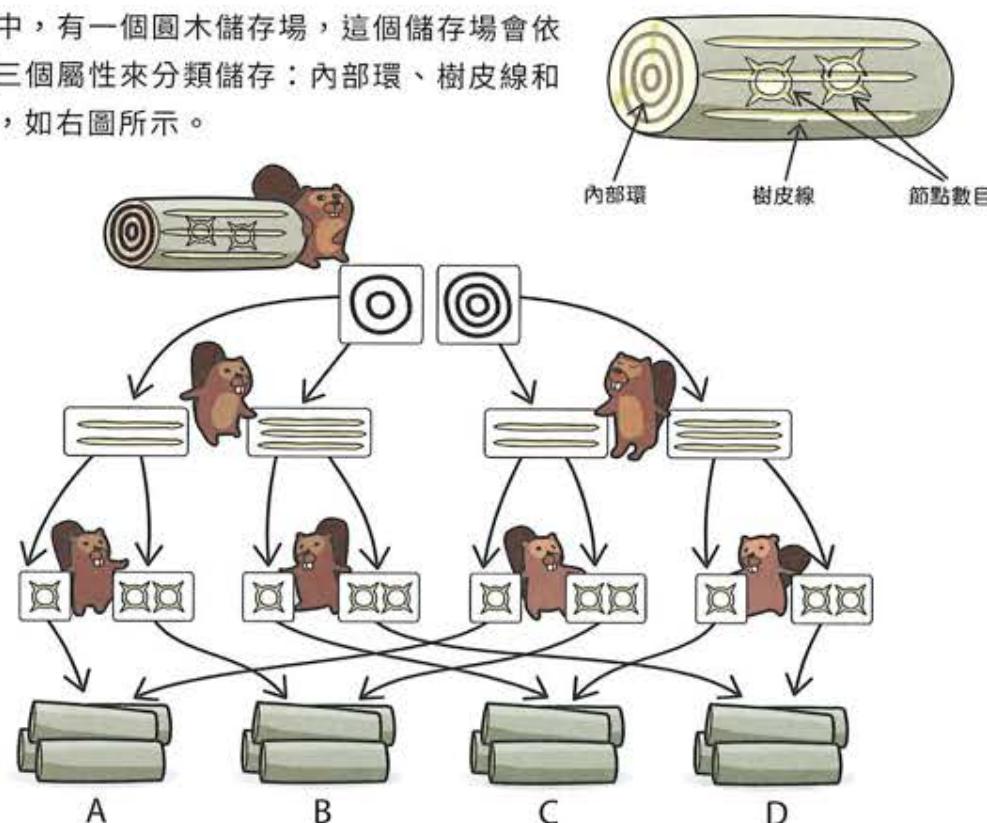
易	中	難			
危險病毒	31	跳躍的袋鼠	15	參觀展覽	19
電動車	37	八卦傳千里	21	遵循規則	55
派對密碼	75	疫情危機	27	機器人的行走	109
跳格子	107	樹獨數獨	81	神奇數字機 II	113
發光面板	117	密碼檢測	97	拈	125



Benjamin/ 易 Cadet/ 易 Junior/ 易 Senior/ 易

1. 圓木分類器

在海狸村中，有一個圓木儲存場，這個儲存場會依據圓木的三個屬性來分類儲存：內部環、樹皮線和節點數目，如右圖所示。



如上圖所示，你可以看出儲存場中如何把圓木分類：從上面第一層的海狸開始按照內部環、樹皮線和節點數目對圓木進行分類，一直分到下面 A、B、C 或 D 四堆圓木中的其中一堆。

上圖顯示的圓木將放置在 D 堆中，這是因為該圓木具有三個內部環（從第一層的海狸分給第二層的第二隻海狸）、三條樹皮線（從第二層的海狸分給第三層的第四隻海狸）和兩個節點（從第三層的海狸分到 D 堆）。

請問下圖顯示的圓木將被發送到哪一堆？



A. A 堆

C. C 堆

B. B 堆

D. D 堆



正確答案是 : C. C 堆

● 被分到 A 堆的圓木有以下兩種類型：

1. 有 2 個內部環、2 個樹皮線和 1 個節點
2. 有 3 個內部環、2 個樹皮線和 1 個節點

● 被分到 B 堆的圓木有以下兩種類型：

1. 有 2 個內部環、2 個樹皮線和 2 個節點
2. 有 3 個內部環、2 個樹皮線和 2 個節點

● 被分到 C 堆的圓木有以下兩種類型：

1. 有 2 個內部環、3 個樹皮線和 1 個節點
2. 有 3 個內部環、3 個樹皮線和 1 個節點

● 被分到 D 堆的圓木有以下兩種類型：

1. 有 2 個內部環、3 個樹皮線和 2 個節點
2. 有 3 個內部環、3 個樹皮線和 2 個節點

這就是為什麼答案是 C。



資訊科學上的意義

此任務的圖稱為一種 **決策樹**，這是資訊科學中用來對資料採用一連串規則檢查以進行決策的概念。決策是電腦經常需要運作的處理，每個規則檢查代表一種評估，再根據評估結果選擇向下適合的路徑。決策樹也經常用於根據資料特性對資料進行分類。

此任務類似於 **電腦網路** 封包的 **分類** 和轉發概念，這樣的處理會發生在例如網路交換器或路由器之類的網路設備中。這些設備會以不同的規則檢查網路封包，並決定如何轉發數據封包。



關鍵字

分類、決策樹、電腦網路

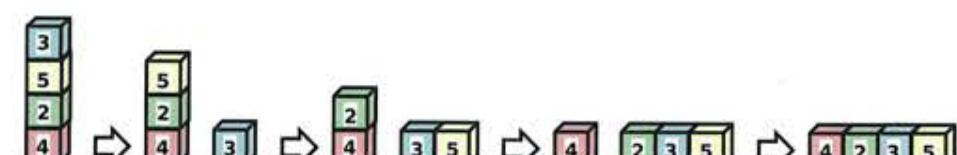
2. 積木塔

海狸小奧有一組積木，每個積木都有一個一位數的數字。

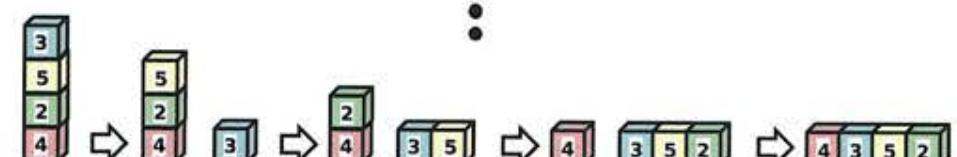
小奧喜歡將積木一個疊著一個形成積木塔，再依序將積木一個個從最上面取下排成積木列。每當小奧取下一個積木，就會把該積木放在已排好的積木列最左側或最右側。所有積木都排好後，積木列就會形成一串數字。



如下圖所示，4 個積木的積木塔 ，可能會形成：2534、4235、… 或 4352 等八種不同的數字。



⋮
⋮
⋮



如右圖，小奧現在用六個積木疊了個積木塔；請問這個積木塔可能形成的最小數字為何？

- A. 563547
- B. 347565
- C. 345567
- D. 354765





正確答案是：B.347565

3
6
5

要組合出最小的數字，應該將積木塔中最小的數字 (3) 排在最左邊。因此在積木 3 以下的兩個數字—6 和 5，應該依序排在已完成的數字的右邊，因此最小的數字應由 3 開頭，65 結尾。

7
4
5
3

積木 3 上面有一個由三個積木所組成的積木塔：547，我們必須將這三個數字安插在 3 和 65 之間。

利用相同的規則，最小的數字 (4) 應該被排在最左邊，積木 4 以下只有一個數字：5，因此先拿下積木 7，再將積木 4 放在最左邊，積木 5 放在最右邊，可以組合出最小的數字：475。

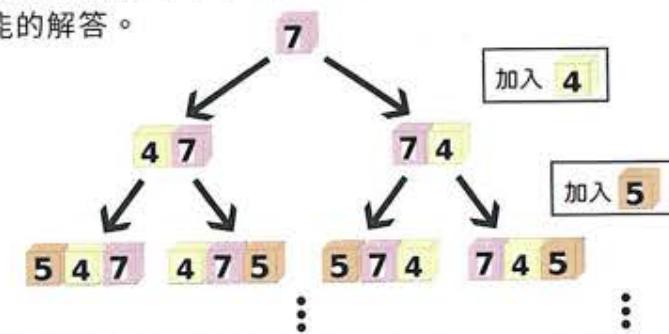
將 475 安插在 3 和 65 之間，可以得到最小數字組合：347565。



資訊科學上的意義

將積木依序疊起來成為一座高塔，而且只能從最上面依序將積木取出，在資訊科學領域中，這是一種常見的線性資料結構，稱為 **堆疊 stack**。堆疊的特性為只有一個開口，因此先進入堆疊的資料，會最後才能被取出，最後放入的資料，最先被取出，我們稱之具有後進先出 Last In First Out, LIFO 的特性。堆疊可以被使用於中序算式 infix notation 和後序算式 postfix notation 的轉換，讓程式更容易計算出算式的答案。

除此之外，在記錄如何得到數字排列的過程，我們利用 **樹** 的結構來表示。如下圖，將積木塔最上面的積木作為根，每增加一個積木，便可向下增加一層，同一層的節點表示以本題操作方式加入新積木後能得到的不同數字排列。最後，將所有積木都加入樹以後，便可以得到這個積木塔可以排列出的所有數字。利用樹，我們可以更有系統分析搜尋可能的解答。



關鍵字

堆疊、樹

3. 圖書館的書

海狸鎮圖書室只有一疊書，每當有海狸來借書，圖書館員就會寫下借書者的名字並把最上面的書借給他。如果有海狸來還書，圖書館員也會記錄還書者的名字並把他還的書放在最上面。

本周一圖書室的書這樣疊著：



本週一開始圖書室借還書的紀錄資料如下：



小凱借了哪一本書？

- A. 《夏綠蒂的網》
- B. 《好奇的喬治》
- C. 《閣樓上的光》
- D. 《哈比人》
- E. 小馬還的書



正確答案是：B. 《好奇的喬治》

小凱借書時會拿到最上面的書，往前回溯就是前一個人（小菲）還的書。小菲第二個借書的，第一個借書的小亞拿走了最上面的《夏綠蒂的網》，之後小菲就借走《好奇的喬治》。所以，小凱借的書也就是《好奇的喬治》。

下表表示是本週圖書室裡那疊書的狀態：

初始 狀態	借書者	小亞	小菲		小馬		小菲
	所借的書	《夏》	《好》		《夏》		《好》
	還書者			小亞		小菲	
	所還的書			《夏》		《好》	

《夏》	藏書的狀態	《好》	《夏》	《好》	《好》	《好》
《好》		《好》	《閣》	《閣》	《閣》	《閣》
《閣》		《閣》	《閣》	《閣》	《閣》	《閣》
《哈》		《哈》	《哈》	《哈》	《哈》	《哈》
《襪》		《襪》	《襪》	《襪》	《襪》	《襪》



資訊科學上的意義

題目中的一疊書就像資訊科學中的 **堆疊**。堆疊是一種抽象的資料結構，我們可以將之想像成只有上方一個開口的容器，容器中的物品一個疊在另一個之上；當我們要存放物品時，只能從上方開口將物品疊在最上方，要拿取物品時，也只能依序從上方開口將最上方的物品取出。就像在此題中，拿取最上面的那一本書。

在堆疊工作中有兩個動作：`push`(把資料放進容器「最上方」) 和 `pop`(把「最上方」的資料移除)。觀察前面所描述的堆疊處理資料的方式，可以發現一個特性：**後進先出 (LIFO)**，簡而言之就是最後加入的那筆資料，接下來會最早被移除。在此題的借還書情境中，最後還回來的書，接下來會最先被借出。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 (\leftarrow) 時，最先看到的頁面就是之前最後被瀏覽過的網頁。

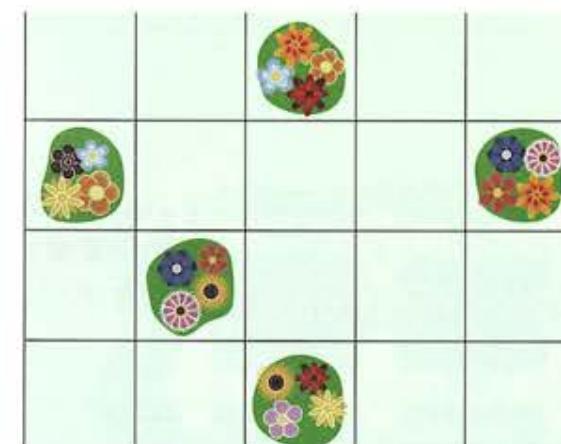


關鍵字

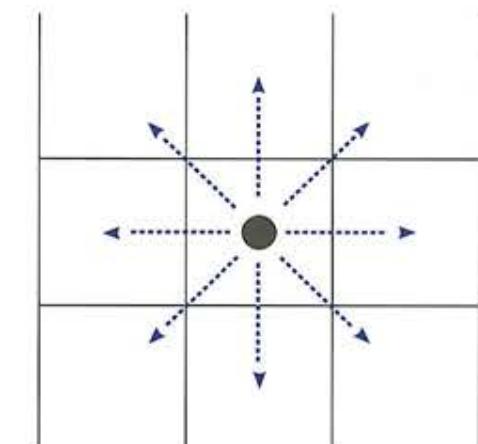
堆疊、後進先出 (LIFO)

4. 灑水器

海狸**小伯**把他的花園切成方格型並種植了幾叢花圃（見下圖一）。為了確保每個花圃都能澆到水，且重複澆到水也沒關係，**小伯**決定在他花園中的方格型空位架設幾台灑水器，每個灑水器能澆到其相鄰的八個方格，如圖二所示。



圖一



圖二

請問**小伯**最少需要架設幾台灑水器才能讓每個花圃都澆得到水？

A. 1

B. 2

C. 3

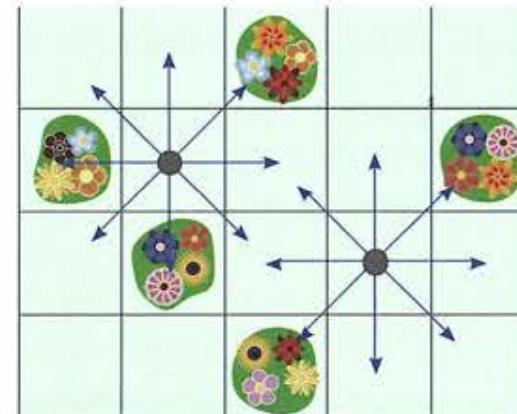
D. 4



正確答案是：B. 架設 2 台灑水器

只用一台灑水器來澆到所有的花圃是不可能的：舉例來說，第二列的兩個花圃距離 4 個方格，因此無法用一台灑水器同時澆到這兩個花圃。

而右圖兩個灑水器的安置位置則可讓所有的花圃都被澆到水。



資訊科學上的意義

在資訊科學的領域，我們經常需要找出 **問題的最佳解決方案**，這代表我們必須先訂定一種評估標準，用來比較各個方案的好壞以找出最佳解。在本題中，要解決的問題是「灑水器要能澆到所有花圃」，評估標準是「使用的灑水器數量越少越好」。

要澆到所有花圃的最簡單解法就是在所有的空格都設置灑水器，或在每個花圃旁都配置一個灑水器，但這樣不僅耗費昂貴，還危害環境。而且，澆太多水反而對一些花不好（這又是另一個更複雜的限制條件）。

本題較為簡單，只要持續嘗試放置位置就能找到最佳解。由於問題解的可能組合較少，就算嘗試所有灑水器的部署方式也是可以在短時間列出來。但隨著花園空間加大、花圃數量增加、若再加上其他限制條件，會使找到最佳解的任務因為組合太龐大而變得更加困難。

演算法中的解題策略，可用來幫助解決複雜問題中找到最佳解或接近最佳解。在現實生活中，使用和本題類似概念來解決問題的實際範例，包括規畫分配一個地區的醫療機構、消防局和衛生人員（數量和位置）等資源。



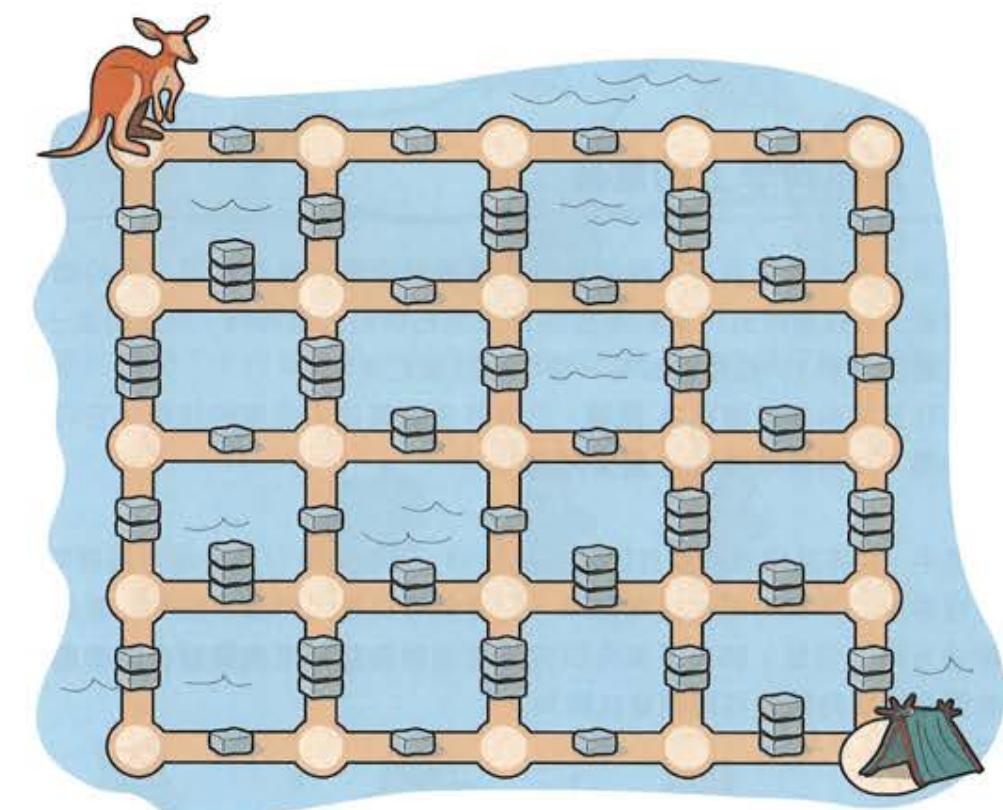
關鍵字

最佳化

5. 跳躍的袋鼠

有隻袋鼠想回家，但她只能沿路徑垂直（上 - 下）或水平（左 - 右）跳躍，而且她的跳躍高度最多只有兩個磚塊。

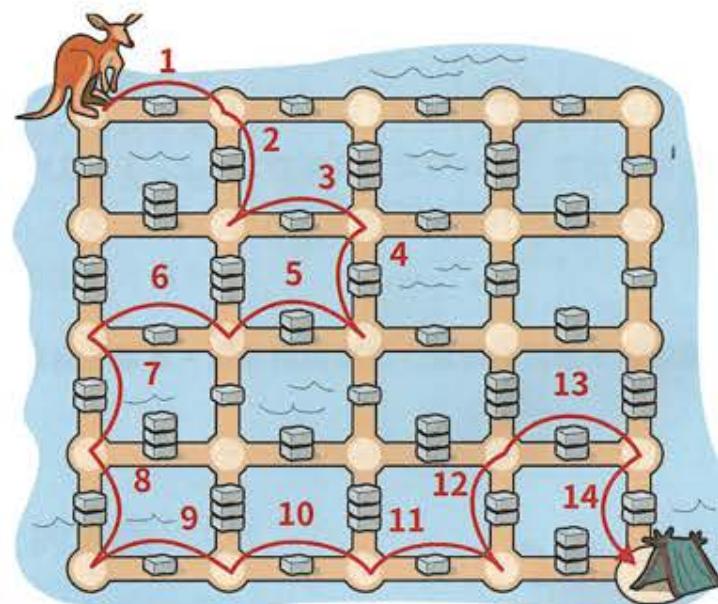
如下圖所示，請問袋鼠最少需要跳躍幾次，才能回到家？（範圍 [0, 40] 的整數）





正確答案是：14 次

解決方案如右圖所示



資訊科學上的意義

要找到解決方案，可依照以下步驟進行：搜索路線時，先選定可以走的路徑一步一步往前走，若往前所有可能的路徑都有三塊石頭陷入僵局時，則返回上一步（也可能不只返回一步），然後嘗試另一種可能往前的走法。

這種方法在資訊科學中被稱為 **回溯**，它是許多演算法中使用的技術。它可以用於解決如數獨、路徑搜尋或組合 **最佳化** 問題。

在此任務中，從終點袋鼠的家往回找尋路徑時，因為大多只有一個可選擇的方向，因此在找尋解決方案時顯得更有效率。這樣做可以讓我們減少回溯，並且更容易找到解決方案。但是，如果不事先研究這個問題的所有可能路徑，就未必是從起點開始還是從終點開始找路徑會比較快。

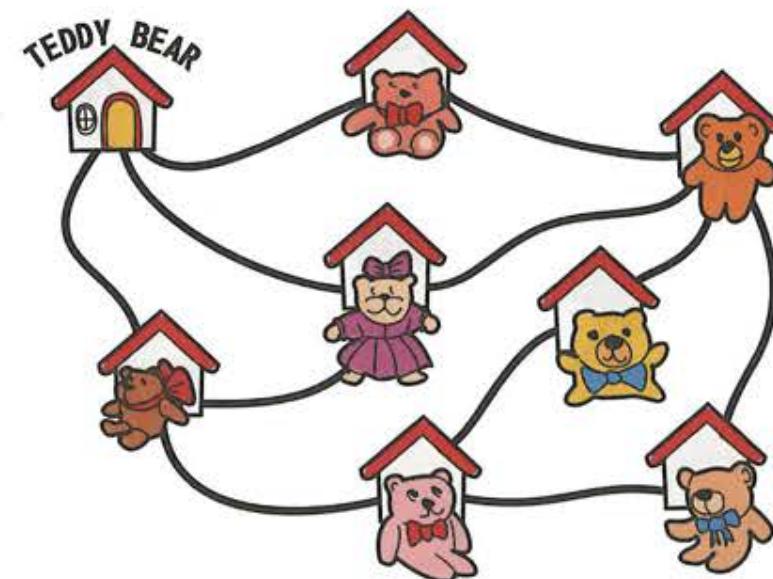


關鍵字

回溯、條件、最佳化

6. 尋訪泰迪熊

海狸一家想拜訪泰迪熊們。牠們從自己的家（名為 TEDDY BEAR）出發，挑了一條路徑，沿路尋訪路上的每個泰迪熊家並留影，最終回到自己的家中。下圖為牠們的城鎮地圖。



海狸一家共拜訪了 4 位泰迪熊，但回家檢查留影照片時發現只拍到下面 3 位泰迪熊。



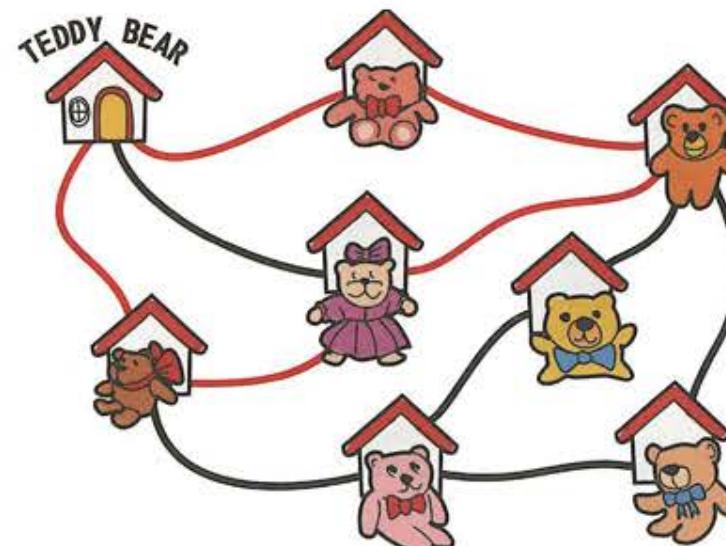
請問，是哪位泰迪熊被忘了拍呢？

- A. A pink teddy bear with a bow tie.
- B. A yellow teddy bear with a blue bow tie.
- C. A red teddy bear with a red bow tie.
- D. A brown teddy bear with a blue bow tie.



正確答案是：C.

下圖中以紅色標示的路徑，是唯一可以拜訪 4 位泰迪熊後再返回家中的路線。



資訊科學上的意義

路徑搜尋是資訊科學中的常見問題，實際生活上使用的導航系統也是路徑搜尋的一種應用。為了解決這個問題，你必須走訪路徑且找出符合條件（在此題中有四個固定拜訪點）的路線。儘管這個任務非常簡單，但隨著地圖擴展，這個問題會變得更難解決。

圖 **graph** 是一種平面圖的表示形式（是由節點—房屋、邊—房屋之間的路組合而成）。在此題中，我們已選擇四個節點（海狸的家和三個已知的泰迪熊家），而任務是必須找出已知由四個節點所構出的連接路線中的另一節點。

一個起點跟終點相同的路徑稱為 **循環路徑 cycle**。就像是搜尋一個從家到學校、然後到游泳池、再到商店，最後回家的路線圖。如果有多個路徑可走，進階的問題是如何能節省成本達到最佳化，例如尋找最短路徑。



關鍵字

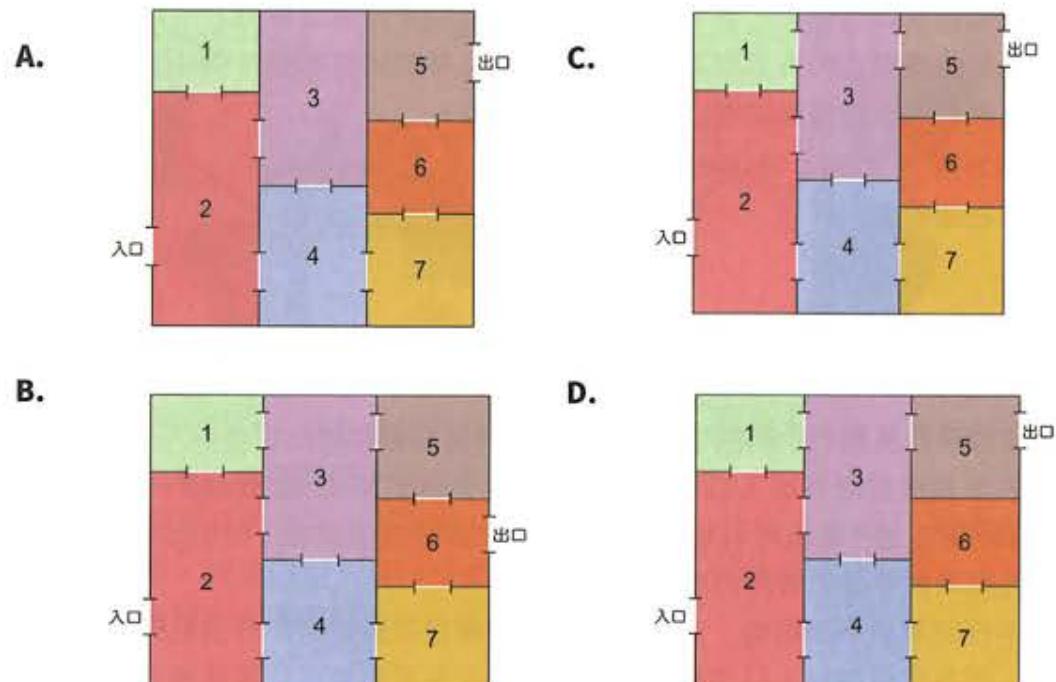
圖、循環路徑

7. 參觀展覽

有一間新建的博物館，館內有編號為 1~7 的七個展覽房間。

博物館希望參觀者能夠不需走回頭路就參觀完所有展覽房間，以提供參觀者良好的參觀體驗，為了決定每個展覽房間的出入口配置，他們提出了四種可能的規劃方案（如下圖）。

請問下列哪一種規劃方案，可以讓參觀者進入博物館後，每個展覽房間都只需要進入一次，就能從出口離開？





正確答案是：C

只有規劃方案 C 能夠不走回頭路就參觀完所有展覽房間。參觀順序依序為 $2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 5$ 。如果有個房間只有一個出入口，就不可能不走回頭路。因為參觀者從某個房間進入這個房間後，將不得不從同個出入口回到原來房間，也就是經過原來的房間兩次。

在規劃方案 A 中，編號 1 的房間只有一個出入口。

在規劃方案 B 中，博物館出口所在的編號 6 的房間，只可能經過編號 5 或編號 7 的房間到達。如果選擇經過編號 5 以進入編號 6 的房間，那麼勢必需要繼續參觀未參觀過的編號 7 的房間後，再次回到編號 6 的房間，才能到達博物館出口；反之亦然，也就是會經過編號 6 的房間兩次。

在規劃方案 D 中，編號 6 的房間只有一個出入口。

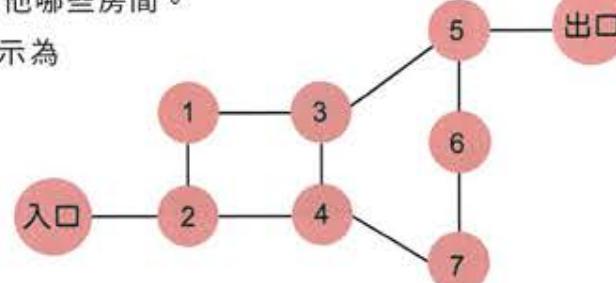


資訊科學上的意義

在利用電腦解決問題前，我們必須先將問題以適合電腦理解的方法表示。為了找到單趟就能參觀完所有房間的路線，重要的是需要知道哪些房間彼此有出入口相連，也就是可以從該房間進出到其他哪些房間。

以規劃方案 C 為例，可以抽象表示為

一個如右圖的圖：



其中圓圈表示房間 (加上進入博物館入口前與離開博物館出口後的「外部房間」)，線條表示兩個房間有出入口相連。這種表示法我們稱為 **圖 graph**，圓圈稱為 **節點**，線條稱為 **邊**。電腦可以利用圖的表示有效解決許多問題，例如找到最短路徑、確定社交平台中使用者間的朋友關係等。

在圖中找到單趟參觀路線，其實就是找到一條沿著邊走訪所有頂點的路線。在這題中，因為房間和出入口的數量很少，即使沒有電腦也可以順利解決；但是如果房間和出入口的數量變多，那麼即使有電腦也可能因為路徑過多而無法有效率地找到結果。

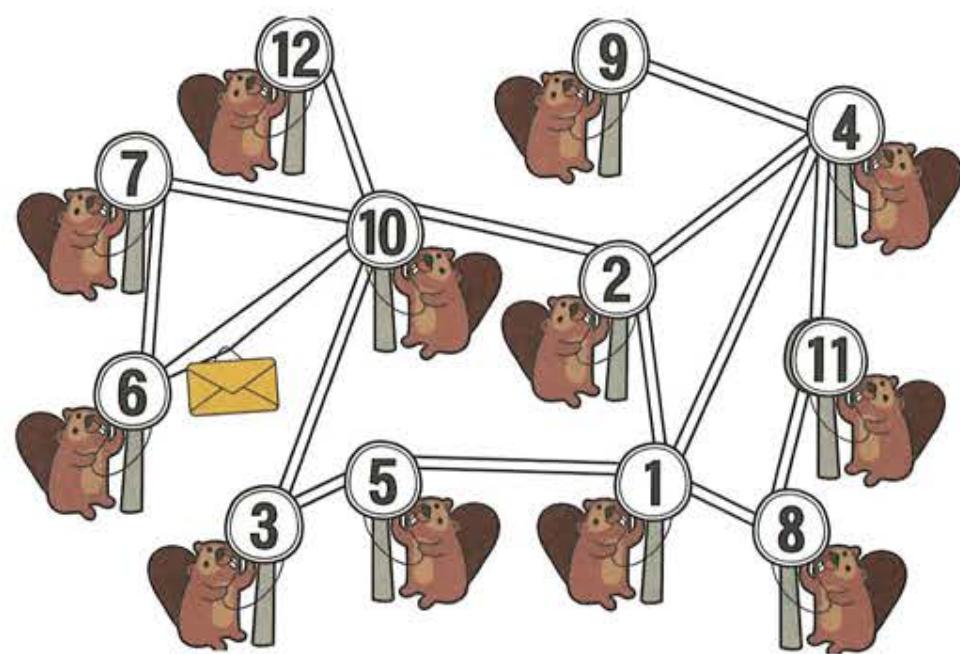


關鍵字

圖、節點、邊

8. 八卦傳千里

一個住著 12 隻海狸的大型部落，部落中的每隻海狸都有自己的傳聲孔，如下圖所示，每個傳聲孔都與海狸用來傳遞訊息的線路相連。



海狸們最喜歡八卦了，總是希望越快聽到八卦越好，每當有海狸耳聞最新八卦，他會馬上通知鄰近有傳聲管道相連的海狸。

舉例來說，若 8 號傳聲孔的海狸聽到最新八卦，他會通知 1 號和 11 號傳聲孔的海狸。接下來，聽到這個八卦的海狸們就會是 2 號、4 號和 5 號傳聲孔的海狸，以此類推，直到所有海狸都知道最新八卦。

如果想讓所有海狸盡快知道最新八卦，你應該先將八卦告訴哪隻海狸？(範圍 [1~12] 的一個整數，表示編號 [1~12] 傳聲孔的海狸。)



正確答案是：2號傳聲孔的海狸

如果把八卦告訴 2 號傳聲孔的海狸，只要 2 個步驟就能通知所有海狸。

步驟一，讓 2 號傳聲孔的海狸通知鄰近傳聲孔的海狸 (4 號、1 號和 10 號)。

步驟二，讓 4 號、1 號和 10 號海狸通知鄰近的其他海狸 (也就是剩下全部的海狸)。

因為所有傳聲孔與 2 號傳聲孔的最大距離值為 2，而且沒有其他傳聲孔的最大距離小於或等於 2，故告訴 2 號傳聲孔的海狸是最佳解。



資訊科學上的意義

許多現實生活中的問題以 圖 graph 表示後，就能有系統的透過電腦程式解決。圖由一組節點 (通常以點表示) 和一組連接這些節點的邊 (通常以線段表示) 組成。在此問題中，潛在的目標就是找到圖的中心。

圖中每個節點 u 的偏心率是指 u 和其他節點距離的最大值。圖的中心 graph center 就是由圖中最小偏心率的節點所形成的集合。此題的圖只存在一個這樣的節點，也就是 2 號傳聲孔的海狸。

尋找圖的中心常用於決定設施位置的問題，設置設施時希望盡可能降低到達設施的最遠距離。例如，將醫院放置在中心點可減短救護車必須行駛的最長距離。

如果像本題一樣僅有一個小圖需要評估，則可以一一列舉算出每個點到其他點的最短距離。對於較大的問題，則可以使用 佛洛依德演算法 Floyd-Warshall algorithm 等更複雜的演算法，有系統計算出兩兩節點間的最短距離，進而找到圖的中心。



關鍵字

圖、圖的中心、圖中點的距離、佛洛依德演算

9. 城堡裡的海狸

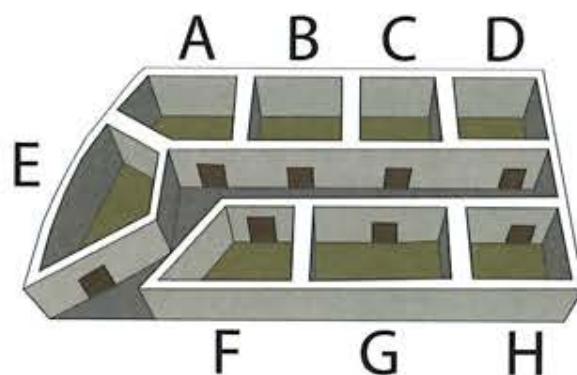
聰明的海狸需要一棵樹 才能在河上建一個水壩，但他只有一根胡蘿蔔 。

今天，是格朗松城堡的交易日，可以在城堡的房間內進行交易。

海狸帶著他的胡蘿蔔 去那裡，希望換成樹 。

在城堡內的每個房間都各有兩種物品交換的方式，如下表所示：

- A 房間： → 或 →
- B 房間： → 或 →
- C 房間： → 或 →
- D 房間： → 或 →
- E 房間： → 或 →
- F 房間： → 或 →
- G 房間： → 或 →
- H 房間： → 或 →



請問海狸必須依序進入哪幾個房間，才可以確保他最終會換得一棵樹 ?

- A. DGE B. GGE C. AGE D. DBC

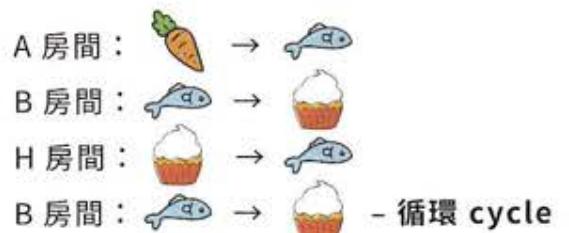


正確答案是：A. DGE

在 D 房間，海狸用他的胡蘿蔔換成蛋捲冰淇淋。之後，他移到 G 房間，將蛋捲冰淇淋交換成戒指。最後，海狸去 E 房間，把戒指交換成樹。

在尋找可能完成任務的房間順序時，可以遵循以下兩種不同的策略。

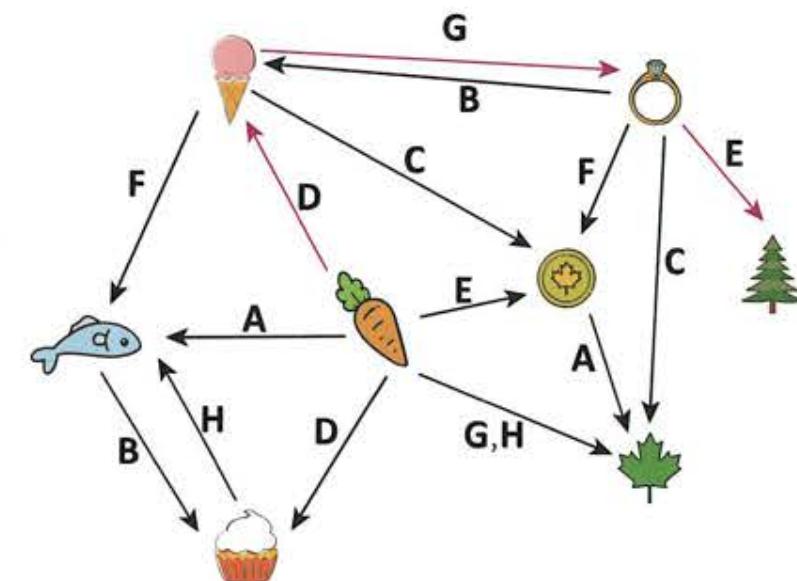
- 第一種策略比較冗長，由海狸擁有的胡蘿蔔開始，胡蘿蔔可以去五個房間 (A, D, E, G 和 H) 交換成 6 種不同物品。在思考過所有的可能性後，將會發現有可能因為未提供停止條件而造成 **循環 cycle** 的狀況，例如：



- 第二種策略是則比較快。顛倒過來思考，將海狸唯一可以換得樹的 E 房間設定為起點。而這棵樹是由戒指交換而來的，但戒指僅能經由 G 房間以蛋捲冰淇淋換得。故在最後步驟中，我們必須確認如何交換，才能獲得蛋捲冰淇淋。為了獲得蛋捲冰淇淋，只能在 B 房間以戒指中進行交換，或者在 D 房間中以胡蘿蔔來進行交換。由於海狸只有一根胡蘿蔔，所以應該由 D 房間開始，於是我們找到從胡蘿蔔換得樹的路徑。

標示各種路徑的表格內容，可用圖 graph 結構來表示。它的每個節點代表要交換的物品，而每個節點間連接線標示的英文字母代表進行交換的房間；以視覺化的圖形來表現出各房間之間的序列的狀態，更容易追蹤胡蘿蔔和樹之間的關係。

經由下圖所示，我們可以輕鬆地驗證出 DGE 是正確的答案。



資訊科學上的意義

這種已知狀態關係的問題，可以使用 **圖 graph** 表示。有向圖中將東西視為一種狀態，圖有向邊表示經過某個動作（去某個房間）會變成另一個狀態，用以展示一個有限狀態機中狀態轉換關係。

這裡使用圖型的 **深度優先搜尋演算法 Depth-First Search**，DFS 是一種用於遍歷或搜尋樹或圖的演算法。這個演算法會儘可能深入往前搜尋位走過的節點。當某個節點 v 的所在邊都已被探尋過，搜尋將回溯到發現節點 v 的那條邊之起始節點。這個過程一直進行到發現已遍歷起點可到達的所有節點為止。這樣就能找出所有由起點有路徑能到達的節點。

此題的演算法是從胡蘿蔔為起點，並沿每個節點（代表一項物品）盡可能地探索，直到找到樹。



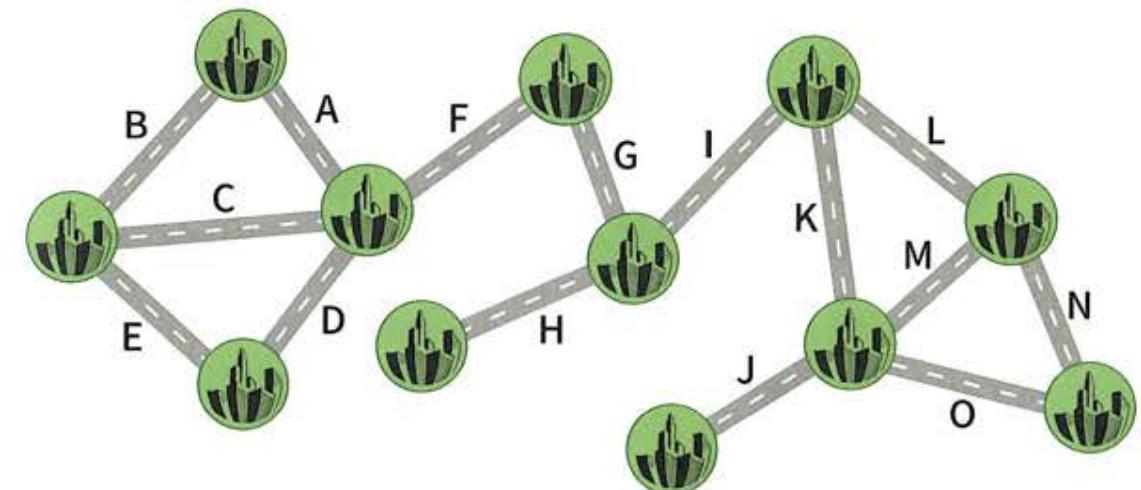
關鍵字

圖、深度優先搜尋



10. 疫情危機

海狸國有 12 個城市，城市間藉由道路（道路標示為 A 到 O）相連，如下圖所示：



兩個城市只要能透過道路直接或間接相連，我們就稱它們位在同個經濟共同體。由上圖可以看出目前 12 個城市都在同個共同體內。

不幸的是，由於疫情爆發，市長們決定要使用路障停止兩條道路通行，以減少城市交流。他們的目標是將海狸國分為三個獨立的經濟共同體，但同時希望分割後共同體內經濟干擾能降到最低，換句話說，目標是讓分割後規模最小的經濟共同體內城市越多越好。

請問需要停止哪兩條道路的通行？（答案由 2 個大寫字母組成）

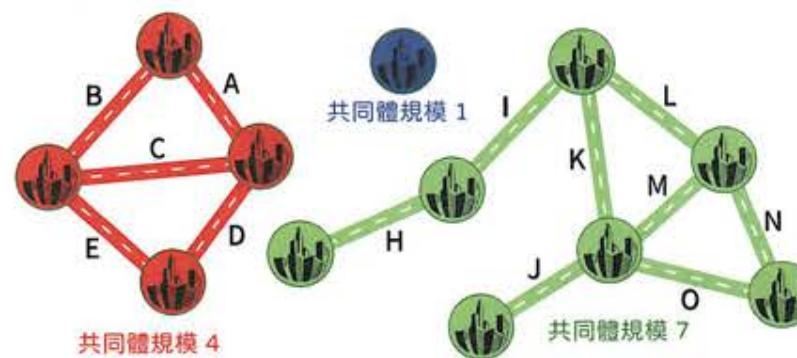


正確答案是：FI

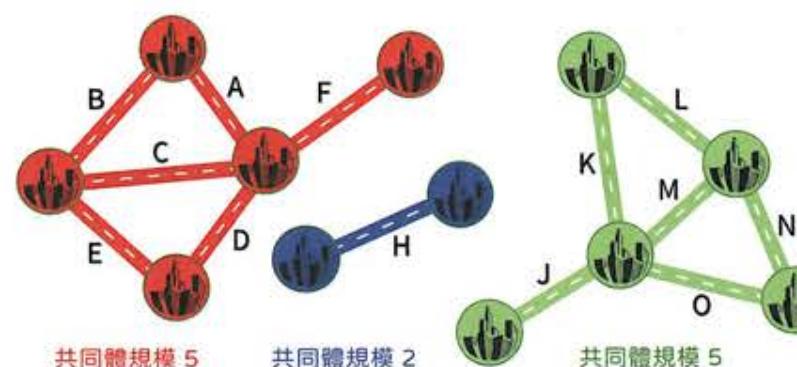
在本題中，有一些道路就算停止也不會造成分割，可以忽略不考慮。例如道路 C 就算停止，原本的人仍可以藉由道路 A 和 B 通行。仔細觀察會發現，只有停止 F、G、H、I 和 J 五條道路之一，才能夠真正造成分割。當然，如果同時停止 A 和 B 兩條道路，也可以真正造成分割；但這同時也會用完兩個路障，導致最後只得到兩個而不是三個共同體。

接著嘗試所有 F、G、H、I 和 J 五條道路取兩條的組合，並檢查分割出的共同體中規模最小者。

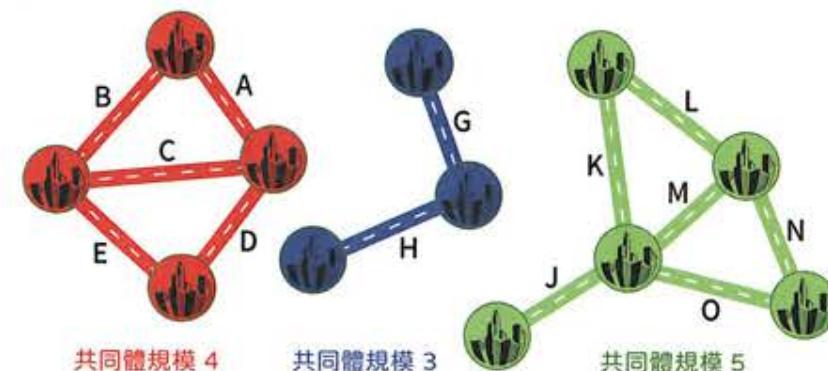
如果停止道路 F 和 G，分割出的共同體規模分別為 4、1 和 7，規模最小者大小為 1（如下圖）。



同理，停止道路 FH、FJ、GH、GJ、HI、HJ 和 IJ，也會導致規模最小者大小為 1。如果停止道路 G 和 I，分割出的共同體規模分別為 5、2 和 5，規模最小者大小為 2（如下圖）。



最後只剩下停止道路 F 和 I 的情況未考慮，在此情況中，分割出的共同體規模分別為 4、3 和 5，規模最小者大小為 3（如下圖）。



在本題中，我們希望分割出的三個共同體中規模最小者包含越多城市越好，因此最後答案為道路 F 和 I。



資訊科學上的意義

本題的任務是在無向圖 undirected graph 中尋找連通塊 connected component，也就是其內任兩點都可藉由至少一條路徑相連的子圖形。要找出連通塊，可以透過 圖形走訪 graph traversal 達成：以某點為起點，有系統地找與其相連的點，以此類推，直到無法找到新的點為止。在本題中也使用到 橋 bridge 的概念，橋代表的是移除後會造成連通塊數量增加的邊。

連通塊分析 connected component analysis 常用於電腦視覺應用，以標記感興趣區域。例如：光學字元辨識中的字符分割；視覺目標追蹤中的前景與背景分離處理以及醫學影像處理中的物件偵測等。



關鍵字

連通塊、圖形走訪、橋



11. 危險病毒

一個由 6 個島嶼 Alpha、Gamma、Beta、Delta、Eta 及 Kappa 所組成的國家，有一種危險的病毒正在 Kappa 島上散播，島民急需醫療補給，但這些醫療用品僅在首都 Alpha 島上生產。

為了防止病毒散播，政府限制所有的運輸路線如下圖所示。每一條路線上的數字是單日運輸物品的載重上限 (公斤) 。

每條運輸路線每天僅能使用一次，但對於各運輸路線的使用順序則沒有限制。例如：從 Beta 島出發，可以運輸 2 公斤的醫療用品至 Gamma 島，3 公斤至 Delta 島以及 5 公斤至 Eta 島。



Kappa 島每日可獲得的最大醫療物品量是多少 (單位 : 公斤) ?

- A. 20
- C. 15
- B. 18
- D. 12



正確答案是：B. 18

從 Alpha 島，可以運輸 10 公斤的藥品到 Gamma 島，另外可運輸 10 公斤的藥品到 Beta 島。從 Gamma 島可運輸 10 公斤的藥品到 Delta 島。

從 Beta 島可運輸有 3 公斤的藥品到 Delta 島，另外運輸 5 公斤的藥品到 Eta 島。

目前 Delta 島上的藥品總重量為 13 公斤，從 Delta 島只能運送 10 公斤的藥品到 Kappa 島，剩下的 3 公斤藥品則運往 Eta 島。

現在 Eta 島上有 8 公斤的藥品將送往 Kappa 島。

Kappa 島獲得的藥品總重量為 18 公斤。



資訊科學上的意義

圖 **graph** 是由節點 **node** 和邊 **edge** 所組成的一種資料結構。圖可用於表示許多現實生活中的問題，城市以節點表示、運輸路線以邊表示，邊上的數值表示網路流量。尋找城市之間的每日最大乘載量就是經典的最大流問題。

現實生活中經常有 **最大流** 的問題，例如捷運路網中，不同線路使用的車廂大小可能不同、掛載的車廂數量也可能不同，導致每條路線的載運量不同，要從 A 地運送大量旅客到 B 地時，人潮的消化速度就取決於路網中的最大流。



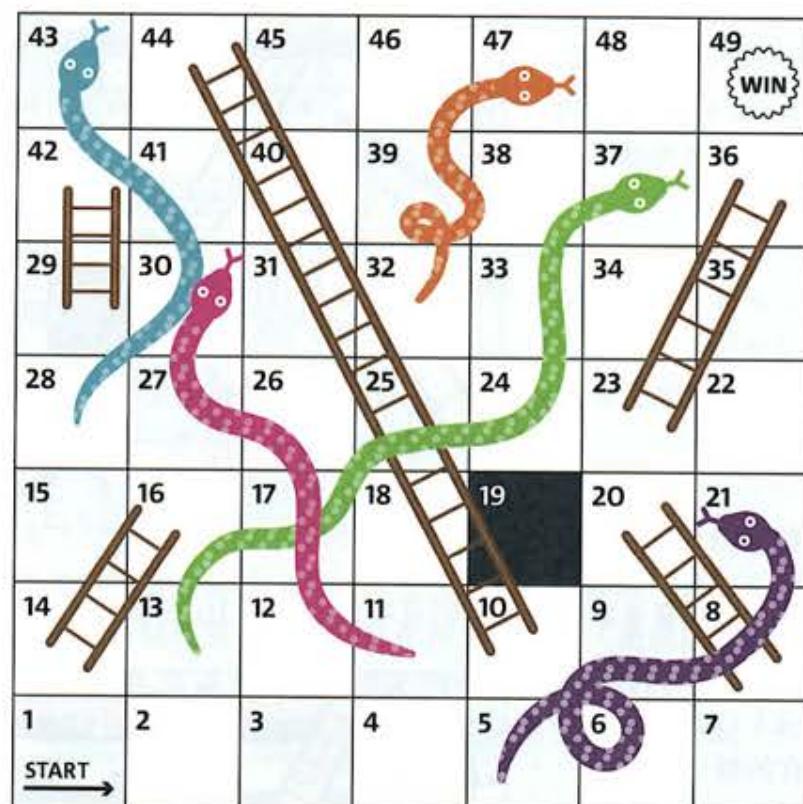
關鍵字

圖、最大流問題

12. 蛇與梯

蛇與梯是依據六面骰的點數（數值為 1 至 6 的整數）來移動的一款桌遊。

每一位玩家從格子 1 為起點，依據骰子的點數從所在位置依格子編號移動，最先抵達格子 49 為贏家。



但是若抵達占有蛇頭的格子，則必須滑至蛇尾所在處；以上圖為例，若玩家抵達格子 21，則滑至格子 5。

相反的，若抵達梯子的底部，則可往上爬到梯子頂端；以上圖為例，若玩家抵達格子 23，則爬到格子 36。

假若玩家位於格子 19，最少需再擲幾次骰子就可抵達格子 49？

- A. 2 次
- B. 3 次
- C. 4 次
- D. 5 次

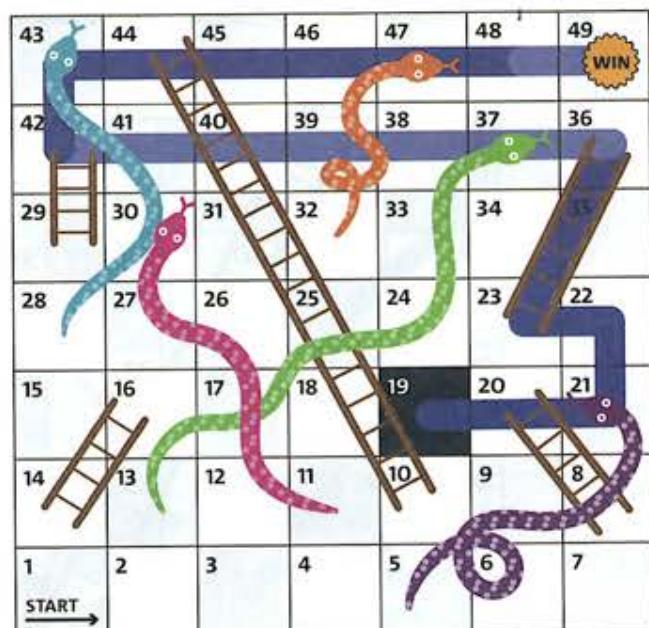


正確答案是：B. 再擲 3 次

若只考慮往前的最短路程，至少需投擲 4 次的骰子。

投擲 1 次，從格子 19 至格子 23，可以往上爬至格子 36。

但從格子 36 距離終點，還有 13 格的距離，因此至少需要投擲 3 次骰子 (6, 6, 1) 才能抵達終點，因此總共是投擲 4 次骰子。

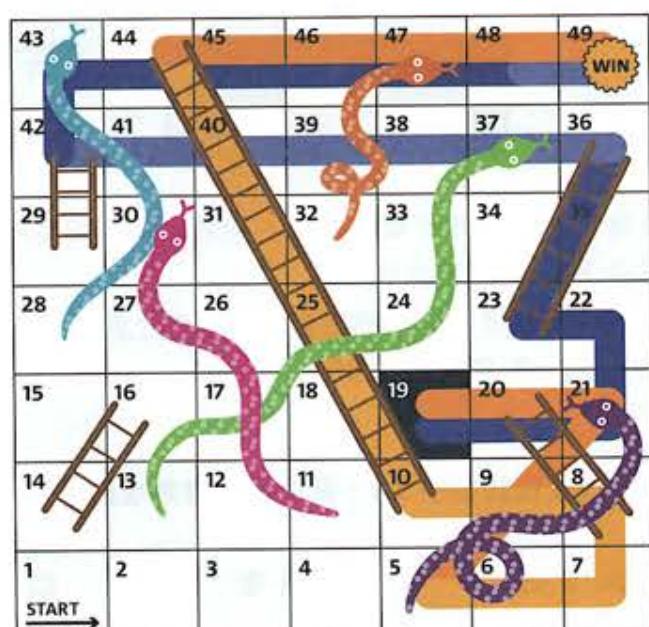


反之，如果有考慮到會後退的路徑，則可以透過格子 10 的梯子抵達格子 44。

首先從格子 19，投擲點數 2 可抵達格子 21，並透過蛇頭滑向蛇尾所在的格子 5。

再從格子 5 透過投擲點數 5 抵達格子 10，並透過梯子爬至格子 44。

最後，透過投擲點數 5 抵達終點，因此總共只需要再投擲 3 次骰子。



此外，並沒有任何方法可以在投擲 2 次的情況下抵達終點。當投擲 1 次的時候，玩家可能從格子 19 抵達的格子有 20、5、22、36、24 及 25(分別對應投擲的點數為 1 至 6)。

從這些格子，皆無法在只有 1 次投擲的機會下抵達終點。



資訊科學上的意義

這是一個關於 **最短路徑** 的問題。

最短路經的問題可以利用 圖 graph 的表示法找出答案。圖是一種抽象的資料結構表示法，由節點以及用來連結節點的邊所構成。在本題中，節點可表示每個格子，格子 i 若可以藉由一次擲骰子的機會到達格子 j 。格子 i 的節點點就有一個邊連結到格子 j 的節點；畫出這個圖後，把每個邊的長度都設為 1，那麼本題將變成從一個節點到另一節點，找出最短路徑的問題。

廣度優先搜尋 breadth-first search 是一種用於搜尋圖型資料結構的演算法。此演算法是從任意指定的起點開始，像樹根生長一樣擴散，先分別探查所有相鄰的節點，再由這些節點探查下一層所有相鄰的次節點。藉由這種方法，此題將由格子 19 的節點開始，經過一層廣度優先搜尋所能探查到的節點，表示是擲一次骰子可到達的格子，看最少經過幾層廣度優先搜尋能探查到終點，即可有系統的找出答案。



關鍵字

圖、最短路徑問題



13. 電動車

藍色電動車充電後可以行駛 4 公里，每次充電需要 3 分鐘。

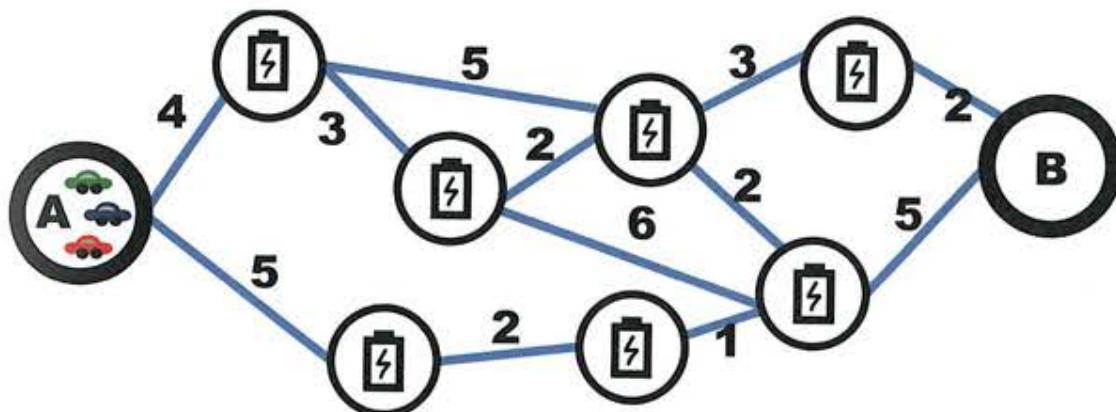
綠色電動車充電後可以行駛 5 公里，每次充電需要 4 分鐘。

紅色電動車充電後可以行駛 6 公里，每次充電需要 5 分鐘。

電動車不必在每個充電站都停下來，不過一旦停下來充電，無論這台電動車本來剩多少電力，它都必須充滿電才能離開。

目前所有電動車都以相同的速度（每分鐘 1 公里）行駛，並在 A 點充滿電。

下面的地圖顯示了 A 點和 B 點之間的所有道路和充電站。數字表示每個充電站之間的距離（以公里為單位）。



依據地圖，哪一種顏色的電動車可以用最短的時間從 A 點行駛到 B 點？

- A. 綠色電動車
- B. 藍色電動車
- C. 紅色電動車
- D. 三種電動車所需時間相同



正確答案是：A. 綠色電動車

- 綠色電動車到 B 點的最短時間為 21 分鐘，其所行經的距離為 { 5, 2, 1, 5 }，總共為 13 公里，花費 13 分鐘，它需要停靠 2 個充電站，分別為第 1 個與第 3 個充電站，共花費 8 分鐘 (4 分鐘 × 2) 的充電時間，因此，綠色電動車總共花費 $13 + 8 = 21$ 分鐘。
- 藍色電動車到 B 點的最短時間為 26 分鐘，其所行經的距離為 { 4, 3, 2, 3, 2 }，總共為 14 公里，花費 14 分鐘，它所經過的充電站都需要充電，合計 4 個充電站，共花費 12 分鐘 (3 分鐘 × 4) 的充電時間，因此，藍色電動車總共花費 $14 + 12 = 26$ 分鐘。
- 紅色電動車到 B 點的最短時間為 23 分鐘，其所行經的距離為 { 5, 2, 1, 5 }，總共為 13 公里，花費 13 分鐘，它需要停靠 2 個充電站，分別為第 1 個與第 3 個充電站，共花費 10 分鐘 (5 分鐘 × 2) 的充電時間，因此，紅色電動車總共花費 $13 + 10 = 23$ 分鐘。



資訊科學上的意義

在點與點間找出 **最短距離**，是生活中常見的問題，例如導航系統，用電腦來解決此問題時，首先要把地圖轉成一個圖 graph 來表示，以節點表示地點，並將地點間的距離紀錄在邊上。

近年來，電腦科學家設計各種演算法協助有效率尋找圖型中的最短路徑。其中一種演算法由荷蘭電腦科學家 Edsger Dijkstra 所發佈，現在被稱為 dijkstra 演算法。這種演算法可以找到從某個起點到其他節點的最短路徑。

但在現實情況下，不能只考慮地點間的最短距離，如同本題會有充電後行駛距離的限制及充電時間的考慮，或是有路段車流速度不同的影響，因而使得要找出最短時間的路徑更不容易。



關鍵字

最短路徑問題



14. 粉刷房屋

海狸街道上的人們認為替他們的白色房屋粉刷不同的顏色將大大地改善街景，因此他們決定根據下面的規則為房屋粉刷：

1. 每間房屋必須粉刷成紅色、綠色或藍色其中一種顏色。
2. 左右相鄰的房屋不能粉刷相同的顏色。
3. 房屋的顏色不能跟街道正對面的房屋相同。

在下圖中，你看到有五間房屋已經被粉刷。



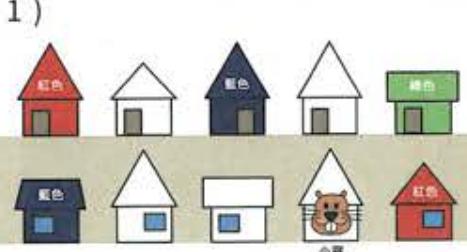
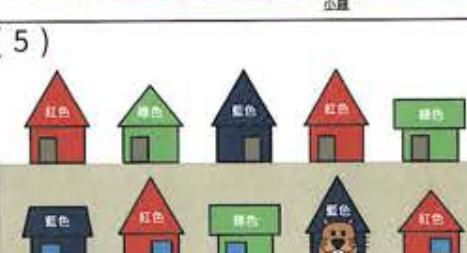
請問海狸**小羅**的房屋可以粉刷成哪一種顏色？

- A. 只能粉刷紅色
- B. 只能粉刷藍色
- C. 只能粉刷綠色
- D. 可以粉刷藍色或綠色



正確答案是：B. 只能粉刷藍色

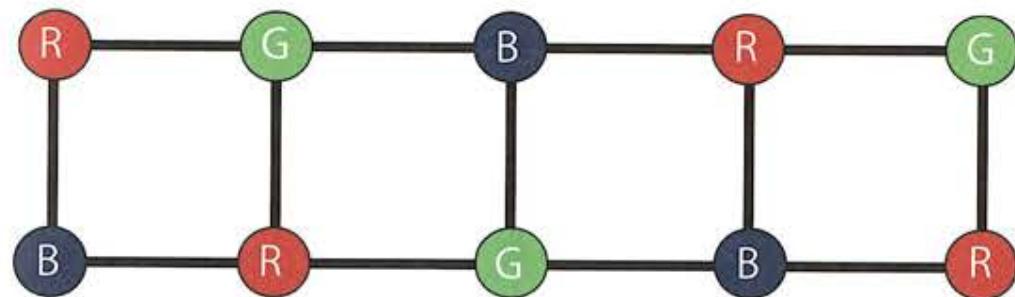
可以透過以下方式來決定正確的顏色：

(1)		從這張圖片開始。
(2)		先從第一排開始決定房屋的粉刷顏色，由於不能跟鄰居顏色相同，因此各只有一種顏色可供選擇。
(3)		接下來考慮第二排的第二間房屋。這間房屋的隔壁是藍色的而它對街的房屋是綠色的，因此它只能粉刷紅色。
(4)		基於同樣原因，第二排的第三間房屋必須粉刷綠色。
(5)		於是 <u>小羅</u> 的房屋就只能粉刷藍色！



資訊科學上的意義

此題中的 10 間房屋及其限制關係可形成所謂的 圖 graph，這是一個常用於資訊科學中表示資料及其連接方式。



一個圖由節點（圖中有顏色的圓圈）和邊（連接這些圓圈的線條）所構成。在此題中，每個節點對應到一間房屋，每個邊對應到兩間房屋有相鄰或在對街的關係。此外，題目要求有邊連接的兩個節點不能同色。在此問題中可用三種顏色來做到這一點，任何圖型都能討論最少需要幾種顏色來對節點著色，但是需要的最少顏色量取決於選擇的圖，未必只用三種顏色就能達到。

實際上，為平面圖（在同一平面上繪製，而不會有交叉邊的圖）著色最少需要多少種顏色是一個著名的問題，答案是 4 種；實際上需要大量的計算列舉才能證明這一點。

在資訊科學中，許多問題可以轉成用圖表示，就可運用現存的許多經典的圖論演算法來解決圖型上的各種問題。



關鍵字

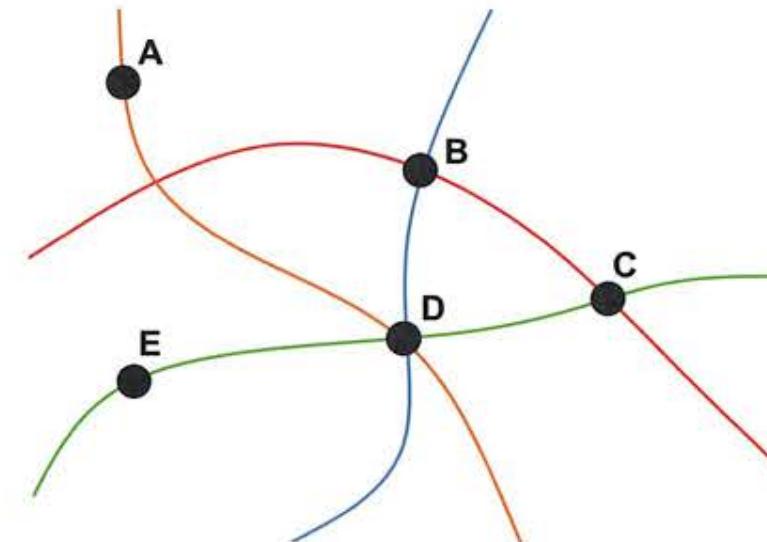
圖、著色問題



15. 城鎮與高速公路

這是 5 個城鎮和 4 條高速公路的地圖。

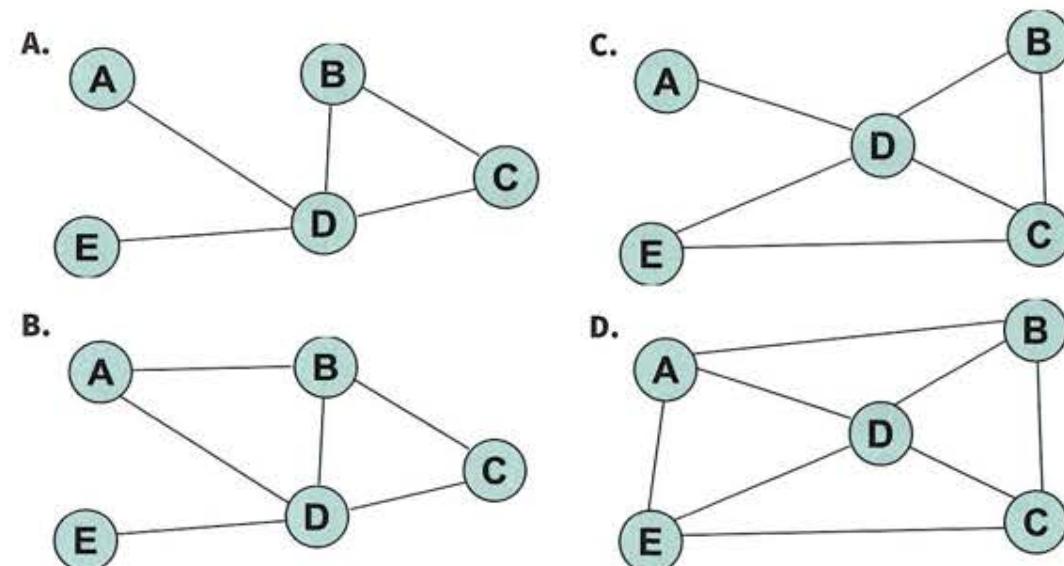
黑點代表城鎮，彩色線代表高速公路。



我們想將上面的地圖改用以下方式畫成新的圖形：

- 以圓圈表示城鎮。
- 以連接兩個圓圈的直線，表示兩個城鎮位於同一條高速公路上。

下列何者為新產生的圖形？





正確答案是：C

仔細研究地圖，我們可以看到：

- A 鎮與 D 鎮位於同一條高速公路上。
- B 鎮與 C 鎮位於同一條高速公路上；B 鎮和 D 鎮位於同一條高速公路上。
- C 鎮與 B 鎮在同一條高速公路上；C 鎮、D 鎮和 E 鎮在同一條高速公路上。
- D 鎮與 A 鎮在同一條高速公路上；D 鎮與 B 鎮在同一條高速公路上；D 鎮、C 鎮與 E 鎮都在同一條高速公路上。
- 最後，E 鎮與 C 和 D 鎮位於同一條高速公路上。

其他的答案都是錯誤的：

- 在答案 A 中，E 鎮和 C 鎮之間的連接不見了。
- 在答案 D 中，A 鎮與 B 鎮之間，以及 A 鎮與 E 鎮之間都有連接，但這些鎮不在同一條高速公路上。
- 同樣地，答案 B 中，E 鎮和 C 鎮之間的連接也不見了。

有些學生可能會犯粗心的錯誤，需要特別注意的是：

- 儘管 A 鎮可以到達 B 鎮，但實際上它們不在同一條高速公路上。
- 儘管 E 鎮和 C 鎮之間另有一個鎮，但它們仍在同一條高速公路上。



資訊科學上的意義

現實生活的問題必須轉換成電腦可以處理的資料形式，程式設計師需要選擇適合的演算法以及對應的資料表示方式。因此，對於程式設計師來說，了解多種資料表示法至關重要，也就是所謂資料結構。

一種常用的資料表示法是 **圖 graph**，圖型是一種由 **節點**（通常是名稱寫在圓圈內或圓圈旁）和 **邊**（連接節點的線）組成的資料結構。圖論則在探討圖型的特性及解決圖型問題的演算法。

正如這個任務所示，當我們把原來地圖中沒有連接任何兩個城鎮的高速公路都去除時，並加上 C 和 E 的邊（在同一條高速公路），就形成對應的圖型了。我們通常會把現實生活中不重要的資料去除，將重要的資料抽象化後記錄下來。



關鍵字

圖、節點、邊



16. 海狸 V.S. 袋鼠

如圖片所示，有五隻海狸在穿越沼澤的一條木樁小道上遇到迎面而來的一大群袋鼠，雙方在相隔一個木樁處停下。



海狸跟袋鼠都不想落入沼澤中，想要留在木樁小路上。還好袋鼠發現了某個木樁旁邊有個石頭，勉強可以讓袋鼠由那個木樁跳到石頭上再跳回來，只可惜石頭和所有木樁一次只能讓一隻袋鼠立足。

海狸和袋鼠都不介意一路返回到岸邊，除了帶頭的海狸**佛瑞德**，若 1 個木樁算 1 步，牠合計最多只願意往後共移動 10 步禮讓袋鼠，之後就要一路往前移動到對岸。

請問在**佛瑞德**堅持之下，最多有幾隻袋鼠可以不必退回岸邊就能通過沼澤？

- A. 10 隻以上
- B. 10 隻
- C. 6 隻
- D. 4 隻
- E. 不到 4 隻



正確答案是：C. 6 隻

除了佛瑞德以外，所有的海狸都可以忽略，因為它們都可以往後退來讓袋鼠跳過，也就是一開始除了佛瑞德，其他海狸都可以先退回到沼澤的另一邊等袋鼠通過。如果要讓袋鼠通過，佛瑞德必須做以下的步驟：

1. 袋鼠先往前一步跳到石頭上。	
2. <u>佛瑞德</u> 往前兩步。	
3. 袋鼠跳回木頭上之後就可以繼續往前走了。	
4. <u>佛瑞德</u> 必須往後退兩步，才能讓下一隻袋鼠跳上石頭。	

依照上圖的步驟五次之後，佛瑞德就已經退後十步了，此時佛瑞德的位置就是他最初遭遇袋鼠的位置，第五隻袋鼠已經通過，第六隻袋鼠可以跳上石頭，佛瑞德接著往前走，因此第六隻袋鼠可以跳回木棧道往前走，所以總共有六隻袋鼠可以在佛瑞德的條件下通過。

本題可以簡化為一個數學公式，如果佛瑞德要讓 k 隻袋鼠通過，他就必須要後退 s 步，所以 $s = 2 \times (k - 1)$ 。解此方程式可得 $k = 0.5 \times s + 1$ 。



資訊科學上的意義

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：每個木樁跟石頭就像變數一樣用來儲存資料，而袋鼠跟海狸就像放置在這些變數裡的資料。

演算法中的步驟，可以更改這些變數內容來反映執行結果。為解決這個問題，處理過程中有幾個相同的步驟重複了好幾次，也是運算思維當中的基本特徵：拆解問題，將原先的問題拆解成比較小的問題後，用重複的步驟解決這些問題。

在設計演算法的過程，將不斷重複步驟的概念找出來轉換為可執行的程式碼，便可用來快速及自動地解決問題。



關鍵字

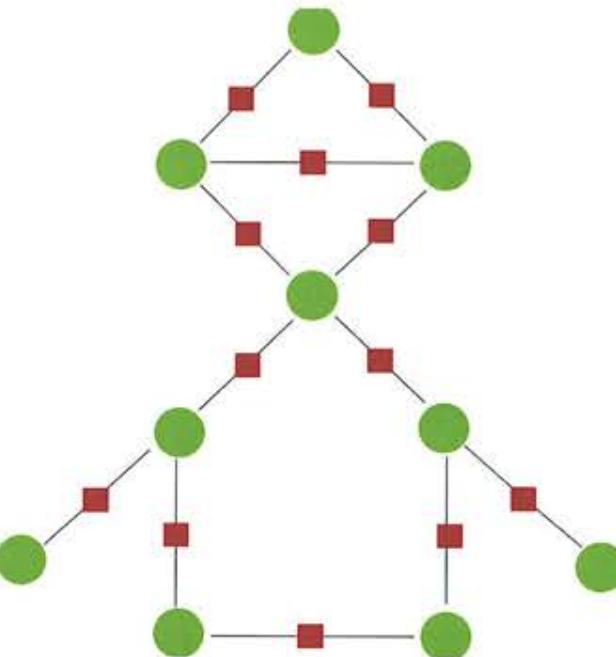
演算法



17. 雜貨商店

由於病毒爆發，許多雜貨店不得不關閉以降低健康風險。但是為了確保食物供應無虞，每個村莊和維持營運的雜貨店之間，要有直接連通的街道。

下圖綠色圓點代表村莊，紅色方形代表雜貨店，連接它們的線代表街道。

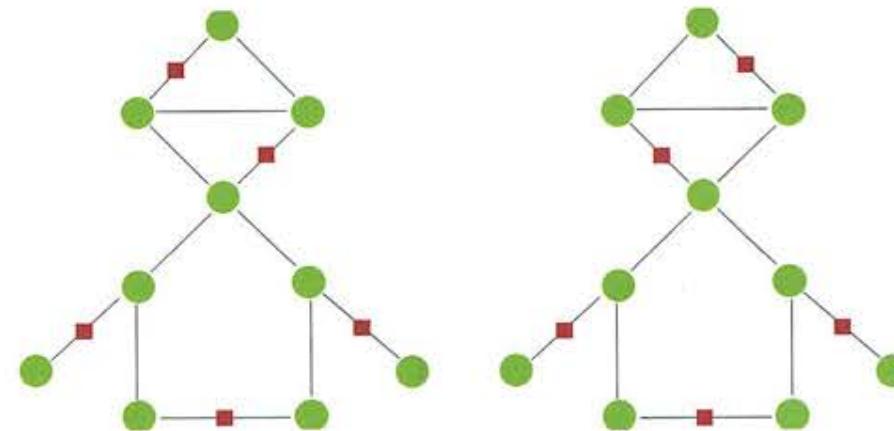


地圖中的雜貨店最少需幾家維持營運，以確保所有村莊的食物供應無虞？(範圍 [0~12] 的整數)



正確答案是：5家

最左邊和最右邊的雜貨店一定需要營業，以供應最左邊和最右邊的村莊。為了供應底部的兩個村莊，底部的雜貨店保持一間營業就足夠。至此，我們選擇了3家雜貨店營業，除了頂部的四個村莊外，所有村莊均已供應。若選擇頂部四個村莊中間的雜貨店，因為此雜貨店沒有連接到上下兩個村莊，我們必須至少再選擇兩家雜貨店。但若選擇正方形對面兩個雜貨店即可供應所有四個村莊（如下圖）。



我們找到了一個完美的解決方案，選擇了5家雜貨店：最左邊和最右邊的兩家，底部的一家和頂部的兩個相對雜貨店。



資訊科學上的意義

這個問題為圖典型的 **最小邊緣覆蓋** 問題。我們必須從一個圖型中找到最小數量的邊，以便圖的每個頂點都被所選擇的其中一個邊連接到（覆蓋）。在這個問題中，村莊對應到需要被覆蓋的節點，有雜貨店的街道是可覆蓋節點的邊。

由於一個邊可覆蓋到最多兩個節點，因此圖若有N個節點，最小邊緣覆蓋至少要包含「 $N/2$ 」個邊，然而要找出這些邊，就必須設計一個演算法來挑選出這些邊。圖論中會探討這些圖上的問題，以及如何運用圖上的特性有效率找出解的演算法。



關鍵字

最小邊緣覆蓋



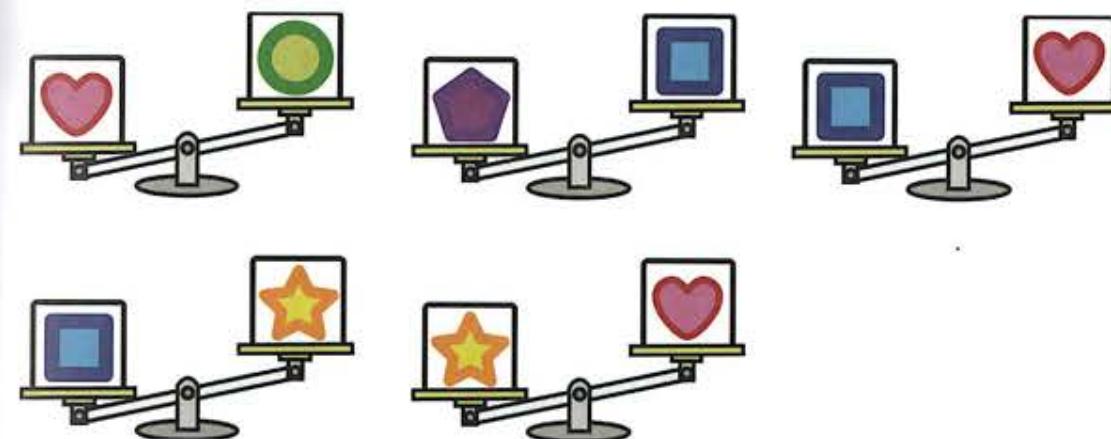
18. 最重的標誌

有五個盒子，每個盒子上都有不同形狀的標誌。你可以使用天秤來比較兩個盒子的重量。

舉例來說： 表示 比 重。



下面顯示五組盒子在天秤上的重量比較結果。



請問最重的盒子是哪一個？

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



正確答案是：C

我們能提前斷定最重的盒子嗎？

最重的盒子在所有的比較中，必須是較重的那個盒子（也就是說，它一定是在天秤較低的一側）。在五次比較中，五角形盒子是唯一不會在天秤較高一側的盒子。由於其他盒子至少有一次在比較中是屬於較輕的盒子，因此我們可以確定那些盒子不是最重的。



資訊科學上的意義

從五個盒子中找到最重的盒子，你必須 **比較五次**！

當盒子很多時，或要再找出最輕的盒子，會導致更多次的比較。如果盒子已按重量排序，找到最重或最輕的箱子會容易得多。

電腦經常需要從大量資料中尋找到特定資料，如果資料已先排序好，這會容易得多，這也就是為何排序是資訊科學中很重要的議題。資訊科學家設計了許多排序演算法，由於電腦經常需要對資料進行排序，因此有效率的排序演算法非常重要。

由於電腦中一個指令同時只能比較兩個值，就好像此任務中的天秤。因此，無論採用何種排序演算法，排序的核心運算都是兩筆資料值的比較，排序演算法使用兩兩比較數值的過程及結果，來決定出資料的大小順序。



關鍵字

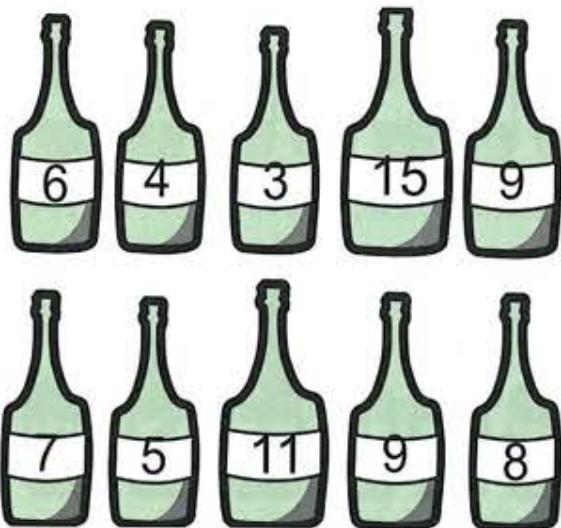
比較、排序



19. 水瓶

海狸傑克在裝瓶工廠工作，他的工作目標就是裝滿愈多瓶子愈好。

他每天都被分派到一缸 50 公升的水，而今天工廠提供了下圖的 10 個空瓶子給他，每個瓶身上都標有瓶子的容量（公升）：



請問海狸傑克今天最多可以裝滿幾個水瓶？（範圍為 [1, 10] 之間的整數）



正確答案是：7 個水瓶

如果你有兩個不同容量的空瓶，希望能夠盡可能裝滿越多瓶越好，則先裝容量較大的那一瓶並非最佳解。針對這個任務所應採取的策略應是先從最小容量的水瓶開始裝，我們可以先把瓶子按照容量由小排到大，接著依序裝瓶直到把水用完即可。



本題按照空瓶容量由小到大依序選擇， $3+4+5+6+7+8+9=42$ ，把水依序裝滿前 7 個空瓶後，還剩下 $50-42=8$ 公升，但剩下的 8 公升已無法再裝滿其他的空瓶了。



資訊科學上的意義

在問題解決時，使用 **貪心演算法** 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。在很多問題中，貪心演算法不一定會產出整體的最佳解，但卻能在合理的時間內得到可接受的最佳解。

一般而言，一個問題要採用貪心演算法，要將本來的問題拆解成找尋答案的步驟，在此題中就是一次裝滿一個空瓶。另外要有一個函式用來評估為達到目標當前狀態的最佳選擇，在此題中是裝滿某個空瓶後的剩下水量。由於目標是裝滿最多空瓶，因此在此題貪婪演算法的策略是每次選最小容量的空瓶裝水，讓剩下的水最多，以便可能再裝更多空瓶。



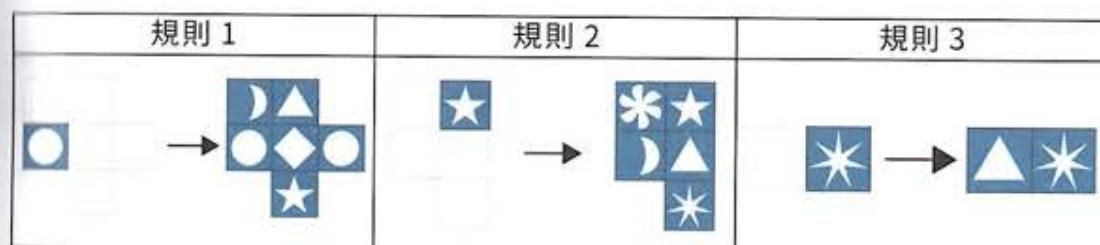
關鍵字

貪心演算法



20. 遵循規則

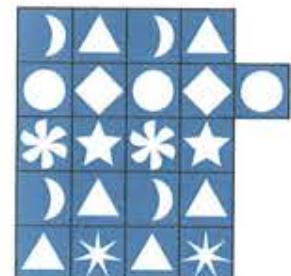
你可以透過以下規則產生漂亮的圖形：



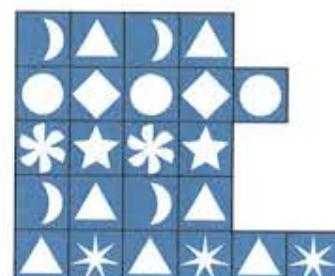
例如，如果你從 開始，使用兩次規則 3，就可以產生

如果你從一個圓形 開始，使用上方的規則，可以產生下列那一個圖形？

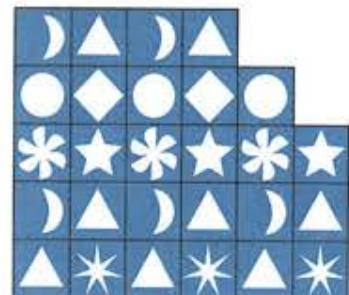
A.



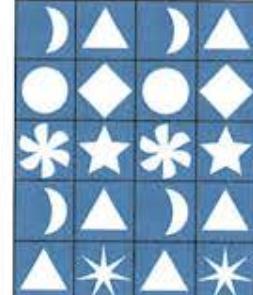
C.



B.



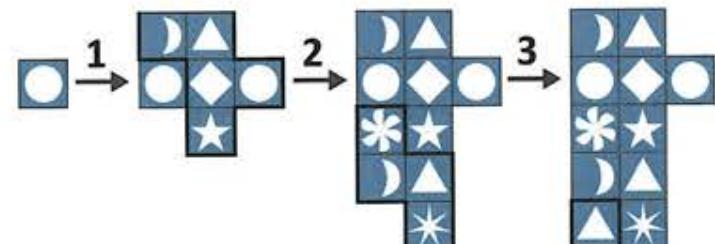
D.



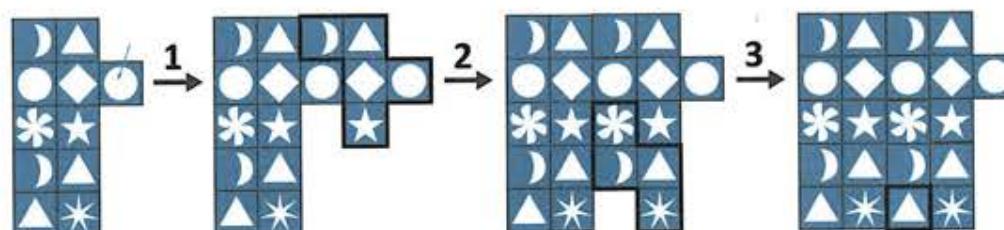


正確答案是：A

下圖顯示如何從一個圓形 開始，得到圖案 A。首先，我們依次使用規則 1、規則 2 和規則 3：



接下來，我們對最右邊的圓形（下圖箭頭所示）使用規則 1，然後再次使用規則 2 和規則 3。



請注意，在這裡我們依序使用規則 $1 \rightarrow$ 規則 $2 \rightarrow$ 規則 $3 \rightarrow$ 規則 $1 \rightarrow$ 規則 $2 \rightarrow$ 規則 3 ，但是這並非唯一的順序，也可以使用其他順序獲得相同的結果。

你需要一些技巧來判斷其他圖形是無法由題目定義的規則產生。當然，你也可以嘗試製作圖形 B、C 或 D，結果每次都失敗，但是，如此一來很難確定自己是否遺漏了其他產生圖形的方式。

圖形 D 是不可能的：請注意，規則 1 是唯一可以生成菱形 的規則。每增加一個菱形 的同時，也會增加一個圓形 。我們從一個圓形 開始，由這些規則產生的圖形中，圓形必定會比菱形多一個。然而在圖形 D 中，圓形和菱形一樣多，因此可以推斷無法使用題目的規則產生 D。

圖形 B 是不可能的：規則 1 是唯一可以生成五角星形 的規則，且五角星形總是在菱形 下方。在圖形 B 中，最右邊的五角星上方沒有菱形，因此可以推斷無法由三個規則產生圖形 B。

圖形 C 是不可能的：同樣地，規則 2 是唯一可以生成七角星形 的規則，且七角星形的在上方必定為三角形 。圖形 C 包含一個七角星形，但是上面沒有三角形，因此可以推斷無法使用題目的規則產生圖形 C。

另外一種更簡單的判斷方式是：在所有規則中，只有規則 1 能夠將圓形向右擴展。因此，最後產生的圖形中最右邊一欄中，必定能找到一個圓形 ，圖形 B、C、D 都不符合。



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**，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作的相同處理。

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。他們經常使用本題在解釋為什麼圖形 B、C 和 D 無法根據規則產生的技巧，這些技巧通常涉及綜合數學理論嚴謹的分析方法或計算的問題，這一部分稱為理論計算機科學 theoretical computer science。



關鍵字

規則、圖形識別



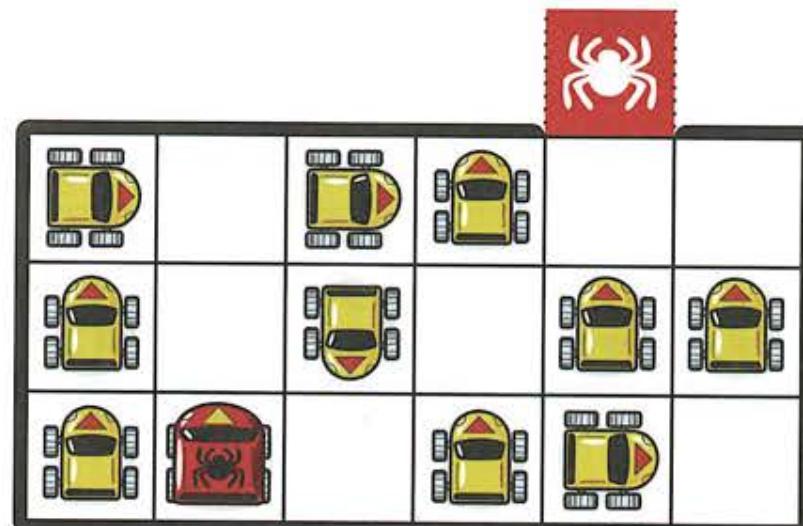
21. 智慧停車

停車場的智慧車可以根據以下規則移動

智慧車可以做以下的一個「動作」：

- 向前移動一格
- 向後移動一格
- 在目前的位置向左轉 (90 度)
- 在目前的位置向右轉 (90 度)

除此之外，只有蜘蛛車才能移動到停車場外的蜘蛛停車格。



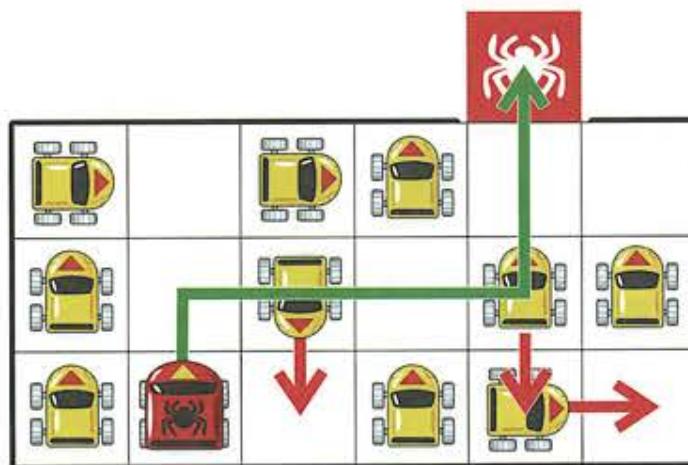
智慧車 (包含蜘蛛車) 最少需要執行多少個「動作」才能將蜘蛛車停到蜘蛛停車格？

- A. 9
- B. 11
- C. 13
- D. 15



正確答案是：B. 11 個動作

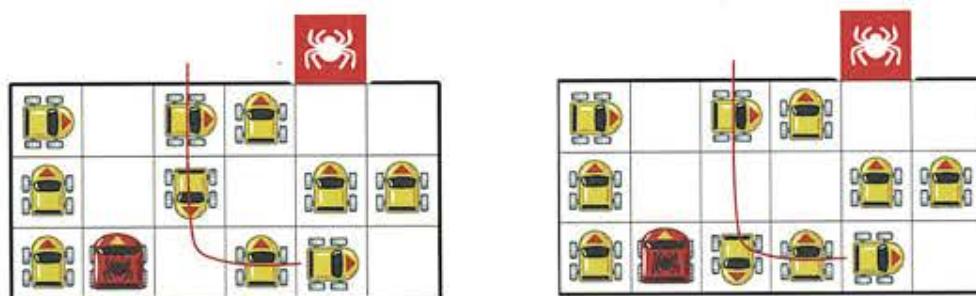
如圖所示，蜘蛛車可以透過 11 個動作到達蜘蛛停車格：



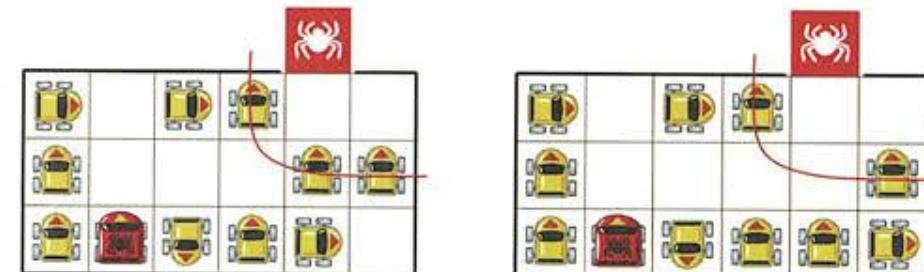
現在仍需證明最少需要 11 個動作。

假設蜘蛛車是停車場中唯一的汽車，要到達蜘蛛停車格，總共須向上移動 3 次，向右移動 3 次，以及旋轉 2 次。這些動作有多種不同的排列順序可以讓蜘蛛車移動到停車格中，無論使用哪種排列順序，蜘蛛車至少需要這 8 個動作才能到達蜘蛛停車格，然而，蜘蛛車移動的路徑上，可能會有其他智慧車擋住，需要移動其他的智慧車。

首先，我們先把會擋住目標位置的智慧車移開，如左下圖中紅線標示擋住目標位置的智慧車。此時只要像右下圖所示，讓其中一輛智慧車往前移動一格，即可移除障礙。



接著，我們要繼續移掉會擋住目標位置的智慧車，如左下圖中紅色線所示，此時需要兩個動作才能移除障礙。



因此智慧車所需的最少動作數為 $8+1+2=11$ 。



資訊科學上的意義

證明一個解法是最佳解（如最少次數）可能是非常困難的事情。

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。在絕大多數的情況下，人工進行窮舉搜尋是不可行的，但是，以電腦系統化的來進行窮舉搜尋卻是相當直觀且快速。

有時，一個問題的可能解法過多，以至於即使使用電腦，窮舉搜尋也需要花很長的時間。發生這種情況時，可以使用一些演算法設計的策略來簡少可找到最佳解所要搜尋的解法，例如，分支界線法 divide and conquer algorithm 或貪婪演算法 greedy algorithm。



窮舉搜尋



22. 零錢包



海狸丹丹去水上樂園的時候，習慣把銅板放在防水袋中，銅板才不容易生鏽。今天出發前，他準備了 63 枚 1 元的海狸幣，然後按照一定的規則把它們分別放進幾個防水袋中；如此一來，需要付錢的時候，不用打開防水袋，只要把其中幾個袋子交出來，就可以購買一個價錢在 1 到 63 元之間的商品。

請問丹丹最少要準備幾個防水袋？（範圍為 [0~9] 的整數。）



正確答案是：6 個防水袋

丹丹必須將以下數量的海狸幣放入各自的防水袋裡面：1、2、4、8、16、32；顯而易見的， $1 + 2 + 4 + 8 + 16 + 32 = 63$ ，所以他可以不需要打開任何一個防水袋就能支付 1 到 63 這個區間的任意海狸幣。例如他想要支付 13 枚，他可以用裝有 1、4 和 8 這三種海狸幣的袋子來組合 13。使用這幾個袋子來組成 1 到 63 的數字組合，以下列舉一些例子：

袋子內的銅板數	1	2	4	8	16	32
3	V	V				
20			V		V	
50		V			V	V
60			V	V	V	V

下表完整的表示了透過這六個袋子（且不需要打開的前提下），如果取用他們其中的幾個袋子來組成 1 到 63 的數字組合，表格的標頭是每個袋子內裝入銅板的數量。之後的欄位如果取用了那個袋子就標示為 1，否則標示為 0。

銅板數	32	16	8	4	2	1
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	0	1	0	1	0
11	0	0	1	0	1	1
12	0	0	1	1	0	0
13	0	0	1	1	0	1
14	0	0	1	1	1	0
15	0	0	1	1	1	1
16	0	1	0	0	0	0
17	0	1	0	0	0	1
18	0	1	0	0	1	0
19	0	1	0	0	1	1
20	0	1	0	1	0	0
21	0	1	0	1	0	1

銅板數	32	16	8	4	2	1
22	0	1	0	1	1	0
23	0	1	0	1	1	1
24	0	1	1	0	0	0
25	0	1	1	0	0	1
26	0	1	1	0	1	0
27	0	1	1	0	1	1
28	0	1	1	1	0	0
29	0	1	1	1	0	1
30	0	1	1	1	1	0
31	0	1	1	1	1	1
32	1	0	0	0	0	0
33	1	0	0	0	0	1
34	1	0	0	0	1	0
35	1	0	0	0	1	1
36	1	0	0	1	0	0
37	1	0	0	1	0	1
38	1	0	0	1	1	0
39	1	0	0	1	1	1
40	1	0	1	0	0	0
41	1	0	1	0	0	1
42	1	0	1	0	1	0



資訊科學上的意義

本題主要在討論 **二進位制系統**，電腦科學領域跟數學領域學習二進位數的方法不同。數學偏重於它的運算跟特性，電腦科學則較關注在如何應用上面。

電腦透過二進位系統來表示各種不同的訊息：文件、照片、聲音、影像、數字，乃至於各種程式軟體，皆可編碼成二進位格式，並且以位元組的單位儲存。

一個位元本身無法表示太多，但是多個位元的組成就能夠用來表示更大的數據。

電腦裡的位元通常會以八個位元為一組的方式組合，即稱為位元組 byte，一個位元組即可以用來表示 0 到 255 的數字。

在這個題目中，丹丹只使用了 6 個位元，因此可以組合出 1 到 63 的數字。



關鍵字

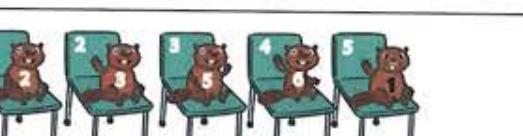
二進位、資料表示、邏輯



23. 最後的贏家

1 到 6 號海狸分別坐在編號由 1 至 6 並排列好的椅子上玩遊戲。開始遊戲之前，會從 1 到 4 之中選擇出 1 個數字。在遊戲開始後，每次裁判拍手，所有海狸將依據預先 所選的數字 往右移動位置，若往右移動時沒有椅子可坐，則移動至最左方的位子來補位。每次海狸們移動完成後，坐在最右邊椅子的海狸將在該回合被淘汰，且最右邊的椅子會被移走。遊戲到最後，仍留在椅子上的海狸是最終贏家。

舉例來說，如果 2 為遊戲開始前所選的數字，那麼海狸 6 就是贏家，詳細解說如下圖所示：

遊戲開始前	
第一次拍手： 4 號海狸淘汰	
第二次拍手： 1 號海狸淘汰	
第三次拍手： 3 號海狸淘汰	
第四次拍手： 5 號海狸淘汰	
第五次拍手： 2 號海狸淘汰	

如果遊戲開始前，所選的數字是 3，請問幾號海狸將成為最後的贏家？（範圍 [1~6] 的整數。）





正確答案是：2 號海狸

遊戲開始前	
第一次拍手： 3 號海狸淘汰	
第二次拍手： 5 號海狸淘汰	
第三次拍手： 6 號海狸淘汰	
第四次拍手： 4 號海狸淘汰	
第五次拍手： 1 號海狸淘汰	

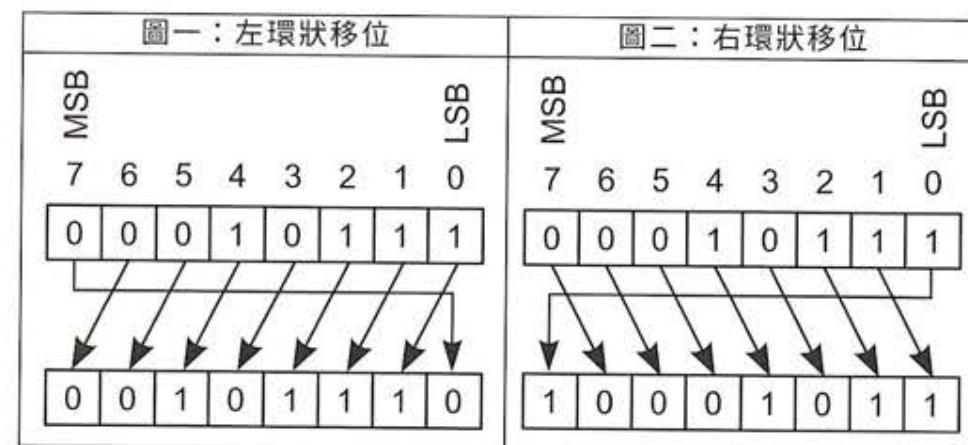


資訊科學上的意義

在數位電腦程式設計，位元運算 是作用在一個或多個位元，或是作用在二進位數字的個別位數上，這是一種快速且簡單的運算，可直接由中央處理器操作，常被用在數值比較或數值的計算上。

位元運算中有一種運算稱作環狀位移 *circular shift*，因為資料左右兩端像是相連在一起成一個環狀，使一整串位元可旋轉：在左移的狀況下，最左端的位元值會移至最右端；在右移的情況下，最右端的位元值會移至最左端。此任務中海狸在遊戲中每次拍手後的移動就相當於右環狀移位，所選的數字是指定要執行幾次右環狀移位。

這種運算對於需要保存資料的所有位元時很有用，且常被用於數位密碼學中。



關鍵字

位元運算



24. 圖書編碼

海狸圖書館裡擺放 10 個書櫃，每個書櫃中擺放 10 個書架，以放置海狸書。圖書館長海狸利布拉為這些書籍編上特殊的數字編碼來表示海狸書的位置，讓找書更方便。這些特殊數字編碼是以書名的海狸文對應到數字，對應方式如下表。

海狸文	□	△	☆	○	◎	◇	♡	☁
數字編碼	1	2	3	4	5	6	7	8
海狸文	+	⊗	⌚	♾	🏔	〰	💧	🌀
數字編碼	9	0	1	2	3	4	5	6

舉例說明，如果書名是 ◎ ◇ ○ ☆ △，那這本書的特殊數字編碼就會是 56432。(如右圖)

↓ ↓ ↓ ↓ ↓
5 6 4 3 2

下列選項中哪本書與下圖的這本紅有一樣的特殊數字編碼？

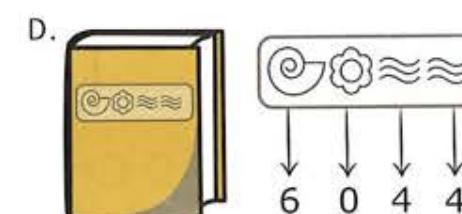
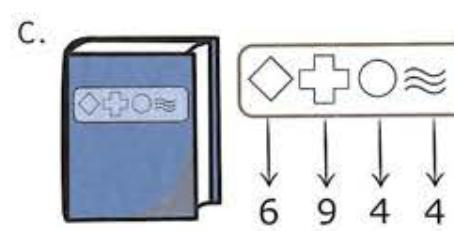
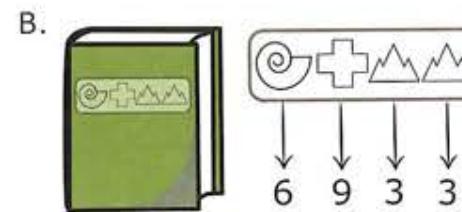
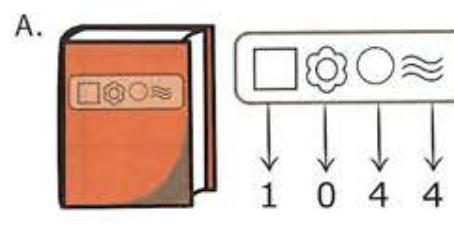
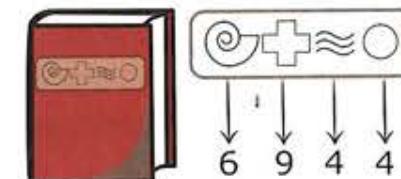


- A.
- B.
- C.
- D.



正確答案是：C.

我們先來看看紅書的特殊數字編碼是什麼：



資訊科學上的意義

在電腦科學中 **雜湊表 hash table** 是一種資料結構，用來儲存需要被頻繁查詢的資料。雜湊表將本來的資料識別稱作關鍵值 key，快速對應到表示資料存放位置的一個索引代碼（通常是數字代碼，稱作 value），以方便找到資料儲存位置。

題目中的對應表就是為書籍的關鍵值書名產生索引代碼的方法。然而，就像題目中的狀況一樣，多本書籍的海狸文書名可能會對應到同一個索引代碼，這種狀況稱為發生 **碰撞**。當為雜湊表設計從關鍵值對應到索引代碼的產生方法時，電腦科學家需要考慮盡可能避免發生碰撞，如果對應到同一個索引代碼的碰撞次數超過一個位置能儲存的資料量，就要設計其它方法來決定存放位置。



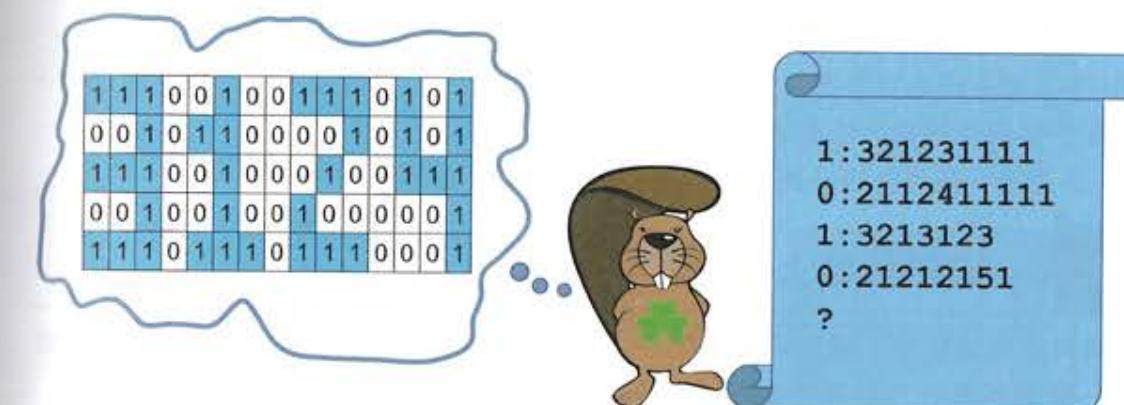
關鍵字

雜湊表、編碼、碰撞

25. 宴會密碼

海狸安安常和她的朋友們傳遞訊息，為了保密，常使用特殊方式將訊息轉換成密文。

安安正在準備一個秘密宴會，宴會入場暗號為 3124。如下圖所示，安安將入場暗號轉換成 5 列的密文。



請問下列何者是密文的最後一列？

- A. 0 : 3131331
- B. 1 : 321231111
- C. 1 : 3131331
- D. 1 : 231231111



正確答案是：C. 1:3131331

密文每一列的第一位數字表示該行是以白色或藍色方塊開頭 (0 表示白色，1 表示藍色)。「：」後面的數字表示白色 (0) 或藍色 (1) 重複的次數，兩種顏色方塊交替出現。入場密碼的最後一列以藍色方塊 (1) 開頭。「：」後面的數字 3，表示有 3 個連續藍色方塊。其次是 1 個白色，3 個藍色，1 個白色，依此類推。在這種情況下，正確的代碼是：

1 : 3 1 3 1 3 3 1

第一個為藍色方塊 : 3 藍 1 白 3 藍 1 白 3 藍 3 白 1 藍



- (A) 不正確，因為以「0：」開頭表示第一個方塊是白色，就像密文的第二列和第四列一樣。
- (B) 不正確，因為它跟密文的第一列一樣：在 3 個藍色方塊和 1 個白色方塊之後，將有 2 個白色方塊，依此類推。
- (D) 不正確，因為它表示的密文的最後一列將以 2 個藍色方塊開頭。



資訊科學上的意義

在資料處理過程中，通常可以運用不同的編碼方式來表示相同的訊息，有時需要從一種 資料表示 方法轉換為另一種資料表示方法。

此任務中引入的 遊程編碼 Run-Length Encoding, RLE 是一種非常簡單的無失真資料壓縮 技術。RLE 編碼使用「變動長度的碼來取代連續重複出現的原始資料」來編碼。在此題中，因為 0 或 1 在同一列中連續出現的計數次數有高於 1，因此紀錄計數值可以節省計數值的儲存 (原來三個 1 只要記錄成一個 3)。在大多數情況下，經過編碼的輸出資料長度小於輸入資料長度，但在最壞的情況下輸出資料長度可能更大。

此題要求學生具備模式辨識和 演算法 法則能力。首先，因為在題目中並未明確給出該編碼方法，學生必須比較示例的相應列來推 RLE 編碼算法，再將此編碼方法應用於最後一列以獲得解答。RLE 編碼不是強加密算法，在觀察一些密碼訊息後，即使不知道相對應的密碼，也不會很難推論出該編碼方法，因此不應將其應用於任何真正重要的秘密。



關鍵字

遊程編碼、資料表示、資料壓縮、演算法、密碼學



26. 派對密碼

麗麗、維維和小傑三人會用一種祕密語言傳遞訊息。

他們分別在紙條上用他們的祕密語言寫下要帶去派對的物品。如下圖所示，祕密被破解後，大家都知道麗麗會帶食物 (FOOD)，維維會帶禮物 (GIFT)。

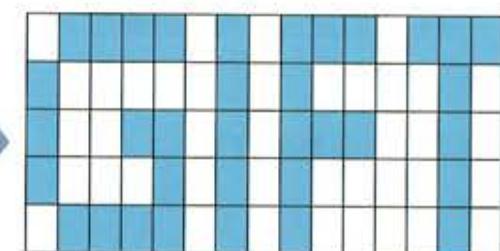
麗麗的紙條

```
1:32131221
1:13111111111111
1:31111111111111
1:13111111111111
1:14131221
```

1	1	1	0	0	1	0	0	0	1	0	0	1	1	0
1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
1	1	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	0	0	1	0	0	0	1	0	0	1	1	0

維維的紙條

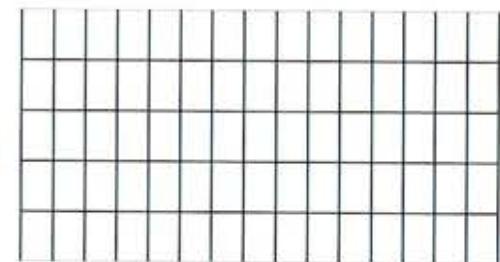
```
0:14111313
1:15111411
1:1221113211
1:1311111411
0:141111411
```



下圖為小傑寫的紙條，請問小傑會帶什麼物品去參加派對？

小傑的紙條

```
0:122123121
1:13111111111111
1:133122111
1:13111111111111
0:1211111111121
```



A. GAME

B. COLA

C. CARD

D. CAKE

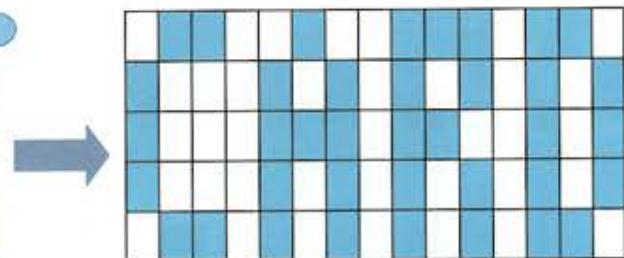


正確答案是：C. CARD

密碼每一行的第一位數字表示該行是以白色或藍色方塊開頭 (0 表示白色，1 表示藍色)。「：」後面的數字表示白色 (0) 或藍色 (1) 重複的次數，兩種顏色方塊交替出現。

小傑的紙條

```
0:122123121
1:131111111111
1:133122111
1:131111111111
0:121111111121
```



小傑密碼的第一行以白色方塊 (0) 開頭。「：」後面的數字 1，表示有 1 個白色方塊，然後 2 個藍色方塊，然後是 2 個白色，1 個藍色，依此類推。

● 密碼的第一行解碼的如下：

0	:	1	2	2	1	2	3	1	2	1
第一個為白色方塊										
: 1 2 2 1 2 3 1 2 1										
白 藍 藍 白 白 藍 白 藍 白										

● 密碼的第二行解碼的如下：

1	:	1	3	1	1	1	1	1	1	1
第一個為藍色方塊										
: 1 3 1 1 1 1 1 1 1 1 1										
藍 白 藍 白 藍 白 藍 白 藍 白 藍										

如果對其餘三行重複這些步驟，則可以解決此題。



資訊科學上的意義

像素是一個很小的顏色方塊，將圖像 image 儲存為電腦可以理解的數字序列，每個像素都有一個特定的數字，該數字告訴電腦該像素應為哪種顏色。若每個像素只用 0 和 1 來代表顏色 (黑或白)，則構成二元圖像。

此題利用二元圖像進行 **資料表示**，並運用二進制壓縮資料，將文字表示為一串編碼。在資料處理過程中，通常可以運用不同的編碼方式來表示相同的訊息。

此任務中引入的 **遊程編碼 Run-Length Encoding, RLE** 是一種非常簡單的無失真資料壓縮 技術。遊程編碼使用變動長度的碼來取代連續重複出現的原始資料來編碼。儘管在大多數情況下，編碼後的輸出資料長度小於輸入資料長度，但在最差的情況下輸出資料長度可能更大。

學生必須通過觀察示例密碼發現編碼方法的規則性，再應用此編碼方法自行解出小傑的密碼所對應的二元圖像。



關鍵字

遊程編碼、資料表示、資料壓縮



27. 樹木數獨

海狸在田地上分 3 列種植 9 棵樹，每 1 列有 3 棵樹。
樹木有三種不同的高度：1()，2() 和 3()。

此外，在每一列（水平線）上和每一行（垂直線）上的三棵樹，高度皆不相同。海狸站在田地周圍每行及每列的位置，觀察從該位置可看見多少棵樹，並將數字寫在一個牌子上（如下圖一），表示行的牌子形狀為 ，列的牌子形狀為 。



圖一

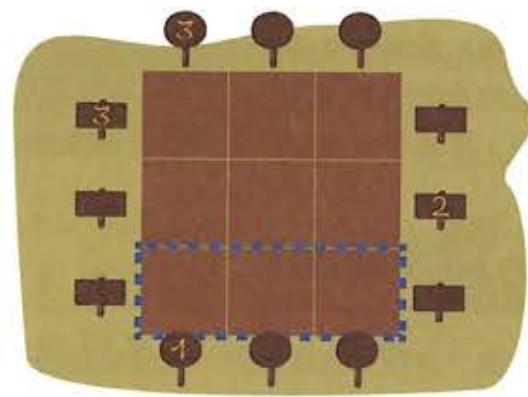


圖二

海狸在觀察樹木有幾棵時，是看不到隱藏在較高樹木後面的樹木的。
例如上圖二，當站在左邊時，高度 1 的樹 () 會被高度 2 的樹 () 挡住，因此看到 2 棵樹；站在右邊時，高度 3 的樹 () 會擋住另外 2 棵樹，只看到 1 棵樹。

如果根據觀察，已寫下數字的牌子如下圖，請問哪個選項是最下面框出來的列中由左到右的樹木高度？

- A. 1()，2() 和 3()
- B. 3()，1() 和 2()
- C. 3()，3() 和 1()
- D. 3()，2() 和 1()





正確答案是：B. 3(🌲), 1(🌲) 和 2(🌲)

本題只有一種解決方法。這兩個牌子記錄 3 的提示足以明確地重建整個田地。如果海狸從一個位置看到所有樹木，則表示樹木的高度必須由按照順序從這個位置持續增加高度 $\text{1}(🌲) \text{, } 2(🌲) \text{, } 3(🌲)$ 。用這種方法能推論出第一列和第一行。

接下來再根據數獨的規則放置其餘樹木。



由於在每一列和每一行中樹的高度都不同，所以只有一種方法可以放置其他樹。中間列右端的位置必須為 $1(🌲)$ ，因為第 2 列有一個 $2(🌲)$ ，且第三行中也已有一個 $3(🌲)$ 。我們得出的結論是，右下角必須為 $2(🌲)$ ，因為 $1(🌲)$ 和 $3(🌲)$ 已經存在第三行，依此類推。唯一的樹木種植位置如右圖。



資訊科學上的意義

此任務與電腦科學計算中的兩個概念有關。第一個概念是檢查符合條件，將資料樣式轉換成一個數字，例如本題中觀察到樹的數目，相當於計算從 **序列** 開頭為遞增數列的長度。第二個概念是從部分訊息重建物件的能力，例如本題中從某一行或某一列，若觀察到樹的數目為 3，則表示從這個位置樹木高度依序為 $1(🌲) \text{, } 2(🌲) \text{, } 3(🌲)$ 。從某一行或某一列的兩端，若觀察到樹的數目皆為 2，則表示樹木的高度從這個位置高度依序為 $2(🌲) \text{, } 3(🌲) \text{, } 1(🌲)$ ，或 $1(🌲) \text{, } 3(🌲) \text{, } 2(🌲)$ 。從某一行或某一列的一端及另一端，若觀察到樹的數目分別為 2 及 1，則表示樹木的高度從這個位置高度依序為 $2(🌲) \text{, } 1(🌲) \text{, } 3(🌲)$ 。上述觀察可以用於物件的 **壓縮** 表示。若從壓縮後的表示法可以重建出跟原來完全相同的資料，不會丟失任何訊息，稱為 **無損壓縮**，否則稱為 **有損壓縮**。



關鍵字

數獨、序列、遞增數列、有損和無損壓縮

海狸有一塊田分割成 4×4 格共 16 塊，他們準備在每一塊小田地上種一棵樹。他們依據以下規則，在 16 塊小田地分別種植高度為 $1(🌲)$, $2(🌲)$, $3(🌲)$ 和 $4(🌲)$ 的樹：

- 規則一：每一列的小田地（每一水平線）每種高度的樹只能種一棵。
- 規則二：每一欄的小田地（每一垂直線）每種高度的樹只能種一棵。

如果海狸在某一列或某一欄觀察樹木（如下圖），他們將看不到隱藏在高大樹木後面的樹木。



在 4×4 格田地每一列和每一欄的末尾，海狸都放置了一個標誌牌，並在標誌牌上寫下了從該位置看見的樹木數量。海狸在標誌牌上寫的樹木數量是正確的，但他有一些樹種錯田地了。



你能找到海狸犯的錯誤並移動種錯的樹木嗎？

- A. (A,X),(D,X)
- B. (B,X),(D,X),(C,Z),(B,Z)
- C. W 列
- D. (A,W),(C,W),(A,Z),(C,Z)



正確答案是：D

當我們查看海狸的種樹說明時，樹木的種植應遵循規則（每一列和每一欄均包含四種高度的四棵樹），但是標誌牌上的數字並無法對應於每個位置可見的樹木數。要解決此問題，必須確認每一列或每一欄標誌牌上的數字與可見的樹木數字是相同的。經觀察確認，第 X 列和第 Y 列，第 B 欄和第 D 欄的樹木是不需要被更動的，因為它們的樹木數量與標誌牌上的數字是符合的。

依標誌牌標示的樹木數量，任何不正確的列或欄，都需要移動一些樹木。從第 W 列和第 Z 列以及第 A 欄和第 C 欄的標示數字來看，看到的樹木數量與標示數字不同，表示樹木放置位置錯誤。

依照此邏輯，我們可以確認錯誤的列與欄交錯處，正是需要移動樹木的田地（下圖中的紅色陰影）。只需移動種植錯誤的樹，正確樹木數量的列欄可保持不變。

在第 A 欄，兩樹交換後的田地均被標示為紅色，兩樹交換後才可保證標誌上的數字是正確的，第 C 欄也是同樣的道理。在這四個位置變更樹的高度後，所有列欄上的樹木高度都已遵守規則（每一列和每一欄均包含四個高度的四棵樹）並且標誌上的數字是正確的。



資訊科學上的意義



解決此任務需要資訊科學家常用的兩個基本能力：第一，是推理能力，從違反標示規則的列和欄，找到種錯位置的樹。第二，使用前述物件知識的部分訊息，重建物件的能力。這可以用於進行資料錯誤糾正。

錯誤檢測 error detection 是資訊科學中的常用技術，在可能遺失或損毀的傳送頻道中，用於確保資料傳輸的可靠性。錯誤檢測技術是可發現資料中有錯誤，**錯誤糾正 error correction** 則除了可檢測錯誤，並可把錯誤的資料重建為其原始和正確的格式。



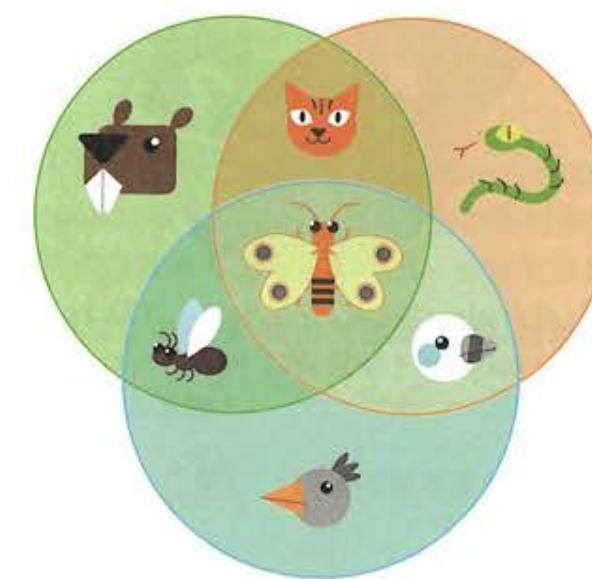
關鍵字

錯誤檢測與糾正



29. 園中動物

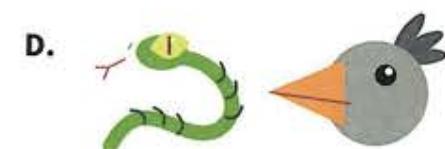
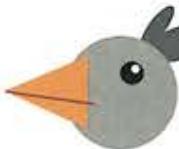
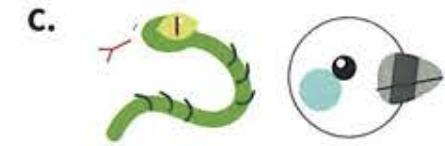
以下的圖表中有三個圓圈，每個圓圈都代表著具有不同特性的動物們：其中一個圓圈代表身上有條紋的動物，另一個圓圈代表超過兩隻腳的動物，最後一個圓圈代表有翅膀的動物。如果一隻動物同時擁有翅膀與條紋，牠就會被放置在那兩個圓圈的重疊處。



你的朋友選了兩種他喜歡的動物，他告訴你：

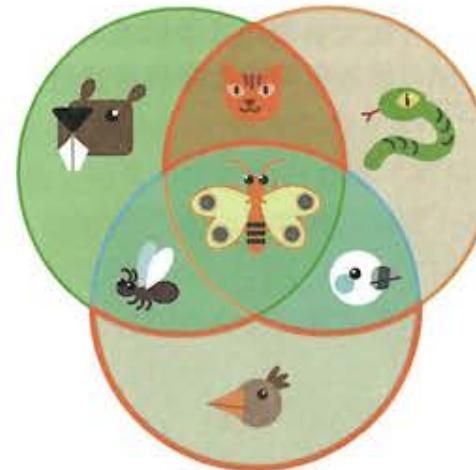
- 我喜歡有翅膀，但是不喜歡有條紋或很多隻腳的動物。
- 我也喜歡沒有翅膀，但是有條紋和很多腳的動物。

請選出你的朋友所喜歡的兩種動物，不用考慮順序。





正確答案是：B.



- 有翅膀，但是沒有條紋或很多隻腳的動物：如圖中位於最底下的鳥。



- 沒有翅膀，但是有條紋和很多腳的動物。



以下的動物都不符合，因為都不完全符合題目的兩個敘述：

- 海狸有很多腳（違反第一個敘述），但沒有條紋（違反第二個敘述）。
- 蛇身上有條紋（違反第一個敘述），但沒有很多腳（違反第二個敘述）。
- 蝴蝶有很多腳（違反第一個敘述），但有翅膀（違反第二個敘述）。
- 苍蠅有很多腳（違反第一個敘述），但有翅膀（違反第二個敘述）。
- 白鳥身上有條紋（違反第一個敘述），但有翅膀（違反第二個敘述）。



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。這個任務裡給定兩個邏輯條件敘述，學生必須根據這兩個邏輯條件敘述刪去不符合條件的動物以找出答案。

每個動物依照題目中的三個特性（條紋、多隻腳、翅膀）分類判斷，都有符合（是）或不符合（非）的判斷結果。文氏圖可用來表示集合的交集狀況，題目中把符合某個特性的動物視為一個集合（圓圈）。蝴蝶符合所有三個分類特性，因此被放在文氏圖的中心。

是 true 或非 false 這兩種數值，稱為 **布林值 boolean**，採用特殊的 **邏輯運算子** and、or、和 not，可以對這兩種數值進行運算，並判斷組合後的結果為是或非。這道題目以特殊的模式使用邏輯運算子，其中第一個敘述是以 or 來合併條紋及多隻腳的條件，經過 not 運算（不喜歡），再跟翅膀的條件以 and 合併。第二個敘述則對翅膀的條件經過 not 運算（沒有翅膀），再以兩個 and 運算來連接三個條件（沒有翅膀、有條紋、多隻腳）。



關鍵字

邏輯運算、布林值

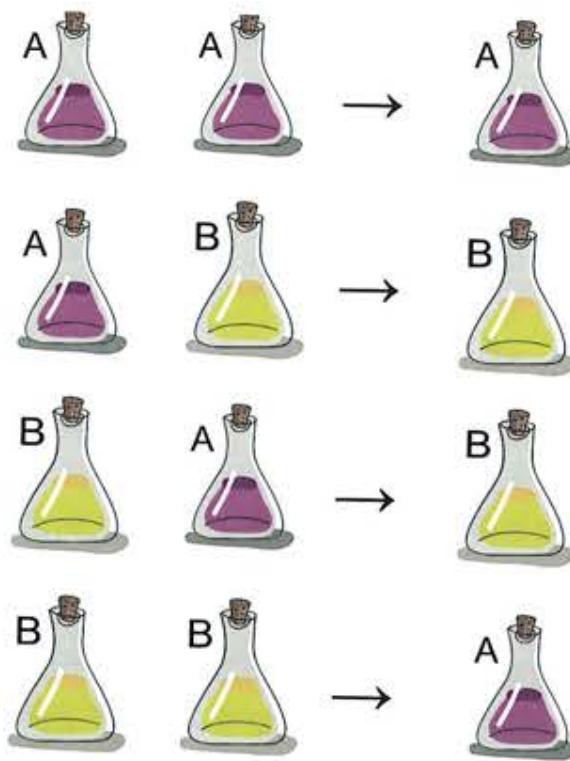


Benjamin/

Cadet/ 易 Junior/ 易 Senior/ 易

30. 魔法藥水

巫師有兩種不同魔法藥水：A 和 B。若他依序將兩瓶藥水倒入大釜鍋內，並且念出咒語，則大釜鍋內的藥水變化規則如下圖所示：



在巫師依序倒入了四瓶藥水後，大釜鍋內的藥水變成了 B 藥水 。請問下列何者是巫師倒入藥水的正確順序？

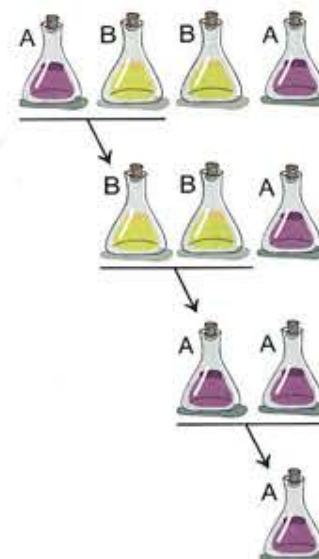
- A. A, B, B, A
- B. A, A, B, B
- C. A, B, B, B
- D. B, B, B, B



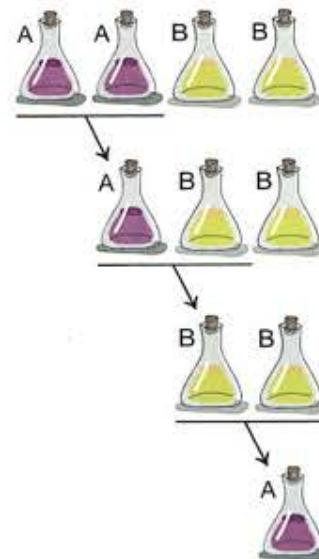
正確答案是：C

我們依照選項給予的順序混合藥水，以下為混合結果：

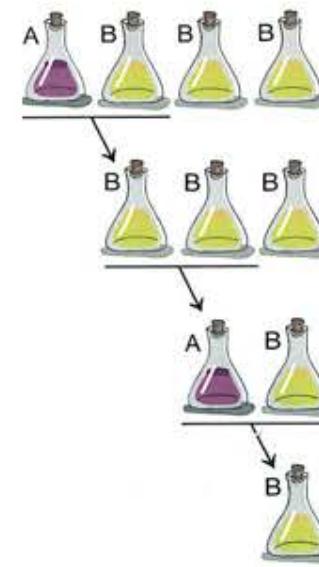
A 選項



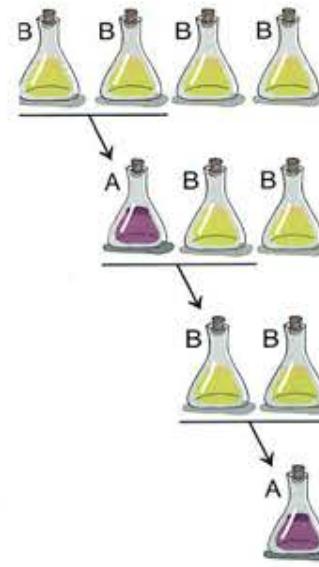
B 選項



C 選項



D 選項



資訊科學上的意義

電腦也是一種電子產品，因此我們會利用電流呈現訊息。當電流通過時，稱之為 **開啟 on**；反之，電流沒有通過的時候，稱之為 **關閉 off**。電腦科學家通常利用數字的 1 或 0 來代表這兩種狀態。我們稱之為 **二進位表示法**，一個資訊單位稱為一位元 bit。

我們可以組合不同的位元達到不同的運算結果，如同巫師將不同的藥水混合得到不同的結果一樣。其中一種運算方法稱為 **互斥或閘 exclusive OR, XOR**，也就是本題目使用的運算方式。其運算方法如下：

$$0 \text{ XOR } 0 = 0$$

$$0 \text{ XOR } 1 = 1$$

$$1 \text{ XOR } 0 = 1$$

$$1 \text{ XOR } 1 = 0$$

生活範例：在樓梯兩邊的盡頭，各有一個開關可以控制同一盞燈。兩個開關均可利用向上切將電燈打開或向下切將電燈關閉。當兩個開關均向上切時，電燈亮，當兩個開關均向下切時，電燈亦是亮的。但若一個開關向上切，另一個開關向下切，則電燈暗。

互斥或閘 exclusive OR gate 是一個電腦的 **邏輯閘**。當只有一個輸入為 1(真)的時候，結果才會輸出 1(真)。若兩個輸入相同，則輸出為 0(假)。互斥或運算可以用來做錯誤偵測與修正。此外，互斥或也有很多用途：

- 可以得知兩個位元是否相同或不同
- 藉由和 1 做互斥或 (\oplus) 的運算，得知二進位數字為奇數或偶數。
例如： $1011 \wedge 0001 = 1010$ ，1011 位元為 1 的有 3 個，和 1 進行互斥或運算後，得到 1010，其位元為 1 的有兩個，1 的總數減少，表示 1011 為奇數；
 $1110 \wedge 0001 = 1111$ ，1110 位元為 1 的有 3 個，和 1 進行互斥或運算後，變成 1111，位元為 1 的有 4 個，1 的總數增加，表示 1110 為偶數。
- 在邏輯電路板，任何邏輯運算均可由互斥或閘所組成。



關鍵字

二元運算、互斥或閘、邏輯閘

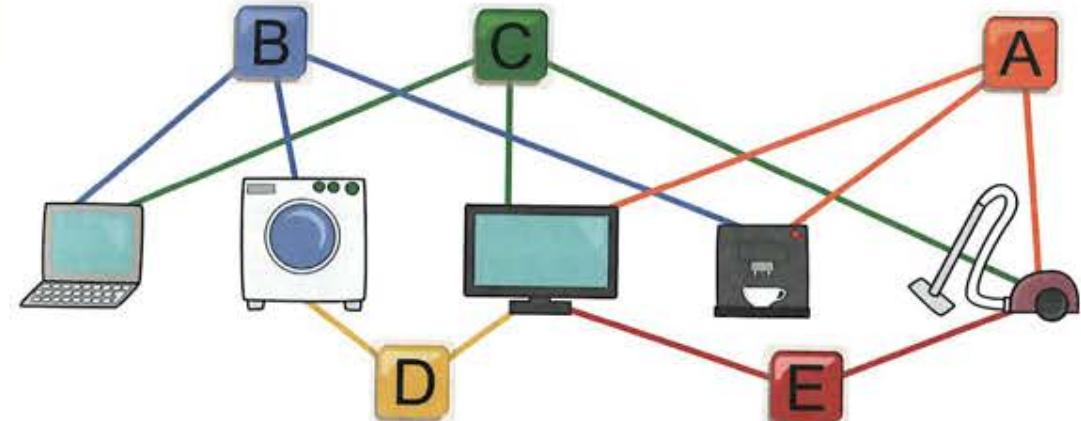


31. 家用設備

海狸太郎的家裡有五種家用設備：電腦、洗衣機、電視、咖啡機和吸塵器。而按鈕 A、B、C、D 和 E 分別控制這些設備的狀態；將開啟的設備關閉，或將關閉的設備開啟。

但是按鈕的設計很特別，每個按鈕同時連接多個設備；所以按下某按鈕後，該按鈕連接的所有設備的開、關狀態都會被改變：

- A 按鈕連接著電視、咖啡機和吸塵器
- B 按鈕連接著電腦、洗衣機和咖啡機
- C 按鈕連接著電腦、電視和吸塵器
- D 按鈕連接著洗衣機和電視
- E 按鈕連接著電視和吸塵器



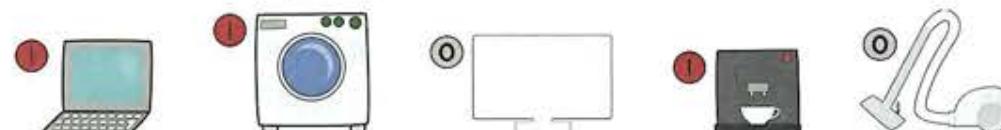
若所有電器原本都關閉著，下列何者按鈕順序，會使最後只有電視和咖啡機開啟著？

- A. (E,C,B,A) C. (D,A,E,C)
B. (C,B,A,D) D. (B,D,C,E)



正確答案是：D. B,D,C,E

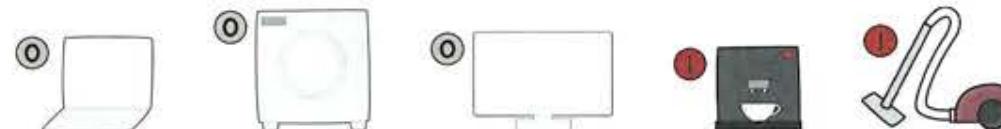
當你按下 B 按鈕的時候，電腦，洗衣機以及咖啡機會打開。



接著當你按下 D 按鈕，洗衣機被關閉，電視被打開。



接著按下 C 按鈕，電腦和電視被關閉，吸塵器被打開。



最後按下 E 按鈕，電視被打開，吸塵器被關閉。



此時只有電腦與咖啡機處理開啟狀態，其他設備則為關閉狀態。

要確認這個方式是正確而其他方式是錯誤的另一個簡單檢驗方式，就是個別查看按鈕與設備的對應關係，如果某個按鈕被按了奇數次，它所控制的設備就會處理開啟狀態，如果按了偶數次，則會讓設備處於關閉狀態，如果以設備的角度來看，當能夠控制它的按鈕被按了奇數次時，它會處理開啟狀態，若被按了偶數次，則這個設備會被關閉。

依照這個想法，分別說明以上的選項：

- (A) 第一組按法 (E, C, B, A) 我們以吸塵器為例子，其中的 (E, C, A) 按鈕是連接著它的。因此最後吸塵器會是開啟狀態，不符合問題的要求。
- (B) 第二組按法 (C, B, A, D) 當中只有 (B, A) 連接著咖啡機，也就是最後的結果會是關閉狀態同樣不符合問題的要求。
- (C) 最後再看第三組按法 (D, A, E, C) 我們以洗衣機為例，發現只有 (D) 按鈕連接著它，也就表示最後洗衣機會處理開啟狀態，同樣不符合問題要求。事實上這一組的按法除了電視以外，其他的設備都會被打開。

(D) 以同樣概念來觀察正確按法 (B, D, C, E)，我們來看最後各種設備的狀態：

- 電 腦：(B, C) 按鈕 => 關閉
- 洗衣機：(B, D) 按鈕 => 關閉
- 電 視：(D, C, E) 按鈕 => 開啟
- 咖啡機：(B) 按鈕 => 開啟
- 吸塵器：(C, E) 按鈕 => 關閉



資訊科學上的意義

我們可以把這個題目中所有的設備都想像成一組特定的數字，每次按下按鈕的時候數字就會改變，但是當下次再按下按鈕的時候，數字就會歸零，如此周而復始。這個想法實際上相當計算機使用的二進位系統，所有的設備都使用兩個數字來儲存跟處理：0 跟 1。舉例來說，計算機對二進位表示的數相加，如果該位的數字超過 1，就會把原本的位數歸零，然後把較高一位的數字加上 1，如下所示：

$$\begin{array}{rcl} A & = 0 & 1 \\ B & = 0 & 1 \end{array} \rightarrow A + B = 1 \ 0$$

$$\begin{array}{rcl} A & = 0 & 0 \\ B & = 1 & 1 \end{array} \rightarrow A + B = 1 \ 1$$

$$\begin{array}{rcl} A & = 0 & 1 \\ B & = 1 & 0 \end{array} \rightarrow A + B = 1 \ 1$$

這個題目當中，如果把數字 1 當成開啟狀態，0 當成關閉狀態，當我們每按一次按鈕時為這個設備加上 1，但只需要考慮其中該位元加 1 後的變化，而不需要考慮進位，就會產生了這樣的關係式：

$$\begin{array}{rcl} 0 + 1 & = 1 \\ 1 + 1 & = 0 \end{array}$$



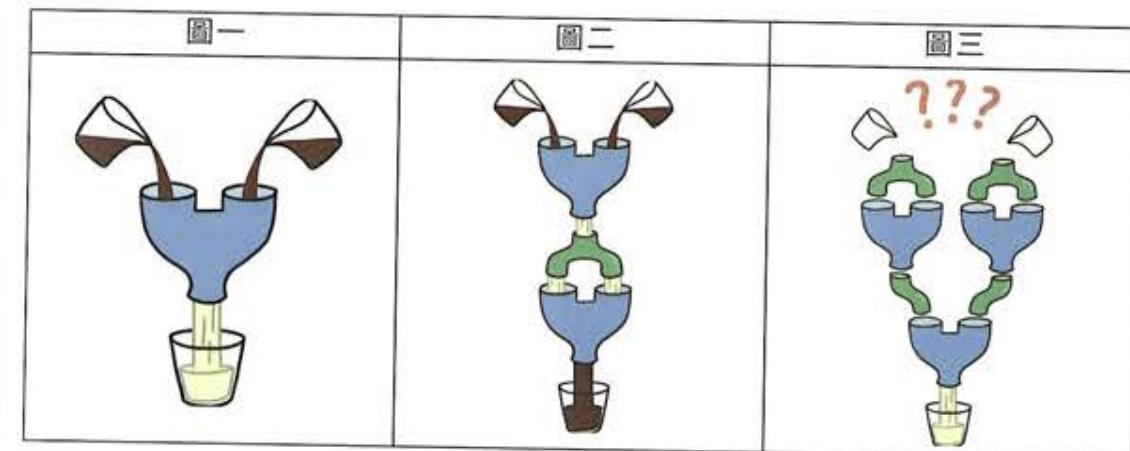
關鍵字

邏輯運算



32. 神奇飲料機

小海狸有一台神奇的機器，根據倒入的飲料會流出不同的飲料，請仔細觀察圖一和圖二中飲料的變化。

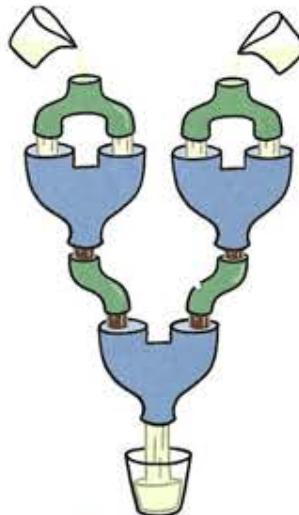


圖三中的機器頂端，應該加入那兩種飲料才能確保最後流出果汁牛奶？

- A. 果汁牛奶，果汁牛奶
- B. 果汁牛奶，巧克力牛奶
- C. 巧克力牛奶，果汁牛奶
- D. 巧克力牛奶，巧克力牛奶



正確答案是：A. 果汁牛奶，果汁牛奶



如左圖，為了最後得到果汁牛奶，第三座 Y 型管的二端皆需倒入巧克力牛奶。換句話說，巧克力牛奶應該來自頂部的兩個 Y 型管。

為了從頂部的兩座 Y 型管中各自獲取巧克力牛奶，應該在每座 Y 型管中至少要有一端倒入果汁牛奶。

依圖所示頂部的 2 座 Y 型管皆各自以綠色連接器連接，倒入的牛奶流入 Y 型管的二端，因此，為了最後可以流出果汁牛奶，頂部的 2 個綠色連接器應該倒入果汁牛奶，果汁牛奶。



資訊科學上的意義

接收一個或多個邏輯值並執行邏輯運算後輸出一個邏輯值的電子電路稱為 **邏輯電路**。而邏輯的基本運算為 **and**、**or**、和 **not**。**NAND** 邏輯閘運算的定義如下。

- $A \text{ NAND } B = \text{NOT}(A \text{ AND } B)$

有趣的是，NAND 可組答出 AND、OR 及 NOT 的運算效果，

- $\text{NOT } A = A \text{ NAND } A$
- $A \text{ AND } B = (A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B) \rightarrow \text{圖 2}$
- $A \text{ OR } B = (A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B) \rightarrow \text{圖 3}$

這些答式可以透過邏輯運算的轉換規則證明它們是相等的。在問題中，果汁牛奶以 0 來表示，巧克力牛奶以 1 來表示，Y 型管為 NAND 運算（有 0 必為 1）；圖 2 顯示僅使用 NAND 電路來表示 AND 運算，圖 3 顯示僅用 NAND 電路來表示 OR 運算。這些邏輯閘可組合出許多運算單元，例如加法器和減法器，邏輯電路設計就是在探討此類問題。



關鍵字

布林代數、邏輯電路、NAND、AND、OR

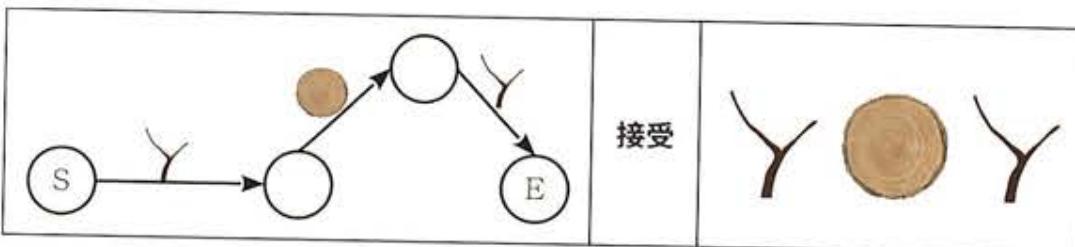
33. 密碼檢測

海狸們使用符號 和 所構成的密碼來保護他們的小屋，小屋設有密碼檢測器來檢測密碼是否合法。

密碼檢測器的檢測流程可以用圓圈和箭頭表示：

- 檢測器必定由圓圈 **S** 開始。
- 檢測器會由左至右每次處理一個輸入的符號。
- 如果正在處理的符號 與 所在圓圈向外的箭頭上符號相符，則根據箭頭方向移動到對應圓圈；否則停止所有檢測，並視該密碼不合法。
- 所有符號都處理完畢時會停止檢測。只有最後停在圓圈 **E** 的密碼是合法，其餘皆不合法。

舉例來說，下圖中的檢測器只能接受一種密碼：



小海狸可以選擇下列的符號組合產生密碼，組合內所有符號都必須使用。
請問哪幾個符號組合可以產生被下圖檢測器接受的密碼？

- | | |
|-----------|-----------|
| A. 4X 8X | B. 2X 7X |
| C. 5X 8X | D. 5X 7X |



正確答案是：A. 4X 8X

選項 A 是對的。觀察題目檢測器的所有流程會發現，合法密碼 的數量必須是 的兩倍。

選項 B 是錯的。因為它包含 2 個 和 7 個 .

選項 C 是錯的。因為它包含 5 個 和 8 個 .

選項 D 是錯的。因為它包含 5 個 和 7 個 .



資訊科學上的意義

本題的密碼檢測器可以用 **確定有限狀態機 Deterministic Finite-State Machine, DFSM** 表示。有限狀態機是一種數學計算模型，圓圈表示狀態，箭頭表示轉換方向。每個時間會位在某個狀態，之後根據輸入的訊號或動作，轉換到下個狀態，這個過程被稱為狀態轉移。

有限狀態機由狀態列表、初始狀態，以及狀態轉移規則構成。

許多現代社會設備都具有狀態機行為，即根據所觸發事件決定執行動作。舉例來說：

- 自動販賣機：當投入硬幣，即更新金額。當投幣金額大於等於貨品金額，即供應產品並找錢。
- 紅綠燈：紅綠燈依前一燈號改變燈號及亮燈時間。
- 順序密碼鎖：要求需要按照正確的順序輸入數字才能夠解鎖。



關鍵字

有限狀態機、DFSM

34. 戲劇演出

有一個童話劇分成兩幕，兩幕演出之間有中場休息。劇中的角色有國王、公主、惡龍及王子。

在同一幕中，如右圖由上至下分別顯示各個角色出場與退場的順序。

標示綠色箭頭 (往右→)：表示角色出場。

標示紅色箭頭 (往左←)：表示角色退場。



請問下列哪個選項是錯誤的？

- A. 公主與王子曾一起出現在舞台上。
- B. 國王與惡龍曾一起出現在舞台上。
- C. 王子在中場休息時間後，才出現在舞台上。
- D. 王子與惡龍曾一起出現在舞台上。



正確答案是：B. 國王與惡龍曾一起出現在舞台上。

A. 選項的敘述是正確的。

因為在中場休息時間後，王子進入舞台中，惡龍離開舞台，接著公主再次進入到舞台中，此時公主與王子一起出現在舞台。

B. 選項的敘述是錯誤的。

因為在惡龍進入舞台前，國王已經離開舞台了，而且不再進入舞台。

C. 選項的敘述是正確的。

D. 選項的敘述是正確的。

因為在中場休息時間後，惡龍再次進入舞台中，接著王子也進入舞台。



資訊科學上的意義

在資訊科學上，我們通常以 **序列** 的形式來表達一段過程（按照一定順序所發生的一連串事件）。此外，將資訊、資料或處理方式以圖形化的圖像表示，可取代複雜的文字敘述方式，便於轉換成電腦能有系統處理的表示法及計算過程。

例如本題中每個角色相當於電腦程式中的一筆資料，箭頭方向表示出場或退場。出場及退場的順序相當於一連串的指令，對這些資料的狀態進行修改。能理解並掌握一連串的指令執行對這些資料狀態的改變，才能確保電腦程式達到預期的效果，因此是程式設計重要的基礎能力。



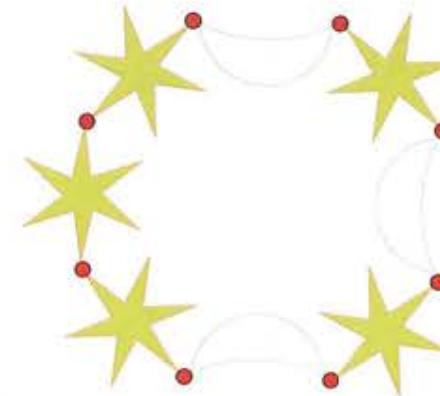
關鍵字

序列執行、變數



35. 星星與月亮

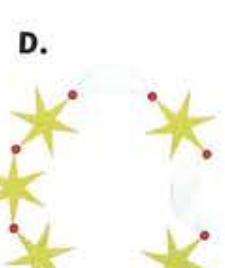
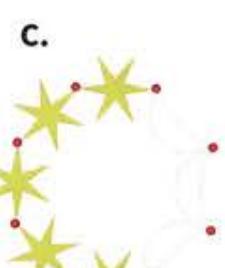
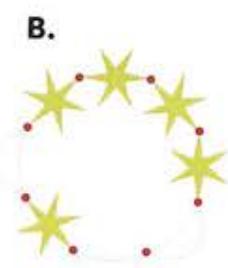
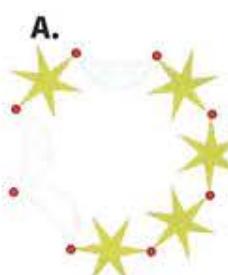
瑪麗想要一條如下圖所示的手鍊，因此她提供以下製作指令給約翰。



- 取一個星星 (★) 和一個月亮 () 並將它們連接在一起。
- 再重複上一個步驟兩次。
- 將三個做好的零件連接成一串。
- 把這串手鍊半成品的一端再接上兩個星星，最後將兩端連接在一起，就完成這條手鍊。

不過如果約翰沒有手鍊的圖片，即使他完全按照指令操作，仍然可能做出一條不一樣的手鍊。

請問下面四條手鍊，哪一條不會是依照瑪麗的指令而製作出的手鍊？

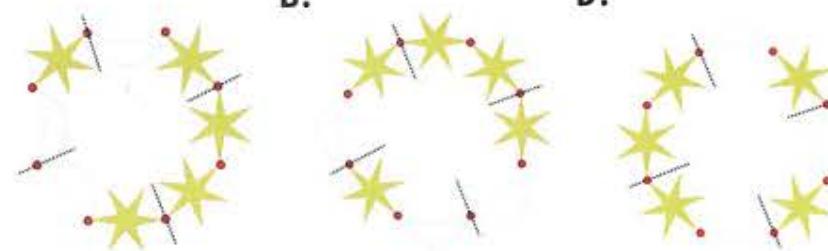




正確答案是：C

下圖顯示另外三個答案都能夠被分割成三組星星跟月亮的組合以及一組有兩個星星的組合。按照指令中第一個步驟，手鍊中每個月亮一定會有一個星星在旁邊，所以手鍊 C 中連續有三個月亮是不可能發生的。

A. B. D.



資訊科學上的意義

當程式設計師向電腦發出 **指令** 時，必須準確指定電腦所要執行的事，否則執行出來的結果可能不是他們所要的。例如，在瑪麗提供的製作指令中，沒有準確地陳述要如何將三組星星跟月亮的組合零件連接在一起。因此有多種可能的執行方式跟結果 (A、B 及 D)。必須指定三個零件接在一起時，月亮只能連接到星星，就能確保做出瑪麗想要的手鍊。

如果有一台能控制手鍊製造機的電腦，瑪麗提供的指令一但不夠明確，電腦通常會停下來不執行，並回應「我不確定你的意思」。不過如果每個步驟的細節都要全部寫清楚，在一樣的步驟需要重複好幾次的情況下，提供指令的過程可能變得很乏味且產生冗長的程式。因此，**電腦程式** 提供一些讓指令更簡潔的方式，例如在瑪麗提供的指令中，可以直接說要重複某個動作幾次，而不需要一次又一次地複製執行這些動作的指令，這樣的指令可以用特定的程式語言讓電腦理解。



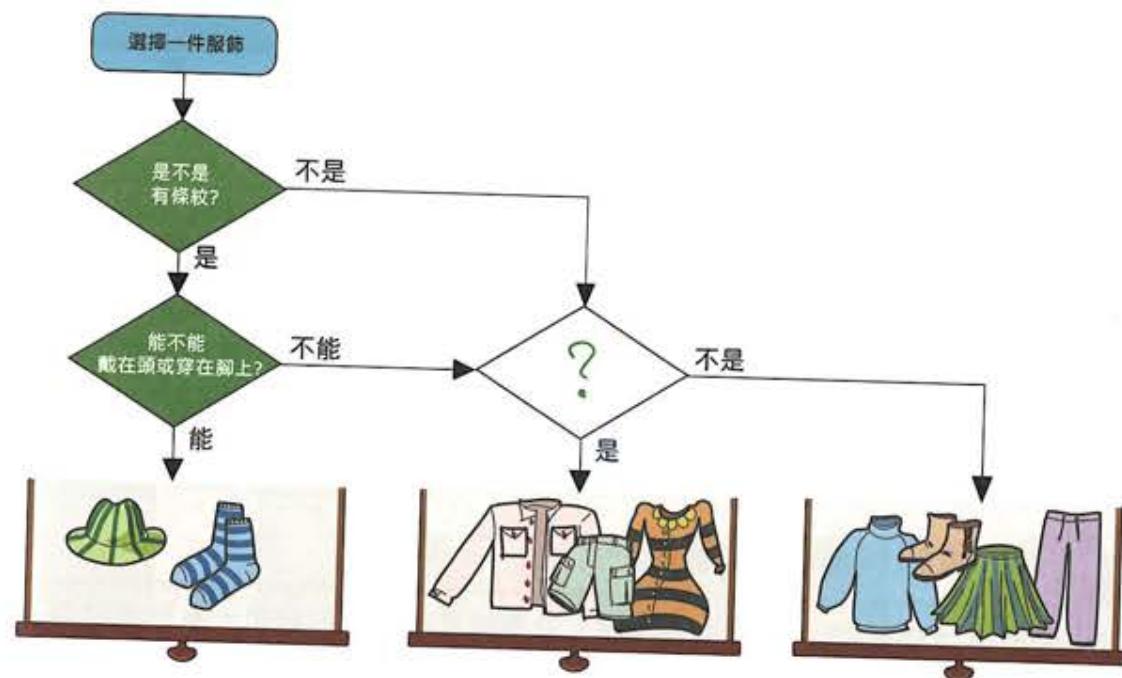
關鍵字

電腦程式、迴圈、指令

36. 物件分類

如下圖所示，寶拉有 3 個抽屜；她利用流程圖來幫助她決定衣物該放到哪個抽屜中。流程圖由最上方的選擇方框開始，提示寶拉選擇一件服飾，接著順著箭頭往前；若遇到菱形問題框，寶拉會順著該問題的答案所指示的箭頭繼續往前，直到決定衣物所屬的抽屜，就可以把這件衣物放進那個抽屜。

接著再回到流程圖最上方，再拿另一件衣物重複流程圖的過程，直到她把所有的衣物都放進抽屜為止。



如果上圖的衣物都按照著流程圖收納至抽屜中，請問下列何者是流程圖當中 (◇?) 代表的問題？

- A. 是否為長袖？
- B. 是否有鈕扣？
- C. 是否有拉鍊？
- D. 是否有口袋？



正確答案是：B. 是否有鈕扣？

如果選擇 答案 (A) 是否為長袖？，那麼中間抽屜的短褲跟右邊抽屜的藍色上衣是不合理的放法。



如果選擇 答案 (C) 是否有拉鍊？，那中間抽屜的那件白色外套跟洋裝都是不合理的放法。



如果選擇 答案 (D) 是否有口袋？，那麼中間抽屜的那件洋裝是不合理的放法。



如果選擇 答案 (B) 是否有鈕扣？，可以透過同時比較中間與右側抽屜的衣物來檢驗這個描述發現是正確的，每個中間抽屜的衣物都有鈕扣，而每個右側抽屜的衣物都沒有。



資訊科學上的意義

寶拉使用的 流程圖 也可以稱為 決策樹。決策樹可以引導你藉由一系列的問題，最後得到某個判斷的結果，這樣的流程單純也因此運用在許多決策或問題解決過程。在編寫程式的過程當中也經常使用流程圖，並且運用三種基本的處理結構，分別是 **循序結構、選擇結構 和 重複結構**。

循序結構會用圓角矩形描繪，選擇結構（或稱決策結構）會使用菱形，而重複結構則會用箭頭指回到之前流程圖的特定位置使其重複執行。

透過流程圖來規劃程式指令的執行過程，編寫程式語言的團隊就能藉此達成共識，並且在編寫程式甚至是在編寫之前就發現並修正邏輯上的錯誤。

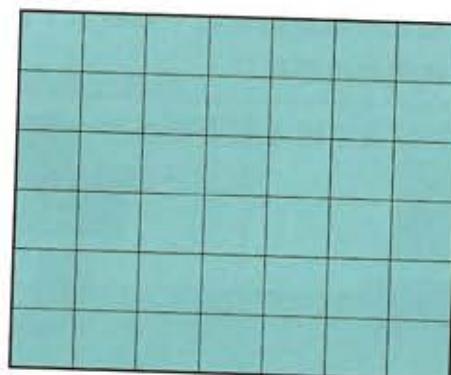


關鍵字

流程圖、決策樹、循序、選擇、重複



37. 不撞牆



掃地機器人正在一間由 6×7 塊正方形瓷磚組成的房間內移動，房間四周為牆壁。掃地機器人每次前進，會朝著其中一面牆的方向，由瓷磚中央移動到下一塊瓷磚中央。

掃地機器人一開機後，就會隨機面朝其中一面牆，接著會執行根據以下指令編寫成的程式：

前進： 依據它現在的方向，移動到下一塊瓷磚。	左轉： 在所在瓷磚上逆時針旋轉 90 度。	右轉： 在所在瓷磚上順時針旋轉 90 度。

現在掃地機器人已內建以下程式，待開機選定隨機牆面之後就會執行：
前進 → 左轉 → 前進 → 右轉 → 前進

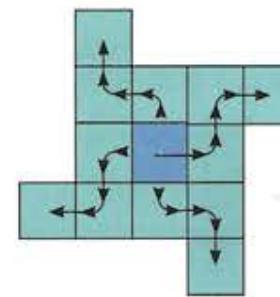
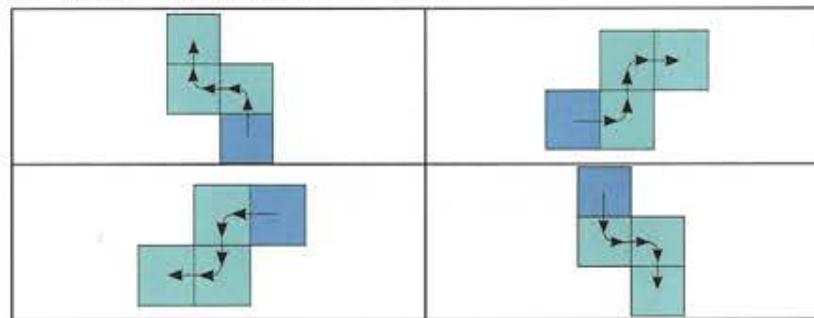
若把掃地機器人隨意放在某塊瓷磚上，執行程式的過程中可能會撞牆。

請問房間中有幾塊瓷磚，可以使掃地機器人不論起始方向為何，執行上面的程式都不會撞牆？（範圍 $[0, 42]$ 的整數）

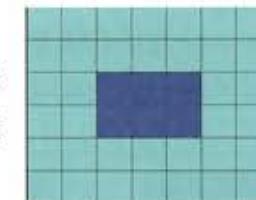


正確答案是：6塊

掃地機器人執行以下程式：前進 → 左轉 → 前進 → 右轉 → 前進。若掃地機器人面朝其中一個方向，執行程式的軌跡過程如下圖所示，但須考量四個方向的移動都可能發生，從哪一塊瓷磚開始也都可以。



若從同一塊瓷磚開始，四個方向的軌跡範圍如左圖所示，如果掃地機器人起始位置在直接接觸牆壁的瓷磚上（移動一步以後就會碰到牆壁的瓷磚，稱為邊界瓷磚），或從相鄰邊界瓷磚的瓷磚上開始，則在執行程式時，掃地機器人（至少）有一個方向（面向最近的牆壁）會碰到牆壁。



如右圖深色區塊所示的 6 個初始位置才能確保掃地機器人無論從哪個方向開始運作程式都不會撞牆，簡而言之，也就是這些在 **中央** 的瓷磚很安全，因為它們離牆壁足夠遠。



資訊科學上的意義

程式設計師在寫程式時必須事先考慮很多狀況，以避免程式執行中發生非預期的錯誤（如本問題中機器人發生撞牆），而導致電腦系統崩潰 crash。因此他們必須檢查並定義程式執行的初始條件，或預先寫好發生錯誤情況的處理。

換言之，程式設計師會進行所謂的軟體測試。軟體測試是一種用來鑑定軟體的正確性、完整性、安全性和品質的過程，有時是由開發人員測試，有時由專門的測試人員測試。



關鍵字

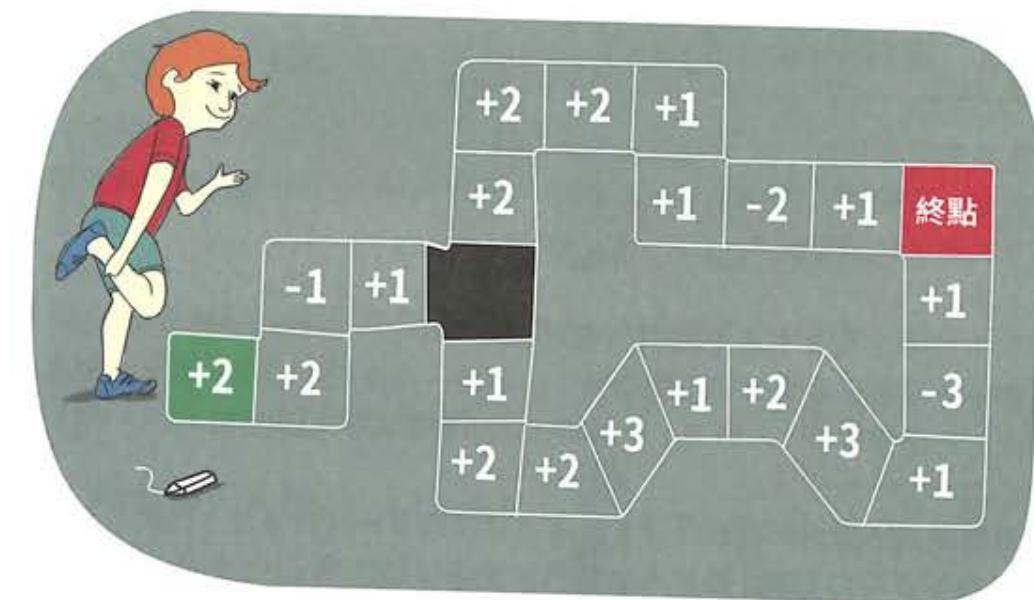
軟體測試



38. 跳格子

小海狸在院子裡畫了一個跳格子遊戲。

小海狸從綠色格子開始遊戲，他依格子裡的數字向前或向後跳動，例如： $+2$ 表示向前跳動兩次， -1 表示向後跳動一次。



小海狸需要在黑色格子寫上什麼，才能使他到達終點？

- A. 左+1
- B. 左+2
- C. 右+1
- D. 右+3



正確答案是：C. 右+1

在 A、B 和 D 的情況下，小海狸無法達到終點：他會不斷來回跳動，永不停止。



資訊科學上的意義

演算法是完成任務的一連串的指令或一組規則。指令可以用多種方式表示，包括自然語言、虛擬碼、流程圖、程式語言等。由於演算法是一連串的明確步驟，因此指令的順序是很重要的。

演算法中使用三種基本控制結構來描述指令：循序、選擇和重複（或迴圈）。在此任務中，重複是最重要的控制結構：小海狸一直在重複相同的基本指令直到完成為止。但是答案 A、B 和 D，小海狸碰巧都會陷入了無窮迴圈。

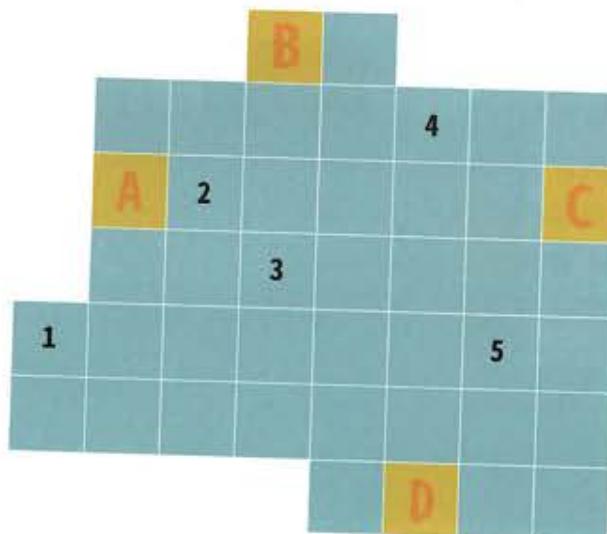
無窮迴圈是指一個迴圈重複無限次，因為它沒有到達終止條件，因此程式結束的條件永遠不會成立（在這題中終止條件是走到終點的格子）。儘管程式設計師可能刻意使用無窮迴圈，但是這是初學程式設計的人常會犯的錯誤。



關鍵字

無窮迴圈、測試

39. 機器人的行走



小海狸使用自己寫好的程式操控 4 個機器人在右圖的格子中移動，圖中 A、B、C、D 代表機器人充電站。

程式可使用下列三種指令組合而成：

- 走一步 – 走到面前的下一個格子



- 往左轉 – 逆時針旋轉 90 度，但不會向前移動



- 往右轉 – 順時針旋轉 90 度，但不會向前移動

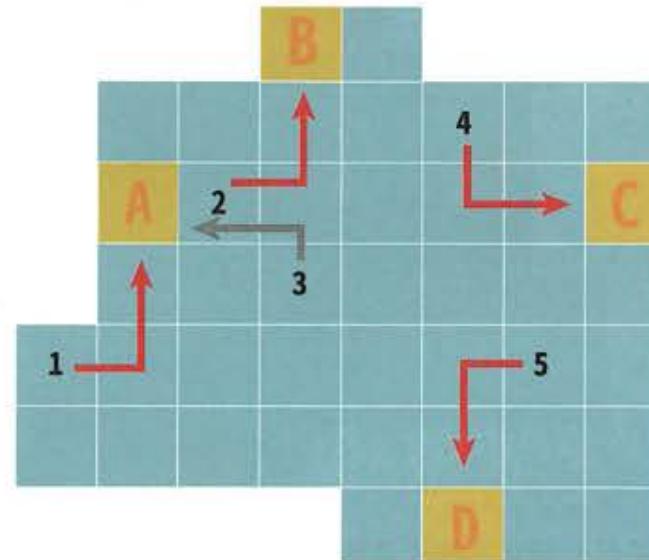
小海狸寫了一個程式：「走一步，往左轉，走一步，走一步」，並將此程式下載到 4 個機器人上。為了讓每個機器人都可以不碰撞彼此地到達某個充電站，小海狸在啟動程式前必須將所有機器人擺放在適當的位置，並調整每個機器人的方向。請注意，程式啟動後，所有機器人會同時開始執行程式。

請問小海狸應該不會在哪一個格子放置機器人？

- | | |
|---------|---------|
| A. 格子 1 | D. 格子 4 |
| B. 格子 2 | E. 格子 5 |
| C. 格子 3 | |



正確答案是：C. 格子 3



執行「走一步，往左轉，走一步，走一步」的程式，機器人的行走路徑會呈現一個左右顛倒的 L 型，並由 L 型較短的邊出發，因此我們可以嘗試將該 L 型填進網格內（上圖箭頭部分）。

要到達充電站 B，只可能由格子 2 出發，如果由格子 4 向上出發，則必定會撞到周圍牆壁。

機器人如果放置在格子 5，根據一開始機器人的方向不同，可能到達充電站 C 或充電站 D。此時我們可以使用刪去法：由格子 4 或格子 5 出發的機器人都可能到達充電站 C，但只有由格子 5 出發的機器人有機會到達充電站 D。因此，放置於格子 5 的機器人必定前往充電站 D，而放置於格子 4 的機器人則前往充電站 C。

最後，機器人在行走時需要注意不能彼此碰撞。對於目的地是充電站 A 的機器人而言，雖然可由格子 1 或格子 3 出發，但如果機器人由格子 3 出發，則將會在踏出第一步時，與由格子 2 出發走向充電站 B 的機器人碰撞，因此充電站 A 只可能由格子 1 出發。



資訊科學上的意義

在撰寫程式時，程式設計人員不僅必須確保程式執行結果正確，同時必須注意程式執行中是否可能出現程式無法正常執行的狀況，這種狀況可能來自於程式執行期間記憶體不足或與另一個程式爭用共享資源。因此，程式設計人員必須對程式設定適當的初始條件，或是嘗試模擬與測試程式執行時的各種狀況，以避免非預期的錯誤造成程式無法正常執行。例如美國航空暨太空總署 NASA 從西元 2004 年開始，在發送指令到火星上的火星漫遊者號前，都會先測試該指令是否在任何狀況下都能成功執行。

另外，在同一臺電腦上若有多個處理器，可以讓多個程式能夠同時地工作，稱為並行程式 **concurrent programming**。但當不同程式在同一時間執行不同動作（移動不同機器人位置），對程式設計人員而言可能會很難管理每個資料的當下狀態。所以必須設定好初始的位置跟方向來避免發生碰撞，或是加上併行程式執行的同步控制機制 **concurrency control** 來避免非預期的錯誤情況發生。



關鍵字

軟體測試、並行計算



40. 神奇數字機 I

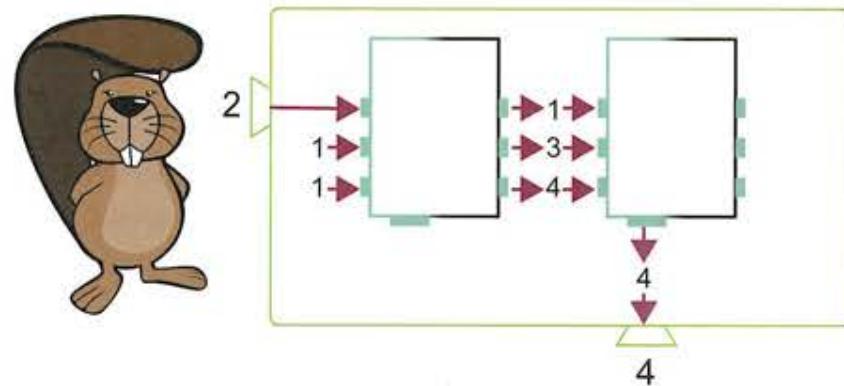
海狸們創造了一個神奇數字機，該機器可以在接收到數字之後，輸出另一個神奇數字。神奇數字機由數個元件組成，每個元件都依以下的規則運作：

接收三個數字，然後依照以下的規則處理每個數字：

- 如果第一個數字是 1，就輸出第三個數字
- 否則：
 - ◎ 把第一個數字減 1，成為新的第一個數字
 - ◎ 把第二個數字加 2，成為新的第二個數字
 - ◎ 把第三個數字加上新的第二個數字，成為新的第三個數字
 - ◎ 依序把這新的三個數字傳給旁邊的元件

當神奇數字機接收到一個數字後，該數字即是第一個零件接收的第一個數字，而這個元件的第二個和第三個數字都設定為 1。若神奇數字機還沒有得到輸出的數字，就會反覆將三個數字以上面的規則傳給旁邊的元件，直到得出輸出的數字為止。

以下圖為例，若使用兩個元件，在神奇數字機接收到數字 2 時，會輸出 4。



若神奇數字機接收到數字 4，那它會輸出什麼數字呢？

- A. 7 B. 10 C. 16 D. 64



41. 神奇數字機 II

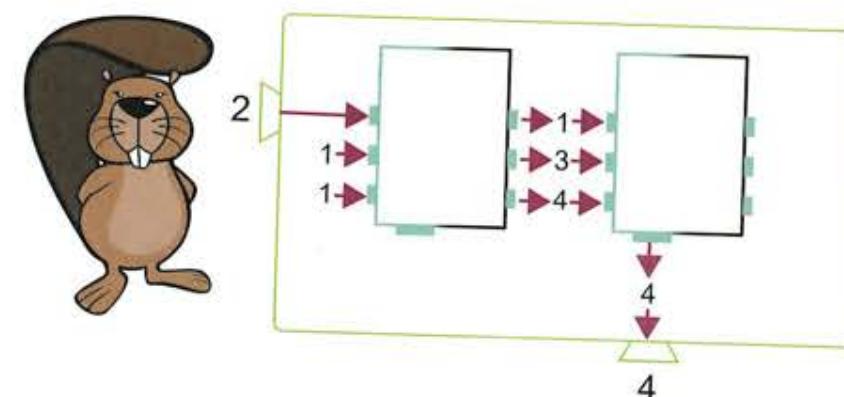
海狸們創造了一個神奇數字機，該機器可以在接收到數字之後，輸出另一個神奇數字。神奇數字機由數個元件組成，每個元件都依以下的規則運作：

接收三個數字，然後依照以下的規則處理每個數字：

- 如果第一個數字是 1，就輸出第三個數字
- 否則：
 - ◎ 把第一個數字減 1，成為新的第一個數字
 - ◎ 把第二個數字加 2，成為新的第二個數字
 - ◎ 把第三個數字加上新的第二個數字，成為新的第三個數字
 - ◎ 依序把這新的三個數字傳給旁邊的元件

當神奇數字機接收到一個數字後，該數字即是第一個零件接收的第一個數字，而這個元件的第二個和第三個數字都設定為 1。若神奇數字機還沒有得到輸出的數字，就會反覆將三個數字以上面的規則傳給旁邊的元件，直到得出輸出的數字為止。

以下圖為例，若使用兩個元件，在神奇數字機接收到數字 2 時，會輸出 4。



下列何者不會是神奇數字機所輸出的數字呢？

- A. 1 B. 50 C. 100 D. 400



40. 正確答案是：C. 16

第一組元件輸入的三個數字是(4, 1, 1)，輸出(3, 3, 4)。

第二組元件輸入的是(3, 3, 4)，輸出(2, 5, 9)。

第三組元件輸入的是(2, 5, 9)，輸出(1, 7, 16)。

第四組元件輸入的三個數字是(1, 7, 16)，輸出16作為最後出輸出結果。

答案A不正確：最後輸出的結果是第三個數字，不是第二個。

答案B不正確：從1開始，不會有增加3的結果。

答案C正確。

答案D不正確：從1開始，不會有增加4倍的結果。



41. 正確答案是：B. 50

如果輸入1，神奇數字機輸出1。

如果輸入2，則輸出 $1+3=4$ 。

如果輸入3，則輸出 $1+3+5=9$ 。

如果輸入4，則輸出 $1+3+5+7=16\cdots$

我們繼續觀察會發現神奇數字機輸出的結果，都是輸入數字的完全平方數，

選項A、C及D都是完全平方數，所以這一題的答案是選項B。

補充說明：

遞迴的特性從數學歸納法可以說明：當輸入 $n=1$ ，輸出 $1 = 1^2$ 為真

假設 $n=k$ ，輸出 $2k + 1 + k^2 = (k + 1)^2$ 為真；新的第三個數字等於上一元件的第二個數字(第 $k+1$ 個奇數)加上第三個數字，又是一個完全平方數亦為真。



資訊科學上的意義

我們常將一個較大問題拆解為相同模式的子問題，即所謂的分而治之 divide and conquer，這也是設計許多重要演算法的關鍵。在資訊科學中，遞迴就是其中一種，解決方法為將一個大問題拆解為多個同樣的問題，但處理範圍較小。函式中會呼叫自己，用自己的程式碼解決一部份的答案，再整合結果得到全體的答案。這個方法可以應用於多種類型的問題，遞迴是資訊科學的核心概念之一。遞迴的威力在於可利用有限的執行敘述來拆解問題並重複執行，這使得即使敘述中沒有任何明確的重複指令，也能隨著資料的狀況繼續，因此遞迴的結束特別重要，它通常是一個可明確判斷的條件檢查，若達到結束條件，函式不再呼叫本身也就不會再增加新的遞迴計算。否則會造成電腦無限的計算而無法中止，且將計算資源耗盡而當機。



關鍵字

遞迴



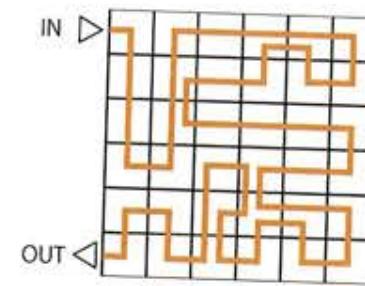
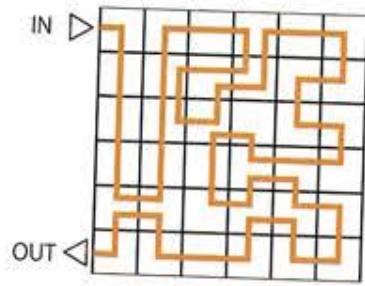
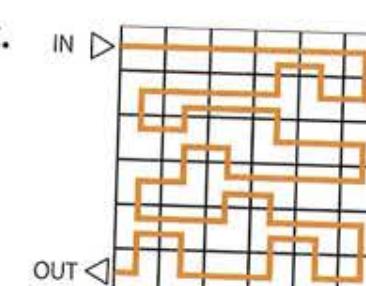
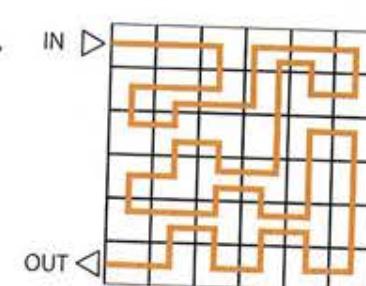
42. 發光面板

Bebras 王國的國王為了慶祝國慶日，決定使用 36 塊發光面板來創作一個巨大的藝術品，放在廣場上展示，如下圖一所示。踩踏每塊面板時，可切換面板開關，將未點亮的面板點亮或將已點亮的面板轉換為未點亮，國王從圖一中的 IN 面板出發，從 OUT 面板結束。他可以前進到鄰近的上、下、左、右面板（非對角線），創造多種發光面板圖案，同一面板上可以踩踏多次。

未點亮面板圖示	點亮面板圖示	圖一
		<p>IN ▷</p> <p>OUT ◁</p>

國王製作了上面圖一的圖案，但他現在想點亮所有面板重新開始。

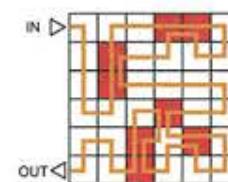
請選擇以下哪條路線，可以點亮所有面板？

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

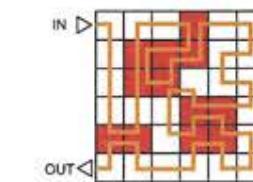


正確答案是：C.

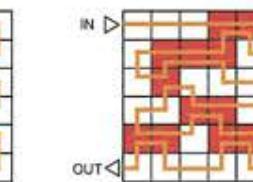
答案 (A)



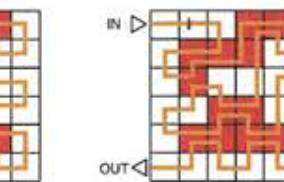
答案 (B)



答案 (C)



答案 (D)



答案 (C) 路徑是踩踏目前已發光的面板兩次，而剩下的未點亮的面板則踩踏一次，最後所經過的面板均會被點亮發光。在檢查所有面板後，把被踩踏兩次的面板著色為紅色，將會發現答案 (C) 紅色區塊的分佈與圖一所示相同。



資訊科學上的意義

像素 pixel 是電腦顯示影像的基本單位，一個像素所能表達的不同顏色數，取決於每個像素以多少位元來表示，例如，本題的每塊發光面板可視為一個像素，每個像素皆以 1 位元表示，0 代表黑色（面板未點亮），1 則代表白色（面板被點亮發光）。若有一影像中每個像素以 8 位元來表示，則該影像可顯示 $2^8 = 256$ 種面板圖案。

具有 **追蹤程式碼** 的能力，在程式設計中很重要。本題可透過一步一步追蹤發光面板的變化方式（即程式處理的數據）來找出答案。但是，追蹤每一步執行後的內容需要花費很多時間，因此透過了解程式指令執行（在此題的發光面板被走訪次數）造成的變化模式，可有效縮小答案範圍。



關鍵字

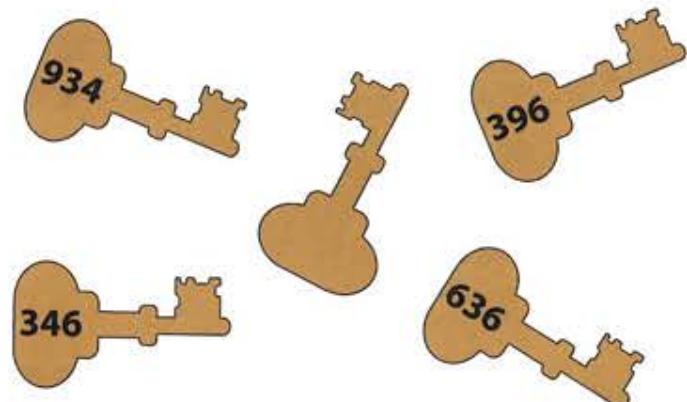
像素、程式碼追蹤

43. 警衛雅克

雅克是公寓大樓的警衛。大樓有五個房間，每個房間都住著一隻海狸，當海狸去上班時會把鑰匙交給雅克。為了不混淆鑰匙，每間房間都有一個專用箱子來保存鑰匙，而且箱子上有房間住戶名字的前三個字母。



為了安全起見，鑰匙上沒有寫住戶名字，但是雅克把住戶名字轉換成三位數字標示在鑰匙上。對於所有鑰匙，同一個字母始終對應同一個數字。有一天所有的箱子掉下來，掉出所有鑰匙，而且其中一把鑰匙的標籤不見了。



請問不見的標籤上應該標示什麼數字？（範圍 [346~934] 的三位整數）



正確答案是：496

箱子 BEB 是唯一一個第一和第三個的英文字母相同者，鑰匙中第一和第三個的數字相同者只有 636，因此這個箱子符合的鑰匙為 636。由此可知英文字母 B 對應的數字為 6，英文字母 E 對應的數字為 3。

箱子 AER 是唯一結尾與其它箱子不同者，鑰匙中第三個數字與其它不同者為 934，因此英文字母 R 對應的數字為 4。根據字母 A 對應到數字 9，現在我們可以將 EAB 配對到 396、ERB 配對到 346。因此剩下的箱子為 RAB，不見的標籤為 496。



資訊科學上的意義

加密 和 **解密** 是電腦科學部分中稱為 **密碼學** 的基本概念。在此任務中，加密過程基於稱作 **瓦茨亞亞那密碼 Vatsyayana Cipher** 的字母替代密碼，但瓦茨亞亞那密碼是將原字母對應到另一字母，這個想法來自公元 4 世紀的**印度文字**。

在此任務中，**瓦茨亞亞那密碼**則用於將特定字母對應到特定數字，且某個字母和某個數字只能成對使用，不允許重複使用。為了加密訊息，每個字母都用成對的數字代替。一旦收到訊息，解密過程就會以配對的字母替換數字。此方法在加密和解密過程，對整個訊息採用固定的替換方式。如今，這是一種不安全的加密方法，因為一旦被取得正確的字母和數字配對，便可以用來解密訊息中的其他內容。



關鍵字

加密、解密、安全性、密碼學

44. 海狸美容院

有四位海狸造型師在美容院工作：琳娜、阿尼、祖里和皮亞。每個造型師都可以執行三種服務，每種服務需要不同的時間來完成。

服務項目	毛皮柔順	牙齒銳化	尾巴上油
所需時間	5 分鐘	8 分鐘	15 分鐘

開店後這四位造型師同時開始服務四位海狸，每位海狸客戶要求的服務都只有一項，如圖所示。當任一位造型師完成他的工作，就會立刻服務下一個排隊等候的海狸。



奧比正在等待造型師幫他擦亮尾巴。請問哪個造型師將服務奧比？

- A. 琳娜
- B. 阿尼
- C. 祖里
- D. 皮亞



正確答案是：C. 祖里

因為所有造型師同時開始工作。

琳娜和皮亞正進行的服務項目都是銳化牙齒，所以會在 8 分鐘後完成。

阿尼尾巴上油的工作將在 15 分鐘後完成。

祖里柔順毛皮的工作，將在 5 分鐘後完成。由於祖里比其他人先完成工作，所以祖里將服務奧比。



資訊科學上的意義

海狸美容院有四位造型師 (琳娜、阿尼、祖里和皮亞)，可在同一時間服務四位不同的海狸。在現代電腦中有數個 **處理器** 可以同時進行工作，我們會說這樣的電腦是可以 **多工處理**。

而電腦的工作方式與海狸美容院的情況非常相似，當電腦的一個處理器完成執行中的 **任務**，它會收到一個新的任務，因此可以有多個任務同時進行，我們稱這種能力為平行化處理。

將工作分配給電腦中不同處理器的方法稱為 **排程**，這種將任務分配到電腦不同處理器並平行執行這些任務的能力是加快電腦計算速度的方式之一，有效運用平行化處理，可以更快速地解決複雜困難的問題。



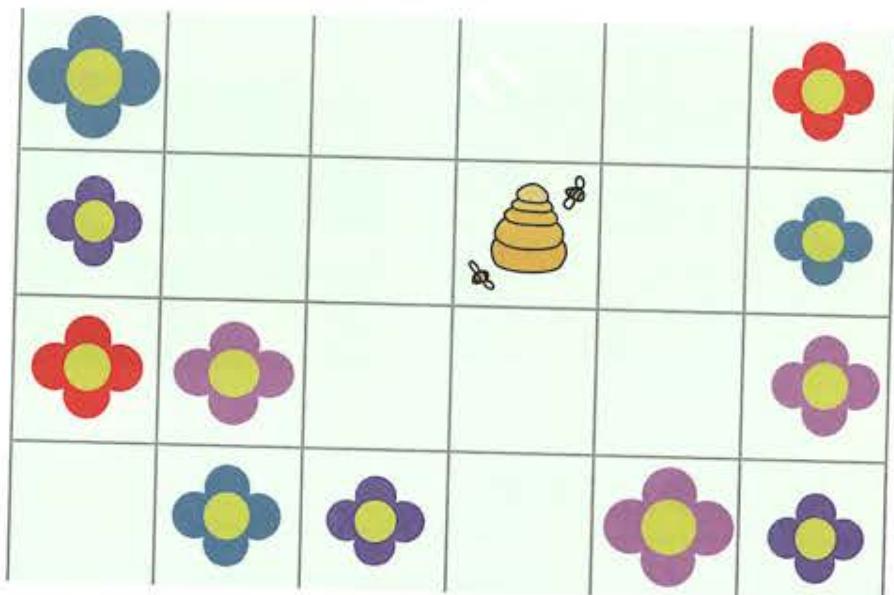
關鍵字

處理器、任務、多工處理、平行性、排程

45. 可以飛到哪？

蜜蜂只能以下圖中的上下左右四個方向飛往另一個格子，如果要飛到斜對角的格子（例如右上角，就要向上再向右各飛一格，或是向右再向上各飛一格），要飛行 2 格的距離。

當蜜蜂從巢中出發採花蜜，從蜂巢起算，最多可以飛行 3 格的距離就要飛回來。



請問圖中的 11 朵花，最多有幾朵花可被蜜蜂們採到？

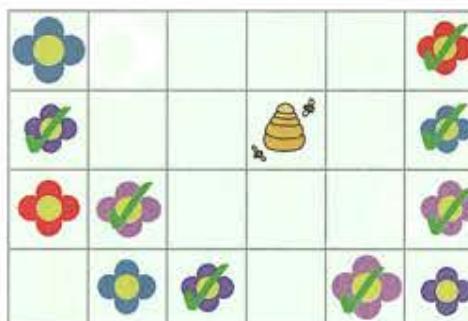
- A. 5
- B. 6
- C. 7
- D. 8



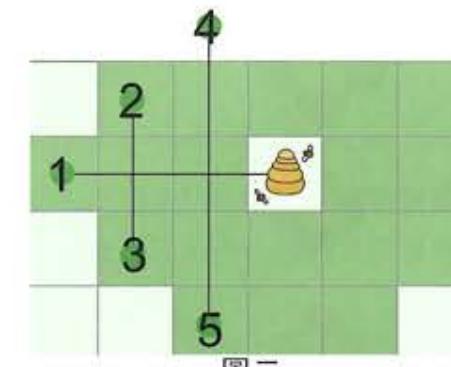
正確答案是：C. 最多有 7 朵花

在三格範圍內的所有花都被打勾標記在下圖一中。

蜜蜂被限定最多只能飛行三格的距離，因此找出距離蜂巢三格內的所有格子。蜜蜂往左飛行 3 格會到達 1 號點，它也可以往左飛 2 格後，往上 1 格 (2 號點) 或往下 1 格 (3 號點)，也可往左 1 格再往上 2 格 (4 號點，在圖外) 或往下 2 格 (5 號點)。採用同樣的方法，往右上與右下移動，就可找出右方距離蜂巢三格內的所有格子，這些格子在下圖二中以淺綠色標示。



圖一



圖二



資訊科學上的意義

蜜蜂的飛行方式是以指令及條件限制來描述，例如向上、向下、向左、向右飛的指令，蜜蜂的飛行距離是條件限制。了解指令及條件限制的內涵就能推斷出執行後能達到的效果及特性，例如飛行範圍。理解一連串指令執行後的效果是資訊科學領域中的重要技能，也是設計演算法的基礎能力。

要注意的是，題目中距離的單位是格子，而不是常見的兩點間的直線距離（也稱為歐幾里得距離）。在此題中，蜜蜂不可往對角或是任意方向移動，被限制只能以水平和垂直方向在格子間移動，稱為直角直線距離 rectilinear distance。因為方格相似於城市街區，這樣的距離也被稱為 曼哈頓距離 manhattan distance 或城市街區距離 city block distance。



關鍵字

曼哈頓距離、計程車幾何

46. 拿



漢賽爾與葛麗特正在玩取石頭遊戲，遊戲的玩法是兩人輪流取石頭，取走最後一塊石頭的人就是遊戲贏家。

每場遊戲都使用 3 塊黑石頭與 7 塊白石頭，每次可選擇取走 1~2 塊的黑石頭或取走 1~3 塊的白石頭。

由葛麗特開始取石頭，請問她在第一回合應如何取石頭，才能確保自己成為最終贏家？

- A. 取走 1 塊白石頭
- B. 取走 2 塊黑石頭
- C. 取走 3 塊白石頭
- D. 不論取走幾塊石頭都可以



正確答案是：C. 取走 3 塊白石頭

先從 3 塊黑石頭開始分析，如果葛麗特選擇移走 1 塊或 2 塊黑石頭，則漢賽爾可以移走剩下所有的黑石頭（若葛麗特移走 1 塊黑石頭，漢賽爾可以移走剩下的 2 塊黑石頭，反之亦然）。此策略會讓漢賽爾成為最終贏家，因此，黑石頭的獲勝策略是全部（3 塊）都不移走。

接著，來分析 7 塊白石頭的獲勝策略，葛麗特可以選擇移走 1 塊、2 塊或 3 塊白石頭。但對於任意 4 塊石頭，無論第 1 位玩家選擇移走多少石頭，第 2 位玩家都能一次移走剩下所有的石頭（若葛麗特移走 1 塊白石頭，漢賽爾可以移走剩下的 3 塊白石頭，以此類推）。因此，白石頭的獲勝策略是留下其中的 4 塊不移走。

綜上所述，為了讓葛麗特成為贏家，她必須留下 3 塊黑石頭及 4 塊白石頭。此策略只有在第一次移石頭時，選擇只移走 3 塊白石頭才有可能達成。若葛麗特沒有在第一次移石頭時取走 3 塊白石頭，則輪到漢賽爾時，他只要採取相同策略，則葛麗特接下來不論如何取石頭，都無法確保一定能獲勝。



資訊科學上的意義

賽局理論 原本是經濟學的一個分支，運用在資訊科學上常常探討的其中一個演算法問題，就是為特定玩家找到獲勝策略。

在本題的情況，我們可以從葛麗特贏得最後一步的情況回溯探討，並以葛麗特為贏家、漢賽爾為輸家的立場分類探討，由此可得葛麗特成為贏家所需維持的特性。這被稱為不變式 invariant，也就是她在每一步的選擇中所需維持不變的特性，而漢賽爾無法透過任何一種移動來改變這樣的特性。如果葛麗特能夠維持此不變式，就可以擁有必勝策略。

這樣的分析方式可以擴展到像西洋棋、圍棋等更複雜的賽局，但是這些遊戲可以選擇的位置更多，我們需要使用其他方式，像是人工智慧，來學習好的策略。**尼姆遊戲 Nim** 又稱為拈，類似本題，兩個遊戲者要輪流從排棋子中其中一排取走一個或多個棋子。

組合賽局理論可以分析出哪一位玩家將會獲勝，以及該玩家必須採取哪些移動策略以確保獲勝。但像是西洋棋因為移動位置更多，較為複雜，所以無法像此問題，在遊戲一開始就能找到必勝策略。



關鍵字

賽局理論、尼姆遊戲