

運算思維導向資訊科技教學

Computational Thinking and Computer Science Education

第五屆師資培育國際學術研討會-各科教材教法
【素養導向師資培育工作坊主題演講】

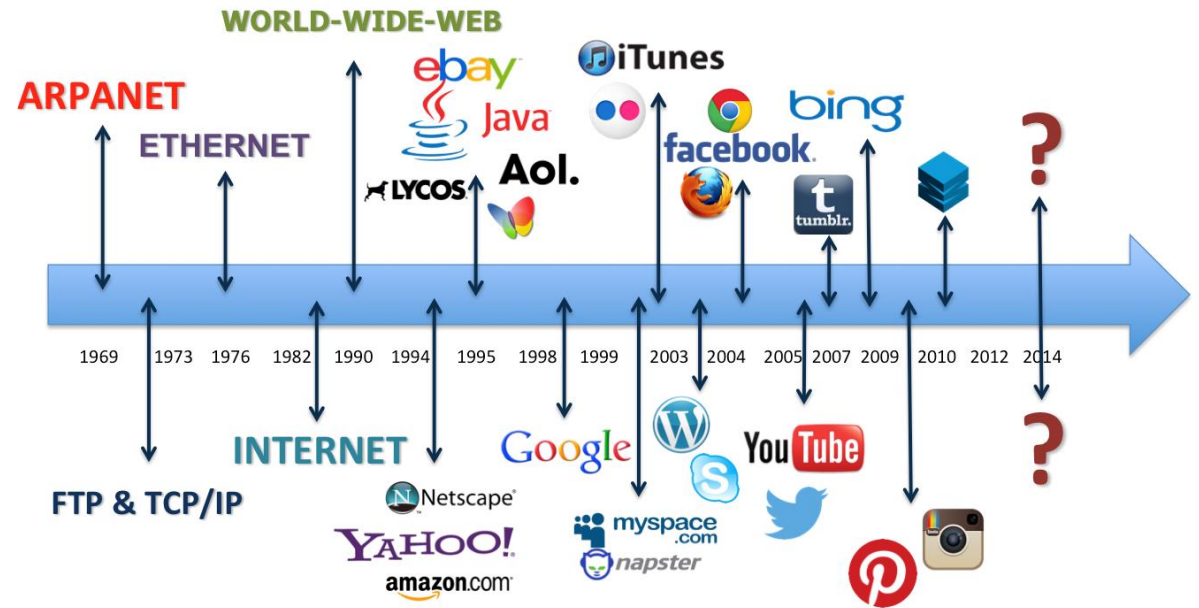
場次一 資訊科技

臺北市立第一女子高級中學 陳怡芬

yfchen@gapps.fg.tp.edu.tw

Since 1994 ~

- 北一女中 1995



所由資訊教師...

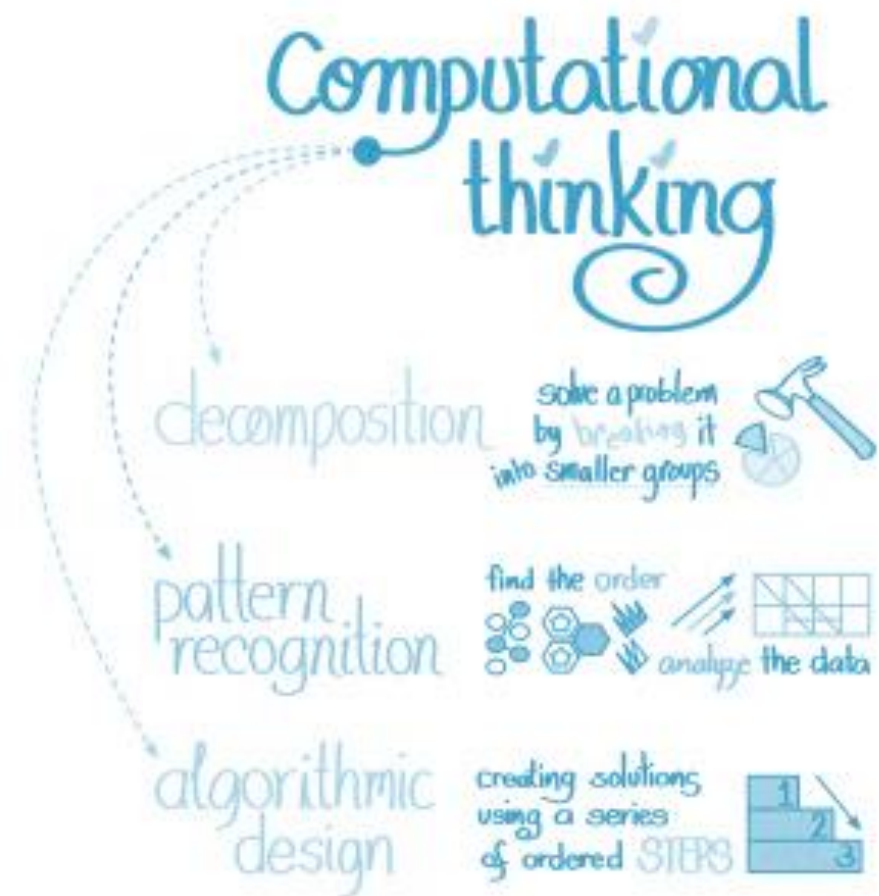
充滿新奇與變化的教學生涯
強化。進化。強化。進化。。

COURSES

- 資訊科技概論-課綱內容(VB,C,C++,APP Inventor, GameMaker...)
- 數理資優班專題研究-程式設計(C/C++/MatLab)、資料結構、演算法、研究方法、研究寫作
- 資訊研究社-程式設計(VB,Java,APP Inventor, C, C++, Python)、網頁設計(HTML,PHP, CSS...)、網管
- 競賽培訓課程-進階程式設計(C)、資料結構、演算法、解題技巧
- 程式設計選修(C/C++)
- 特色選修-問題解決與程式設計(C/C++/Python/MSLogo/Scratch...)
- 特色選修-運算思維與程式設計(C/C++/Python/Scratch...)
- 教師研習 – 基礎電腦應用、基礎程式設計、網頁設計、網站架設(MOODLE, Blog)、教學網站管理、資訊融入教學應用...

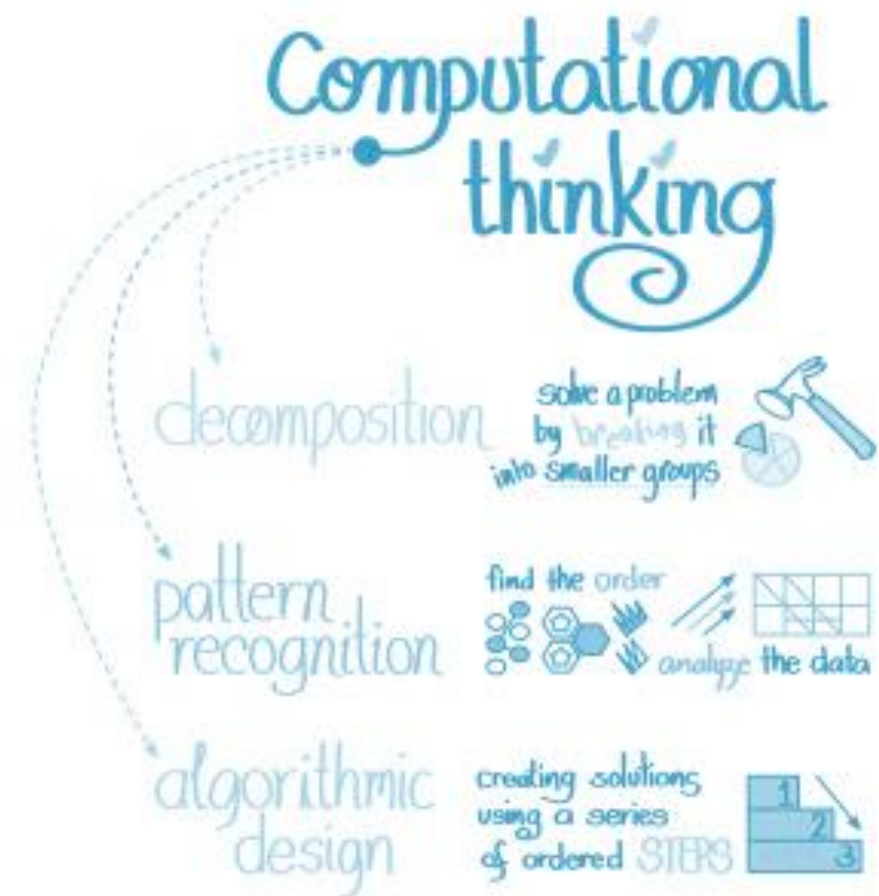
OUTLINE

- 資訊科技教育發展
- 運算思維概念與教學重點
- 生活中的運算與思考
- 運算思維導向課程設計
- 運算思維教材資源

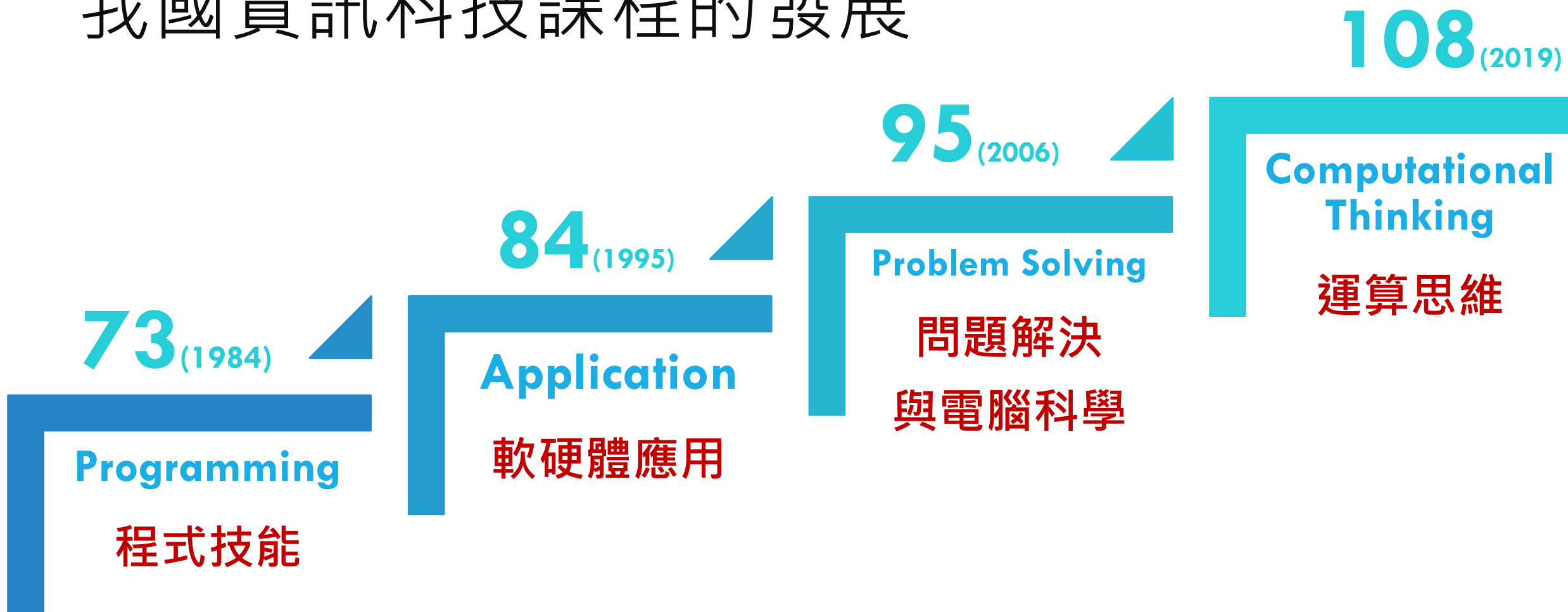


OUTLINE

- 資訊科技教育發展
- 運算思維概念與教學重點
- 生活中的運算與思考
- 運算思維導向課程設計
- 運算思維教材資源



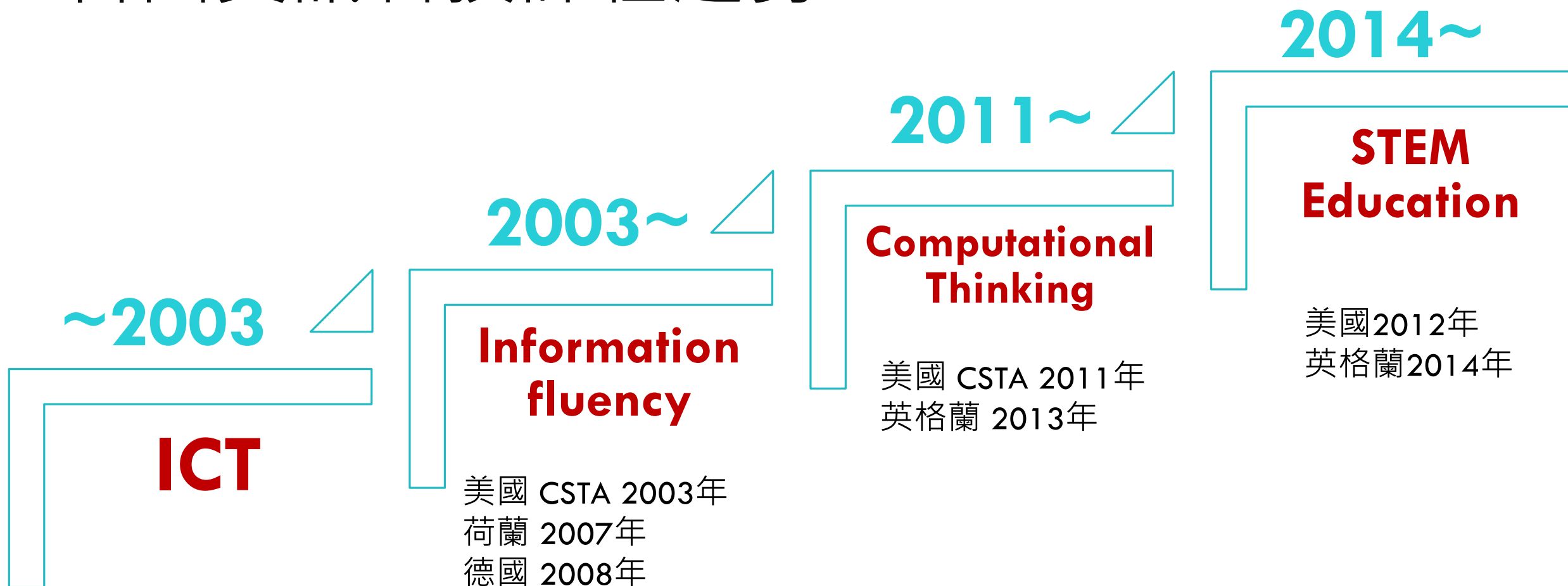
我國資訊科技課程的發展



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

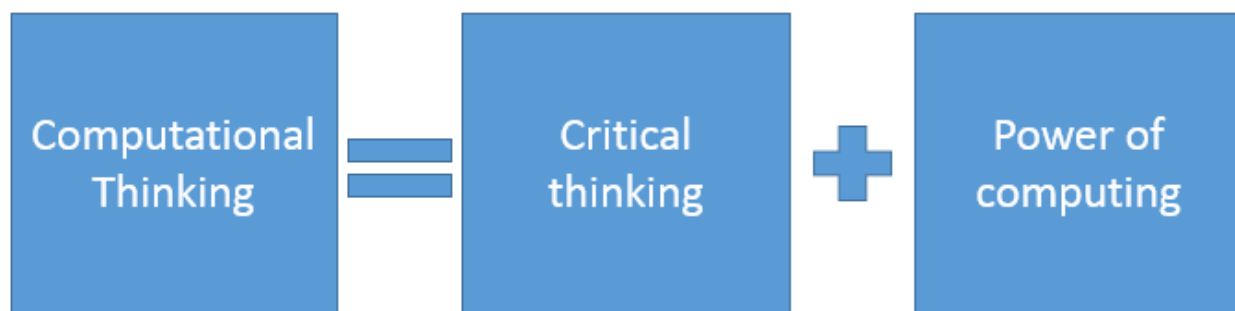
(吳正己, 2010)

各國資訊科技課程趨勢

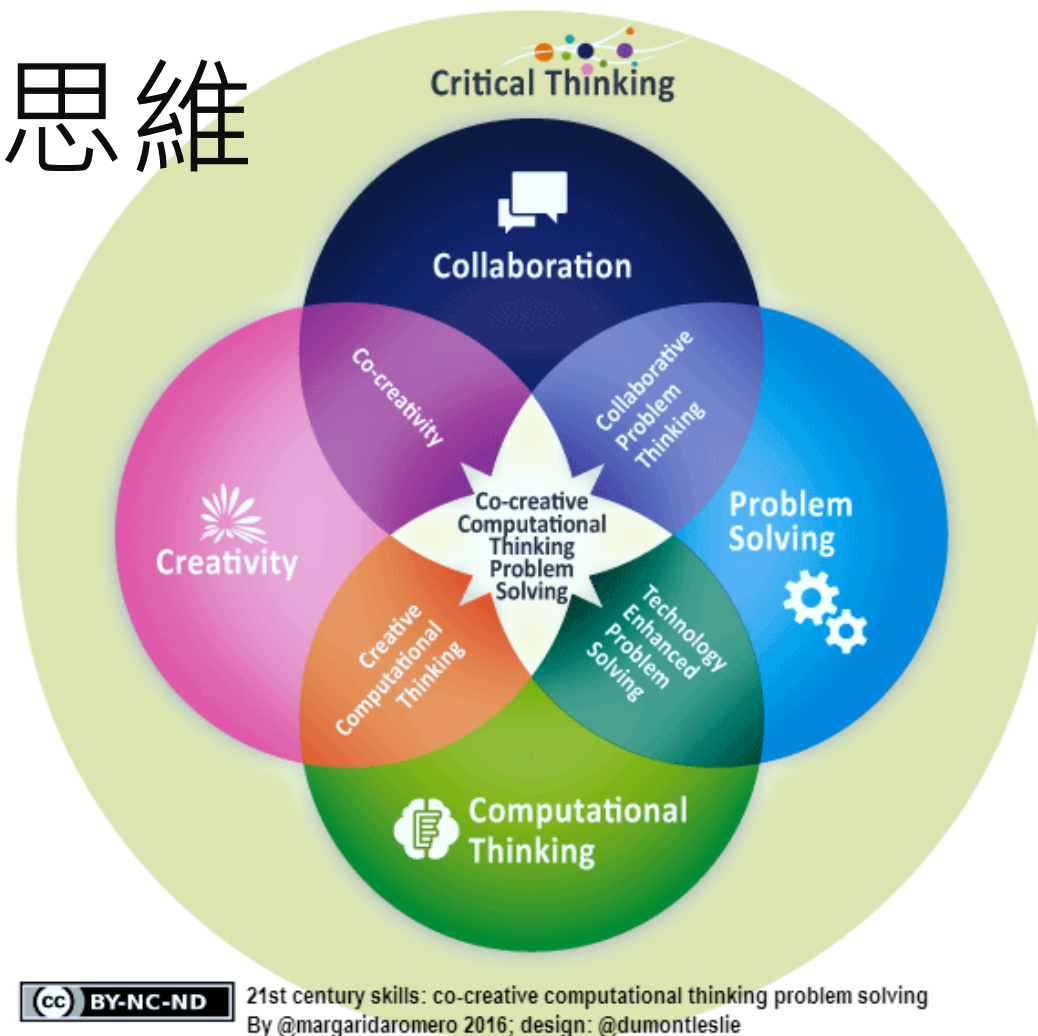


隨著資訊科技的發展與教育理念的變遷，各國不斷修正資訊科技教育的目標與內涵

新課綱核心素養 - 運算思維

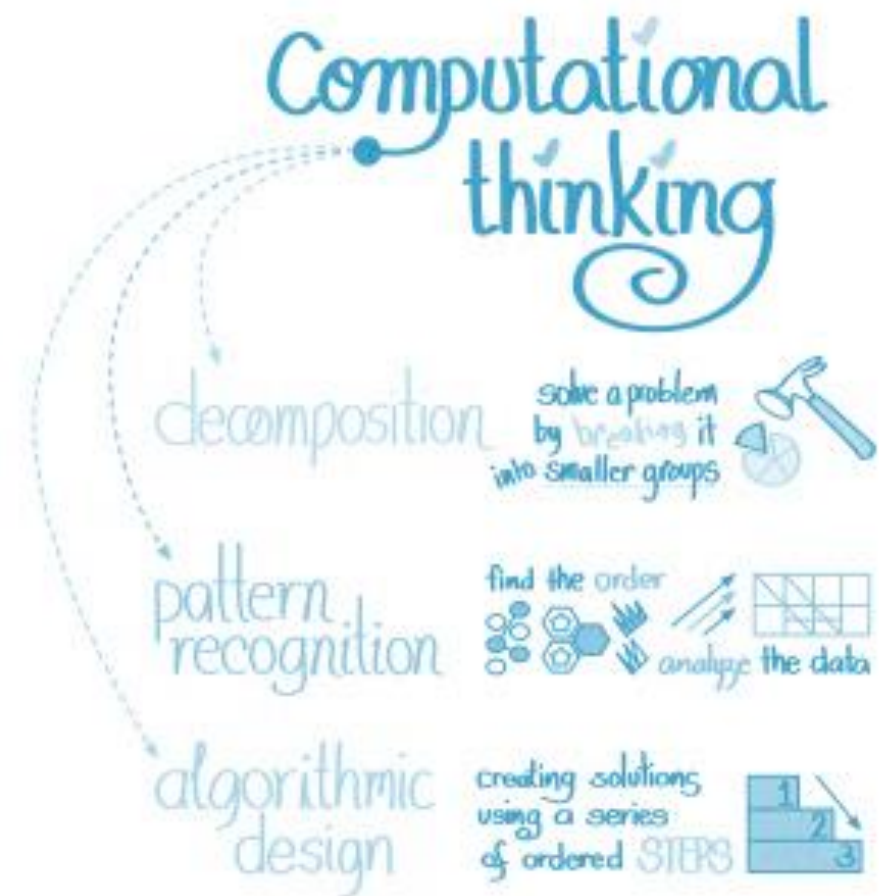


- ◆ 透過電腦科學相關知能的學習
培養邏輯思考、系統化思考等運算思維
- ◆ 藉由資訊科技之設計與實作
增進運算思維的應用能力、解決問題能力、團隊合作以及創新思考



OUTLINE

- 資訊科技教育發展
- 運算思維概念與教學重點
- 生活中的運算與思考
- 運算思維導向課程設計
- 運算思維教材資源



COMPUTATIONAL THINKING

DECOMPOSITION

Breaking big problems into smaller, easier to manage problems



PATTERN RECOGNITION

Analyze & look for a repeating sequence



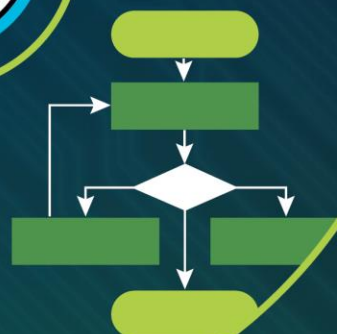
Remove parts of a problem that are unnecessary and make one solution work for multiple problems

ABSTRACTION



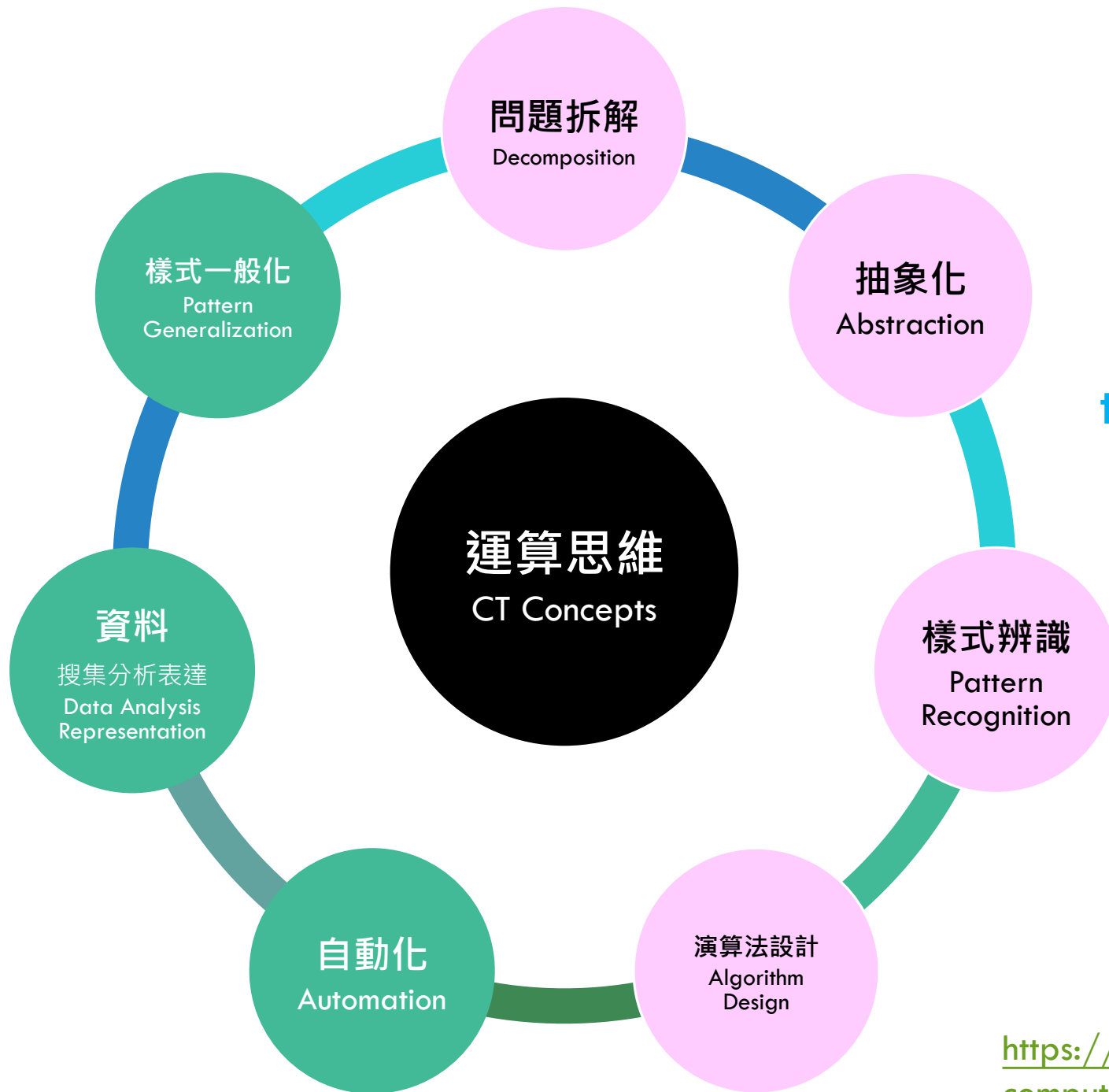
Step-by-Step instructions on how to do something

ALGORITHM DESIGN



“**Computational Thinking** is the **thought processes** involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.”

— Cuny, Snyder, Wing



CT concepts are the **mental processes** (e.g. abstraction, algorithm design, decomposition, pattern recognition, etc) and **tangible outcomes** (e.g. automation, data representation, pattern generalization, etc) associated with solving problems in computing.

Computational Thinking
for Educators

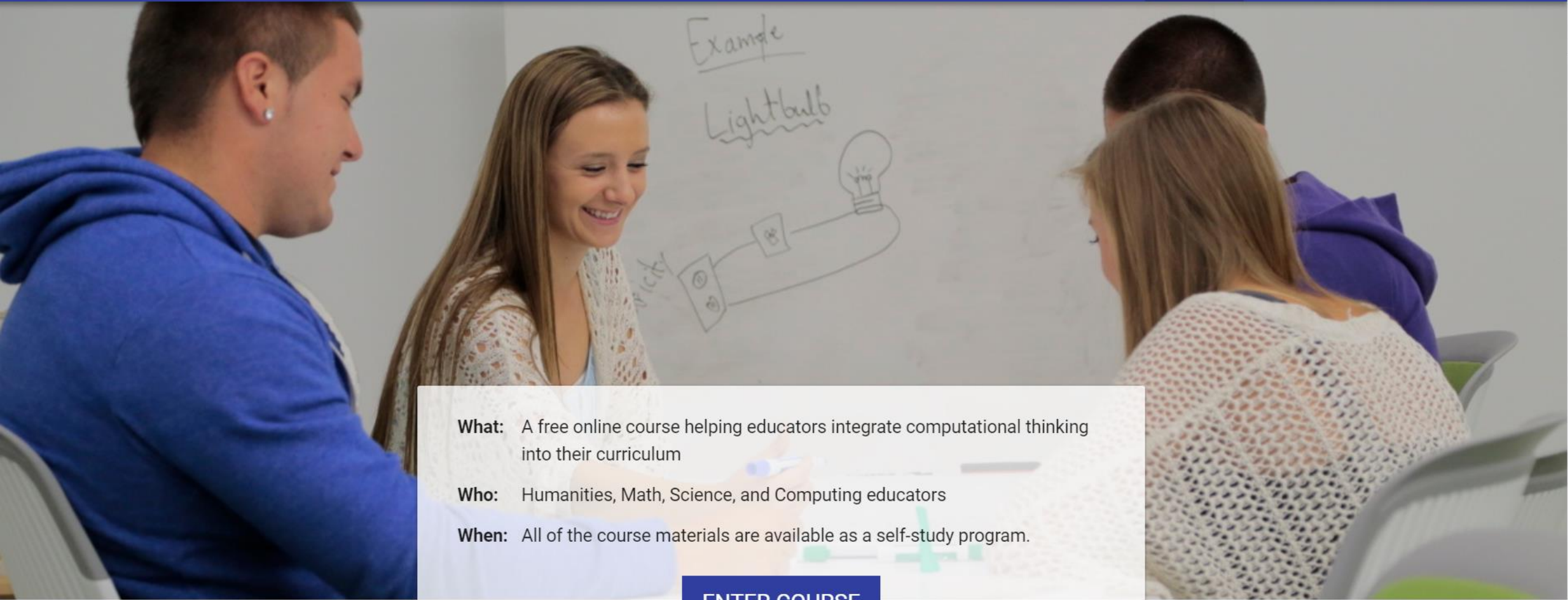


<https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>



COMPUTATIONAL THINKING COURSE WITH GOOGLE

☰ Menu



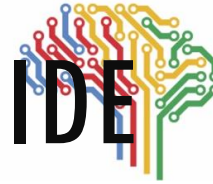
What: A free online course helping educators integrate computational thinking into their curriculum

Who: Humanities, Math, Science, and Computing educators

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

[ENTER COURSE](#)

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



COMPUTATIONAL THINKING CONCEPTS GUIDE

CONCEPTS, DEFINITIONS, AND TEACHING TIPS

- **Decomposition** is breaking down data, processes, or problems into smaller, manageable parts
- **Pattern Recognition** is observing patterns, trends, and regularities in data
- **Abstraction** is identifying and extracting relevant information to define main idea(s)
- **Algorithm Design** is creating an ordered series of instructions for solving similar problems or for doing a task
- **Automation** is having computers or machines do repetitive tasks
- **Data Collection** is gathering information
- **Data Analysis** is making sense of data by finding patterns or developing insights
- **Data Representation** is depicting and organizing data in appropriate graphs, charts, words, or images
- **Pattern Generalization** is creating models, rules, principles, or theories of observed patterns to test predicted outcomes

COMPUTATIONAL THINKING COURSE WITH GOOGLE

Introducing Computational Thinking

[What is Computational Thinking?](#)
[How the Course Works](#)

Exploring Algorithms

[Overview](#)
[CS: Traveling](#)
[Humanities: Words Over Time](#)
[Math: Cellular Automata](#)
[Science: Genomics](#)
[Lesson](#)

Finding Patterns

[Overview](#)
[CS: Data Compression](#)
[Humanities: Music](#)
[Math: Turtle Geometry](#)
[Science: Classifying](#)
[Lesson](#)

Developing Algorithms

[Overview](#)
[CS: Towers](#)
[Humanities: Chatbot](#)
[Math: Calculator](#)
[Science: Bouncing Ball](#)
[Lesson](#)

Applying Computational Thinking

[Project Overview](#)
[Project, Part I](#)
[Project, Part II](#)
[Conclusion](#)
[Final Survey](#)

Computational Thinking

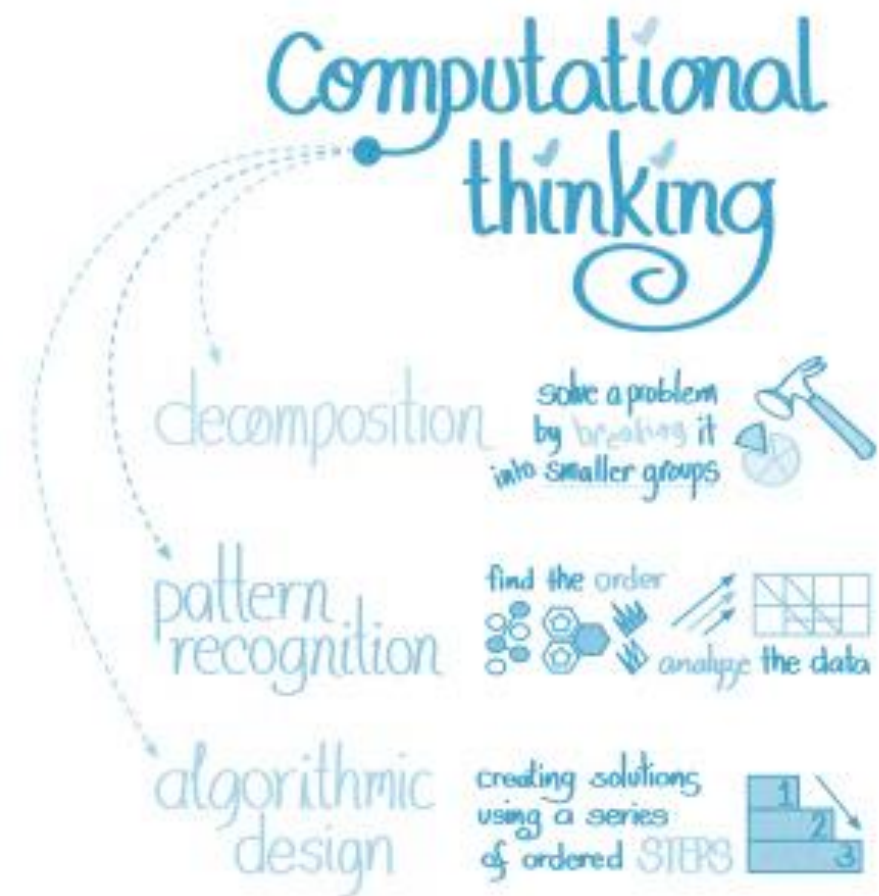


for Educators

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

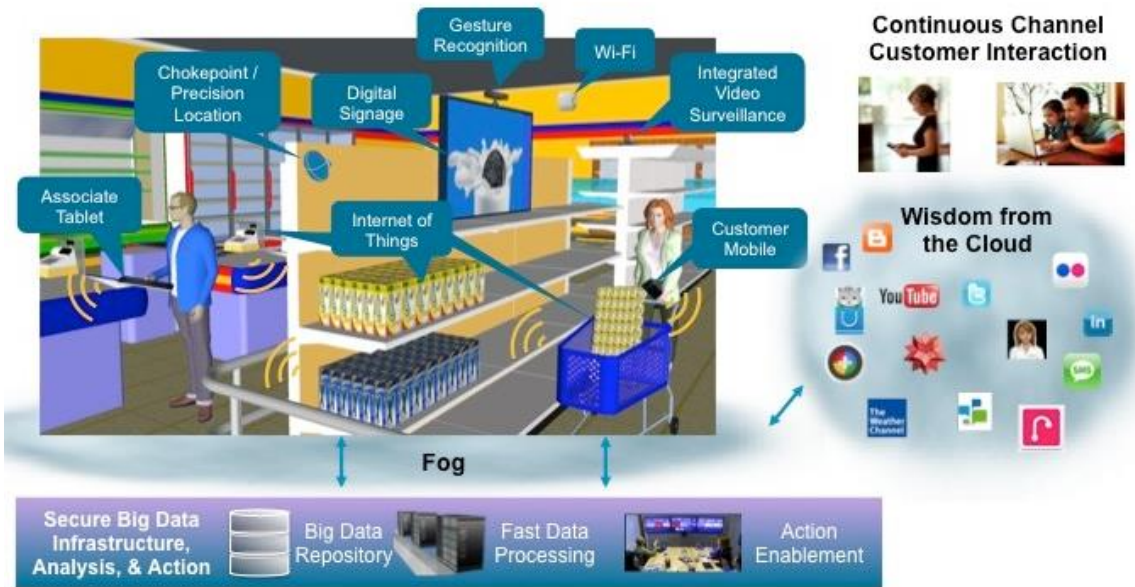
OUTLINE

- 資訊科技教育發展
- 運算思維概念與教學重點
- 生活中的運算與思考
- 運算思維導向課程設計
- 運算思維教材資源



生活中無處不在的運算與思考

Capturing Store Insights for Timely Engagement



Paul Schottmiller - March 19, 2013

Amazon warehouse robots

<https://www.youtube.com/watch?v=3eQAFVetNGI>

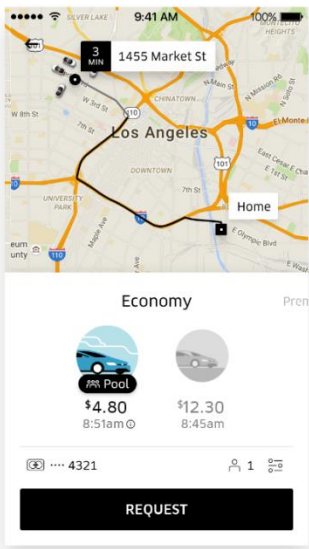


REQUESTING A RIDE IS AS EASY AS

1

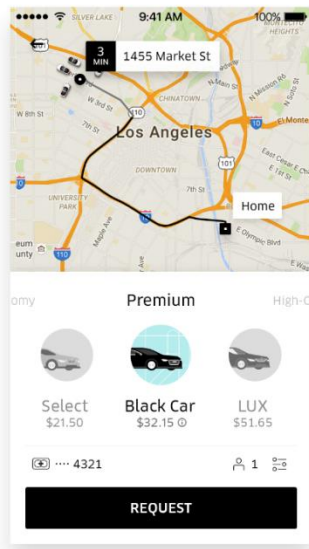
2

3



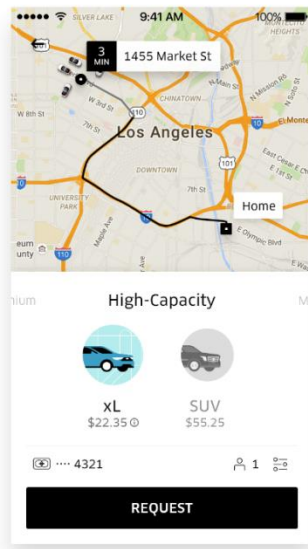
Economy

Price/time tradeoffs and low cost products



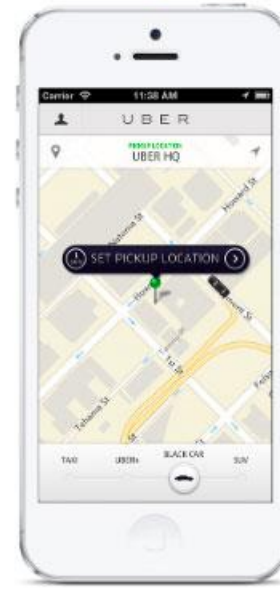
Premium

Nice cars and premium services



High-Capacity

Products that can hold 5 or more people



PARTNER APP W/ UBER BUTTON



GET /v1/products
/v1/estimated_price
/v1/estimated_time

REQUEST



POST /v1/requests

REQUESTING



DELETE /v1/requests/request_id/

SUCCESSFUL POST?

Yes

GET /v1/requests/request_id/

CHECK STATUS

Success

RESPONSE STATUS

accepted
arriving

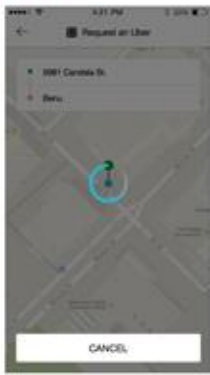
DRIVER EN ROUTE



VEHICLE PICKER

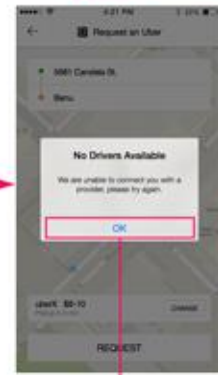


CANCELLING



Tap OK

REQUEST: NO DRIVERS AVAILABLE

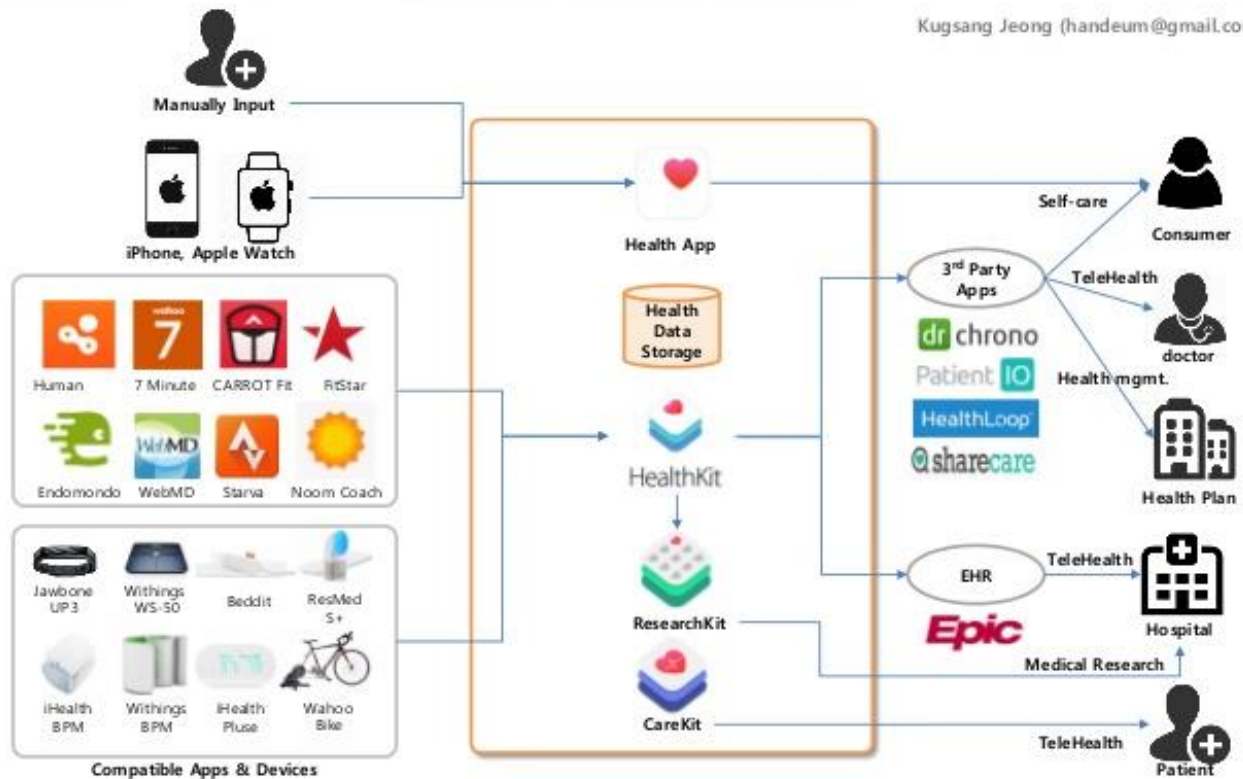


APPLE WATCH



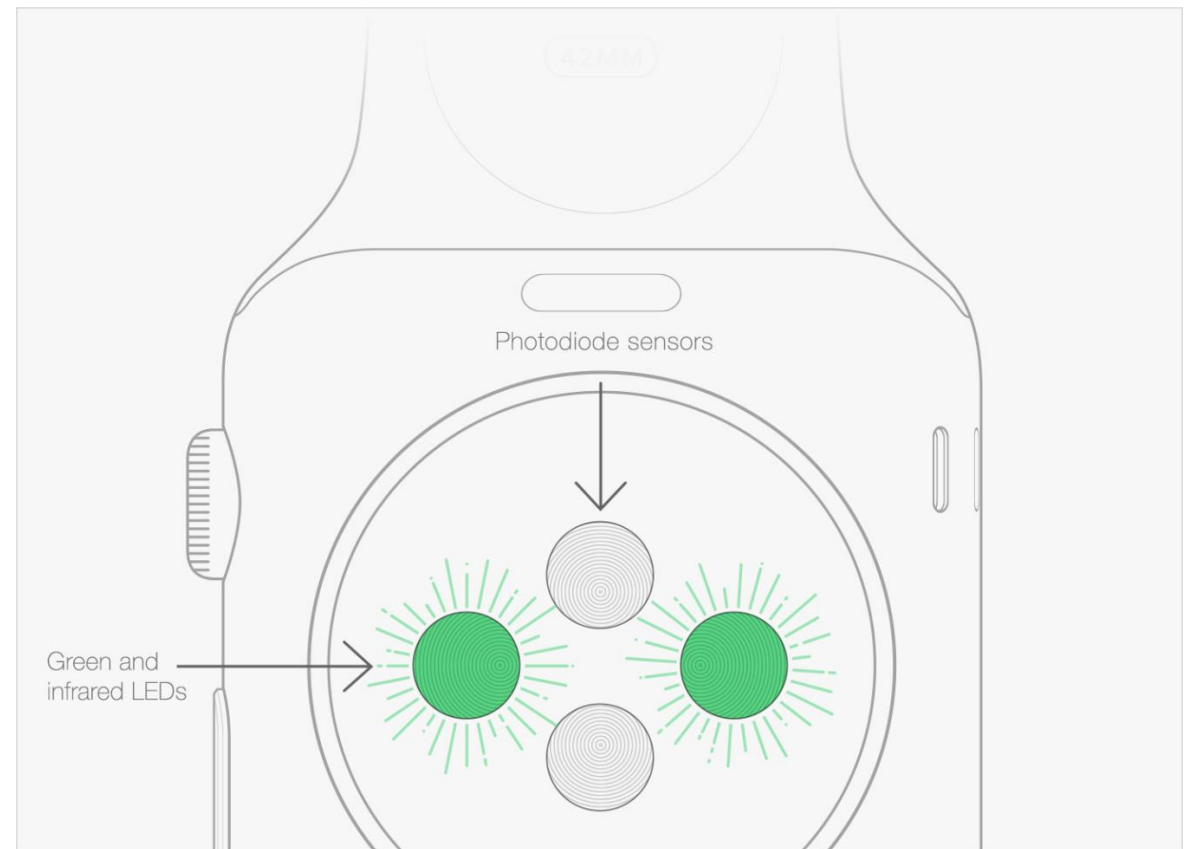
Apple Health Ecosystem

Kugsang Jeong (handeum@gmail.com)

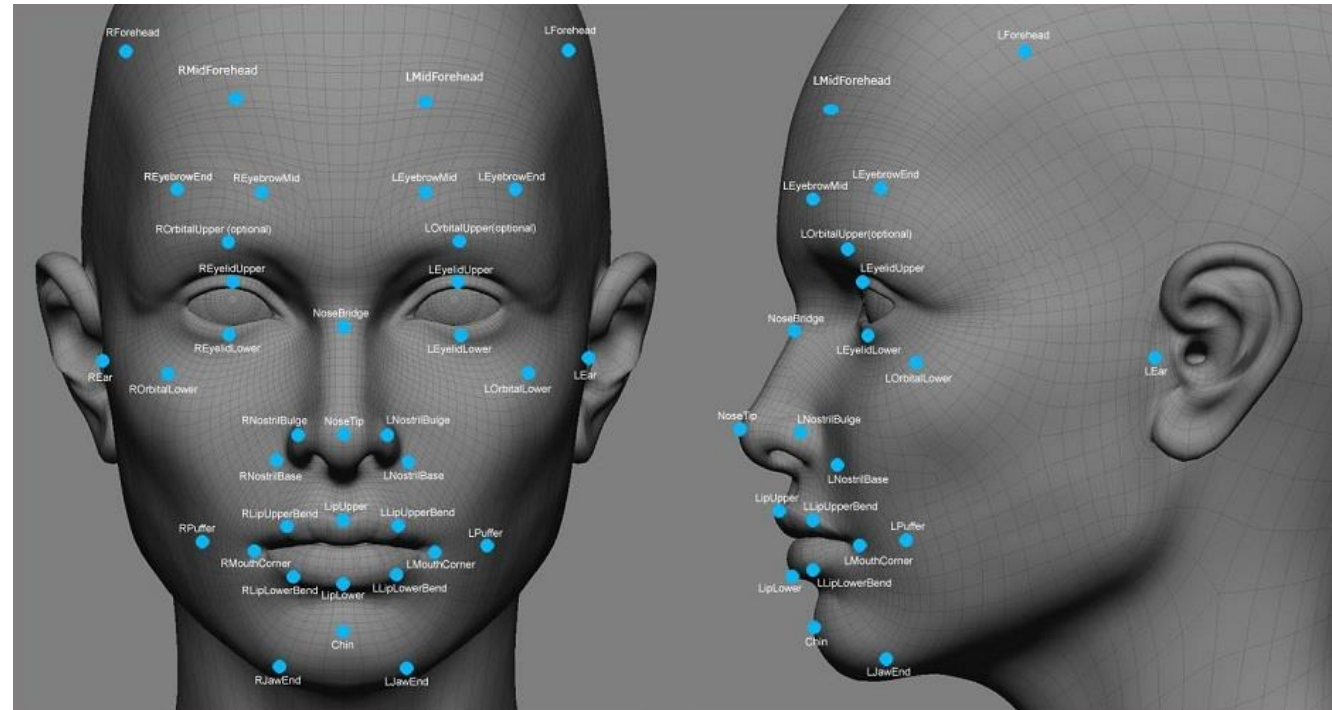


APPLE WATCH SENSOR

- How to Detect our body information?
 - 加速度感應器、陀螺儀
 - 心率感應器
- **LED 光測器**
 - 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 更細緻地分辨心跳的頻率或血液的組成，分辨使用者的身份



IPHONE X FACE ID

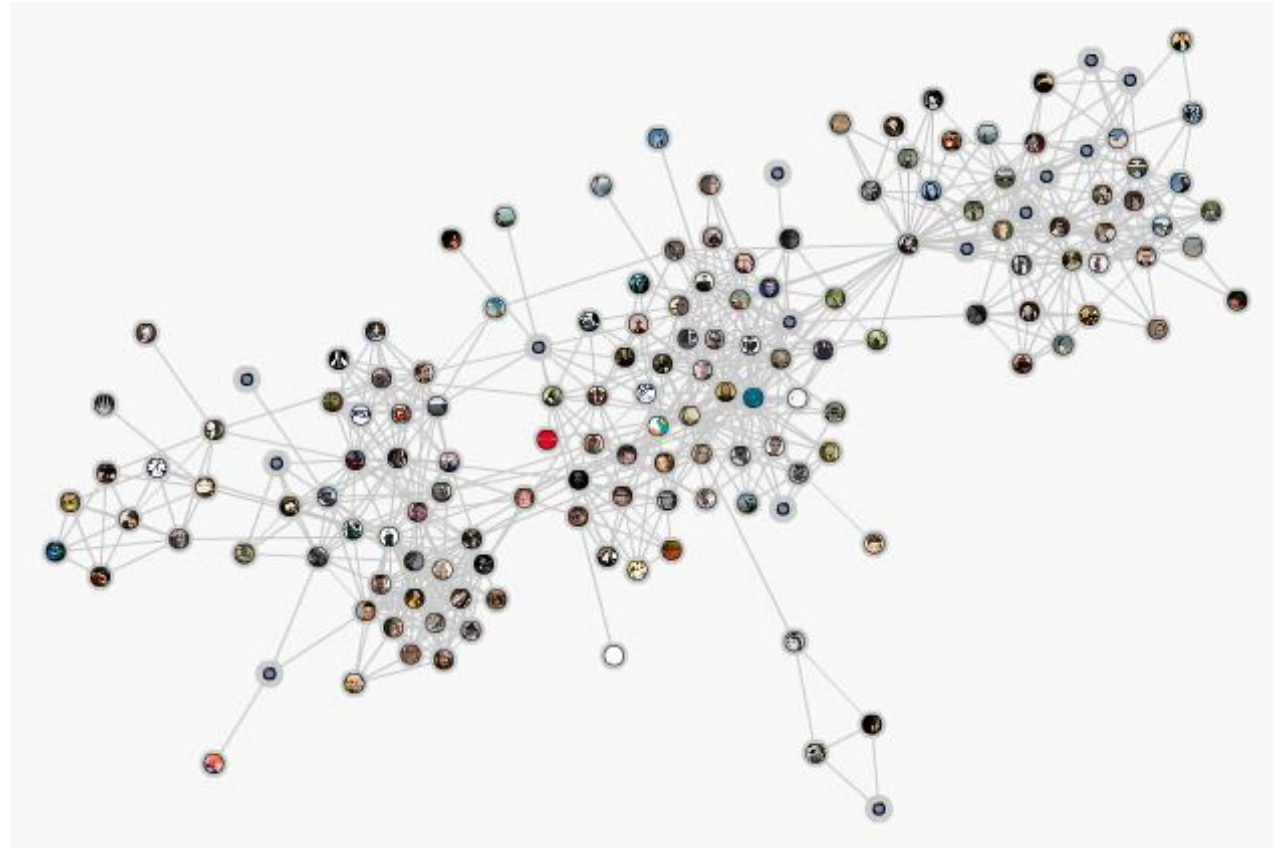
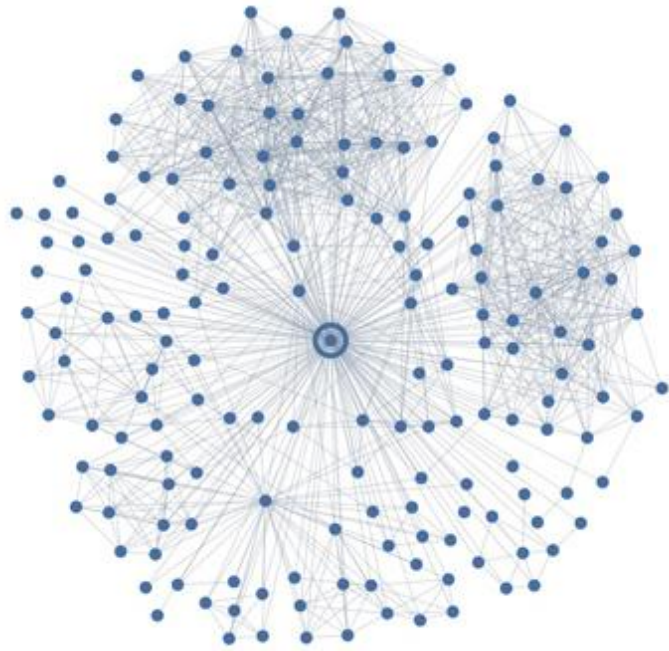


IPHONE X SENSORS

Ambient light sensor
Speaker
Proximity Sensor
Microphone
Flood Illuminator
7MP Camera
Infrared Camera
Dot Projector



SOCIAL GRAPH

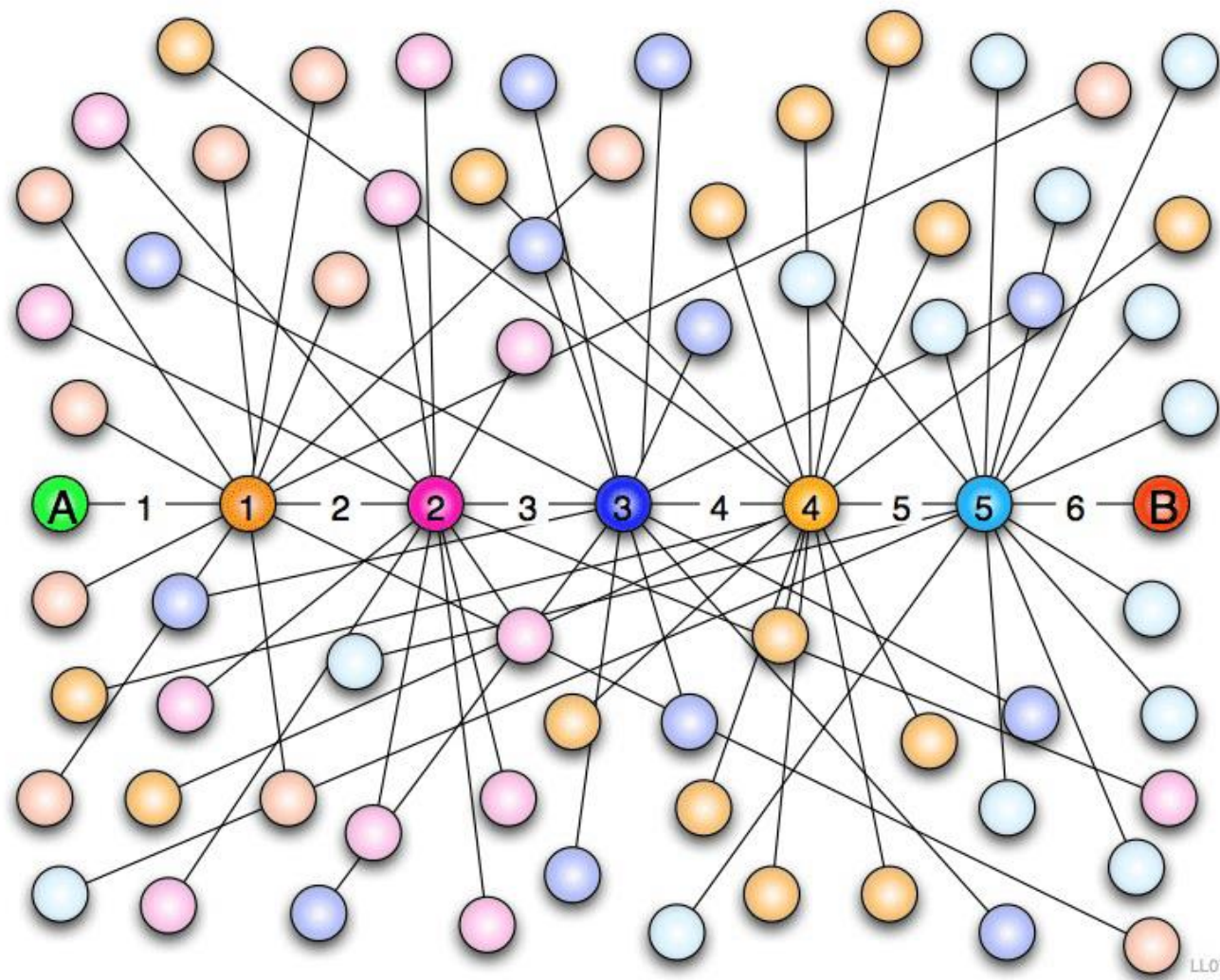


六度分隔理論

SIX DEGREES OF SEPARATION

此理論認為世界上任何互不相識的兩人，只需要很少的中間人就能夠建立起聯繫。

哈佛大學心理學教授 斯坦利·米爾格拉姆於1967年根據這個概念做過一次連鎖信實驗，嘗試證明平均只需要5個中間人就可以聯繫任何兩個互不相識的人。

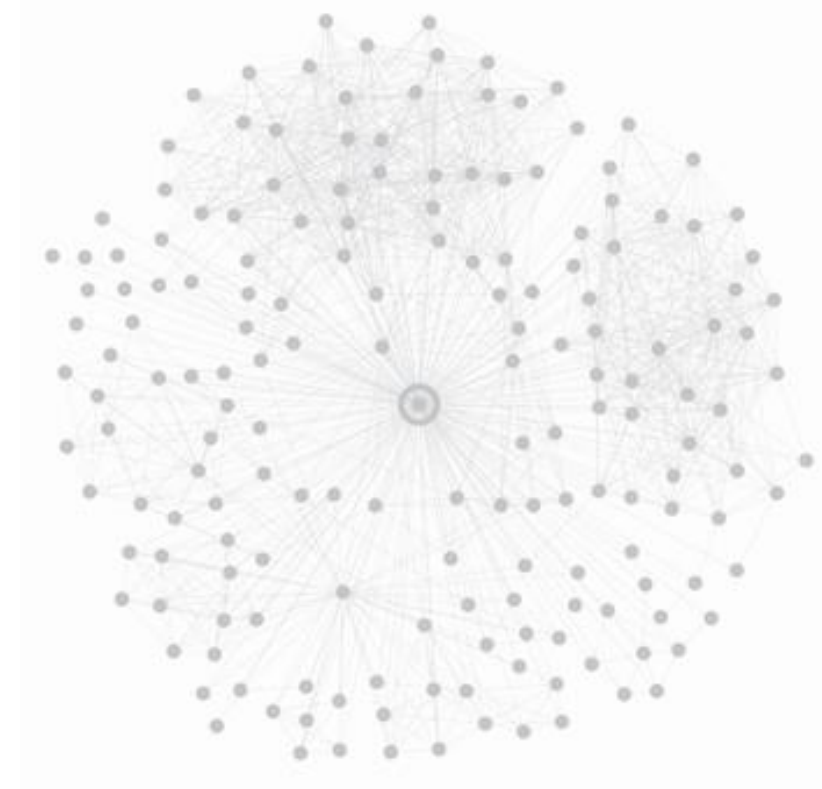


FACEBOOK **3.57** VS. **6** DEGREES OF SEPARATION (2016)



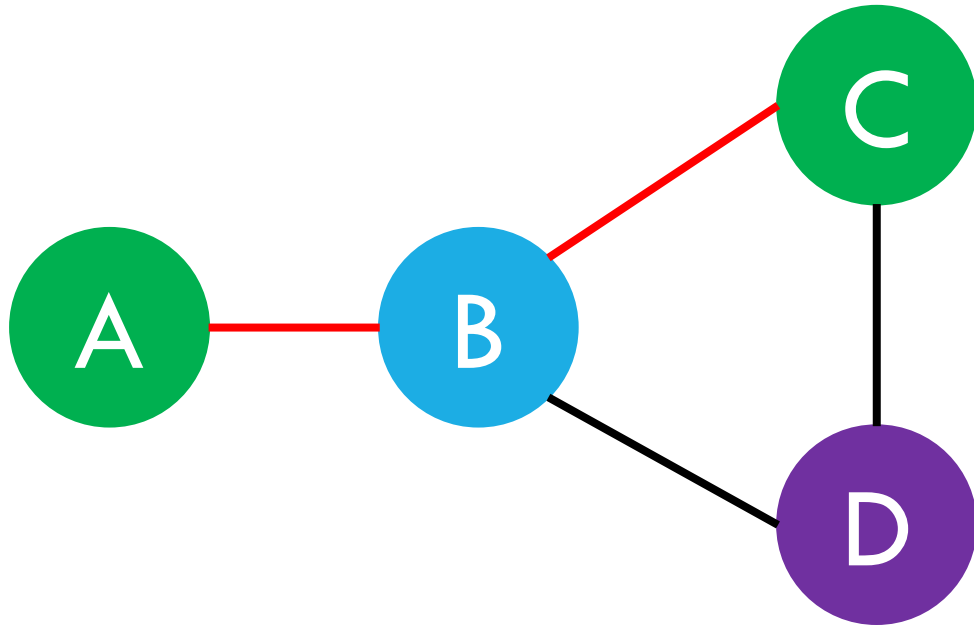
MATCH MAKER

1. *Anny, Bill, Cherry, David* are friends.
2. *David* knows *Anny, Bill, Cherry,* and *Eva*.
3. *Eva* knows *David, Frank* and *Grand*.
4. *Frank* and *Grand* are good friends.



Anny and Grand, Introduce by ?

RELATIONSHIP



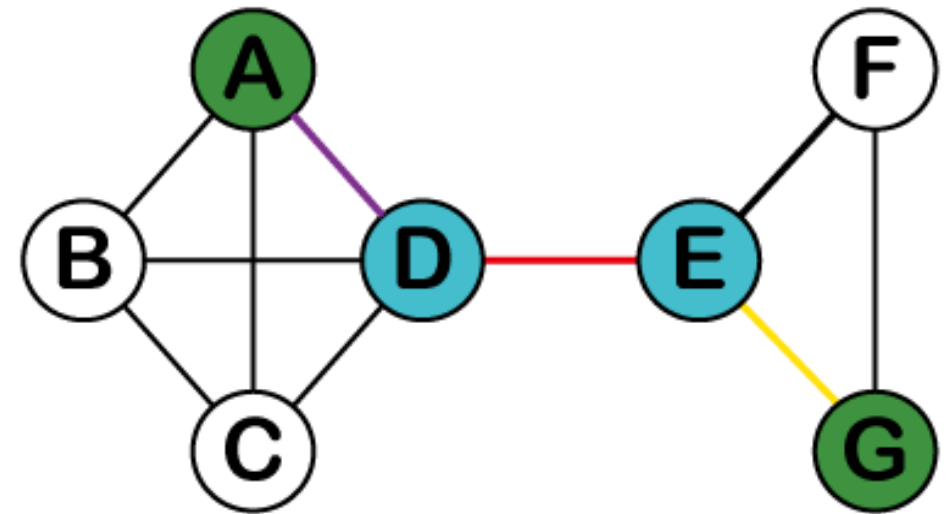
	A	B	C	D
A		0		
B	0		0	0
C		0		0
D		0	0	

MATCHMAKER

1. Anny, Bill, Cherry, David are friends.
2. David knows Anny, Bill, Cherry, and Eva.
3. Eva knows David, Frank and Grand.
4. Frank and Grand are good friends.

Anny and Grand, Introduce by ?

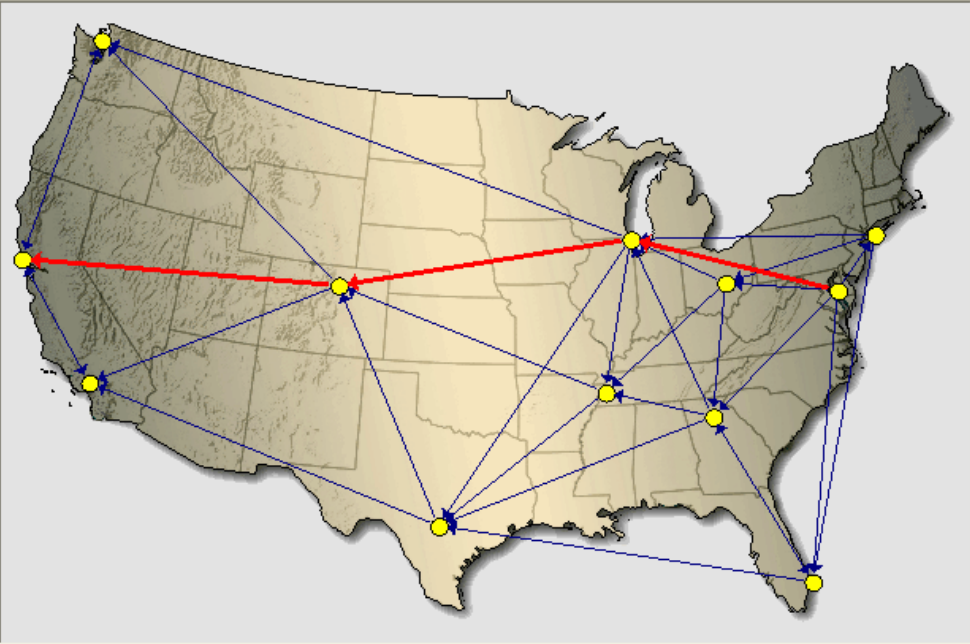
	A	B	C	D	E	F	G
A		○	○	●			
B	○		○	○			
C	○	○		○			
D	○	○	○		○		
E				●		○	●
F					○		○
G					○	○	



SHORTEST PATH

Shortest Path Optimization Using MPL OptiMax

Shortest Path Model (With MPL OptiMax)



Starting City
Washington

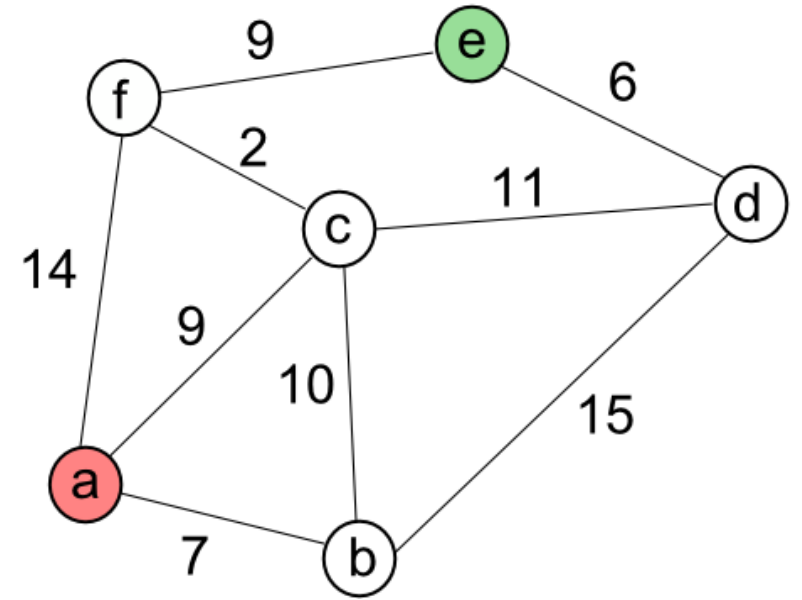
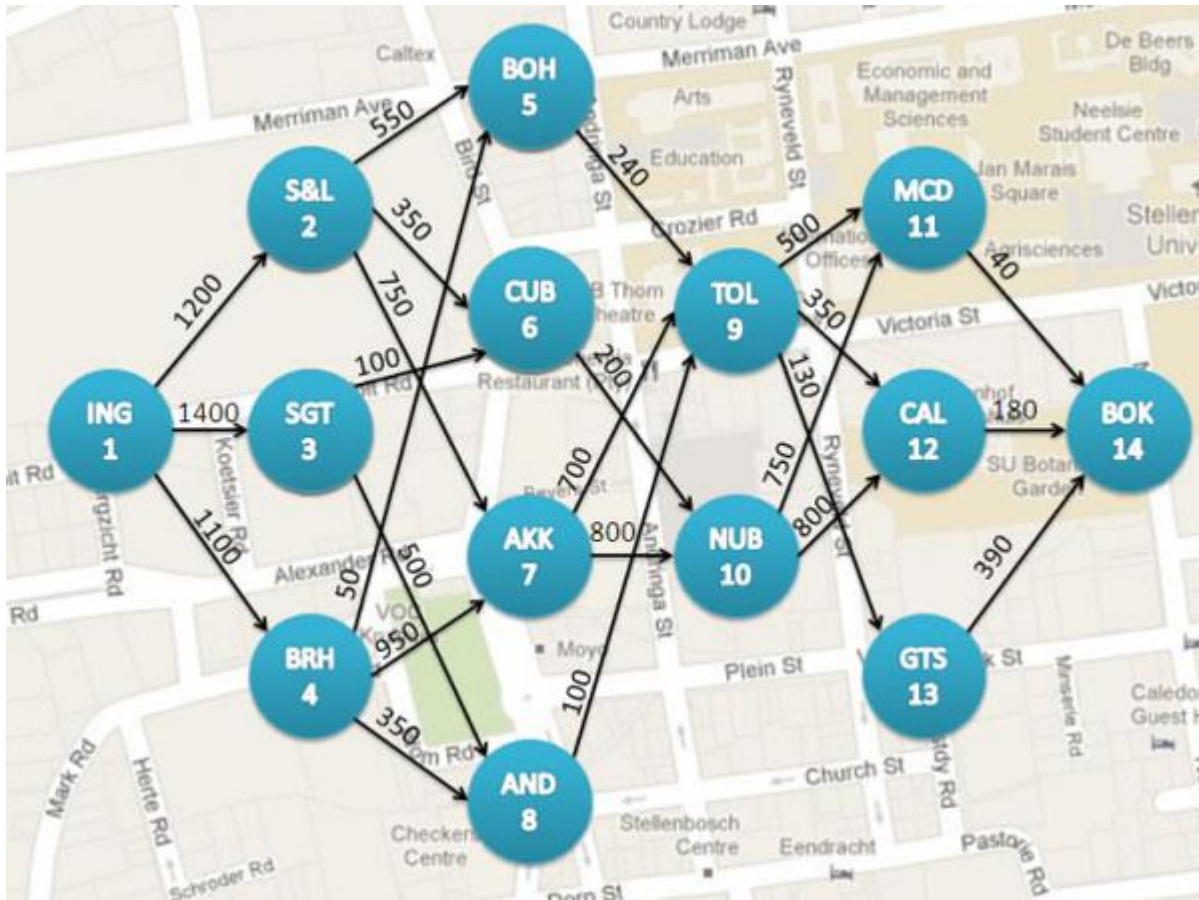
Ending City
San Francisco

Optimal solution found

Itinerary
Washington to Chicago (715 mi.)
Chicago to Denver (1011 mi.)
Denver to San Francisco (1257 mi.)
Total Distance: 2983 miles

Route Information				
	Start City	End City	Distance	Path
▶	Washington	New York	237	0
	Washington	Columbus	420	0
	Washington	Atlanta	632	0
	Washington	Miami	1043	0
	Washington	Chicago	715	1
	New York	Columbus	556	0
	New York	Miami	1281	0
	New York	Chicago	821	0
	Columbus	Atlanta	571	0
	Columbus	Chicago	361	0
	Columbus	Memphis	596	0
	Atlanta	Miami	661	0
	Atlanta	Chicago	716	0
	Atlanta	Memphis	394	0
	Atlanta	San Antonio	993	0
	Miami	Atlanta	661	0
	Miami	San Antonio	1381	0
	Chicago	Memphis	536	0
	Chicago	San Antonio	1202	0
	Chicago	Denver	1011	1
	Chicago	Seattle	2072	0
	Memphis	San Antonio	728	0
	Memphis	Denver	1095	0
	San Antonio	Denver	931	0
	San Antonio	Los Angeles	1372	0
	Denver	San Francis	1257	1
	Denver	Los Angeles	1023	0
	Denver	Seattle	1303	0
	San Francis	Los Angeles	380	0
	San Francis	Seattle	820	0
	Los Angeles	San Francis	380	0
	Seattle	San Francis	820	0

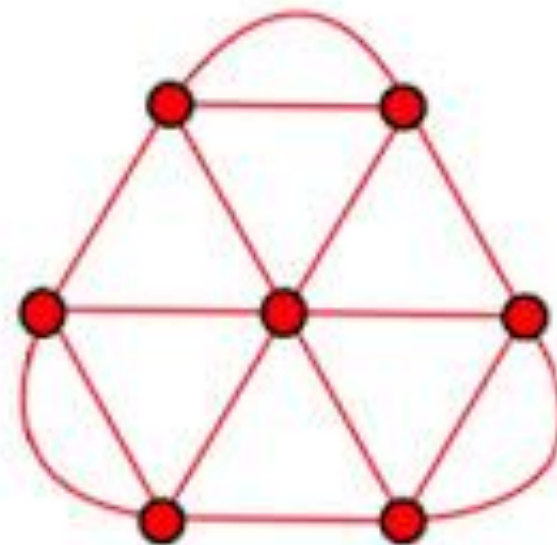
SHORTEST PATH & ROUTING PROBLEM



PATH ARRANGEMENT

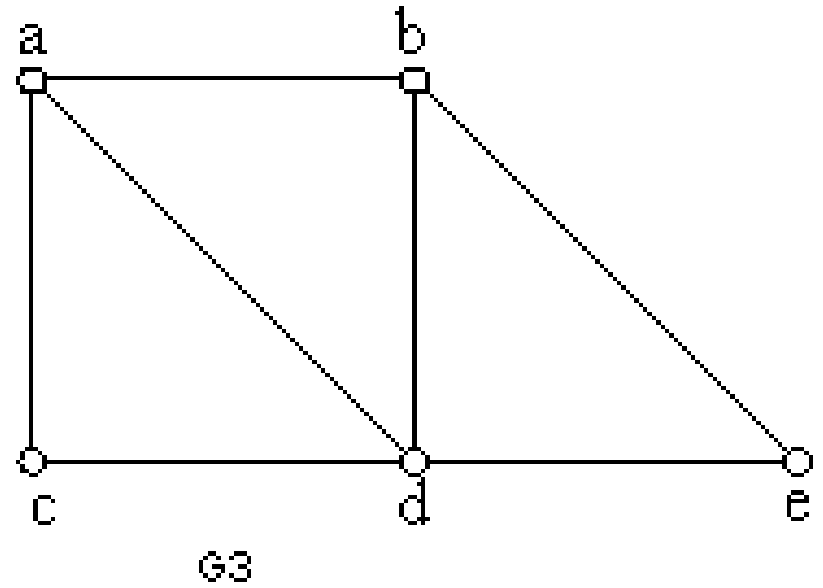
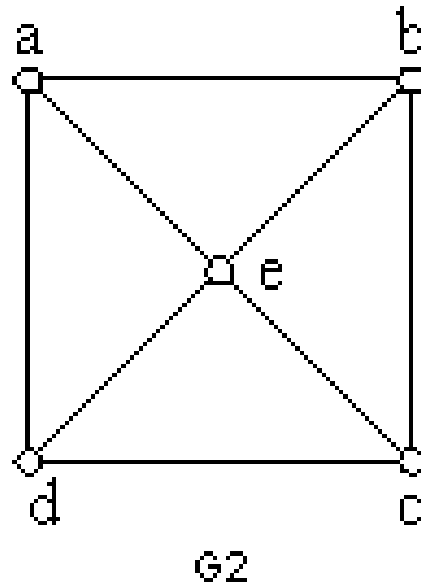
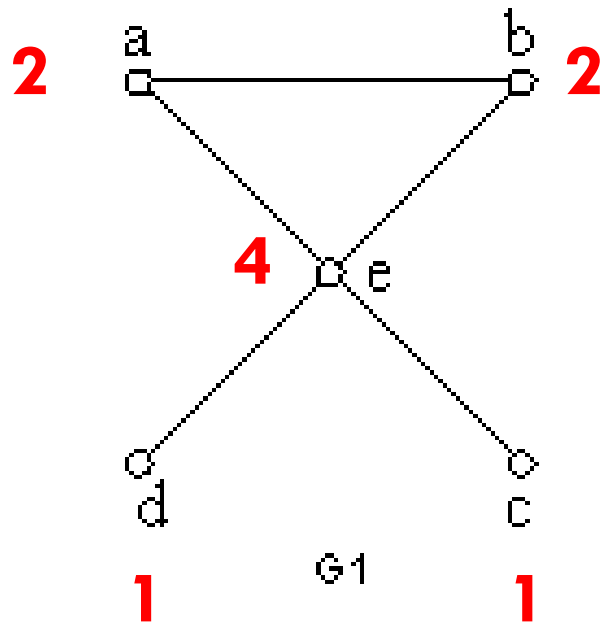


展覽地圖

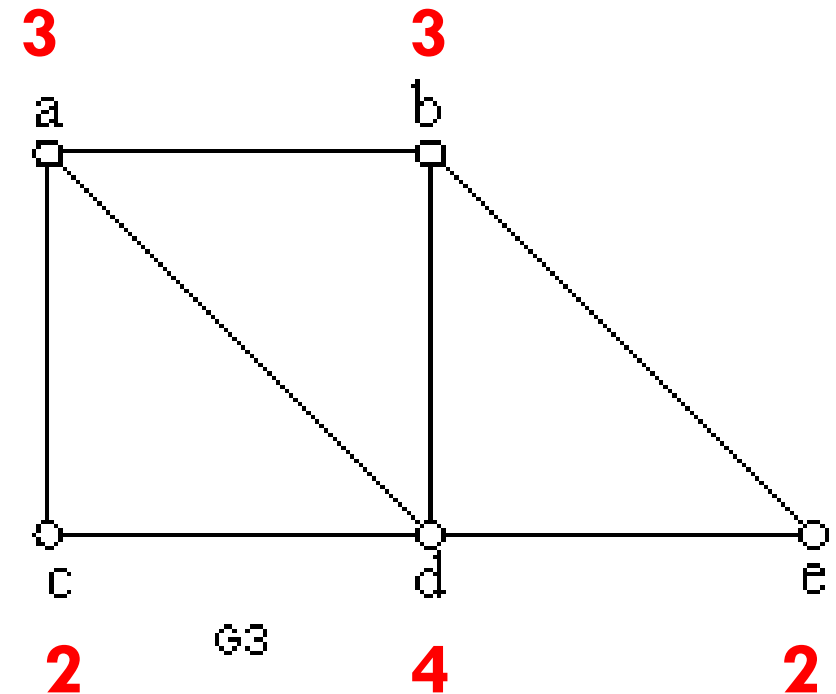
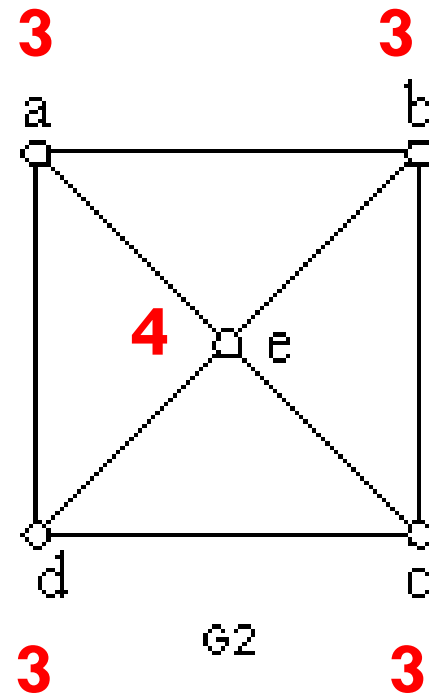
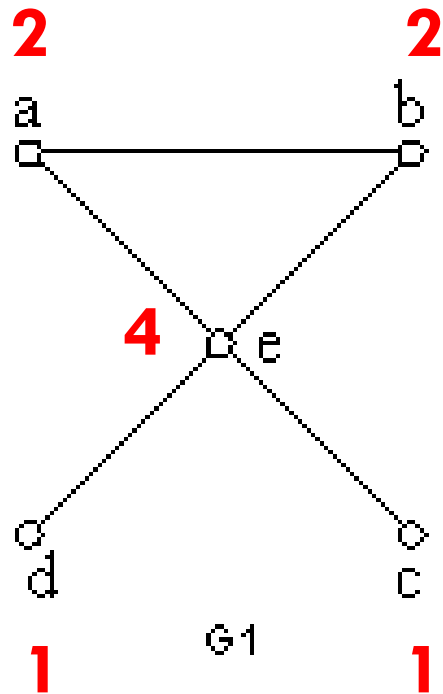


圖形 E

IN DEGREE / OUT DEGREE



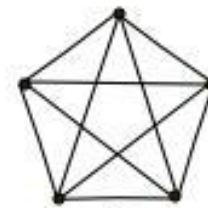
IN DEGREE / OUT DEGREE



PATH ARRANGEMENT



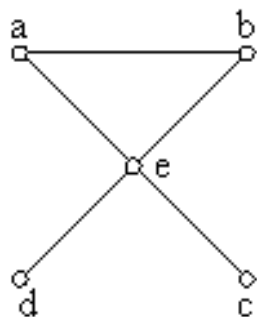
9



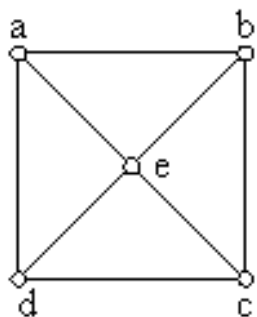
10



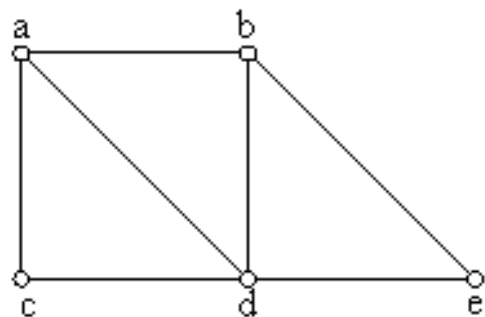
11



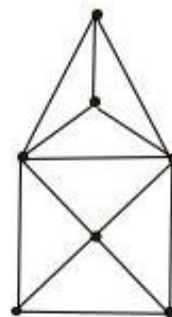
G1



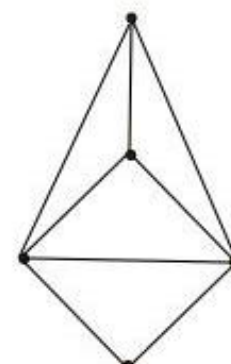
G2



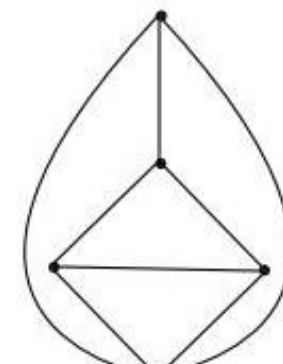
G3



12



13



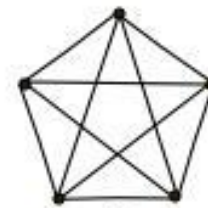
14



PATH ARRANGEMENT



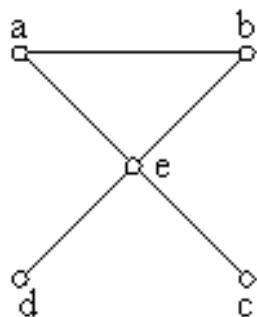
9



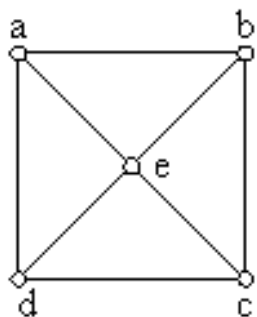
10



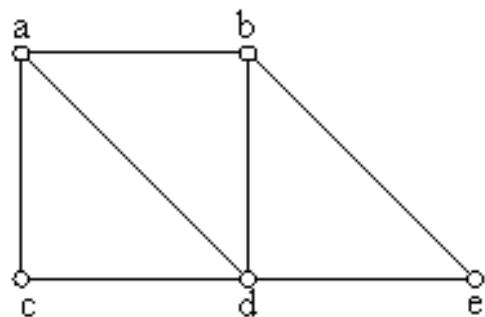
11



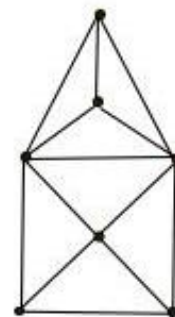
G1



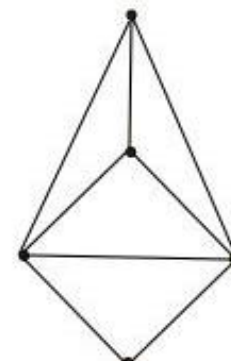
G2



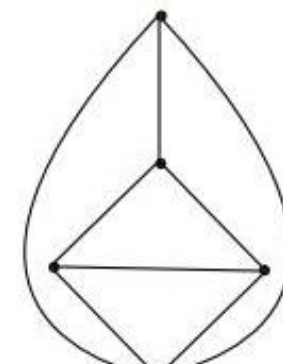
G3



12



13

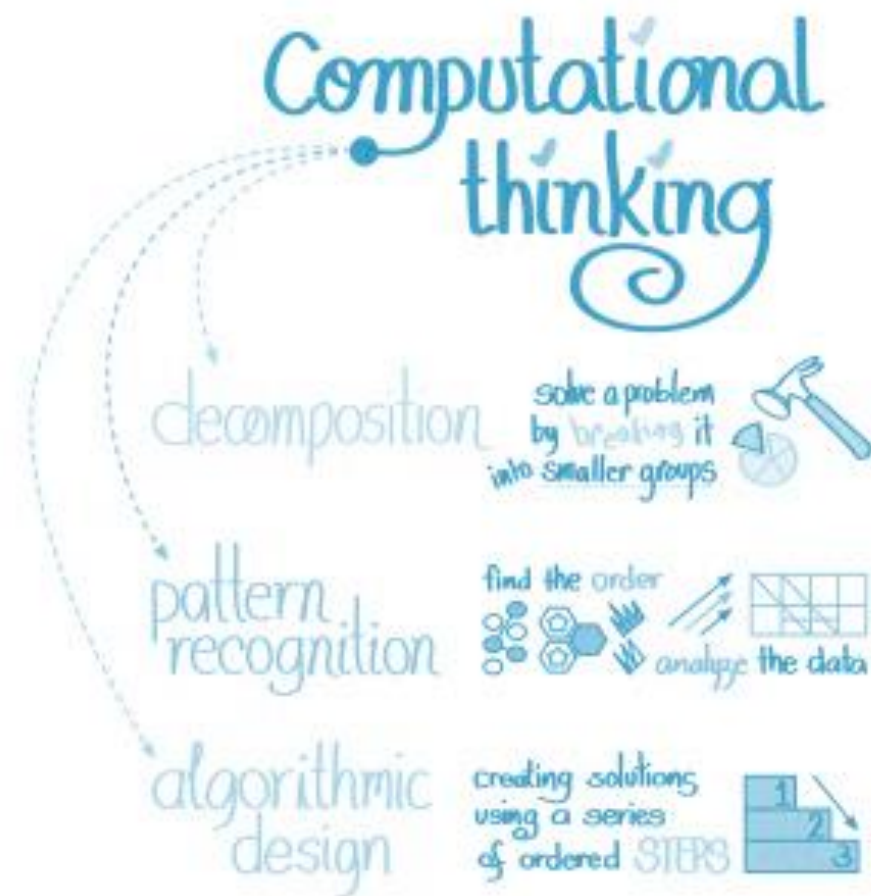


14



OUTLINE

- 資訊科技教育發展
- 運算思維概念與教學重點
- 生活中的運算與思考
- 運算思維導向課程設計
- 運算思維教材資源



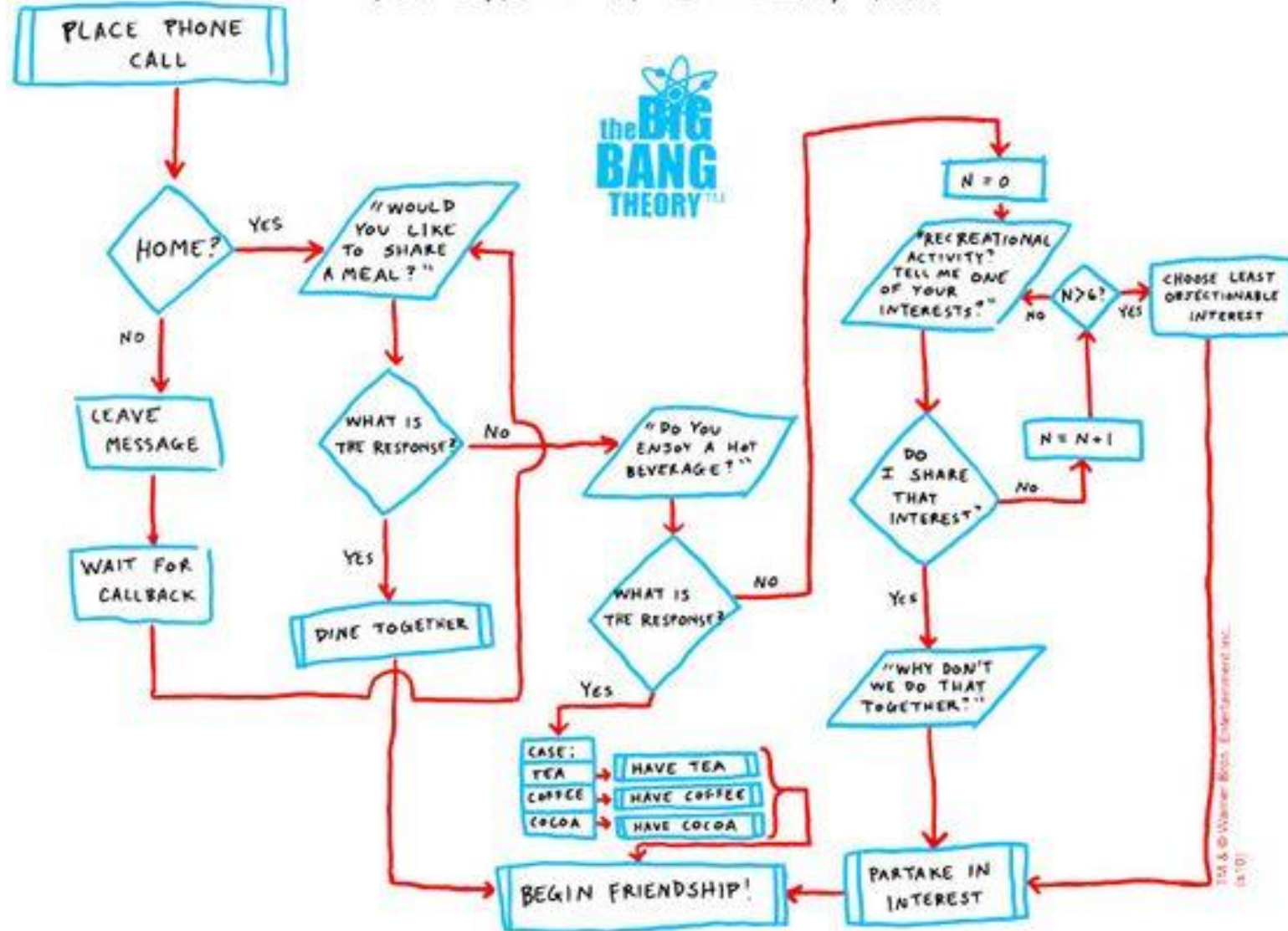


視覺化程式設計-運算思維導向教學

以樂曲創作與
碎形繪製為例

THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, Ph.D



Computational Thinking

(A) $AA \underline{A} B C B \rightarrow A \underline{A} B C B \rightarrow \underline{A} B C B \rightarrow \underline{B} C B \rightarrow \underline{B} B \rightarrow \underline{B}$

(B) $A B C A B C \rightarrow B C A B C \rightarrow B C B C \rightarrow B B C \rightarrow B B \rightarrow B$

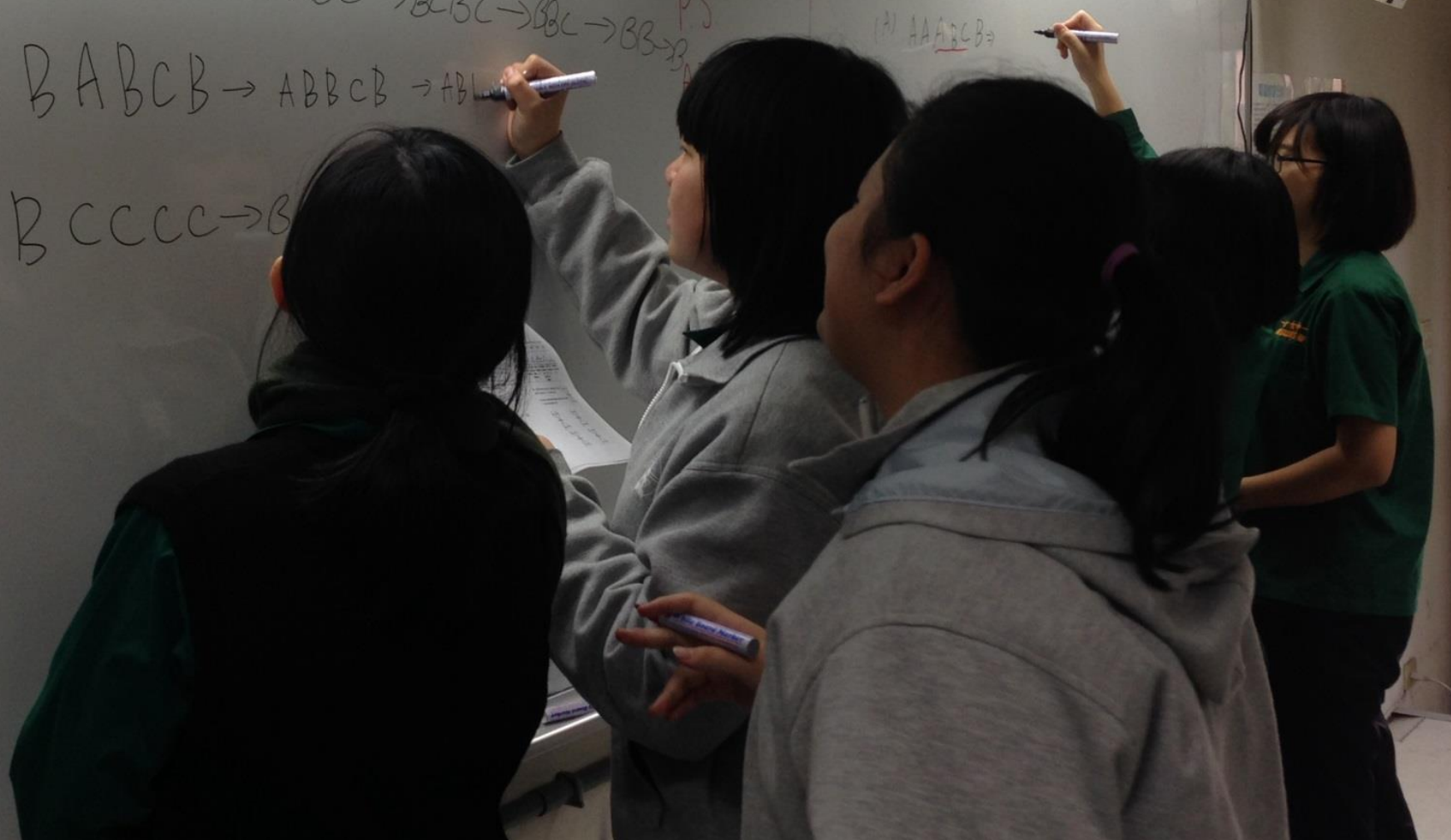
(C) $A B A B C B \rightarrow A B B C B \rightarrow A B$

(D) $A B C C C C \rightarrow B$

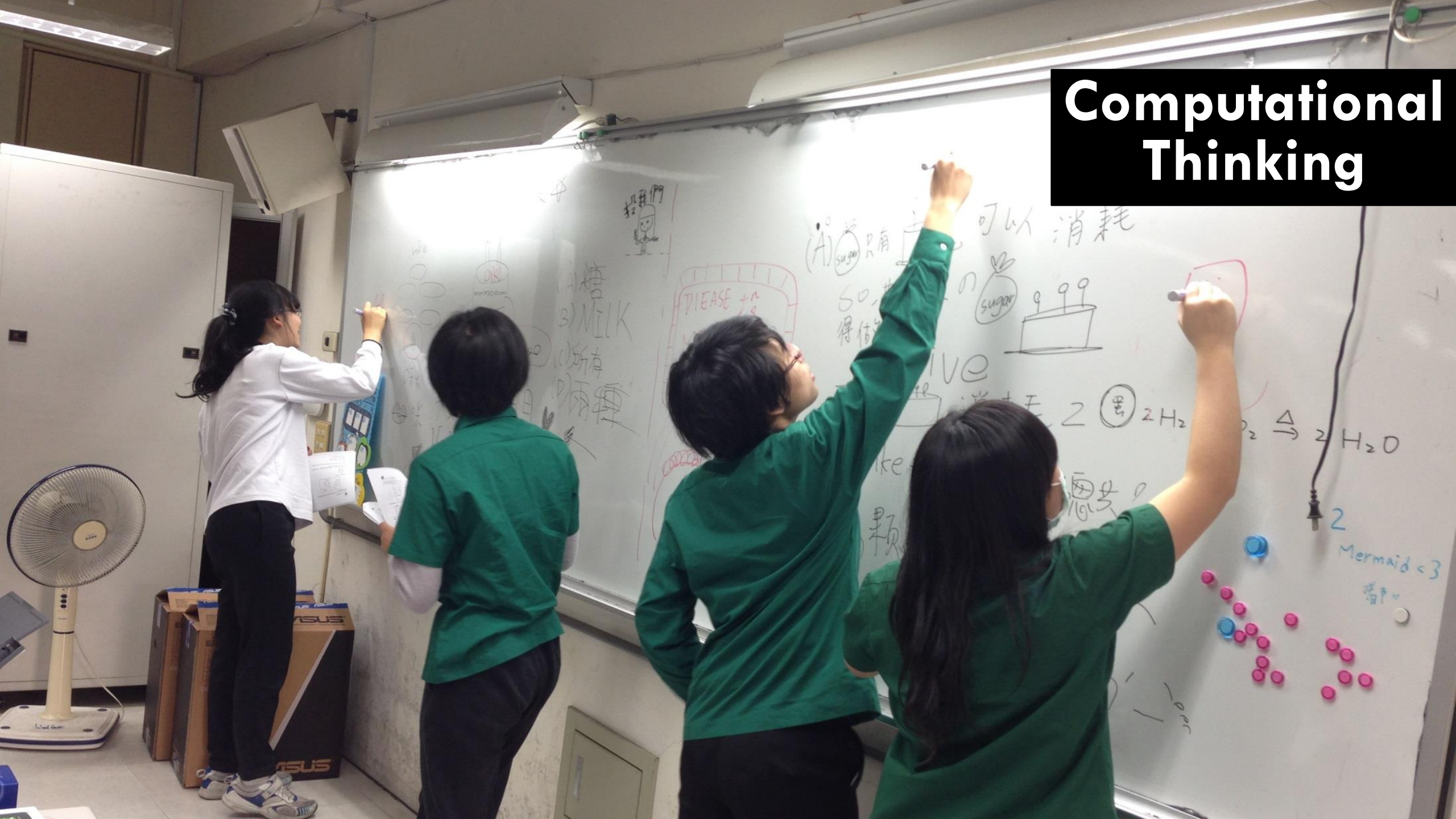
50 資研
28X29TH

嘉均 嘉宇

培澤



Computational Thinking

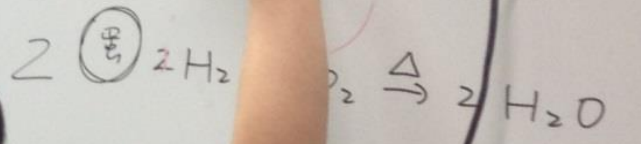


我們們

- (A) 糖
- B) MILK
- (C) 所有
- D) 兩

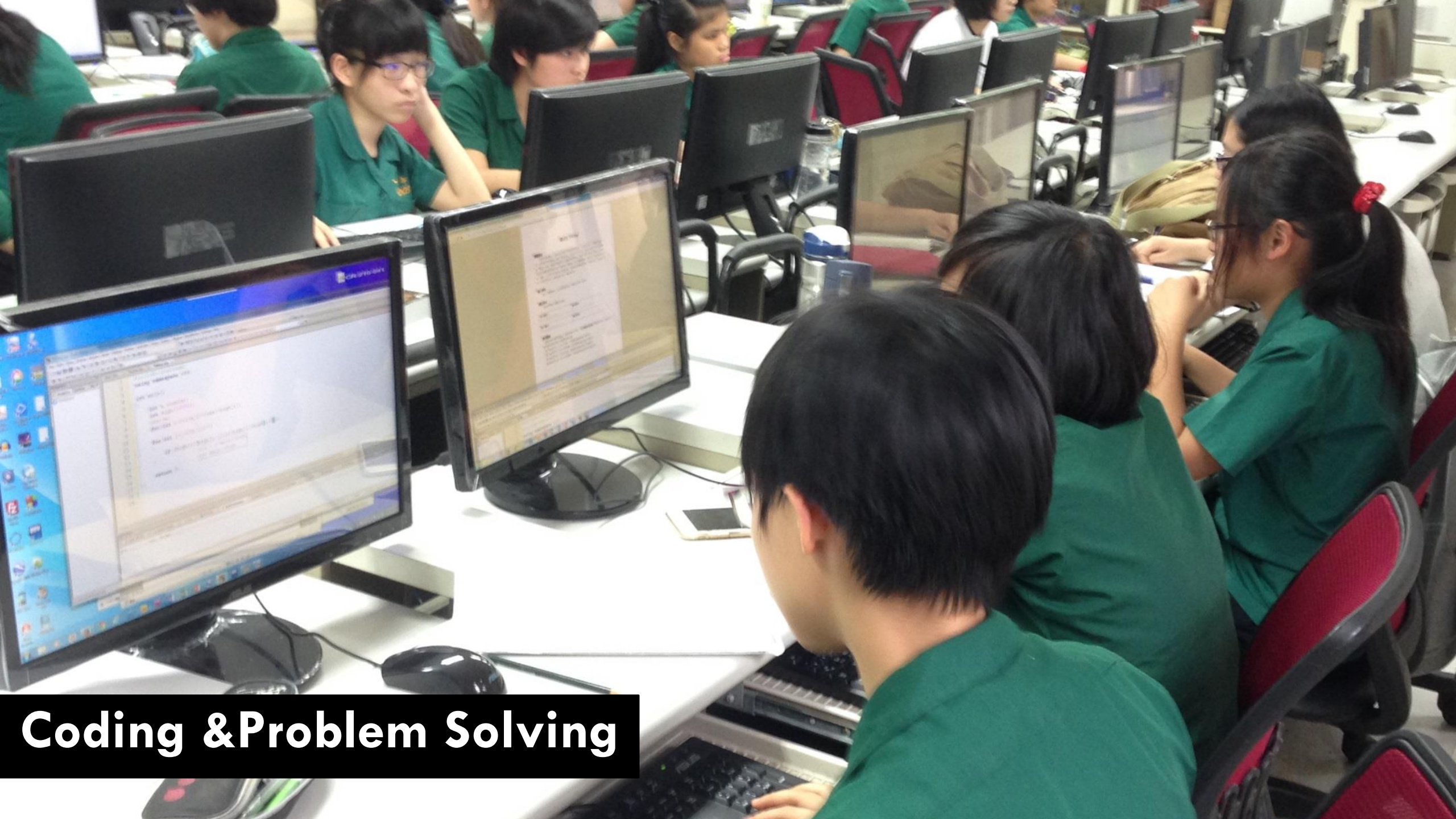


(A) 糖 只有 可以 消耗
So, 糖 的 sugar 999
得估



2 Mermaid < 3
塔





Coding & Problem Solving

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：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：選修『問題解決與程式設計』

時數：4小時

視覺化程式工具：

MSLOGO 、 Python Turtle

重複結構與模組化程式設計

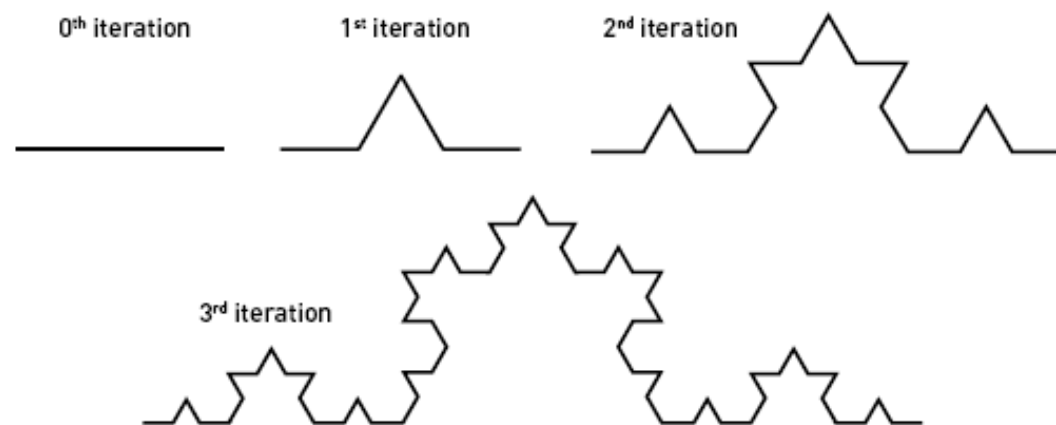
- 透過觀察樂曲(聽覺)與幾何圖形(視覺)的規律性
- 讓學生體驗分解問題、樣式辨識、模型化與抽象化 運算思維歷程

A 

A' 

B 

A' 





CT COURSE DESIGN FRAMEWORK

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

活動設計

觀察分析



資料表示
程式概念



程式實作

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

教學設計

教學活動

活動內容

視覺化樣式
展示與操作

引發學生動機

自動化音樂演奏
自動演奏鋼琴與音樂盒音樂欣賞

提問與討論

引導學生思考如何自動演奏-
自動化音樂演奏的原理與發展

識譜與演唱

引導學生思考人類唱歌的歷程

運算思維視覺化

樂句結構分析

學習單-標示樂句名稱

樂句與函式對應

範例程式操作-播放樂句函式

程式概念說明

函式概念與說明
控制流程概念與說明

範例程式操作-辨識樂句函式
範例程式操作-依樂曲演奏流程置放樂句

語法說明
與程式實作

提問與討論-
音符資料表示法

引導學生思考音符代號的意義
學習單-音符資料與數值對應

程式創作

兒歌實作
兒歌變奏曲實作

範例程式操作-修改樂曲演奏流程、音符、樂器與演奏速度，創作變奏曲

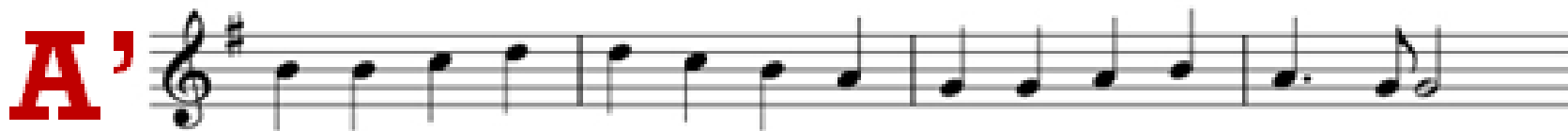
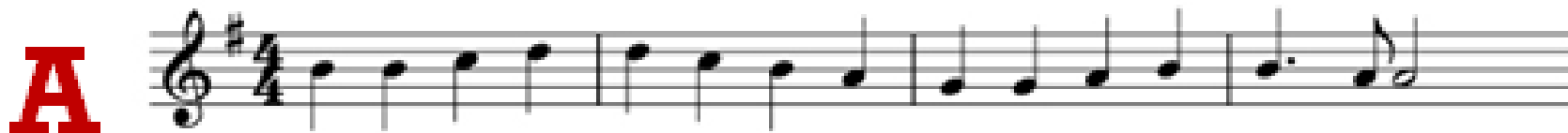
觀摩與討論

展示學生作品

觀摩與學習

CT: DECOMPOSITION / PATTERN RECOGNITION

ABSTRACTION / ALGORITHM

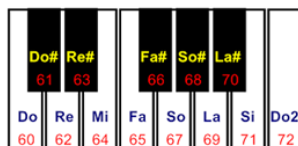


CT: DECOMPOSITION / PATTERN RECOGNITION

ABSTRACTION / ALGORITHM

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



定義 A

- 彈奏音符 71 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 72 1 拍
- 彈奏音符 74 1 拍
- 彈奏音符 74 1 拍
- 彈奏音符 72 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 67 1 拍
- 彈奏音符 67 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 71 1.5 拍
- 彈奏音符 69 0.5 拍
- 彈奏音符 69 2 拍

定義 A'

- 彈奏音符 71 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 72 1 拍
- 彈奏音符 74 1 拍
- 彈奏音符 72 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 67 1 拍
- 彈奏音符 67 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 69 1.5 拍
- 彈奏音符 67 0.5 拍
- 彈奏音符 67 2 拍

定義 B

- 彈奏音符 69 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 71 1 拍
- 彈奏音符 67 1 拍
- 彈奏音符 69 0.5 拍
- 彈奏音符 71 1 拍
- 彈奏音符 72 0.5 拍
- 彈奏音符 67 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 71 0.5 拍
- 彈奏音符 72 0.5 拍
- 彈奏音符 71 1 拍
- 彈奏音符 69 1 拍
- 彈奏音符 67 1 拍
- 彈奏音符 69 1 拍

當按下 空白鍵 鍵

playstyle

A

A'

B

A'

音樂專題~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 metronome marking of 'J = 120'. The score is written for a single melodic line. The first system starts with a key signature of one flat (B-flat). The melody is simple and repetitive, consisting of eighth and quarter notes. The second system continues the melody. The third system introduces a new melodic phrase. The fourth system continues the melody. The fifth system concludes the piece with a final cadence.

The image shows a Scratch project page for '崖上的波妞-2' (The Girl on the Cliff-2). The project is created by '張思婕and丁紫庭' (Zhang Si-jie and Ding Zi-ting) and is based on the work of 'ting0101095'. The project has 18 scripts and 2 characters. The main scene features a cartoon character with red hair and a green flag on its head, floating in the water. A jellyfish is also present. The text 'Please press the jellyfish' is displayed at the bottom of the scene. The project was shared on 30 October 2016 and last modified on 30 October 2016. The project has 0 stars and 1 heart.

<https://scratch.mit.edu/projects/123769932/>

定義 A

- 演奏音階 72 1 拍
- 演奏音階 69 0.5 拍
- 演奏音階 65 1 拍
- 演奏音階 60 0.5 拍
- 演奏音階 60 0.5 拍
- 演奏音階 60 0.5 拍
- 演奏音階 62 0.5 拍
- 演奏音階 65 0.5 拍
- 演奏音階 70 0.5 拍

定義 B

- 演奏音階 70 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 70 0.5 拍
- 演奏音階 69 0.5 拍
- 演奏音階 65 1 拍
- 演奏音階 69 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 62 0.5 拍
- 演奏音階 64 0.5 拍

定義 B'

- 演奏音階 70 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 70 0.5 拍
- 演奏音階 69 0.5 拍
- 演奏音階 65 1 拍
- 演奏音階 69 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 64 1 拍

定義 C

- 演奏音階 65 1 拍
- 演奏音階 60 0.5 拍
- 演奏音階 65 0.5 拍
- 演奏音階 67 2 拍
- 演奏音階 67 1 拍
- 演奏音階 60 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 69 2 拍

定義 F

- 演奏音階 65 0.75 拍
- 演奏音階 65 0.25 拍
- 演奏音階 65 0.5 拍
- 演奏音階 65 0.5 拍
- 演奏音階 74 0.5 拍
- 演奏音階 72 1.5 拍
- 演奏音階 62 0.75 拍
- 演奏音階 62 0.25 拍
- 演奏音階 62 0.5 拍
- 演奏音階 62 0.5 拍

定義 G

- 演奏休息 1 拍
- 演奏音階 69 1 拍
- 演奏音階 70 1 拍
- 演奏音階 72 1 拍
- 演奏音階 65 1 拍
- 演奏音階 65 1 拍
- 演奏音階 67 1 拍
- 演奏音階 69 1 拍
- 演奏音階 67 4 拍

定義 H

- 演奏音階 60 0.5 拍
- 演奏音階 65 0.5 拍
- 演奏音階 65 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 67 0.5 拍
- 演奏音階 69 0.5 拍
- 演奏音階 69 0.5 拍
- 演奏音階 70 0.5 拍

CT PROCESS

定義 Chorus

- C
- D
- C
- D'
- E

定義 Verse

- A
- B
- A
- B'

定義 Bridge

- F
- G
- H

當角色被點擊

- style
- Verse
- Chorus
- Bridge
- Verse

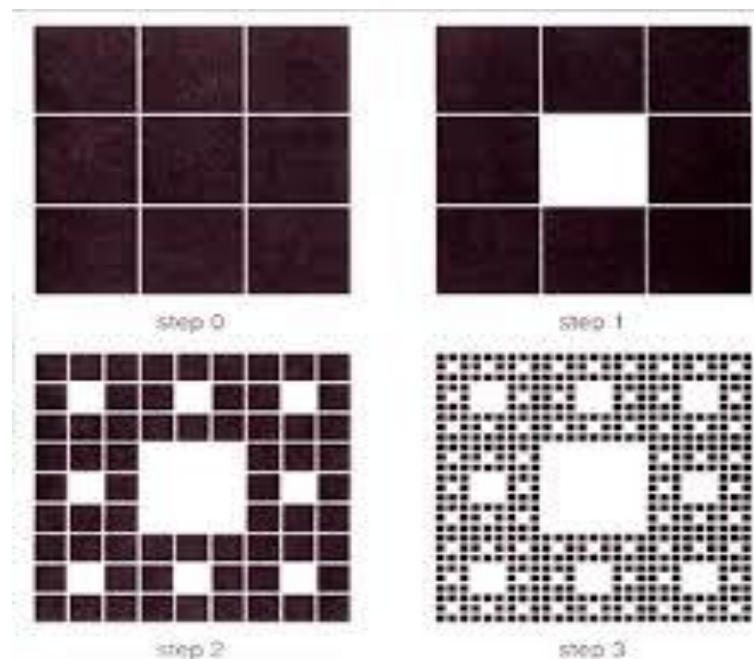
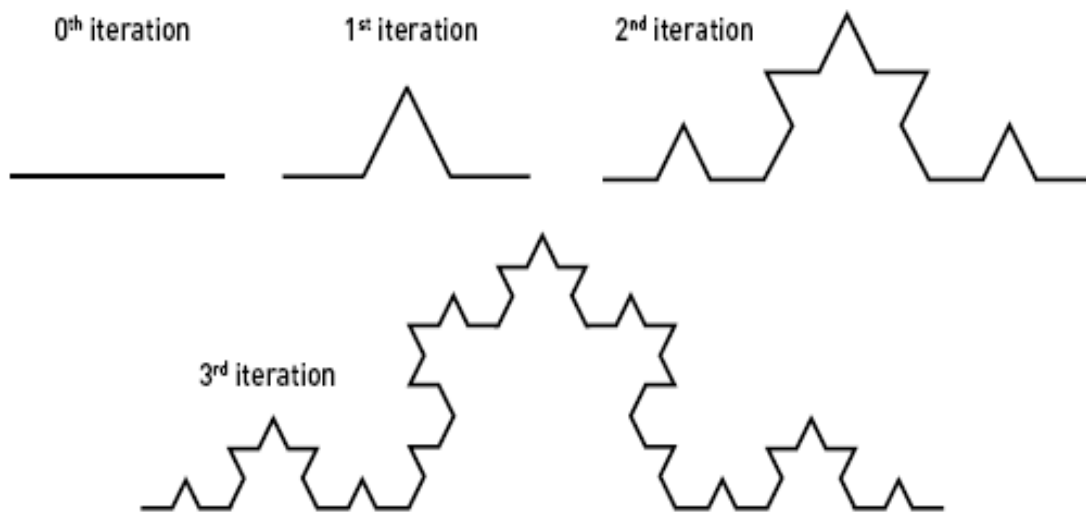
v: -8;

崖上的波妞
(單旋律)

Moderate ♩ = 120

碎形~尋找大自然的密碼

FRACTALS – CODE IN NATURE



- 引導學生觀察多種碎形圖形的產生規則
- 尋找圖形樣式

- 透過碎形自我相似性建構體驗
- 體會遞迴函式的抽象概念

INSTRUCTION DESIGN

<http://ct.fg.tp.edu.tw/?p=65>

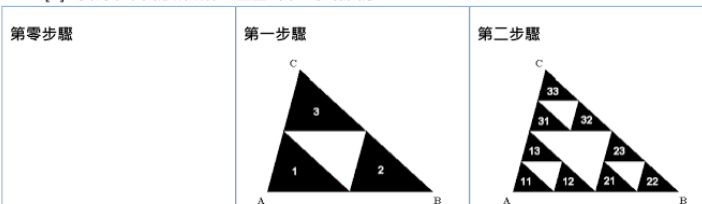
[1] 試寫下下圖形成規律，並畫出第四步驟圖形



[2] 試寫下下圖形成規律，並畫出第三步驟圖形



[3] 試寫下下圖形成規律，並畫出第三步驟圖形



視覺化程式設計-碎形繪圖專題

學習活動 3 : Geometric drawing by programming

一、藝術家(Artist)：運用已知程式積木完成繪圖任務

步驟	任務說明	完成作品	程式積木
1.	階段 10 的第 1 關 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。		程式積木 向前移動 100 像素 向右轉 90 度 向左轉 90 度
2.	階段 10 的第 2 關- 用 120 度、60 度及 150 像素當做邊長，畫出這顆鑽石吧		程式積木 向前移動 100 像素 向右轉 90 度 向左轉 90 度

[2] 繪製邊長 50 的正方形 (Square2.py)

觀察上述程式碼，完成下列問題

- 重複的段落是：
- 重複次數是：_____

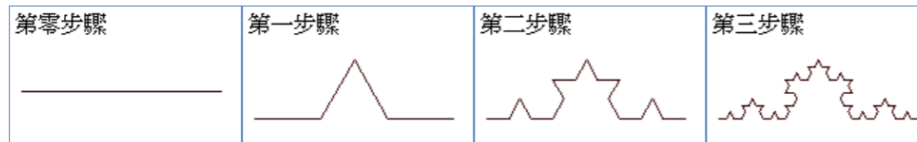
```
from turtle import *
canvas = Screen()
sarah = Turtle()
```

```
for i in range(4): #repeat four times
    sarah.forward(50)
    sarah.left(90)
```

```
canvas.exitonclick()
```



二、觀察 Koch Curve



```
def f(t, length, depth):
```

```
    起始元：
    if depth == 0:
        t.forward(length)
```

```
    生成元：
    else:
        f(t, length/3, depth-1)
        t.left(60)
        f(t, length/3, depth-1)
        t.right(120)
        f(t, length/3, depth-1)
        t.left(60)
        f(t, length/3, depth-1)
```

第零步驟



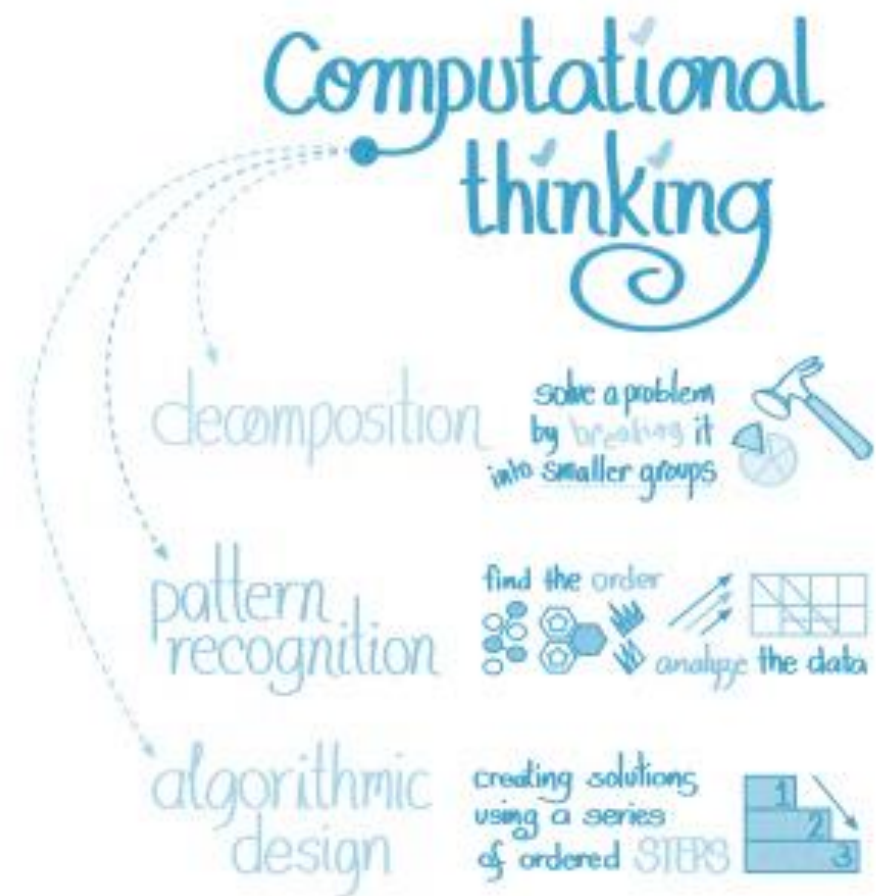
第一步驟



討論：程式呼叫 `f(Koch, 150, 2)` 時，請畫下其 `f` 函式呼叫的歷程

OUTLINE

- 資訊科技教育發展
- 運算思維概念與教學重點
- 生活中的運算與思考
- 運算思維導向課程設計
- 運算思維教材資源

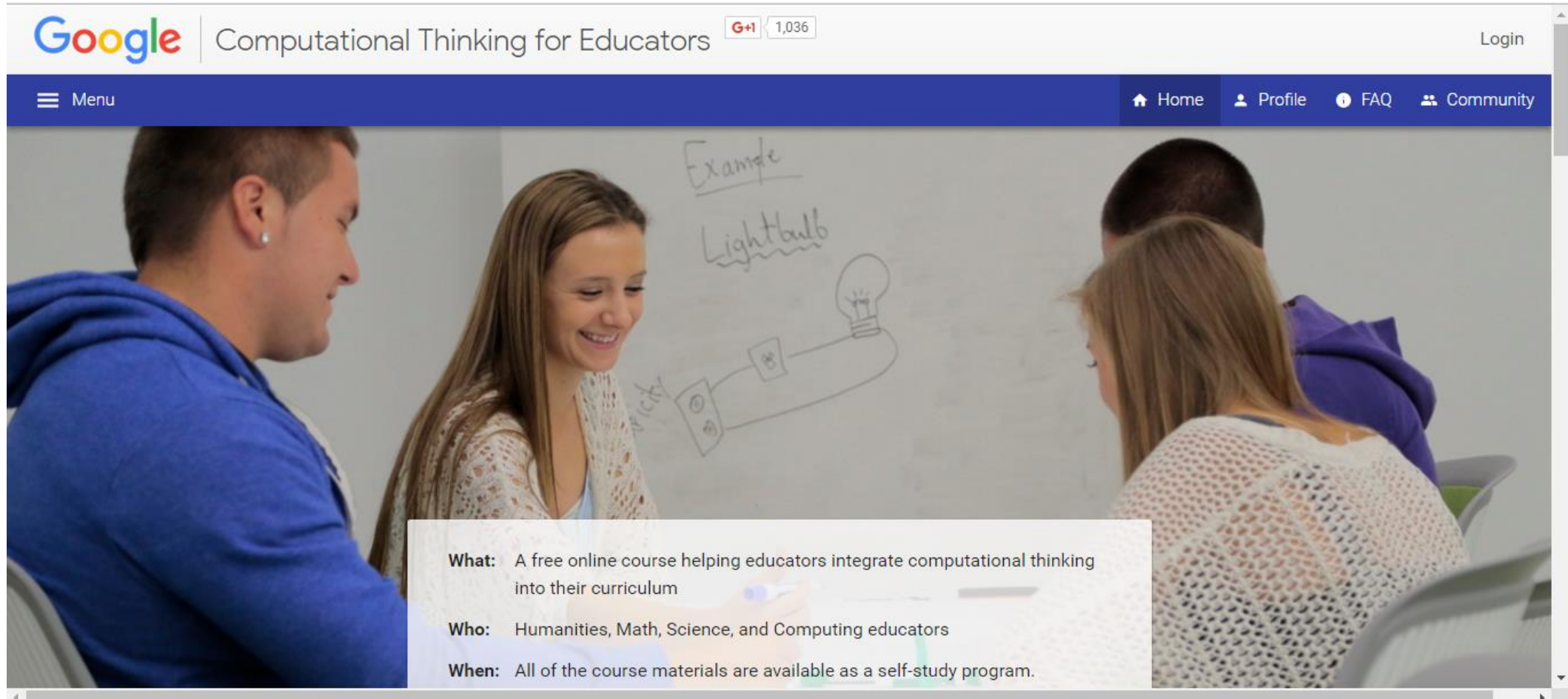




**INCORPORATE COMPUTATIONAL THINKING (CT)
INTO YOUR CURRICULUM**

MATERIALS

COMPUTATIONAL THINKING FOR EDUCATORS



The image shows a screenshot of a Google+ page for 'Computational Thinking for Educators'. The page features a blue header with the Google logo, the course title, a G+ icon, and a count of 1,036. Below the header is a navigation bar with 'Menu', 'Home', 'Profile', 'FAQ', and 'Community' options. The main content area displays a photograph of four people (two men and two women) gathered around a whiteboard. The whiteboard has the word 'Example' written at the top, followed by 'Lightbulb' and a hand-drawn circuit diagram. A text box is overlaid on the bottom right of the photo, providing details about the course.

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

<https://code.org/>

ANYBODY CAN LEARN COMPUTER SCIENCE

Every student in every school should have the opportunity to learn computer science

Take the diversity pledge

I agree

Join 2,180,230 others



Students

Explore our courses

Try Code Studio

Find a local class

Other online courses



Educators

Teach your students

Elementary school

Middle school

High school



Hour of Code

Anybody can learn. Start today

Try the Hour of Code

Host an Hour of Code

263,525,499 served



Advocates

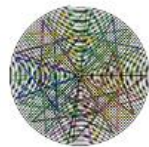
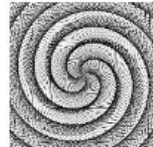
Support diversity in computing

See the stats

Get involved

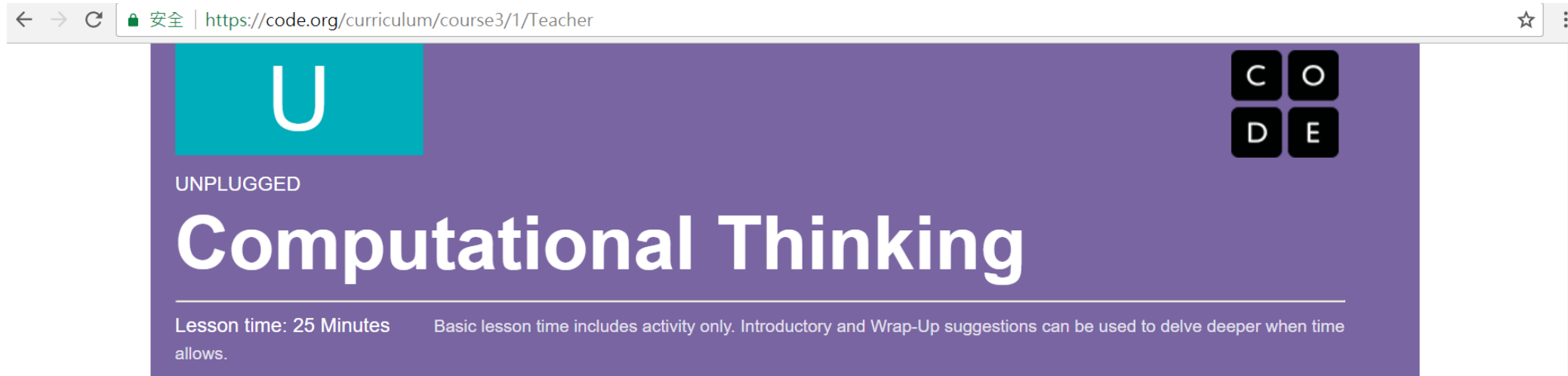
T-shirts, hats, & more

Code Studio student creations



<https://code.org/curriculum/course3/1/Teacher>

UNPLUGGED COMPUTATIONAL THINKING



← → ↻ 安全 | <https://code.org/curriculum/course3/1/Teacher> ☆ ⋮

U

C O
D E

UNPLUGGED

Computational Thinking

Lesson time: 25 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

For this activity, no instructions are provided. Instead, students will use examples of what imaginary players have done to figure out how to play the game. This lesson gives students the opportunity to practice the four arts of computational thinking (decomposition, pattern matching, abstraction, and algorithms) in one cohesive activity.

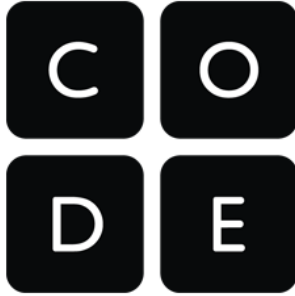
TEACHING SUMMARY

Getting Started - 15 minutes

- 1) [Vocabulary](#)
- 2) [Figuring it Out](#)

Activity: Computational Thinking - 25 minutes

- 3) [Computational Thinking](#)



TEACHING GUIDE

<https://code.org/curriculum/course3/1/Teacher>

MATERIALS, RESOURCES AND PREP

For the Student

- One die per group
- One [Computational Thinking Kit](#) per group
- Pens, Pencils, & Scissors
- [Computational Thinking Assessment](#) for each student

For the Teacher

- [Lesson Video](#)
- This Teacher Lesson Guide
- Print one [Computational Thinking Kit](#) per group
- Print one [Computational Thinking Assessment](#) for each student

GETTING STARTED (15 MIN)

1) Vocabulary

This lesson has four new and important words:

New Words!

<p>Decompose Say it with me: De-COM-pose <i>Break a problem down into smaller pieces</i></p>	<p>Pattern Matching Say it with me: Pat-ern Mat-ching <i>Finding similarities between things</i></p>
<p>Abstraction Say it with me: Ab-strac-shun <i>Pulling out specific differences to make one solution work for multiple problems</i></p>	<p>Algorithm Say it with me: Al-go-rithm <i>A list of steps that you can follow to finish a task</i></p>

<http://comphinking.csie.ntnu.edu.tw/>

教育部 運算思維推動計畫

教育部 運算思維推動計畫

HOME 活動資訊 活動花絮 資源共享 相關報導 關於本計畫 計畫行事曆 友站連結

You are here: Home > 資源共享 > 國中資訊教師運算思維增能研習 教學簡報檔

中小學教師運算思維增能研習 合格種子教師名單
Read more

中小學教師運算思維增能研習 教學簡報檔
Read more

國中資訊教師運算思維增能研習 合格種子教師名單...
Read more

國中資訊教師運算思維增能研習 教學簡報檔

資訊科技科課綱介紹-課綱精神

資訊科技科課綱介紹-課綱簡介

運算思維、程式設計與資訊科學 I

運算思維、程式設計與資訊科學 II

視覺化程式設計語言

程式設計工具體驗

專題製作與實施規劃

資訊科學主題_物聯網

資訊科學主題_資料科學

資訊科學主題_演算法

資訊科學活動介紹

資安、隱私、智財教材教法匯整

<http://ct.fg.tp.edu.tw/>

COMPUTER SCIENCE COURSE DESIGN WITH CT

← → ↻ ⓘ ct.fg.tp.edu.tw 🔍 ☆ ⋮

為了讓所有資訊教育工作者更能了解運算思維與資訊科學教育的內涵，我們從2017年5月起架設本課程分享網站，邀請一群對資訊教育有熱情的專家、學者與中小學教師，嘗試為運算思維導向資訊課程設計進行分享、改作、實戰、修正與再分享。歡迎每一位和我們有共同理想的教育工作者加入我們。您可以下載任何一份教案與教材，針對不同的學習者進行教材改寫，並用相同方式分享



COMPUTATIONAL THINKING

in Computer Science Course Design



<http://ct.fg.tp.edu.tw/>

COMPUTER SCIENCE COURSE DESIGN WITH CT

模組化程式設計 | 我的音樂盒

模組化程式設計 | 碎形~尋找大自然的密碼

數位人文 | 網路爬蟲、分析與視覺化

人工智慧 | 五子棋AI設計

資料編碼 | 看不懂的情書

結構化程式設計 | 終極密碼

資料科學 | 演算法設計 | 圖靈的紙條

模組化程式設計 | 猴子吃香蕉

程式設計專題 | 智慧型停車格停車系統製作

...



<http://paperplane-tw.weebly.com/>

紙飛機計畫：開放資訊教育平台



[首頁](#) [教案](#) [團隊](#) [支持紙飛機](#) [聯絡我們](#)

開放給所有人的
程式教育

紙飛機計畫：開放資訊教育平台

演算法 學畫畫

<http://paperplane-tw.weebly.com/>



紙飛機計畫
交大電子 X 程式設計

用演算法
畫畫

活動名稱：

用演算法畫畫 ALGORITHMIC DOODLE ART

教學單元：

演算思維 (algorithmic thinking)

計算建模 (computational modelling)

自然中的演算法 (algorithms in nature)

電腦製成圖像 (computer generated images)

遞迴 (recursion)

計算思維 (computational thinking)

數學：

分數 (fractions)

(給小孩的練習和探索)

譯者：許連逢

版權聲明：

Computer Science activities with a sense of fun: Created by Paul Curzon, and Peter McOwan Queen Mary University of London for Teaching London Computing: <http://teachinglondoncomputing.org>

本教材以創用 CC 3.0 姓名標示 - 非商業性 - 相同方式分享釋出 (creative commons 3.0 BY-NC-SA)



原文教案與教材連結：



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>

十二年國民基本教育課程綱要總綱(教育部發布版)(2014.11.28)。取自 <http://www.naer.edu.tw/files/15-1000-7944,c639-1.php?Lang=zh-tw>

[十二年國民基本教育課程綱要科技領域 \(草案 \) - 國家教育研究院\(2016.2\)](http://www.naer.edu.tw/ezfiles/0/1000/attach/92/pta_10229_131308_94274.pdf)。取自 http://www.naer.edu.tw/ezfiles/0/1000/attach/92/pta_10229_131308_94274.pdf

Thank you