

創新思考與資訊科技教學

陳怡芬 2017.05.20

資訊科學教學法 | 106 年資訊科技科師資增能學分班

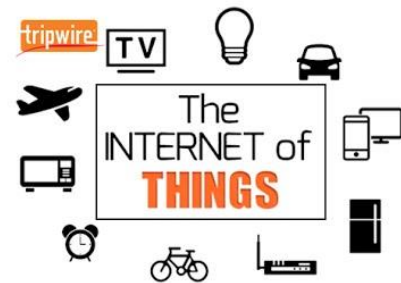
anny83.chen@gmail.com



這是一個運算的時代



[Paul Schottmiller](#) - March 19, 2013



Risks of The Internet of Things 頻道建立者：[Tripwire, Inc.](#)



GHETTO UBER DRIVER
頻道建立者：[Latrell G.](#)



The [Apple Watch](#), released in 2015.



Amazon warehouse robots



Uber & the Cloud:

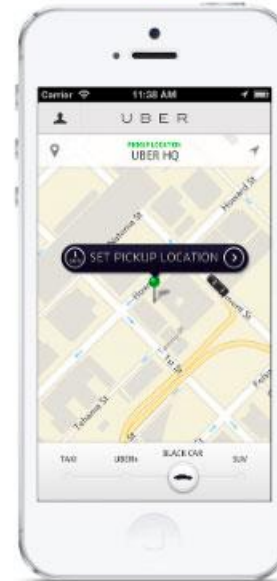
Leveraging Technology to Increase Efficiency



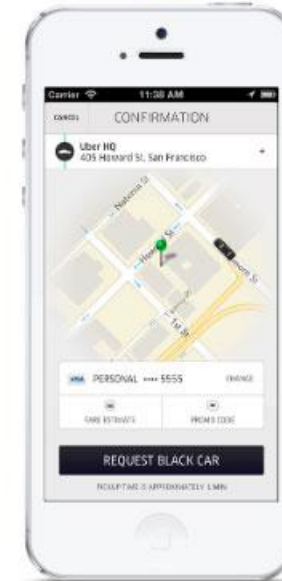
UBER

REQUESTING A RIDE IS AS EASY AS

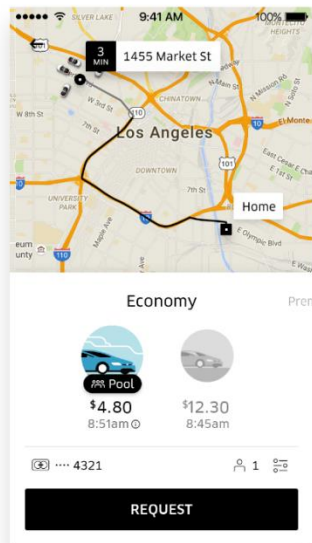
1



2

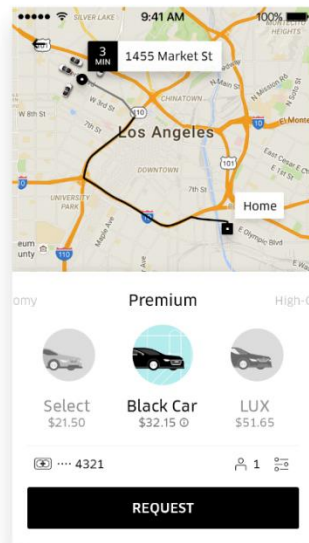


3



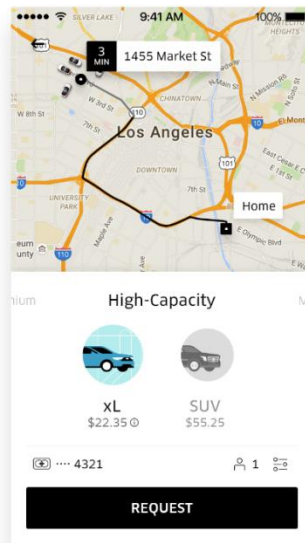
Economy

Price/time tradeoffs and low cost products



Premium

Nice cars and premium services



High-Capacity

Products that can hold 5 or more people





CAB DISPATCH IN WEB BROWSER

assign cab to customer
cab & customer position monitoring
order management
storing selected order info

two way
real-time
communication



CUSTOMER APP

ordering
cab position
cab driver communication



CAB DRIVER APP

availability
dispatch order
customer position
customer communication

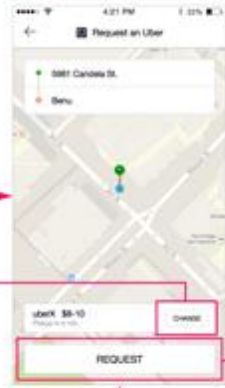


PARTNER APP W/ UBER BUTTON



GET /v1/products
/v1/estimates/price
/v1/estimates/time

REQUEST



Tap Uber Button

POST /v1/requests

REQUESTING



Tap Request

SUCCESSFUL POST?

Yes

GET /v1/requests/request_id-

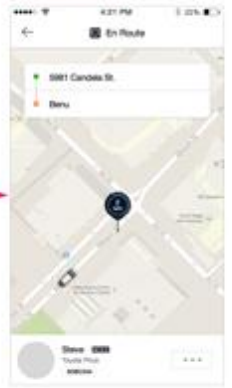
CHECK STATUS

Success

RESPONSE STATUS

accepted
arriving

DRIVER EN ROUTE



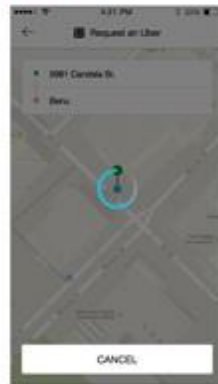
Tap Change

VEHICLE PICKER



DELETE /v1/requests/request_id-

CANCELLING



Tap Cancel

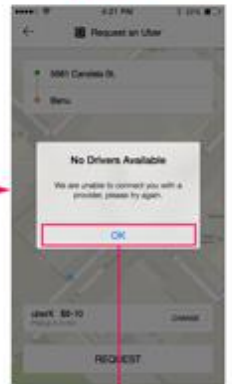
No

RESPONSE STATUS

no_drivers_available

Failed

REQUEST: NO DRIVERS AVAILABLE



Tap OK



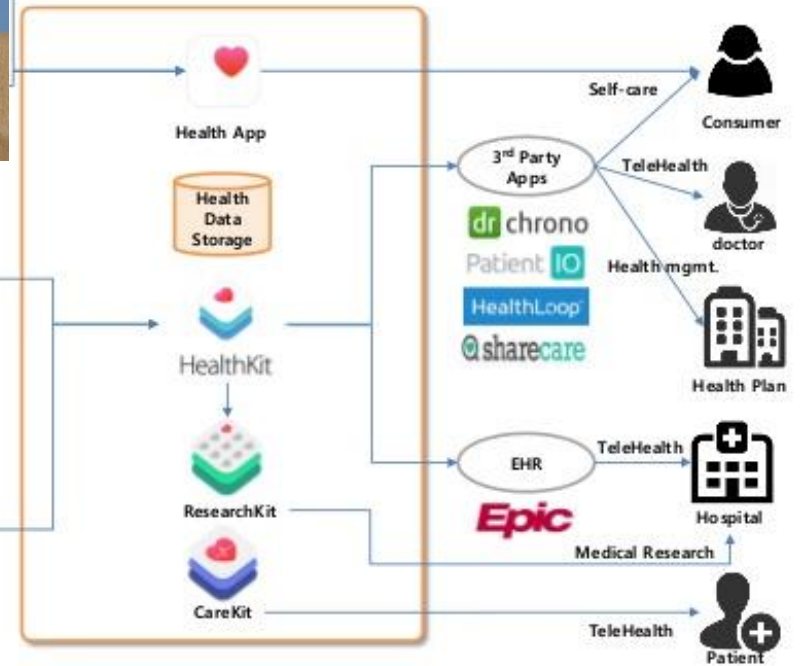


Apple Health Ecosystem

Kugsang Jeong (handeum@gmail.com)



Compatible Apps & Devices

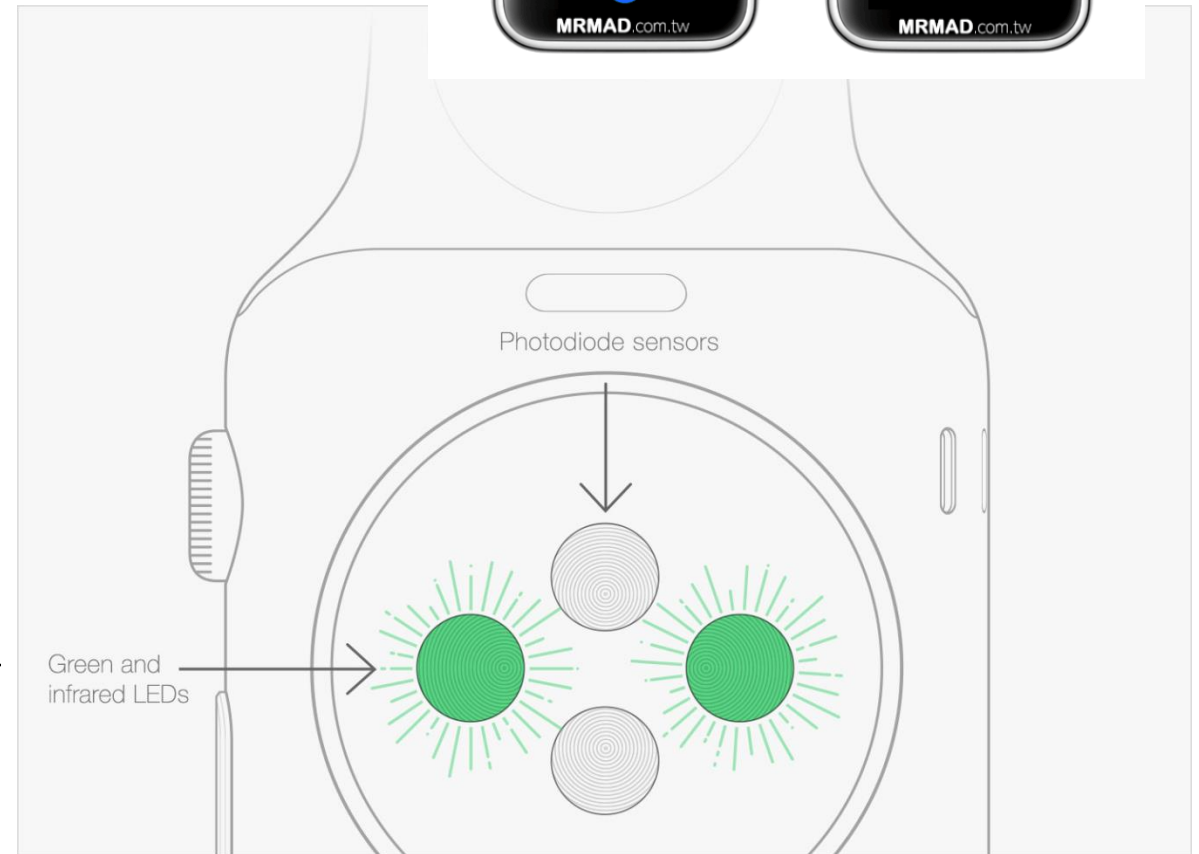


APPLE WATCH SENSOR

目前 Apple Watch 小巧的機身其實只有兩類感應器：與方位相關的加速度感應器、陀螺儀，以及心率感應器。

因此這兩種技術就是 Apple Watch 目前用來偵測身體資訊的主要途徑。

在蘋果的一項專利中，它們描述了一種新方法來強化心率感應的效果。眾所皆知，蘋果使用的是 **LED 光測器**，透過打綠光在皮膚上、藉由血液的吸收量差異來計算心跳的頻率，因此，這個專利的內容便是透過更細緻地分辨心跳的頻率、或是血液的組成，來分辨使用者的身份，換言之，就是透過心率，來達成與 Touch ID、虹膜辨識相同的效果。



善用**運算思維**解決日常生活問題

- 美國卡內基梅隆大學教授 Jeannette M. Wing 認為在基礎語言能力中應該加入電腦運算的因素，在讀、寫和算數之外，還需要該加上電腦運算的概念：「**電腦運算思考的技巧，並不是只有電腦科學家的專利，而是每個人都應該具備的能力及素養。**」(Wing, 2006)
- 具備運算思維能更善用運算解決日常生活問題
- 日常生活與運算的關係愈來愈密切
 - 社交網路、智慧型居家
 - 醫療 交通
 - 購物

運算思維是各領域人才的重要能力

■ 科學與工程領域

- 利用運算**模擬**建築結構，以確認安全性
- 利用運算**預測**氣象，以增加準確性

■ 人文與社會領域

- 利用運算**分析**並優化廣告投放策略
- 利用運算**分析**人口老化趨勢與醫療資源分布

■ 藝術領域

- 利用運算**建構**三維動畫
- 利用運算**創作**數位音樂



未來人才需求

- 具備善用運算方法與工具解決問題的能力 → **運算思維**



- 具備創新與動手實作的能力 → **程式設計**

- 問題解決
- 溝通表達
- 合作共創

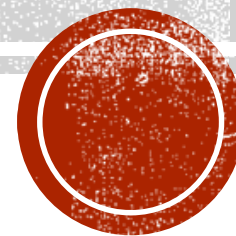
```
C Program for Bisection Method Source Code
1 #include<stdio.h>
2 #include<math.h>
3 float fun (Float x)
4 {
5     return (x*x*x - 4*x - 9);
6 }
7 void bisection (float *x, float a, float b, int *itr)
8 /* this function performs and prints the result of one iteration */
9 {
10     *x=(a+b)/2;
11     ++(*itr);
12     printf("Iteration no. %3d X = %7.5f\n", *itr, *x);
13 }
14 void main ()
15 {
16     int itr = 0, maxitr;
17     float x, a, b, allerr, x1;
18     printf("\nEnter the values of a, b, allowed error and maximum iterations:\n");
19     scanf("%f %f %f %d", &a, &b, &allerr, &maxitr);
20     bisection (&x, a, b, &itr);
21     do
22     {
23         if (fun(a)*fun(x) < 0)
24             b=x;
25         else
26             a=x;
27         bisection (&x1, a, b, &itr);
28         if (fabs(x1-x) < allerr)
29         {
30             printf("After %d iterations, root = %6.4f\n", itr, x1);
31             return 0;
32         }
33         x=x1;
34     }
35     while (itr < maxitr);
36     printf("The solution does not converge or iterations are not sufficient");
37     return 1;
38 }
```

美國二十一世紀關鍵能力聯盟訂定二十一世紀的關鍵能力包含：批判性思考與解決問題、溝通、合作共創、以及創造力 (Partnership for 21st Century Skills-P21, 2007)

1078課綱

資訊科技課程的理念

以**運算思維**為主軸



- ◆ 透過電腦科學相關知能的學習
培養邏輯思考、系統化思考等運算思維
- ◆ 藉由資訊科技之設計與實作
增進運算思維的應用能力、解決問題能力、
團隊合作以及創新思考

我國資訊科技課程的發展

我國資訊科技教育經過幾次的變革，逐漸由**操作技能導向**的課程演變為**高階能力導向**之課程（吳正己，2010）

■ **程式技能導向**的資訊科技教育 73

- 高中選修「電子計算機簡介」

■ **軟硬體應用導向**的資訊科技教育 84

- 國中必修「電腦」
- 高中選修「電腦」

■ **問題解決與電腦科學導向**的資訊科技教育 93~106

- 國中93年全面實施的九年一貫課程
- 高中95年頒布暫綱，98年正式頒布課綱，將「電腦」更名為「資訊科技概論」

■ **運算思維導向**的資訊科技教育 107-108~

- 國中必修「資訊科技」
- 高中必修「資訊科技」



各國資訊科技課程趨勢

隨著資訊科技的發展與教育理念的變遷，各國亦不斷修正資訊科技教育的目標與內涵。

- **培養學生基本ICT技能 (~2003) :**

- 培養國民資訊與通訊技術 (Information and Communication Technology, ICT) 相關技能

- **培養學生高層次的技能 (2003~) :**

- 包含：創造及創新、溝通及合作、研究及資訊運用 (information fluency)、批判性思考、問題解決及決策、數位公民與科技操作及概念

- 美國 CSTA 2003、荷蘭2007、德國電腦科學組織2008年

- **運算思維視為貫穿整個資訊科學課程的主軸 (2011~) :**

- 美國 CSTA 2011年、英格蘭 2013

- **強調科學、科技、工程及數學跨科際整合之STEM學習 (2014~):**

- (Science, Technology, Engineering, and Mathematics) 教育趨勢

- 美國於2012年推展新科技教育計畫，英格蘭於2014年

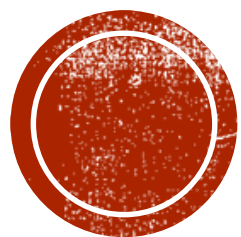


運算思維教學

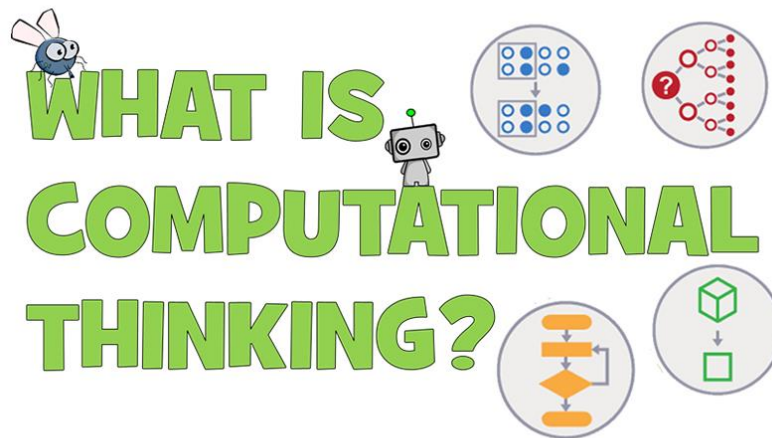
如何將運算思維的培養落實於資訊科技課程教學

- 釐清運算思維的定義
- 提供教學方法的建議





什麼是運算思維



什麼是運算思維-定義

- 運算思維 \neq 資訊科技使用能力
- 運算思維 \neq 程式設計
- 運算思維 \neq 資訊科學

→ 那麼什麼是運算思維？

運算

- 運算是「一種需要演算過程（**algorithmic processes**）、能從演算過程獲益或能產出演算過程，並有確切目標的活動」

ACM, 2005

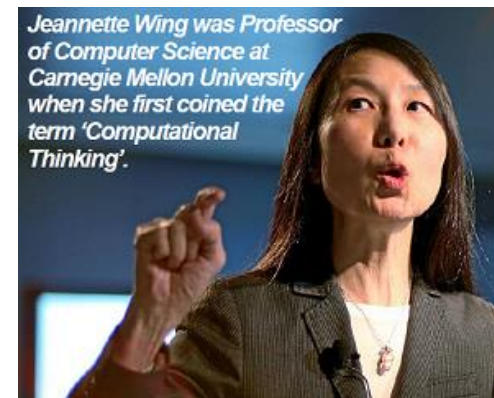
包含：

- 為了特定目的設計與建置電腦軟硬體系統
- 處理、結構化並管理各種資訊
- 利用電腦進行科學研究
- 讓電腦系統更具智慧
- 產生與使用通訊或媒體
- 搜尋與蒐集與特定目的相關之資料
- ...

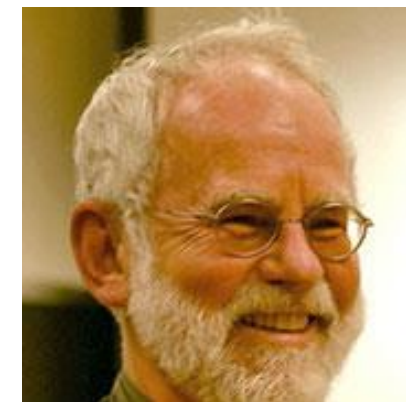


什麼是運算思維-定義

- “運算思維是利用電腦科學的基本概念進行問題解決、系統設計與人類行為理解的**思維模式**” (Wing, 2006)



- “運算思維讓我們能擁有電腦科學家面對問題時所持有的一種的**思維模式**” (Grover & Pea, 2013)



資料來源：

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.

什麼是運算思維-定義

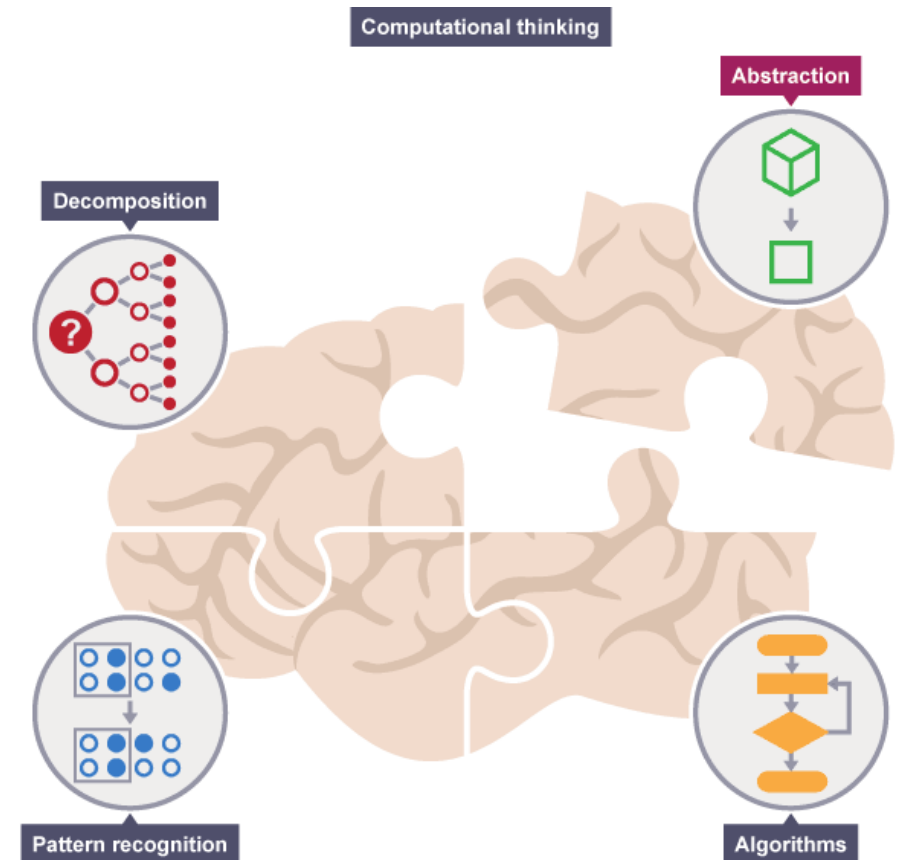
- Google 對運算思維的定義
 - 運算思維是一種利用運算解決問題所需之「心智歷程」(抽象化、演算法設計、解析、樣式辨識等)與「具體產出」(自動化、資料表示、樣式一般化等)

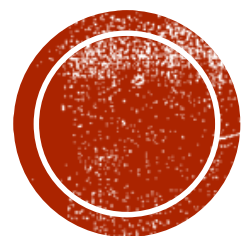


取自教育部運算思維推動計畫

運算思維應包含的元素

- 問題分解 (problem decomposition)
- 抽象化 (abstraction)
- 模式化與模擬 (modeling and simulation)
- 演算法思維 (algorithmic thinking)

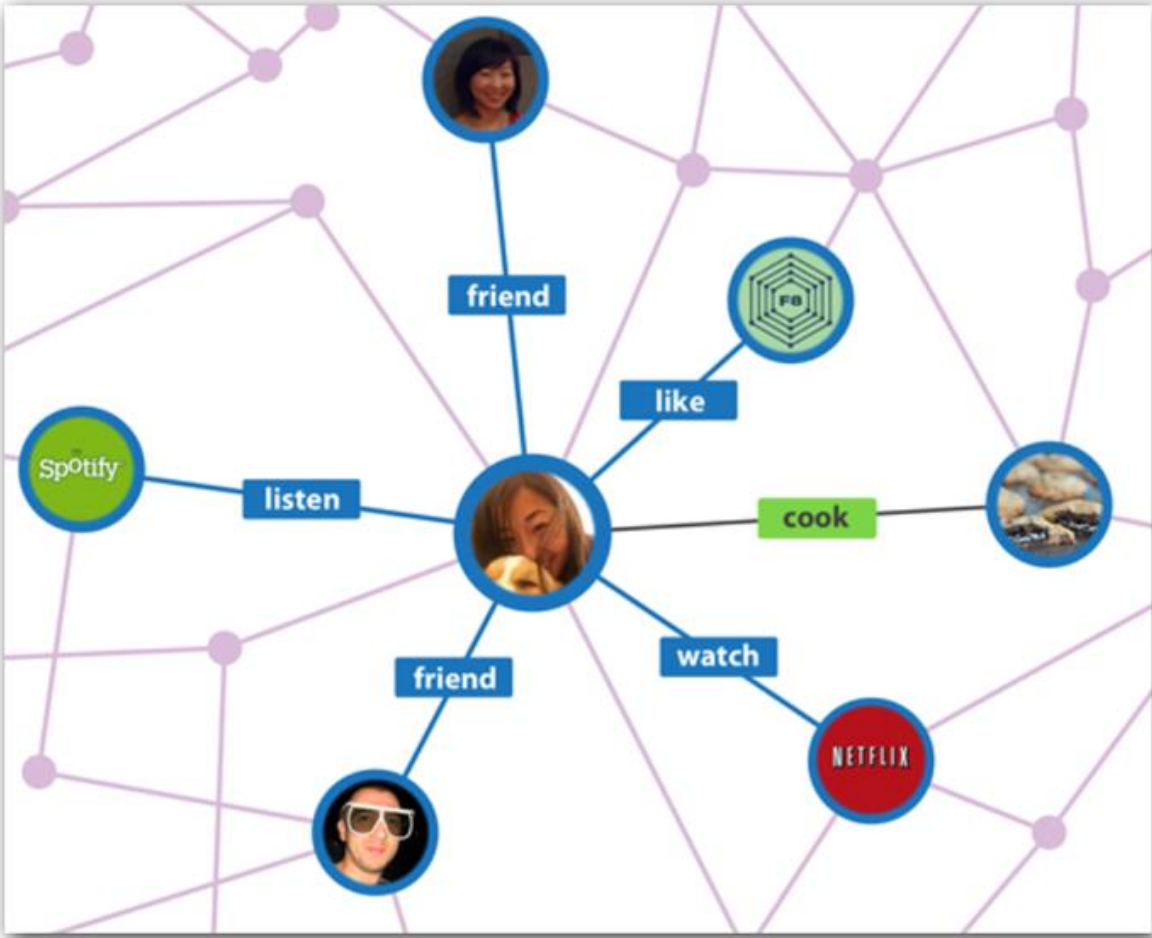




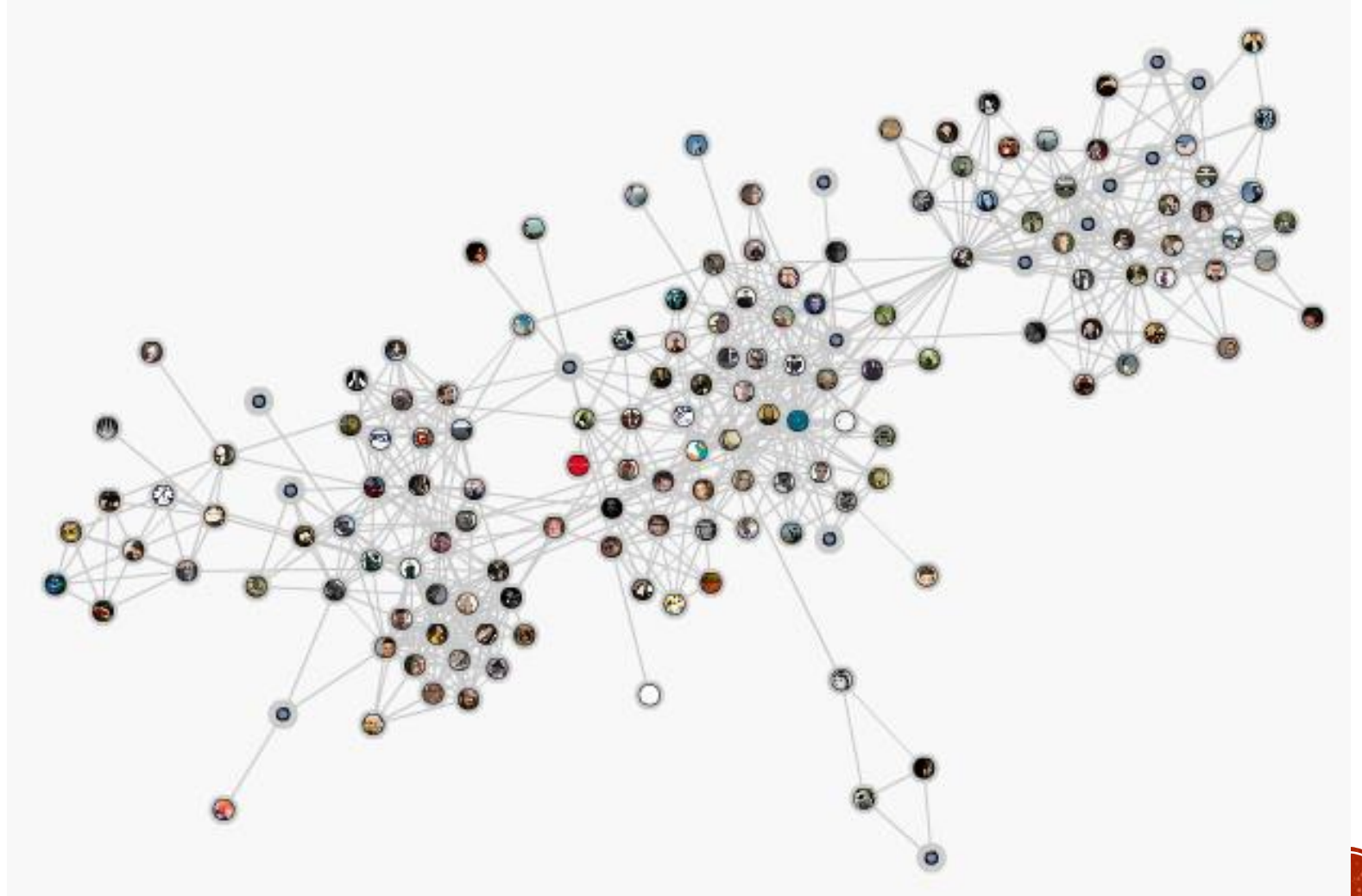
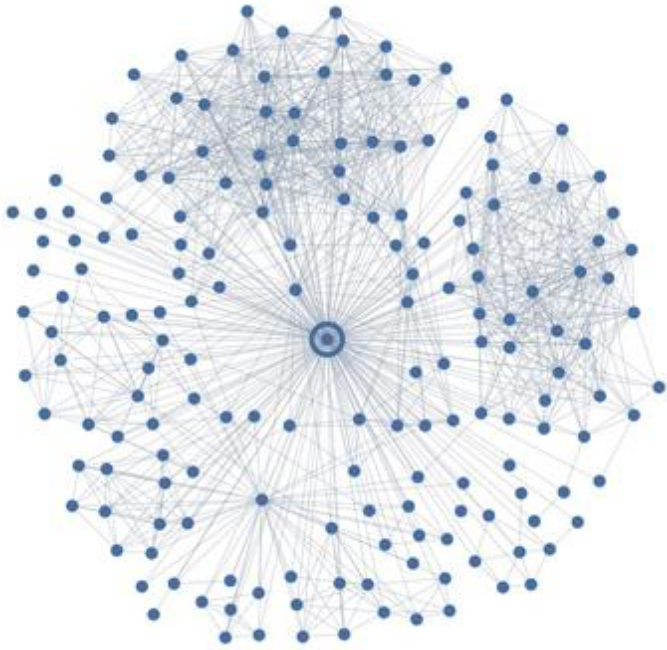
運算思維與程式設計



SOCIAL GRAPH

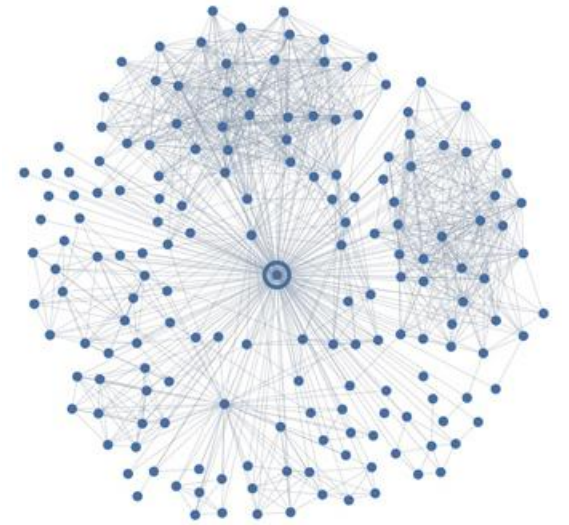


SOCIAL GRAPH



SOCIAL NETWORK

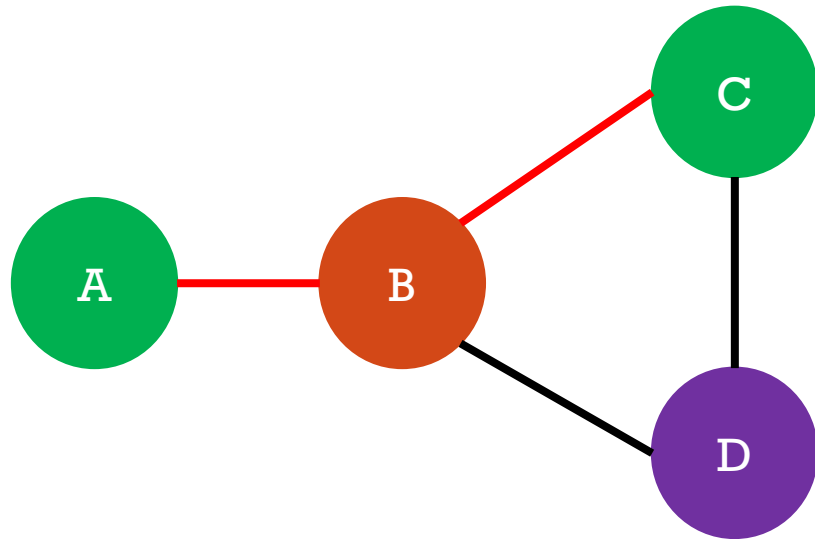
- Anny, Bill, Cherry, David 四個彼此是好朋友
- David 除了認識 Anny, Bill 和 Cherry，還認識 Eva
- Eva 除了認識 David，也認識 Frank 和 Grand
- Frank 和 Grand 是好朋友
- 請問，Anny 想認識 Grand，應該透過誰來介紹呢？



SOCIAL NETWORK

社交網絡、圖型、相鄰矩陣

- 左圖用來顯示Anny, Bill, Cherry, David彼此間的友誼關係，右圖則將左圖記錄成友誼關係表。
- A只認識B這一位朋友，而B則認識A、C及D三位朋友。
- 如果A想要認識C，則必須經過B的介紹。



	A	B	C	D
A		○		
B	○		○	○
C		○		○
D		○	○	



SOCIAL NETWORK

- 下圖是Anny, Bill, Cherry, David , Eva, Frank 和 Grand 7個人的友誼關係表
- 如果Anny想要認識Grand，請問她最少須經過幾個人介紹？

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

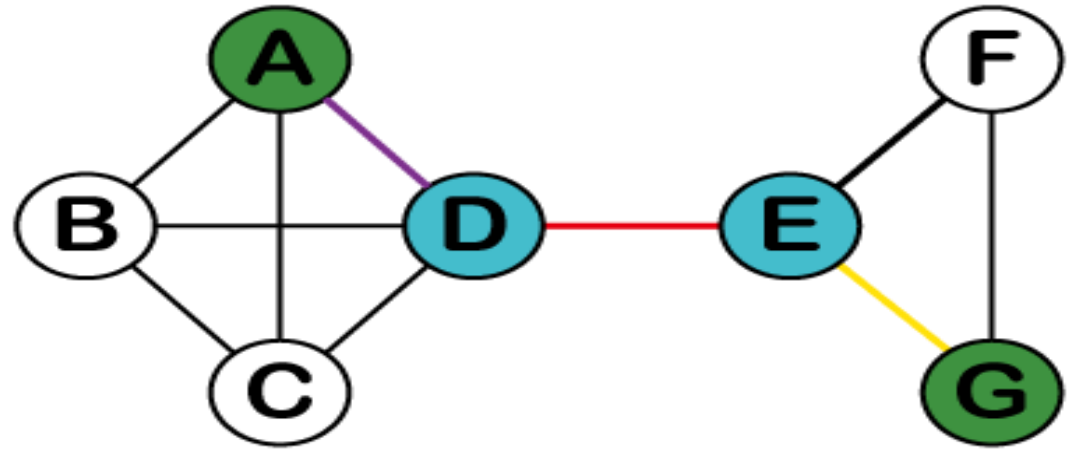
	A	B	C	D	E	F	G
A		○	○	○			
B	○		○	○			
C	○	○		○			
D	○	○	○		○		
E				○		○	○
F					○		○
G					○	○	



SOCIAL NETWORK

- 因為A的所有朋友都不是G的朋友，所以不可能只經過1個人介紹。
- 根據友誼關係表，A至少需要2個朋友來介紹。重新建立社交網路圖。

	A	B	C	D	E	F	G
A		○	○	●			
B	○		○	○			
C	○	○		○			
D	○	○	○		○		
E				●		○	●
F					○		○
G					○	○	



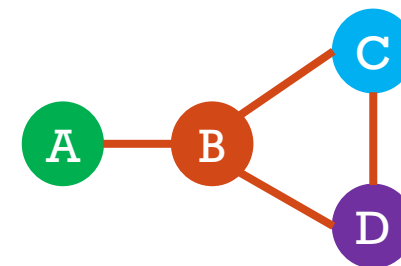
- 由圖得知如果A想要認識G，他至少必須經由D及E這兩位的介紹才行。



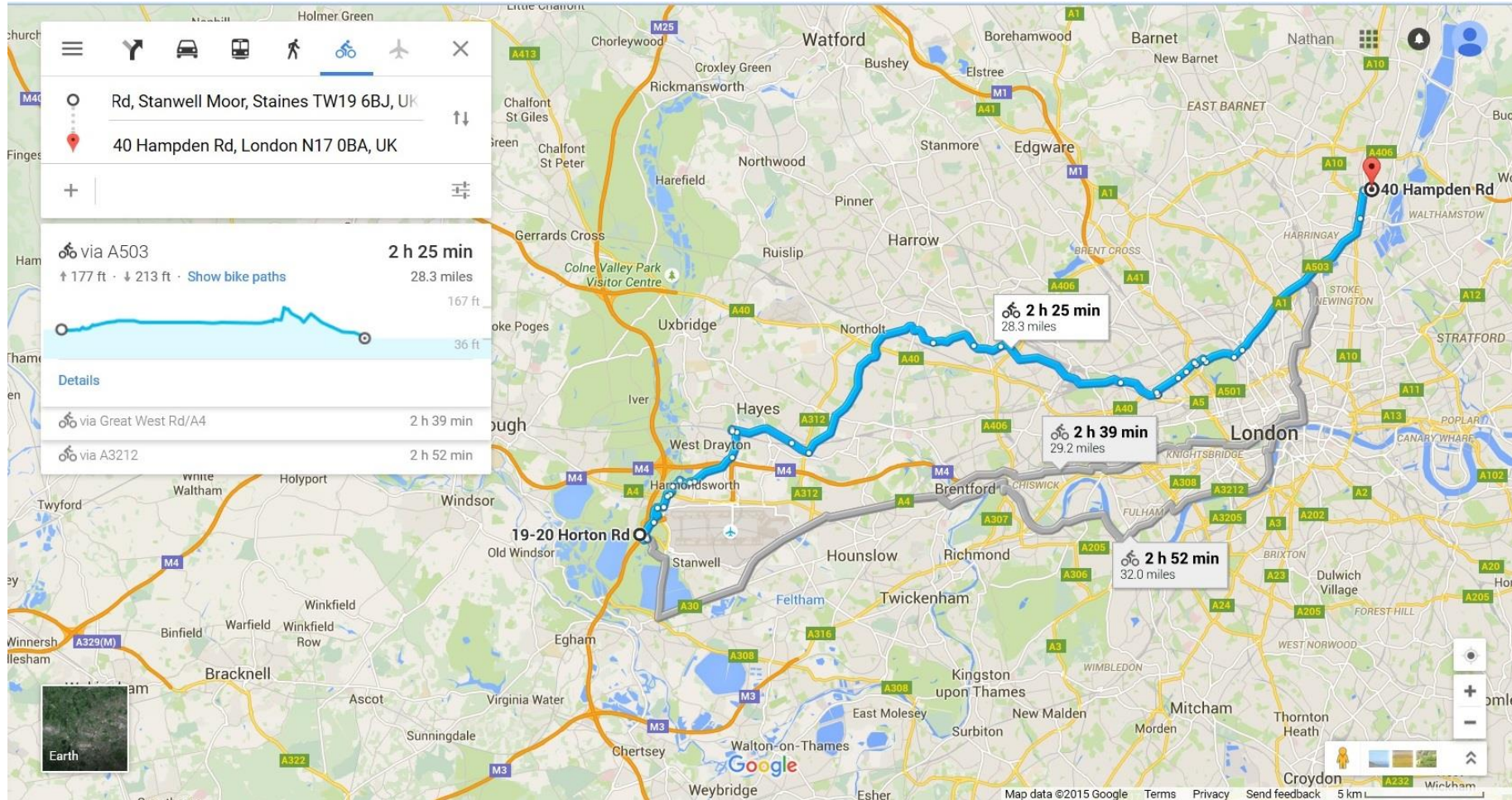


SOCIAL NETWORK

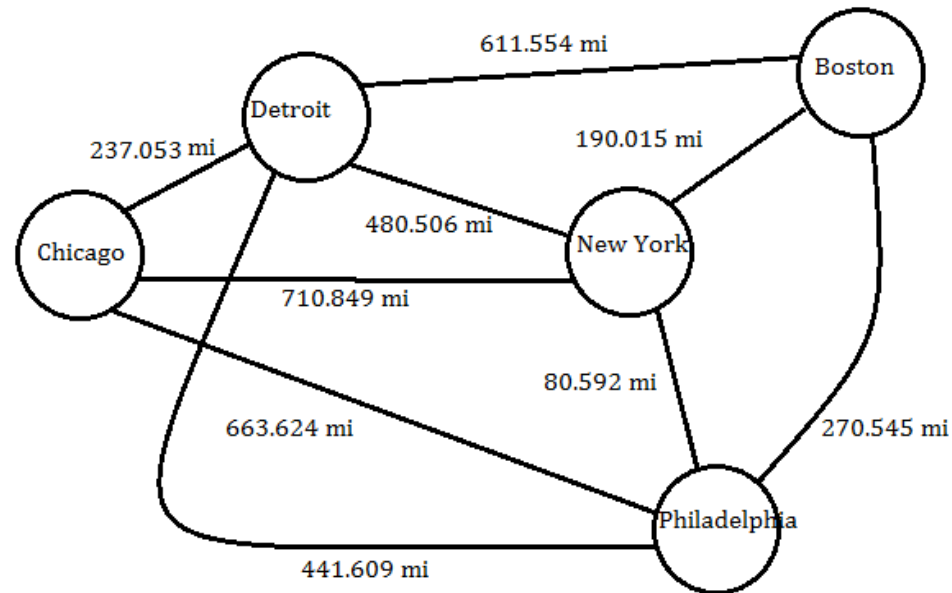
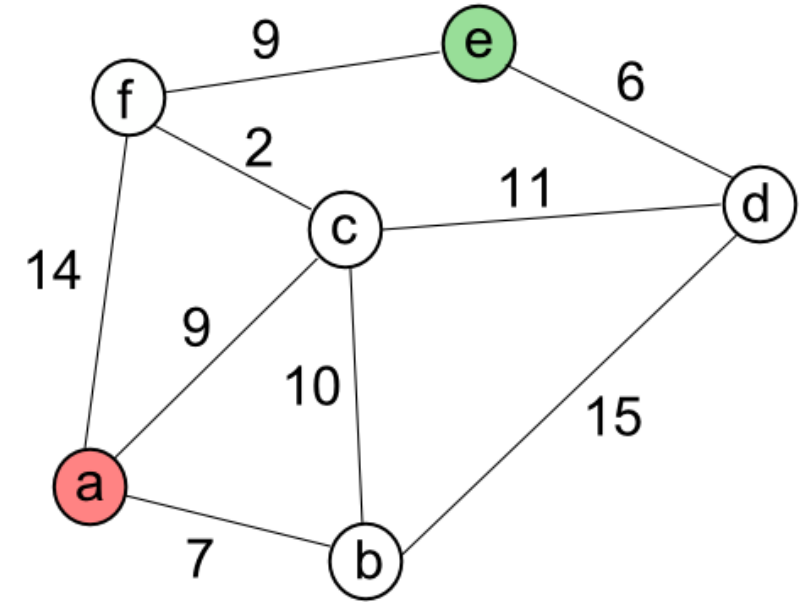
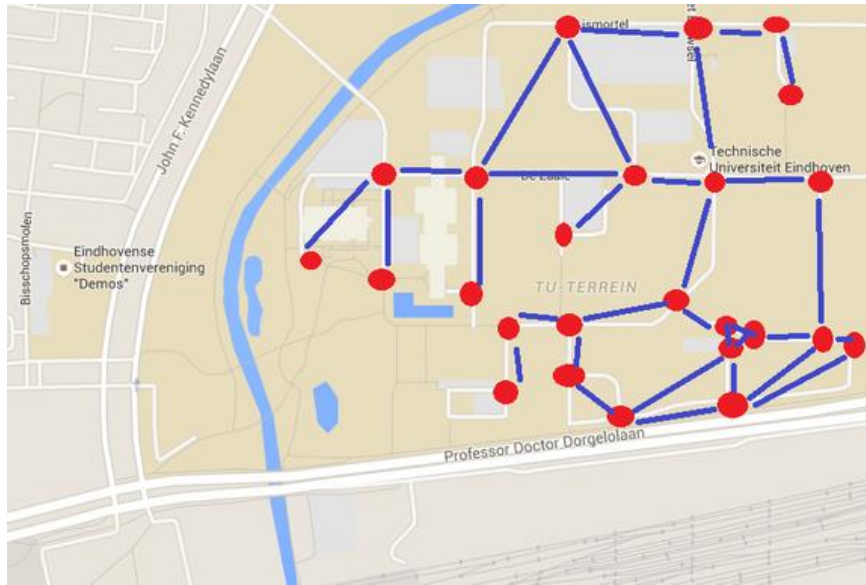
- **圖 (Graph)** 經常用來表示資料間的**關聯性**
- 此題中的圖用來表示友誼關係的視覺化呈現。
- 電腦程式中則經常採用**相鄰矩陣**來表示；
- 根據兩筆資料在矩陣中對應的行與列，該矩陣位置中若有記錄標示，表示兩者間存在**連接關係**，否則兩者不存在**連接關係**。從相鄰矩陣可以很快地計數出每筆資料和其他多少筆資料有相連
- 在本題中可表示每個人的朋友數。此外，可以套用**圖型走訪演算法**，找出節點間的**間接連結關係**及**最短連接路徑**，還可用來分析出關係較緊密的子圖結構（對應到社群）或圖形結構中的**核心節點**（對應到社群**關鍵人物**）。



SHORTEST PATH



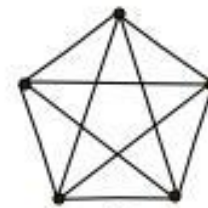
SHORTEST PATH & ROUTING



EULER PATH- 動線規畫



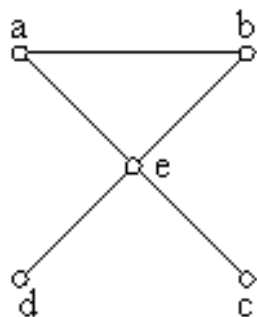
9



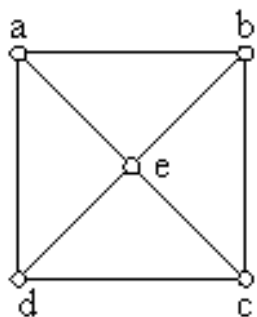
10



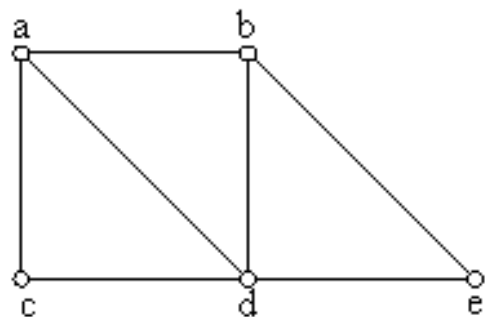
11



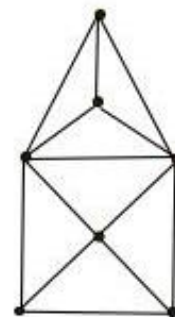
G1



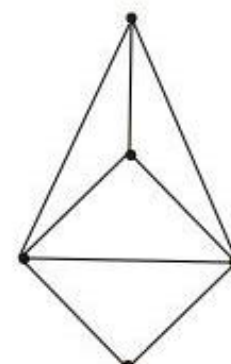
G2



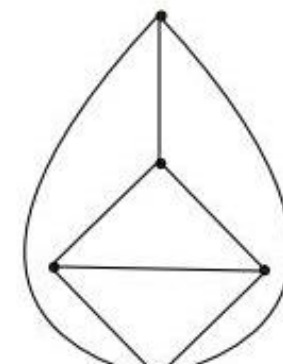
G3



12



13



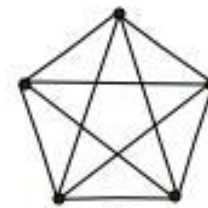
14



EULER PATH- 動線規畫



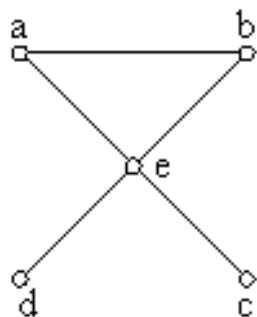
9



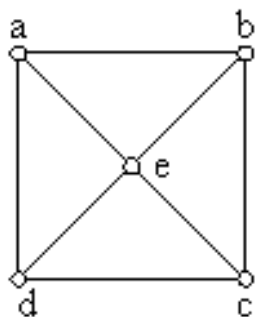
10



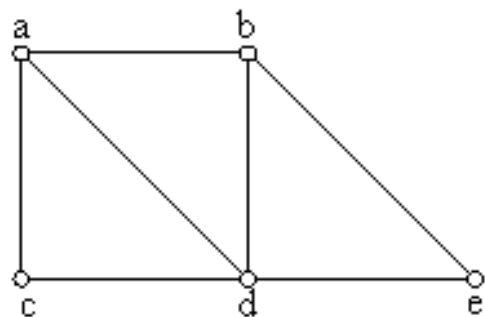
11



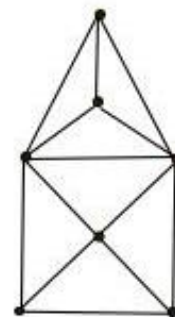
G1



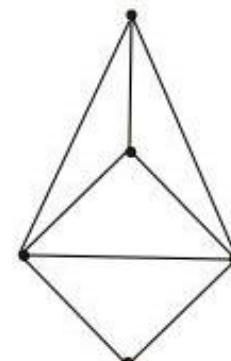
G2



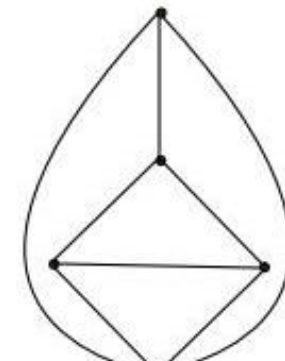
G3



12



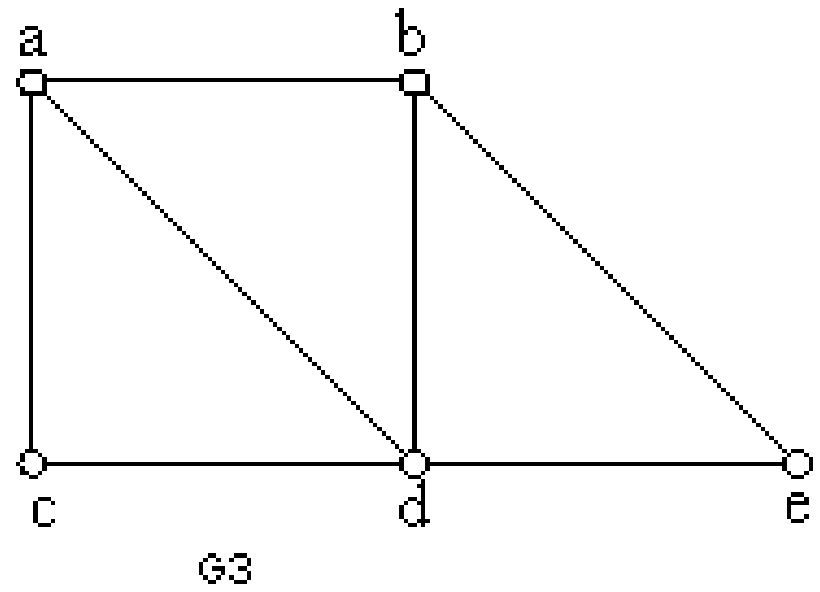
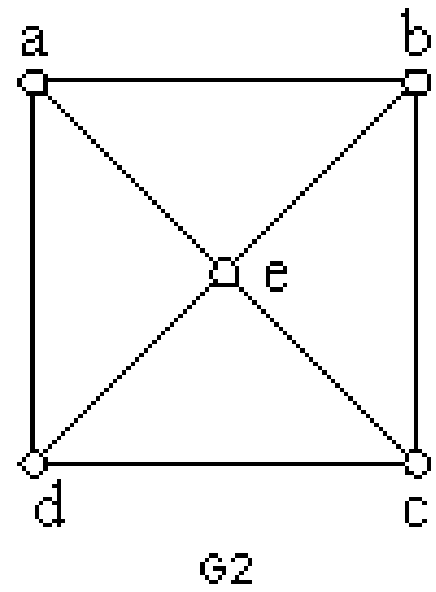
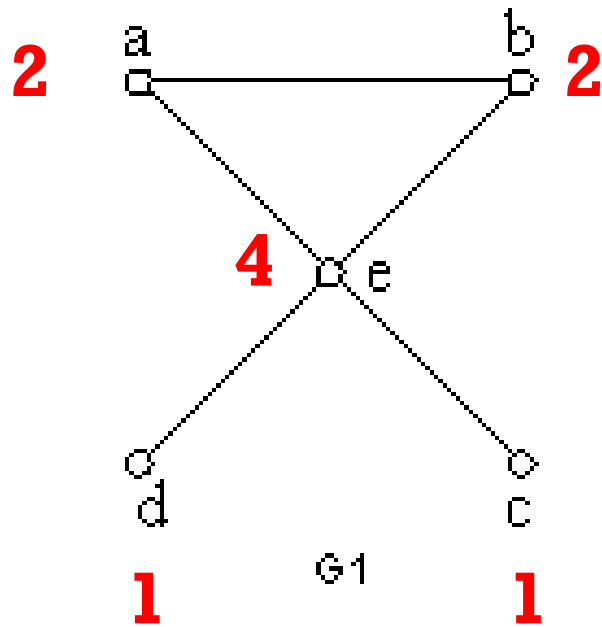
13



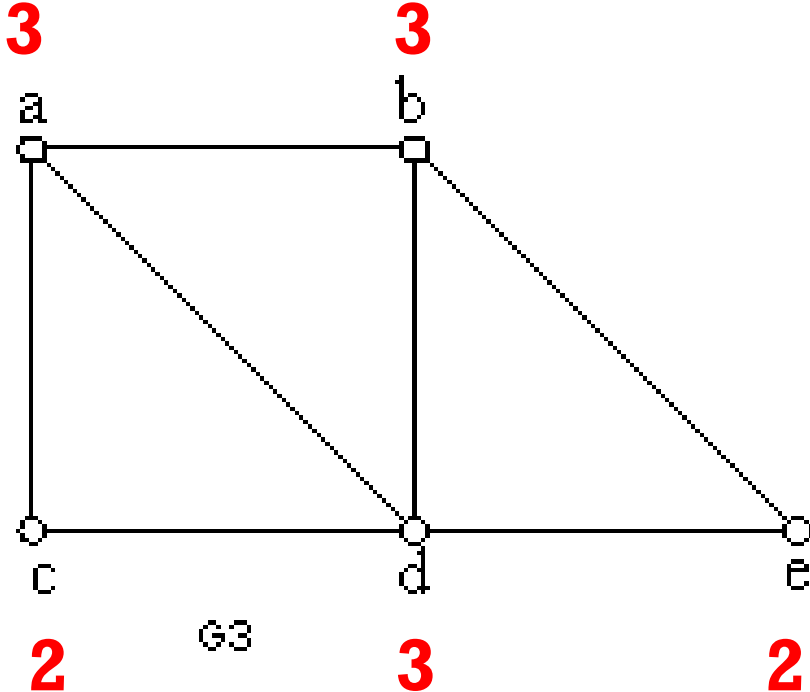
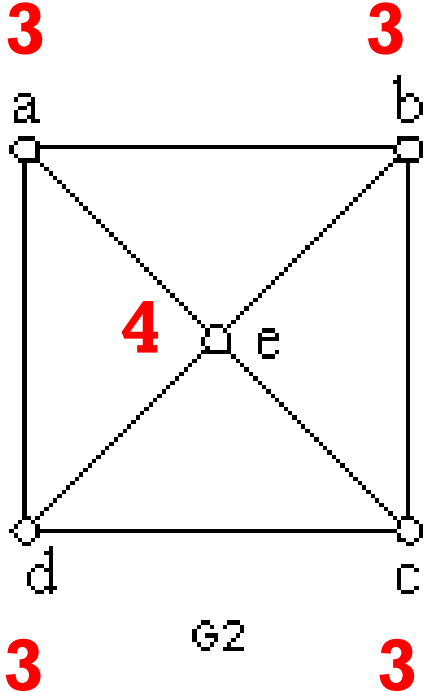
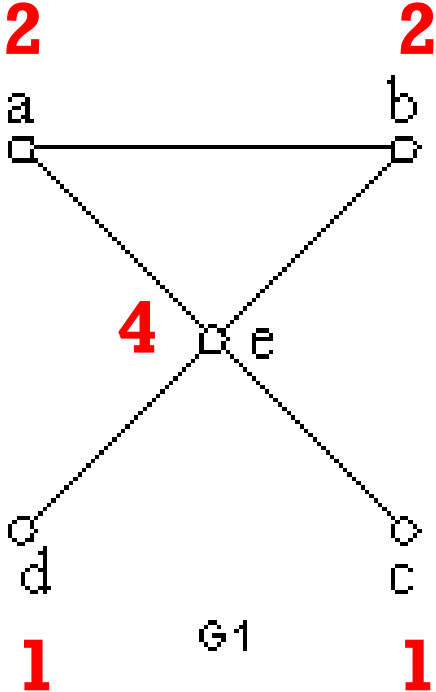
14



IN DEGREE / OUT DEGREE



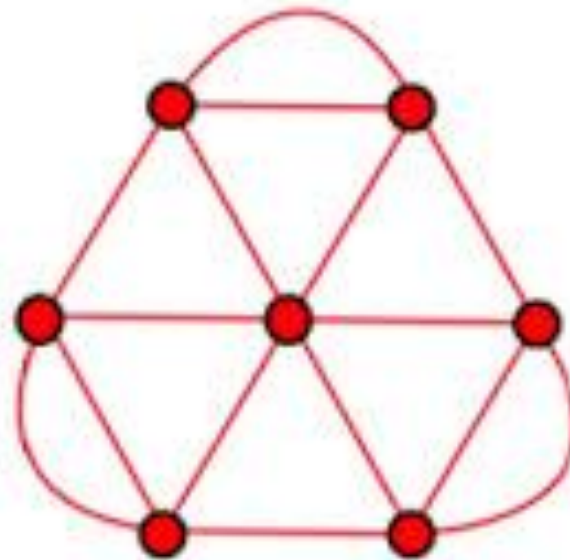
IN DEGREE / OUT DEGREE



EULER PATH-動線規畫



展覽地圖



圖形 E



科學、工程等研究無法與運算分離

- 例如：進行天文研究時，天文望遠鏡所取得的大數據需要進行分析，或是氣象模型或工程材料的大量模擬等
- 因此在進行科學與工程教學時，導入運算思維，可幫助學生有效利用運算工具並藉由各種運算思維：
- 抽象化、系統化資訊處理、符號表示、演算法表達與流程控制、模組化、效能限制及除錯等，能結構化地解析並處理科學與工程領域的複雜問題。



視覺化程式設計之運算思維導向教學- 以**樂曲創作**與**碎形繪製**為例

臺北市立第一女子高級中學 資訊學科教師 陳怡芬

學習內容：重複結構與模組化程式設計

透過觀察樂曲(聽覺)與幾何圖形(視覺)的規律性

讓學生體驗

分解問題、樣式辨識、模型化與抽象化 等運算思維歷程

進而處理較複雜而龐大的任務。

音樂作專題~引導學生分析重複樂句結構，以函式積木實作樂句，以程式流程描述樂曲進行，藉以體會模組化程式設計的概念

崖上的波妞

(單旋律)

Moderate J = 120

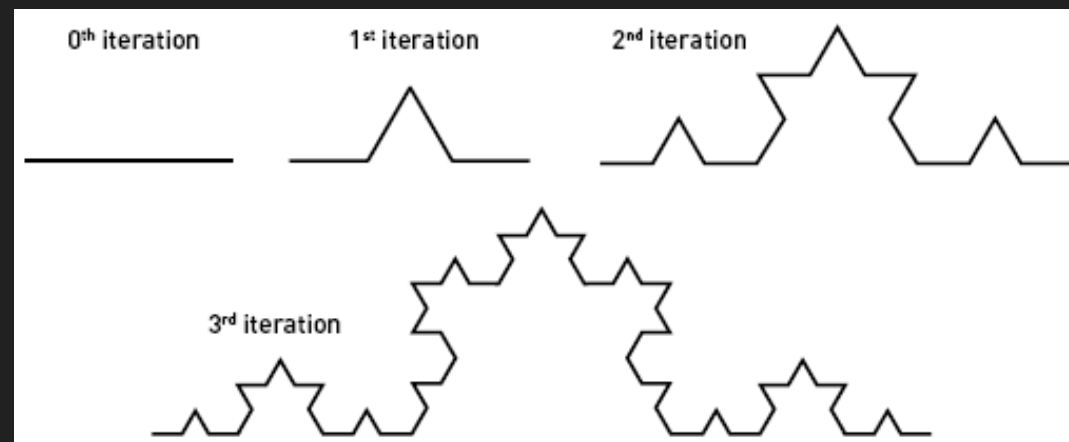
The image shows a musical score for the piece '崖上的波妞' (The Girl on the Cliff). It consists of five systems of music, each with a treble clef and a 4/4 time signature. The tempo is marked as 'Moderate' with a quarter note equal to 120 beats per minute (J = 120). The score is written for a single melodic line. The first system includes a 'LUI' (Lullaby) marking. The music features a simple, repetitive melodic pattern that is characteristic of the film's score.

This image shows a screenshot of a Scratch project page. The project title is '張思婕and丁紫庭-崖上的波妞-2' (Zhang Si-jie and Ding Zi-ting - The Girl on the Cliff-2), created by 'ting0101095'. The project has 18 scripts and 2 characters. The main stage shows a cartoon character with red hair and a green flag on its head, swimming in the ocean. A blue jellyfish is also visible. The text 'Please press the jellyfish' is displayed at the bottom of the stage. The right sidebar contains the following information:

- 操作說明** (Operation Instructions):
 - 崖上的波妞
 - 樂器: 長笛
 - BPM: 120
 - BY: 張思婕 & 丁紫庭
- 備註與謝誌** (Notes and Acknowledgments):
 - 樂譜來源: 張思婕
 - 程式: 丁紫庭
 - 背景圖源: http://www.gamebase.com.tw/gb_img/1029008/90768839/
 - 水母圖源: 張思婕

At the bottom of the page, it shows '© 分享於 3 十月 2016' and '最後修改於 3 十月 2016'. There are also icons for stars (0), hearts (1), eyes (2), and a tree (1).

『碎形幾何繪圖專題』引導學生觀察多種碎形圖形的產生規則，尋找圖形樣式，並透過對碎形自我相似性的理解與建構體驗，進一步體會遞迴函式的抽象概念。



運算思維導向課程架構

運算思維

問題拆解
(Decomposition)

尋找規則
(Pattern Recognition)

模式化
(Modeling)

抽象化
(Abstraction)

演算法設計
(Algorithm Design)

樂曲創作

解析樂句

從規律性尋找
樂曲的重複樣式

以數值模型
記錄音高與節拍資料

以函式描述樂句

運用模組化程式設計完成
自動化樂曲演奏

碎形繪圖

解析複雜圖形

從規律性中尋找
圖形重複樣式

以數學模型
描述圖形物件

以函式描述圖形物件

運用模組化程式設計完成
碎形圖繪製

觀察
分析

資料表示
程式概念

程式實作

課程設計與實施

一、『樂曲創作專題』

- 對象：北一女中高一學生60人
- 課程：特色選修『運算思維與程式設計』
- 時數：3小時
- 視覺化程式工具：
MIT Scratch Online Editor

二、『碎形幾何繪圖專題』

- 對象：北一女中高一學生60人
- 課程：特色選修『問題解決與程式設計』
- 時數：4小時
- 繪圖程式工具：
MSLOGO、Python Turtle

Decomposition / Pattern Recognition Abstraction / Algorithm

運用樂句積木完成音樂盒

Three columns of Scratch blocks defining musical phrases:

- 定義 A:** 16 blocks of notes with durations: 71 (1), 71 (1), 72 (1), 74 (1), 74 (1), 72 (1), 71 (1), 69 (1), 67 (1), 67 (1), 69 (1), 67 (1), 69 (1), 71 (1), 71 (1.5), 69 (0.5), 69 (2).
- 定義 A':** 16 blocks of notes with durations: 71 (1), 71 (1), 72 (1), 74 (1), 74 (1), 72 (1), 71 (1), 69 (1), 67 (1), 67 (1), 69 (1), 67 (1), 69 (1), 71 (1), 69 (1.5), 67 (0.5), 67 (2).
- 定義 B:** 16 blocks of notes with durations: 69 (1), 69 (1), 71 (1), 67 (1), 69 (1), 67 (1), 71 (0.5), 72 (0.5), 71 (1), 67 (1), 67 (1), 69 (1), 71 (0.5), 72 (0.5), 71 (1), 69 (1), 67 (1), 69 (1), 62 (2).

Four staves of musical notation in 4/4 time:

- A:** Treble clef, key signature of one sharp (F#), notes: G4, A4, B4, C5, B4, A4, G4, F#4, E4, D4, C4.
- A':** Treble clef, key signature of one sharp (F#), notes: G4, A4, B4, C5, B4, A4, G4, F#4, E4, D4, C4.
- B:** Treble clef, key signature of one sharp (F#), notes: G4, A4, B4, C5, B4, A4, G4, F#4, E4, D4, C4.
- A':** Treble clef, key signature of one sharp (F#), notes: G4, A4, B4, C5, B4, A4, G4, F#4, E4, D4, C4.

Scratch block: 定義 playstyle
設定節奏為 120 bpm
設定樂器為 1

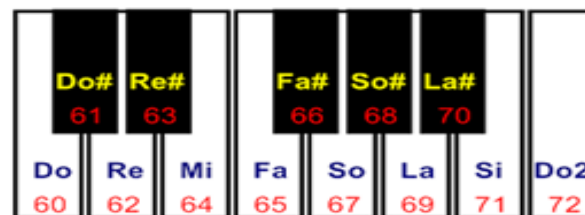
Scratch block: 當按下 空白鍵 鍵
playstyle
A
A'
B
A'



樂曲創作專題- Abstraction

MUSIC NOTATION

以數值符號表達8小節音樂



Note	Beats	Note	Beats
	4 beats		6 beats
	2 beats		3 beats
	1 beat		1½ beats
	½ beat		¾ beat

第1小節

play note **71** for **1** beats
 play note **71** for **1** beats
 play note **72** for **1** beats
 play note **74** for **1** beats



第2小節

play note _____ for _____ beats
 play note _____ for _____ beats
 play note _____ for _____ beats
 play note _____ for _____ beats



樂曲創作專題- Algorithm Design

崖上的波妞 (單旋律)

Moderate ♩ = 120

定絃 B

- 鋼琴音符 70 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 70 0.5 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 62 0.5 拍
- 鋼琴音符 64 0.5 拍
- 鋼琴音符 65 0.5 拍
- 鋼琴音符 67 2 拍

定絃 B'

- 鋼琴音符 70 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 70 0.5 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 64 1 拍
- 鋼琴音符 65 2.5 拍

定絃 C

- 鋼琴音符 65 1 拍
- 鋼琴音符 60 0.5 拍
- 鋼琴音符 65 0.5 拍
- 鋼琴音符 67 2 拍
- 鋼琴音符 67 1 拍
- 鋼琴音符 60 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 69 2 拍

定絃 D

- 鋼琴音符 69 0.5 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 70 0.5 拍
- 鋼琴音符 72 0.5 拍
- 鋼琴音符 74 1 拍
- 鋼琴音符 72 0.5 拍
- 鋼琴音符 69 1 拍
- 鋼琴音符 65 0.5 拍
- 鋼琴音符 67 2 拍

定絃 D'

- 鋼琴音符 69 0.5 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 70 0.5 拍
- 鋼琴音符 72 0.5 拍
- 鋼琴音符 74 1 拍
- 鋼琴音符 72 0.5 拍
- 鋼琴音符 69 1 拍
- 鋼琴音符 60 0.5 拍
- 鋼琴音符 65 2 拍

定絃 E

- 休息 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 69 2 拍
- 鋼琴音符 72 2 拍
- 休息 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 62 0.5 拍
- 鋼琴音符 60 1 拍
- 鋼琴音符 69 2 拍
- 鋼琴音符 67 1 拍

定絃 G

- 休息 1 拍
- 鋼琴音符 69 1 拍
- 鋼琴音符 70 1 拍
- 鋼琴音符 72 1 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 65 1 拍
- 鋼琴音符 67 1 拍
- 鋼琴音符 69 1 拍
- 鋼琴音符 67 4 拍

定絃 H

- 鋼琴音符 60 0.5 拍
- 鋼琴音符 65 0.5 拍
- 鋼琴音符 65 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 67 0.5 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 69 0.5 拍
- 鋼琴音符 70 0.5 拍

定絃 Chorus

- C
- D
- C
- D'
- E

定絃 Verse

- A
- B
- A
- B'

定絃 Bridge

- F
- G
- H

管角色被點擊

- style
- Verse
- Chorus
- Bridge
- Verse

管角色被點擊

- 定絃 style
- 設定節奏為 120 bpm
- 設定樂器為 12

管角色被點擊

- 選擇 訊息1
- 設定旋轉方式為 左-右
- 或復 15 次
- 面向 -90 方向
- 等待 1 秒
- 面向 90 方向
- 等待 1 秒
- 選擇 2

分析、討論、創作、測試



Drawing Projects

碎形~尋找大自然的密碼

Fractals – code in nature



- 本專題引導學生觀察多種碎形圖形的產生規則(rules)，學習尋找圖形樣式(finding patterns)。透過對碎形(fractals)的自我相似性與建構步驟的認識，鼓勵學生自行創造圖形規則，設計屬於自己的碎形。
- 專題重點為重複結構與與函式的學習。活動中透過重複樣式的辨識與碎形圖形的建構，讓學生體驗分解問題、樣式辨識、模型化與抽象化等運算思維歷程，處理複雜而龐大的任務。
- Concepts covered：數學、幾何、碎形、藝術、設計、自我相似性、程式設計、迴圈、疊代、函式、遞迴函式、運算思維

I. 觀察大自然存在的圖形規律

(一) 大自然界存在的神秘圖形規律 Fractals & patterns (50 分鐘)

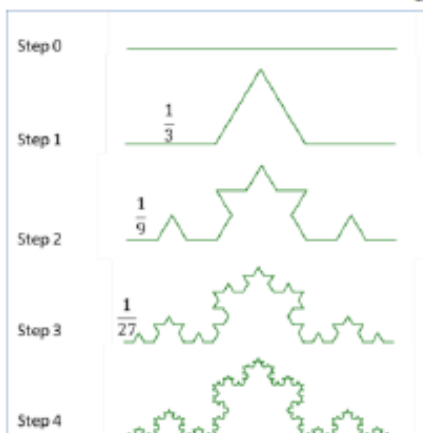
本活動旨在引導學生觀察多種碎形圖形的產生規則(rules)，學習尋找圖形樣式(finding patterns)的方法

步驟	教學活動	活動內容	教材/學習單
1	引起動機	教師展示自然界中的各式神秘碎形，引導學生思考其規律	簡報
2	學生分組討論	學生分組討論碎形的樣式(patterns)與形成規律(modeling)	學習單
3	分享碎形的樣式	學生分享碎形的樣式(patterns)與形成規律(modeling)	學習單 線上平台
4	認識碎形(fractals)與樣式(patterns)		簡報

II. 手繪碎形

(二) 手繪碎形 Drawing Fractals(50 分鐘)

本活動旨在培養學生尋找 pattern 的能力，透過碎形圖的生成練習，促進學生對自我相似性(self-similarity)與遞迴(recursion)概念的理解。

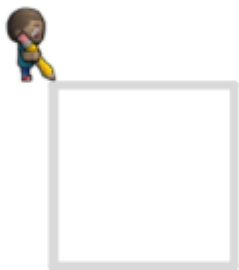



步驟	教學活動	活動內容	教材/學習單
1	引起動機	教師提問 碎形(fractals)具備什麼特性? 樣式(patterns)的意義是什麼?	簡報
2	Finding Pattern	由老師給定明確學習任務，請學生找出規律數列或圖形的第 n 項(number patterns, picture patterns, shape patterns, number sequence patterns...)	學習單
3	繪製碎形(I)	Koch curve - Snowflake 雪花曲線 	學習單

III. 寫程式畫幾何圖形

(三) 寫程式畫幾何圖形 Geometric drawing by programming (50 分鐘)

本活動旨在複習程序性流程控制(1-3 關)，引導學生運用重複結構進行流程簡化(4-7 關)，並學習以幾何圖形函式(8-10 關)建立複雜而規律的圖形。

活動任務：完成 Code.org / 畫家(Artist) 10 個小單元 或利用 scratch / Logo program 完成任務

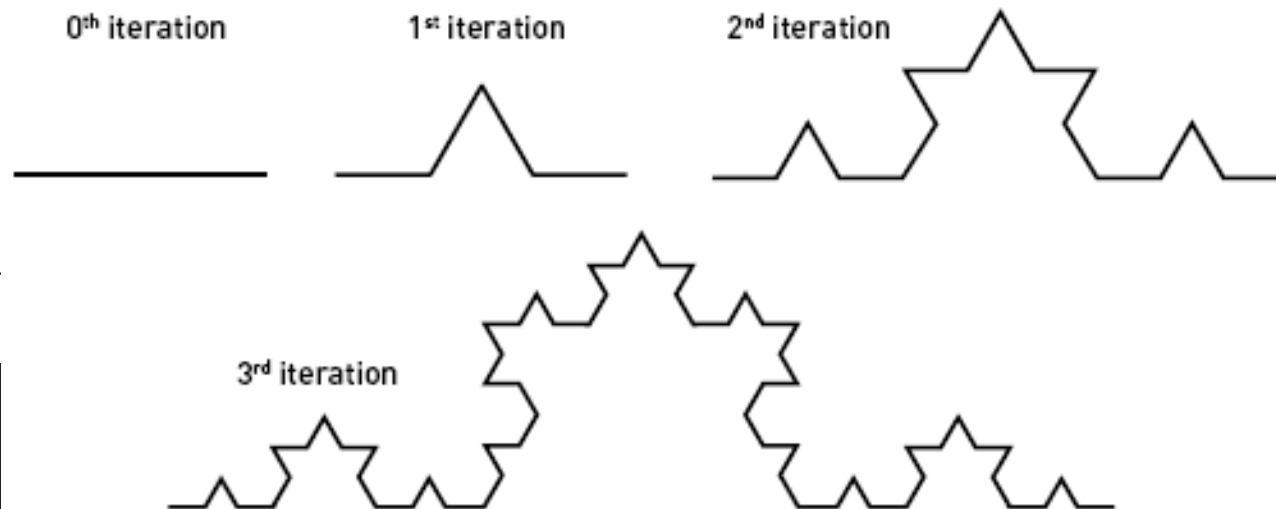
步驟	任務說明	完成作品	程式積木
1.	<p>階段 10 的第 1 關</p> <p>歡迎光臨藝術家！首先，讓我們試著用 "向右旋轉" 及 "向前移動" 積木，畫一個簡單的正方形吧。每邊邊長為 100 像素。</p>		<p>程式積木</p> 
2.	<p>階段 10 的第 2 關-</p> <p>用 120 度、60 度及 150 像素當做邊長，畫出這顆鑽石吧</p>		<p>程式積木</p> 

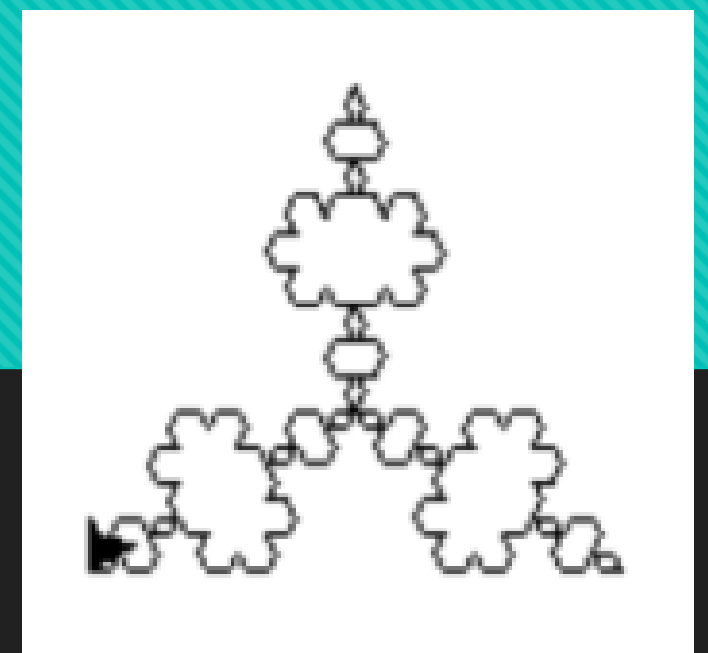
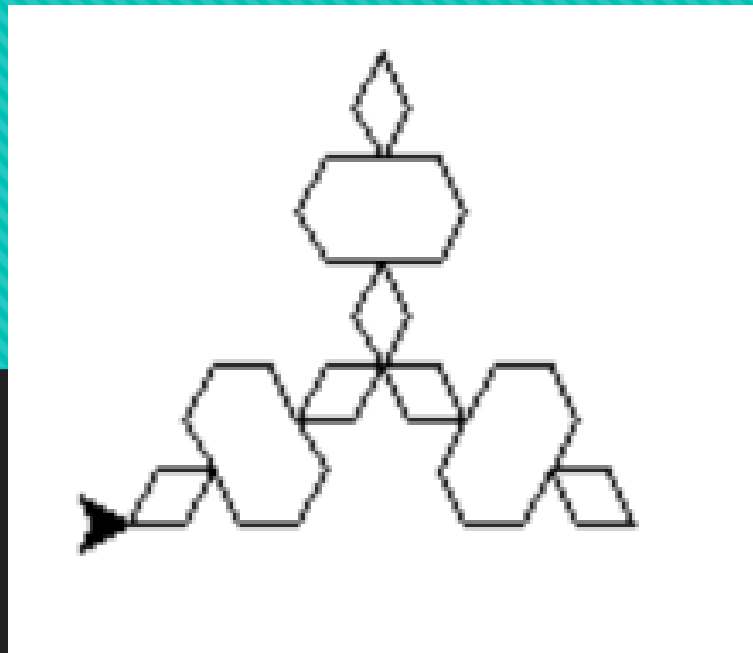
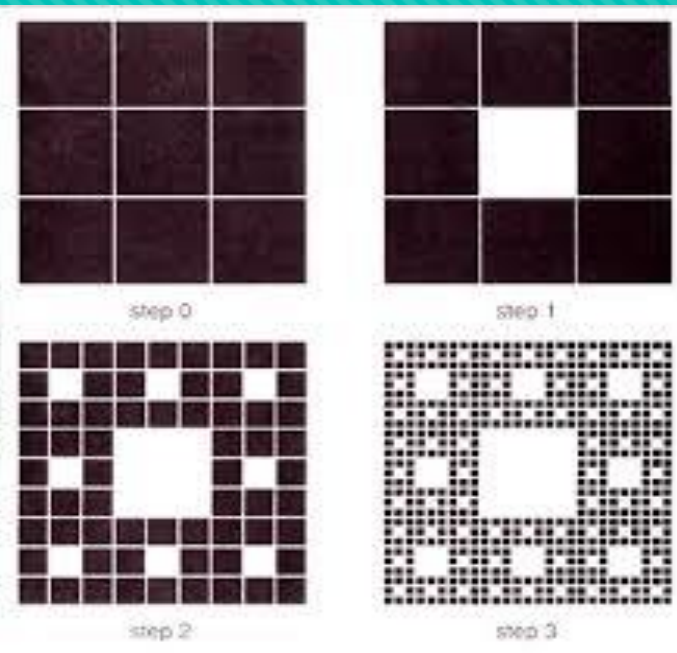
IV. 寫程式畫碎形

(四) 寫程式畫碎形 Drawing Fractals by programming(50 分鐘)

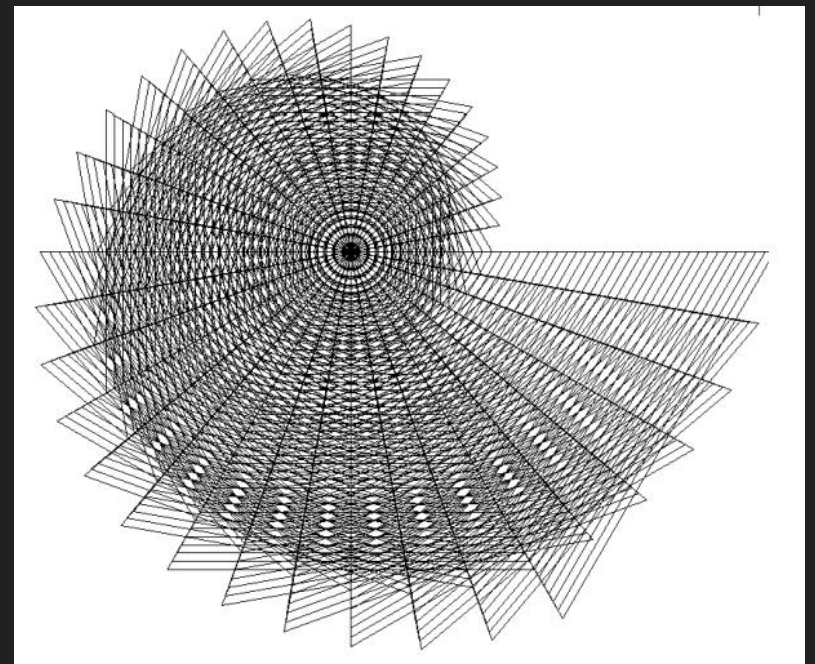
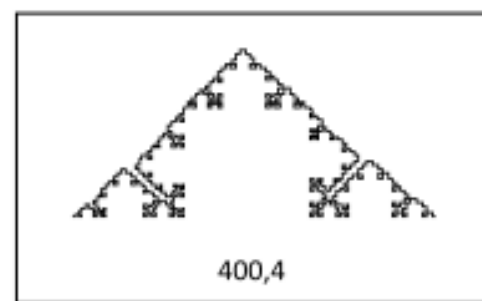
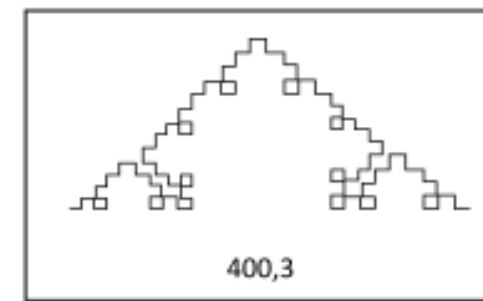
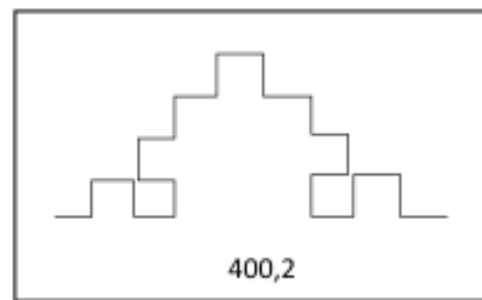
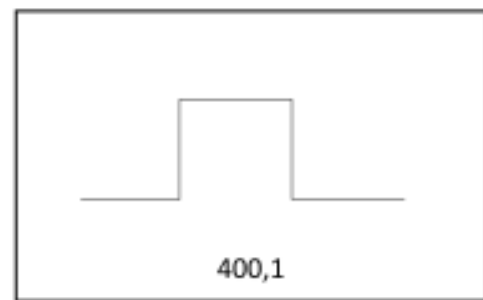
本活動旨在應用傳值函式繪製不同尺度的幾何圖形。引導學生在實作碎形的過程中體會遞迴函式的能力，並依據不同參數的設定實現碎形的自我相似性，完成碎形圖形的繪製。

```
def f(length, depth):  
    if depth == 0:  
        forward(length)  
    else:  
        f(length/3, depth-1)  
        right(60)  
        f(length/3, depth-1)  
        left(120)  
        f(length/3, depth-1)  
        right(60)  
        f(length/3, depth-1)
```





```
def f(t, length, depth):
    if depth == 0:
        t.forward(length)
        return
    else:
        f(t, length/4, depth-1)
        t.left(90)
        f(t, length/4, depth-1)
        t.right(90)
        f(t, length/3.5, depth-1)
        t.right(90)
        f(t, length/4, depth-1)
        t.left(90)
        f(t, length/3.5, depth-1)
```



運算思維概念評量 (I)

10-

I. Pattern Recognition / Model / Abstraction

請以公式或文字描述數列規律性，並完成資料列



i	1	2	3	4	5	6	7	8	9	10	11	12	13
a[i]	3	6	9	12	15	18	21	24	27	30	33	36	39
Rule	$a[i] = 3 + 3(i-1)$												
a[i]	20	18	16	14	12	10	8	6	4	2	0	-2	-4
Rule	$a[i] = 20 - 2(i-1)$												
a[i]	2	2	6	6	8	8	10	10	12	12	14	14	16
Rule	2 的倍數從 2 開始, 重覆寫同一數兩次												
a[i]	1/2	2/3	3/4	4/5	5/6	6/7	7/8	8/9	9/10	10/11	11/12	12/13	13/14
Rule	$a[i] = \frac{i}{i+1}$												

運算思維概念評量 (II)

[3] 織布機花色

	R	G	R			R	G	R			R	G	R			R	G	R	
	G	B	G			G	B	G			G	B	G			G	B	G	
	R	G	R			R	G	R			R	G	R			R	G	R	
	R	G	R			R	R	R			G	G	G			R	G	R	
	G	B	G			R	R	R			G	G	G			G	B	G	
	R	G	R			R	R	R			G	G	G			R	G	R	
	R	G	R			R	G	R			R	G	R			R	G	R	
	G	B	G			G	B	G			G	B	G			G	B	G	
	R	G	R			R	G	R			R	G	R			R	G	R	

Ans:

$$X = \begin{array}{|c|c|c|} \hline R & G & R \\ \hline G & B & G \\ \hline R & G & R \\ \hline \end{array}$$

⇒

$$Y = \begin{array}{|c|c|c|} \hline R & R & R \\ \hline R & R & R \\ \hline R & R & R \\ \hline \end{array}$$

$$Z = \begin{array}{|c|c|c|} \hline G & B & G \\ \hline G & G & G \\ \hline G & G & G \\ \hline \end{array}$$

X X X X 換行 X Y Z X 換行 X X X X

↓

X 4

X Y Z X

X 4.

運算思維概念評量 (III)

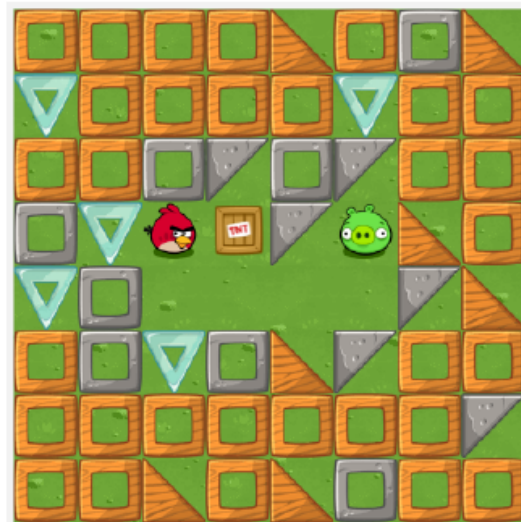
III. Structure Programming 【向左走。向右走】

壞壞小豬偷走 Angry bird 的蛋了，請幫助它來抓小豬吧～

Angry bird 每一次出發，都是面向右邊的喔～

```
moveForward();    //前進
turnRight();      //向右轉
turnLeft();       //向左轉
```

[1] 小豬在這裡，該如何走才能抓住它呢?(小心! 別碰到炸藥囉～) 答案是：_____



(A)

```
moveForward();
turnRight();
turnRight();
moveForward();
moveForward();
turnLeft();
```

(B)

```
moveForward();
turnLeft();
turnRight();
moveForward();
moveForward();
turnLeft();
```

(C)

```
turnRight();
moveForward();
turnLeft();
moveForward();
moveForward();
moveForward();
turnLeft();
moveForward();
```

(D)

```
turnRight();
moveForward();
moveForward();
turnLeft();
moveForward();
moveForward();
turnRight();
moveForward();
```

運算思維概念評量 (III)

III. Structure Programming 【向左走。向右走】

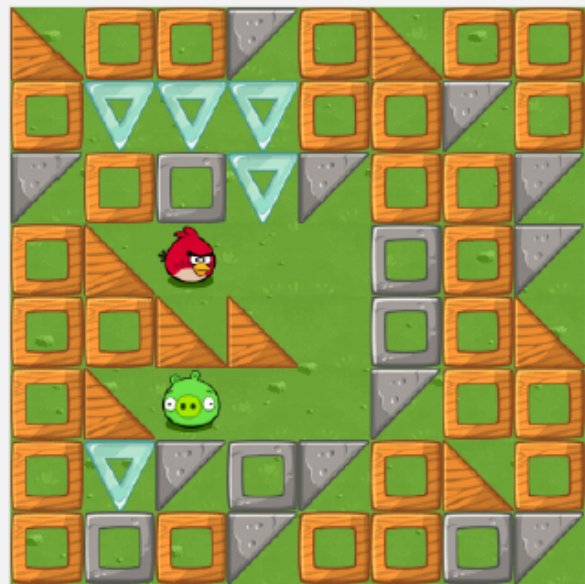
壞壞小豬偷走 Angry bird 的蛋了，請幫助它來抓小豬吧~~

Angry bird 每一次出發，都是面向右邊的喔~~

```
moveForward();    //前進  
turnRight();      //向右轉  
turnLeft();       //向左轉
```

IV. Loop concepts

[2] 小豬躲起來了，該如何走才能抓住它呢？答案是：_____



(A)

```
Repeat(2)  
{moveForward();}
```

(B)

```
turnRight();  
Repeat(2)  
{moveForward();}
```

(C)

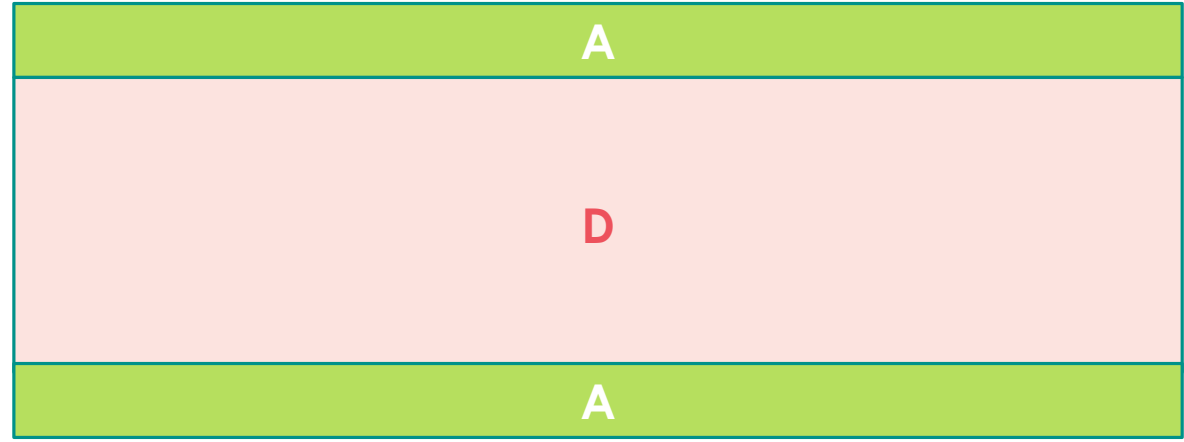
```
Repeat(2)  
{moveForward();}  
turnRight();  
Repeat(2){moveForward();}  
turnRight();  
Repeat(2){moveForward();}
```

(D)

```
moveForward();  
moveForward();  
turnLeft();  
moveForward();  
moveForward();  
turnRight();  
moveForward();  
moveForward();
```


運算思維概念評量 (IV)

V. Function concepts 【自動織布機】



解法：

定義花色塊 <u>A</u>	定義花色塊 <u>B</u>	定義花色塊 <u>C</u>	定義花色塊 <u>D</u>
定義花色塊 A = repeat (2) { repeat (27) { paint G } 換行 }	B = paint R paint Y repeat (5) { paint B } paint Y paint R	C = repeat (9) { paint W }	D = repeat (9) { B C B 換行 }

運用花色塊重新編寫織布機指令

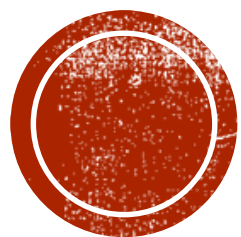
A
D
A

視覺化程式設計 讓抽象思考歷程變得具體

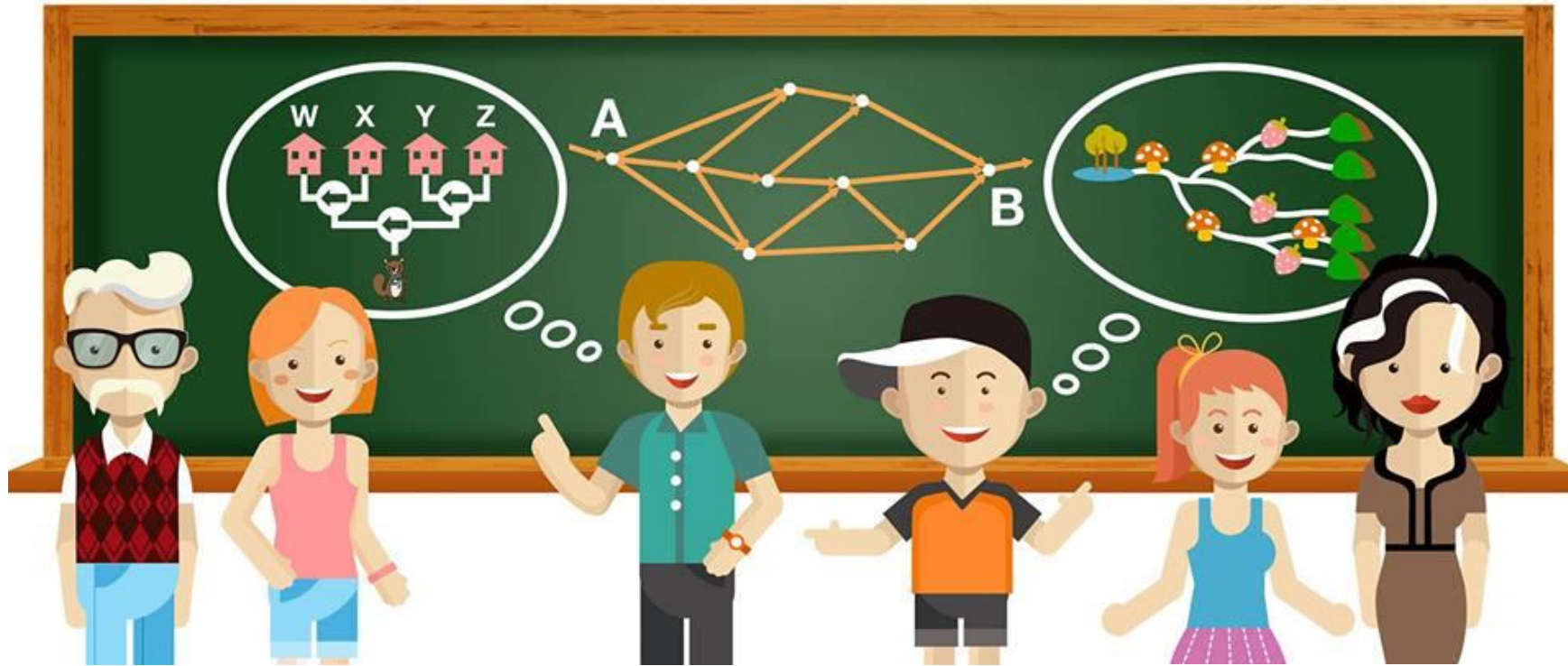
- 傳統教學中模組化程式設計、函式與重複結構的實例偏重於數學應用，多數的學生要直接進行抽象思考並完成任務，難度頗高。
- 本教學模組嘗試設計生活中的音樂與碎形幾何為課程活動，讓學生從聽覺(音樂創作)與視覺(繪圖實作)中實際體會以函式簡化重複樣式，運用模組化程式描述任務可系統化的解決複雜問題的歷程。

活動設計需克服與考慮的問題

- 『音樂創作專題』在試教遭遇到的困難是學生的識譜能力（雖然小學音樂課程有識譜的學習活動），因此，本教學模組提供「五線譜」與「簡譜」兩種形式的樂譜，並建議1-2人一組，讓學生彼此支援，補足識譜落差
- 『碎形幾何繪圖專題』初施行在高中程式設計選修課程，學生已具備C++語言的能力，因此運用的是指令式程式語言MSLOGO與Python Turtle
- 國中生因尚未接觸指令語法，建議以視覺化程式設計工具學習概念
- 以Code.org的「藝術家」關卡任務讓學生從自學中體驗，再以視覺化的MIT Scratch線上程式編輯器，拖接積木的方式完成程式設計



運算思維導向 教材設計參考資源



相關新聞
活動訊息
研習活動
教學資源
社群

...



運算思維

Published by 擲鴨林 [?] · December 25, 2015 at 1:02pm ·

叮叮噹~叮叮噹~

祝大家聖誕快樂 🎅

還記得去年風靡一時的【Frozen 冰雪奇緣】嗎？

快來跟Anna和Elsa在遊戲的過程中一邊感受佳節氣氛，一邊學習程式設計吧！

遊戲入口：<https://studio.code.org/s/frozen/stage/1/puzzle/1>



Frozen - Hour of Code Introduction

Let's use code to join Anna and Elsa as they explore the magic and beauty of ice.

<http://studio.code.org/s/frozen/>

YOUTUBE.COM



資訊科學探索網

聯絡我們

讚 分享 112 人說這讚。成為你朋友中第一個說讚的人。

最新消息 計畫介紹 教材資源 參與教學教師

最新教材資源

- > [相片化妝師](#)
- > [電腦的運作](#)
- > [下一個籃球明星就是你](#)
- > [數位浮水印](#)
- > [小叮嚀如何數數兒？二進位表示法](#)
- > [心想事成：有效率的資料搜尋法](#)
- > [海盜尋費記：有限狀態機簡介](#)
- > [熊貓走迷宮：路線搜尋](#)

2012/3/16

資訊科學與應用專題研習

南區：101/4/3（星期二），國立臺南第二高級中學
 中區：101/4/10（星期二），國立臺中第二高級中學
 東區：101/4/17（星期二），國立花蓮女子高級中學
 北區：101/4/17（星期二），國立臺灣師範大學附屬高級中學
 更多資訊請見[高中資訊學科中心研習活動](#)

2012/3/11

新探索課程來囉

第二期探索課程已製作完成，敬請前往「教材資源」觀看下載。
 國中新課程：[電腦的運作](#)、[相片化妝師](#)

最新教材資源

- > [相片化妝師](#)
- > [電腦的運作](#)
- > [下一個籃球明星就是你](#)
- > [數位浮水印](#)
- > [小叮嚀如何數數兒？二進位表示法](#)
- > [心想事成：有效率的資料搜尋法](#)
- > [海盜尋費記：有限狀態機簡介](#)
- > [熊貓走迷宮：路線搜尋](#)
- > [圖片搜得妙](#)
- > [網路駭客追緝令](#)
- > [你猜我在寫什麼？手寫數字辨識](#)
- > [聽言識人：語音辨識](#)

more



主題式探索課程

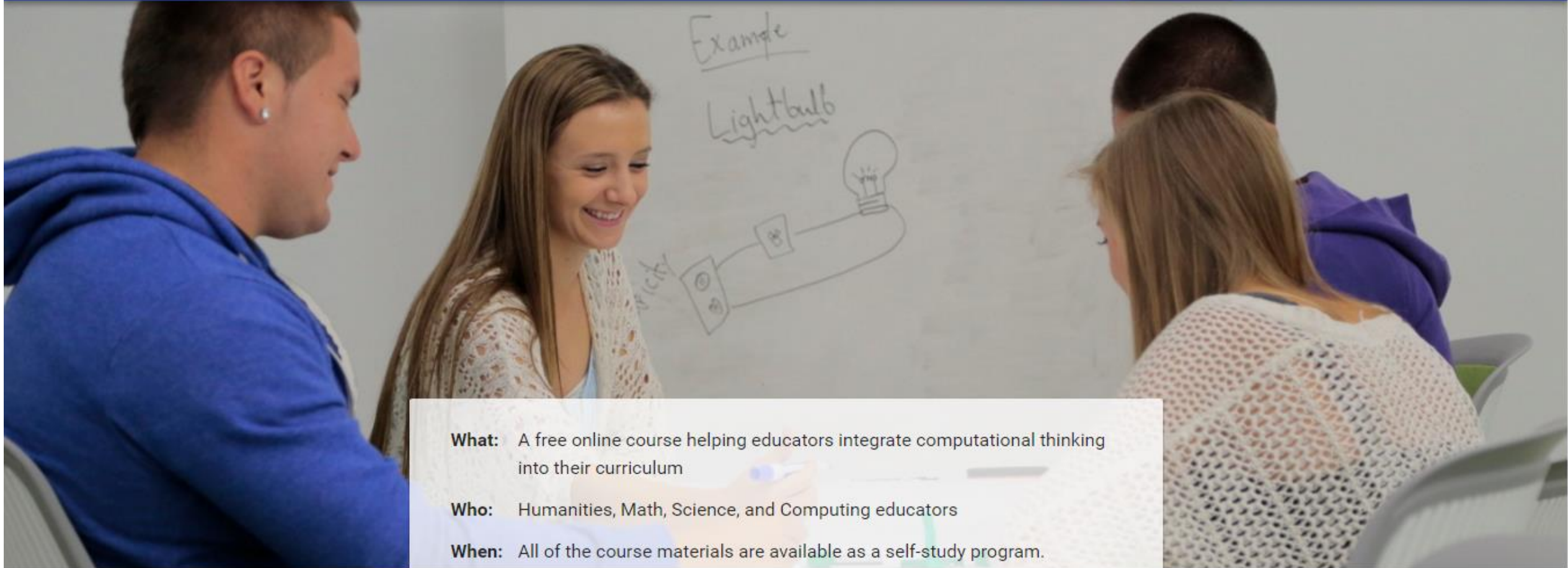
○ 國中

- 小叮噹如何數數兒：二進位表示法
- 心想事成：有效率的資料搜尋法
- 海盜尋寶記：有限狀態機簡介
- 路線搜尋與問題解決

○ 高中

- 手寫數字辨識
- 圖片搜得妙
- 網路駭客追緝令
- 聽音識人：語音辨識





What: A free online course helping educators integrate computational thinking into their curriculum

Who: Humanities, Math, Science, and Computing educators

When: All of the course materials are available as a self-study program.

https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true



改變世界的九大演算法： 讓今日電腦無所不能的最強概念

- **Nine Algorithms That Changed the Future: The Ingenious Ideas That Drive Today's Computers**
- **John MacCormick**
- 引言：讓今日電腦威力無窮的神奇概念



九大演算法

1. 搜尋引擎的索引 (*search engine indexing*)
2. 網頁排序 (*page rank*)
3. 公鑰加密 (*public-key cryptography*)
4. 錯誤更正碼 (*error-correcting codes*)
5. 模式辨識 (*pattern recognition*，如手寫辨識、聲音辨識、人臉辨識等等)
6. 資料壓縮 (*data compression*)
7. 資料庫 (*databases*)
8. 數位簽章 (*digital signature*)
9. 一種如果存在的話將會很了不起的偉大演算法，並探討電腦能力的極限



作者挑選演算法的標準

1. **每天會被一般電腦使用者用到的演算法**
2. **必須能解決現實世界的具體問題**，例如壓縮檔案，或是透過繁忙的網路連線精準無誤地傳輸檔案。這項標準把大學的電腦科學課程中最重要的一些演算法排除在外，包括快速排序法（quicksort）等排序演算法、Dijkstra的最短路徑演算哪等圖形演算法，以及雜湊法（hash table）等資料結構，這些無疑都是了不起的演算法且符合第一項標準，但這些演算法是可被應用到各種問題上的通用演算法，本書將聚焦在某些特定問題的演算法上，因為這些演算法要解決的問題，對一般電腦使用者來說較為明確。
3. **主要是與電腦科學理論有關的演算法**。作者聚焦在電腦科學理論，是由於作者撰寫本書的部分動機在於一般人對電腦科學的認知失衡，人們普遍認為電腦科學是跟寫程式（軟體）和機器設計（硬體）有關，事實上，電腦科學中的一些最美的概念是抽象的、理論性的概念。
4. **美、簡潔、優雅**。本書中所提的演算法，其精隨都是利用一些聰明的「技法」（trick）來解決問題。



運算思維各領域教學模式

- Barr 與 Stephenson (2011) 為資訊科學、數學、科學、社會科學及語言藝術提供運算思維的應用範例，教材編纂者可根據教學主題發展適合的教材與教學活動供學生練習運算思維
- **Computer Science Unplugged** (Bell、Alexander、Freeman、Grimley, 2009) 強調不使用運算工具也可以學習資訊科學的活動，其目的是為彰顯「思維」之重要性。透過運算思維的學習，學生可在學習其他學科領域時，擁有更結構化、程序化、抽象化的思考模式，但必須強調的是，其他領域的學習或不使用運算工具的學習活動雖亦可輔助運算思維的培養



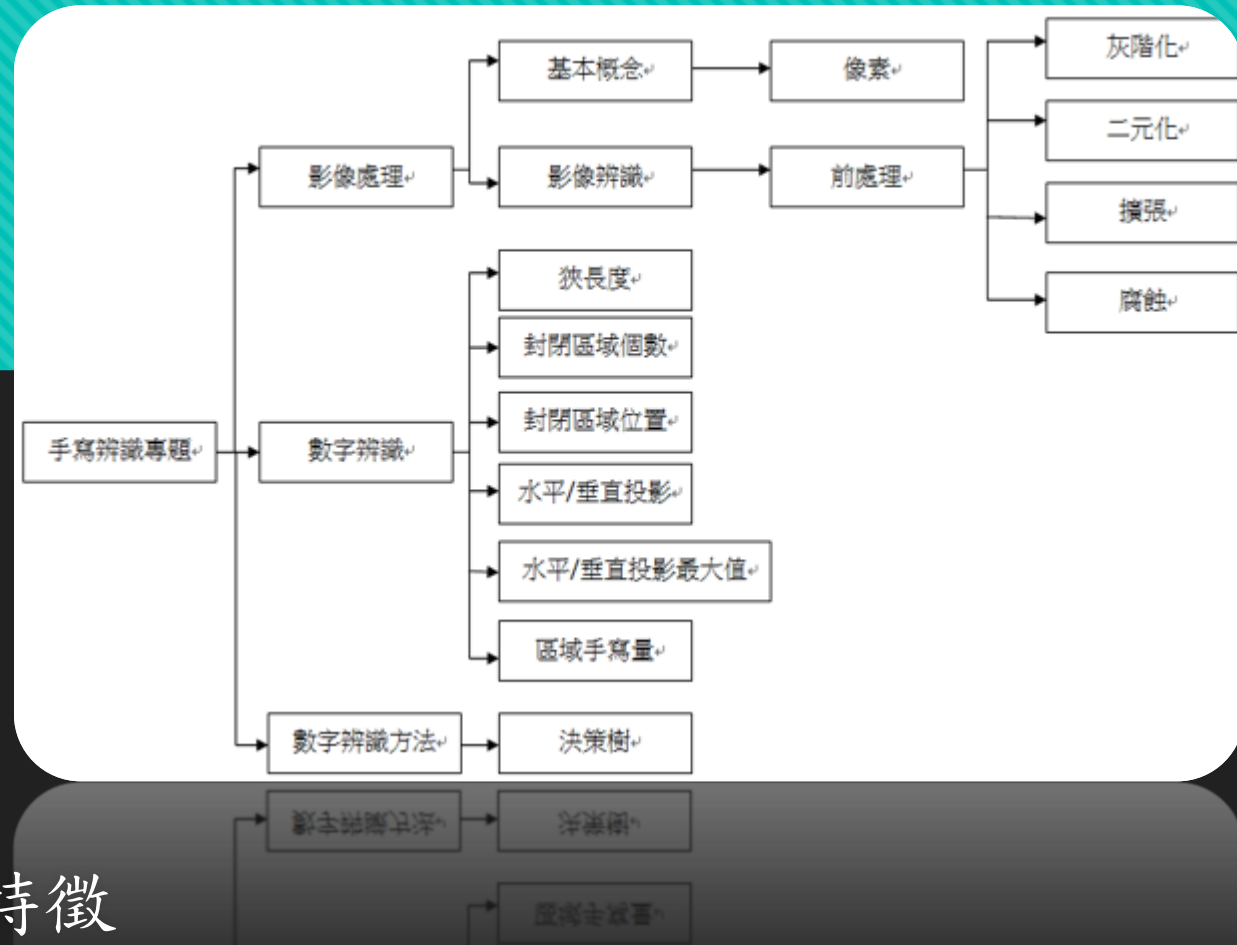
應用運算思維於高中資訊教學設計 ~手寫辨識系統

賴和隆 臺北市立中正高級中學教師

臺灣師範大學 李忠謀 教授

學習目標

1. 了解數位影像的基礎知識
2. 了解影像處理的基本概念
3. 知道數位影像前處理的方法
4. 認識用以分辨手寫數字的影像特徵
5. 了解什麼是決策樹，以及決策樹是如何去分辨不同特徵的數字
6. 能夠自行建構用以分辨手寫數字之決策樹（產出）



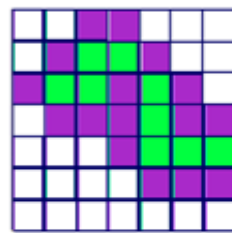
運算思維歷程

- 經由設計，將相關概念分解、抽象化及模組化
- 透過系統中統計圖表與決策樹的方式呈現，將電腦內部運算具體化。

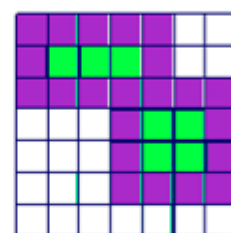
學習實驗單：

目標：能夠判斷或產生影像經過擴張前處理完的結果。

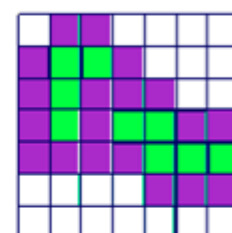
(一)請將各圖經過擴張前處理完的結果，填寫在表格內。



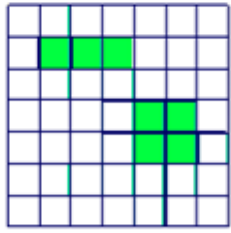
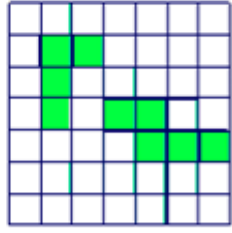
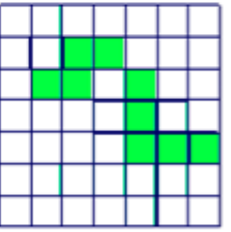
(a)



(b)



(c)

原圖			
結果代號			

手寫數字辨識系統操作手冊

A. 環境設定

1. 請下載"HandwritingRecognition.jar" 程式檔及"com.zip" 壓縮檔。
2. 將"com.zip" 檔解壓縮。
3. 將"HandwritingRecognition.jar" 程式檔及"com" 資料夾儲存在 D 槽內。
4. 請注意，儲存的路徑中不能有中文。
5. 必須先確保電腦可以上網。

B. 執行步驟

1. 請在"HandwritingRecognition.jar" 程式檔上點選滑鼠左鍵兩下，就會出現圖 1 的執行畫面。

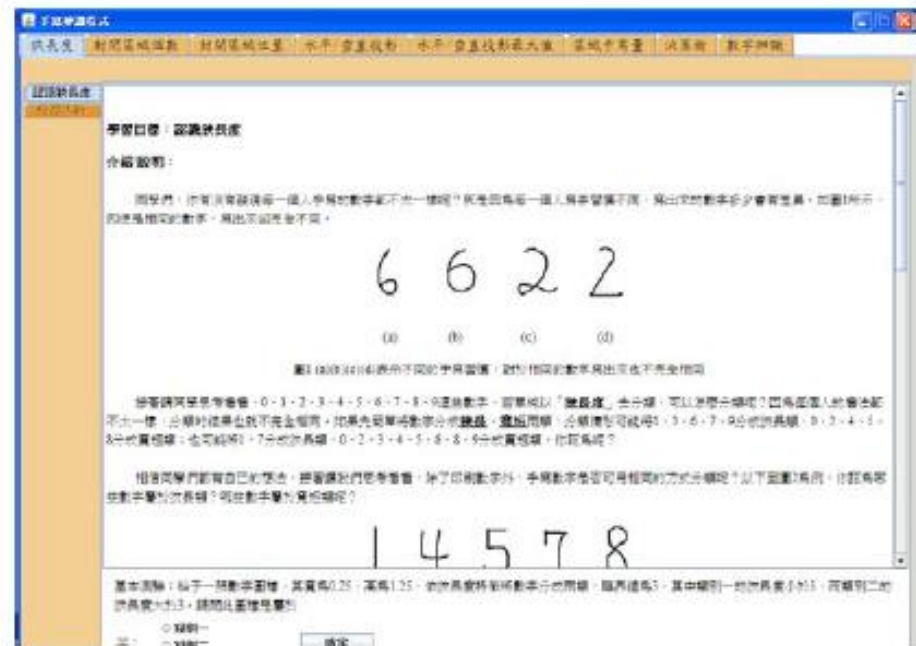


圖 1 手寫數字辨識系統的執行畫面

2. 請選取欲執行的分頁。分頁列表如下圖 2 所示。

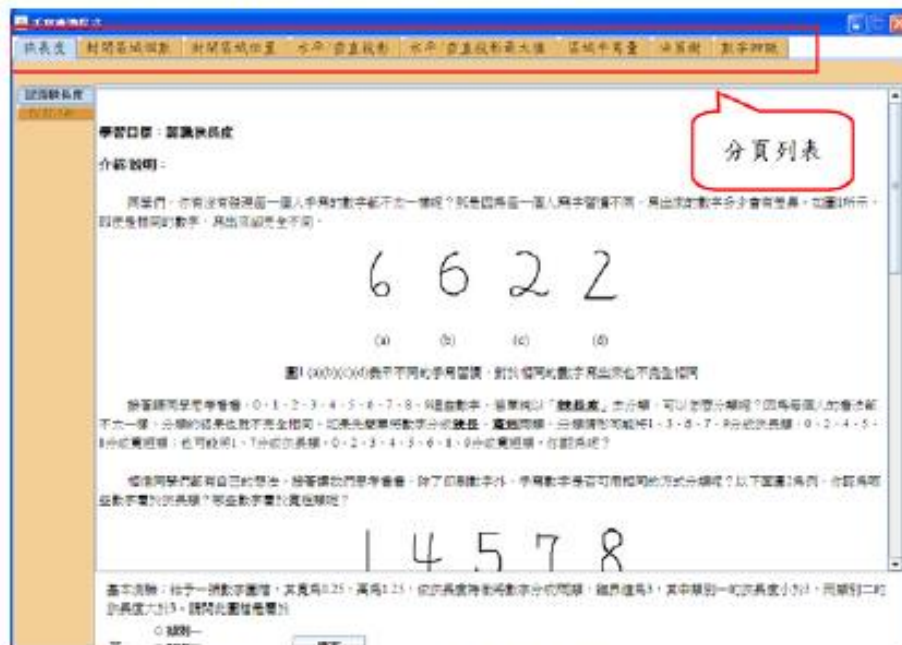
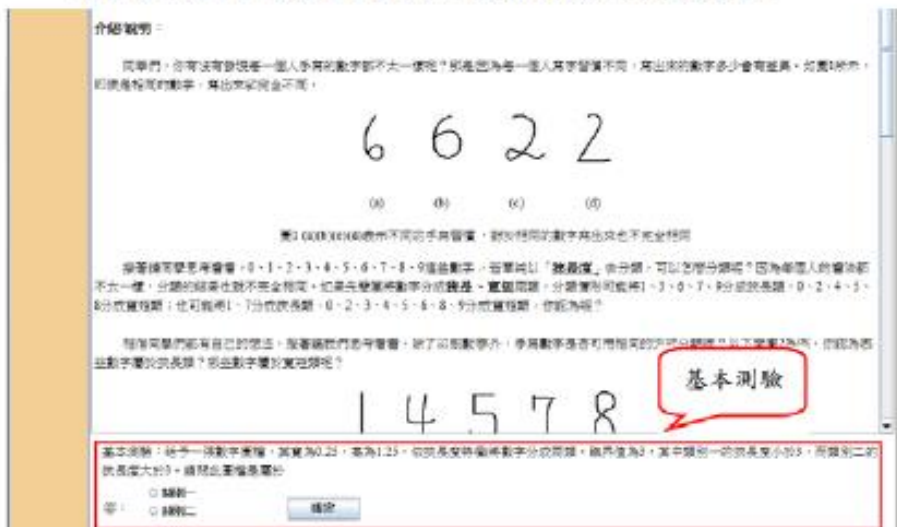
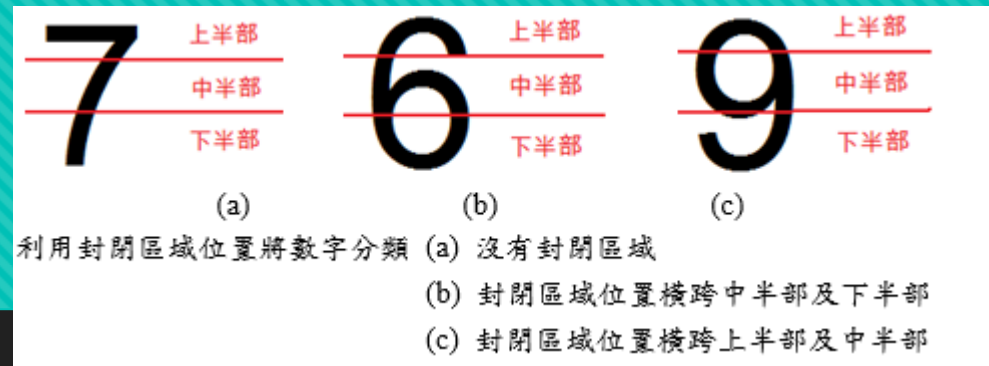


圖 2 紅色框框處為分頁列表

3. 欲進行學習活動之前，請先回答分頁下面的"基本測驗"。如下圖 3 所示。若回答正確，即可進行學習活動；若回答錯誤，請重新挑戰。



運算思維設計分析



一、觀察既有現象形成問題 (Decomposition)

在教學設計中，先教導「像素」、「二元化」、「灰階化」、「擴充」、「腐蝕」等影像數位化及影像處理的基本概念，奠立基本知識。再透過大量的手寫數字經由掃描輸入，由同學觀察既有現象，並經由先前學習的影像處理基本概念形成問題。

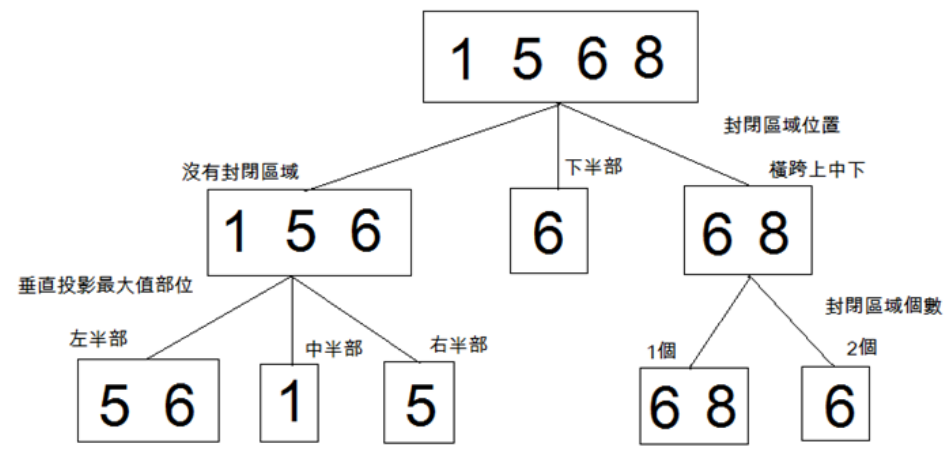
二、有邏輯地組織和分析資料 (Pattern Recognition)

在觀察的過程中練習分類、嘗試藉由分析找出規律，如封閉區域位置、封閉區域個數等。

三、透過抽象化表達資料，例如模型或模擬 (Abstraction)

其中「二元化」、「灰階化」、「擴充」、「腐蝕」等作法，實務上都有其演算法，但是，在操作時，並沒有介紹真正的演算法或撰寫程式碼，而是將其模組化。再以「狹長度」為例，其以每一個數字中高度與寬度的比例大小作為數字的特徵，這就是一種資料表達的抽象化，如此，可以針對我們所要辨識的手寫數字進行量化工作，並設定臨界值，就可以讓我們將每一個手寫數字依據臨界值進行分類，而「封閉區域位置」、「封閉區域個數」也都是抽象化表達手寫數字資料的方法。

運算思維設計分析



(一) 透過演算法將解決方法自動化 (Algorithm Design)

手寫辨識的依據，來自於大量手寫資料抽象化的統計值，並利用建立決策樹的方法，進行數字辨識的工作，如此的步驟與方法，透過學習系統的設計讓學生藉由操作了解其演算方式及體會大量計算透過計算機達成的效率，並理解解決方法自動化的實踐。

(二) 確認、分析、實作可行的解決方案，以達到最有效益的步驟與資源之整合為目標

在教案設計中，除了經由系統的設計與學習手冊引導進行，最後更經由使用者親自建立決策樹已進行數字辨識，整合既有的步驟與資源，讓學生可以確認、分析、實作可行的解決方案。

(三) 一般化和轉化這個問題解決過程到各式各樣的問題

在整個課程的最後，若時間允許，嘗試要學生思考26個英文字母如果要由電腦辨識，既有的方法是否夠用，是否需要建立新的模組。又同樣是辨識，人臉辨識與交通號誌辨識又將會面臨如何不同的處理方式，能否轉化手寫數字辨識的運算思維，或者要建立新的思考方式！

結語-素養導向教材開發

- **學習**是以**素養培養**為目標
- 不是侷限於某一主題或單元的知識或能力指標
- **教材**設計亦當如此
- **強調 big ideas**
- 才是內化素養的學習、思維能力的培養



結語-UNDERSTANDING BY DESIGN

重視理解的課程設計

重要概念 (Big ideas)

- 學科內容的「核心」概念
- 能有效連結學科內容的事實與技能
- 教學設計時不能只將知識內容照本宣科
- 而是要解構、綜合後再理解、建構

主要問題 (Essential questions)

- 生活中經常面對與解決的重要問題
- 學習者藉由主要問題來掌握big idea並做出相關決策
- 切中學科知識主題與引起學習動機的問題

UbD是美國的教育專家Grant Wiggins與Jay McTighe於1998年所提出

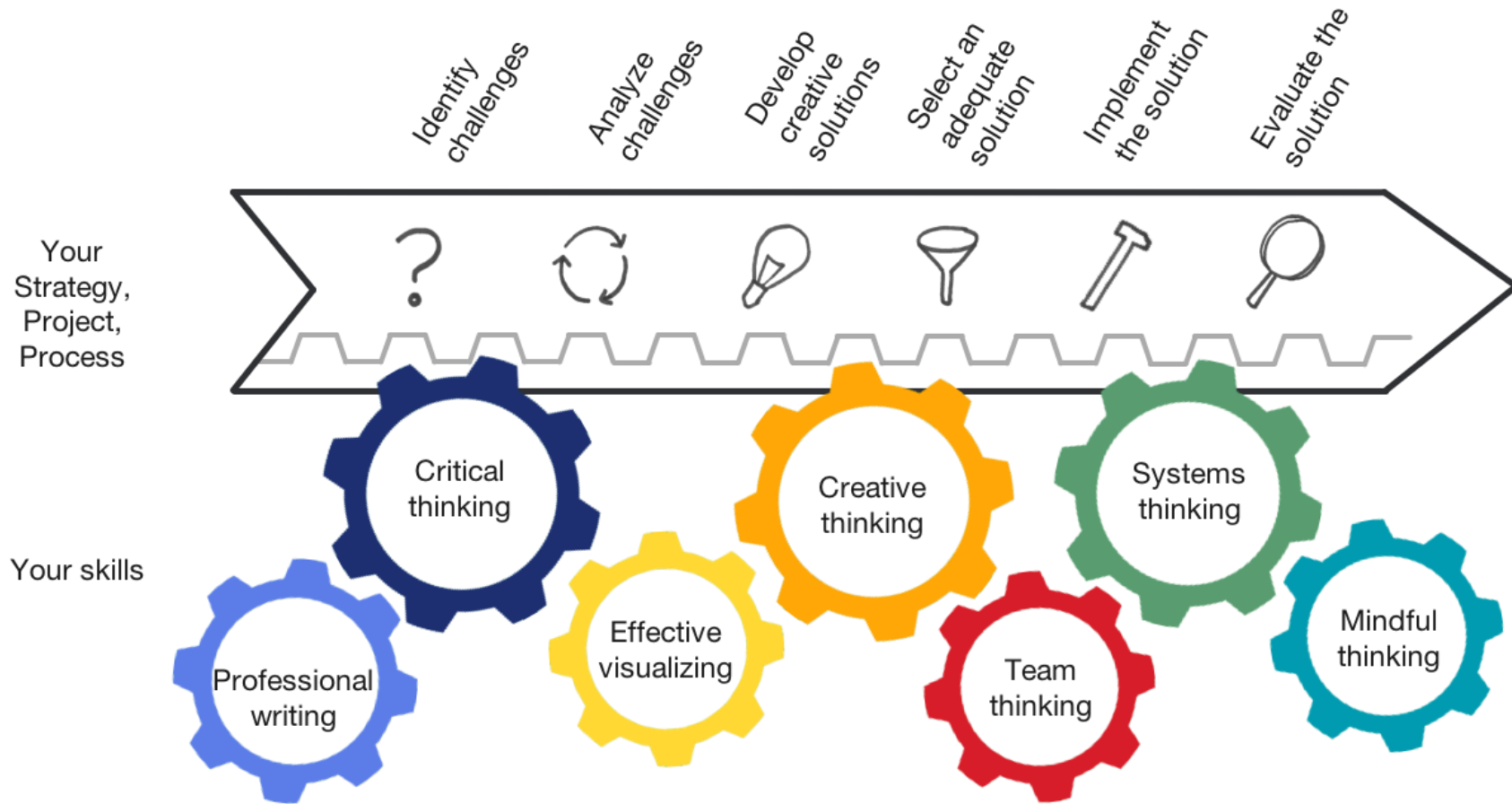
利用Big ideas 引導學習者積極學習



REFERENCE

- Barr, V., & Stephenson, C. (2011). *Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?* ACM Inroads, 2, 48–54. Retrieve from http://www.amanyadav.org/CEP991A/wp-content/uploads/2014/08/Barr_Stephenson_2011.pdf
- Wing, J. M. (2006). *Computational thinking*. Communications of the ACM, 49(3), 33-35. Retrieve from <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- 林育慈、吳正己（2016）。運算思維與中小學資訊科技課程。國家教育研究院教育脈動電子期刊，201608(6)。取自 <http://pulse.naer.edu.tw/Home/Content/02287aac-dc26-4ad4-b87e-2881e942dc16>
- 賴和隆（2016）。應用運算思維於高中資訊教學設計之分享。國家教育研究院教育脈動電子期刊，201606(6)，195-206。取自 <http://pulse.naer.edu.tw/Home/Content/6fe1eedf-10a1-4e1e-890e-dbbec8ce0647>
- 國高中資訊科學探索課程學習活動（2012）。取自 <http://csexp.csie.ntnu.edu.tw/>（資訊科學探索網）
- Google for Education(2015). *Exploring computational thinking*. Retrieved from <https://www.google.com/edu/resources/programs/exploring-computational-thinking/>





In our workshops, we help you tackle your challenges with efficiency

&

And strengthen your skills that are decisive for managerial success

