程式人

用十分鐘搞懂《離散數學》

（集合論、布林代數、排列組合、狀態機、圖論、遞歸方程式）

外加《代數結構》與《作業研究》

陳鍾誠

2016 年 1 月 22 日

還記得念大學的時候

- 資訊系必須念《離散數學》
- 同學說這是資訊系必念的數學

那我問他

- 為甚麼叫離散數學呢？

他說

- 就《離離散散》的數學阿！
- 你沒看整本書東一塊西一塊的嗎？

當時

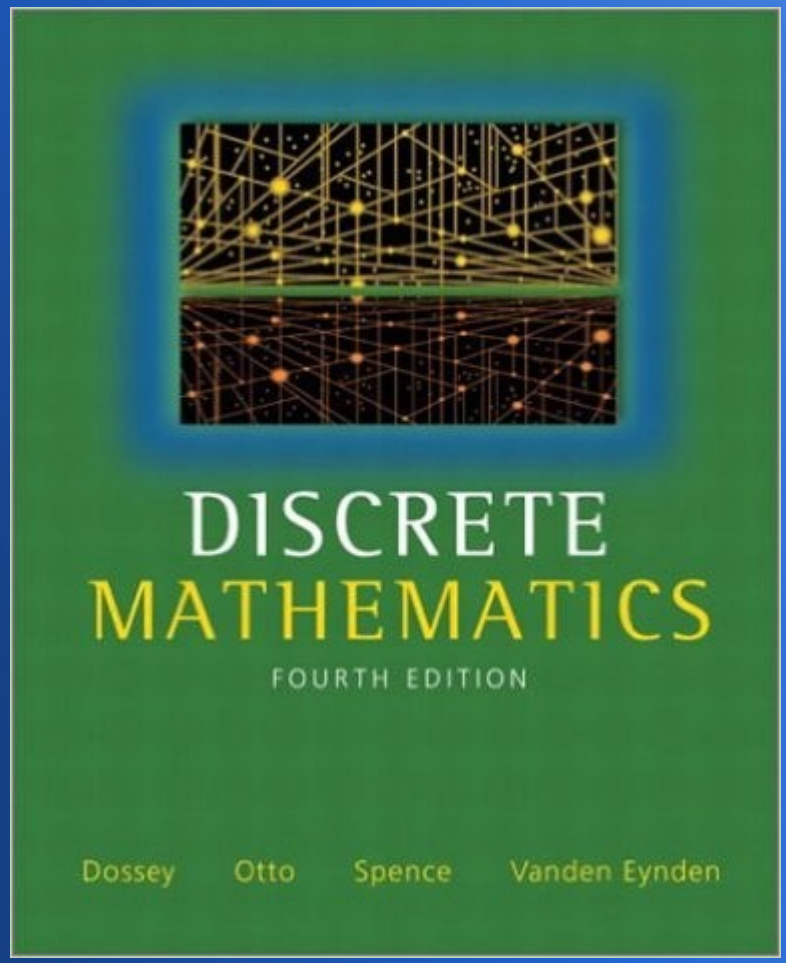
- 我們是用《劉炯朗》寫的書
- 雖然劉炯朗在澳門長大，但寫的書卻是英文的，我們只知道作者叫做 C. L. Liu 。

問題是、不管你念哪一本

- 內容真的就是東一塊西一塊的。
- 就讓我們來看看幾本離散數學的目錄好了！

John A. Dossey 的離散數學

- 第一章 組合問題與方法入門
- 第二章 集合、關係與函數
- 第三章 圖論 第四章 樹論
- 第五章 配對 第六章 網路流量分析
- 第七章 計數技術
- 第八章 遞迴關係與生成函數
- 第九章 組合電路與有限狀態機

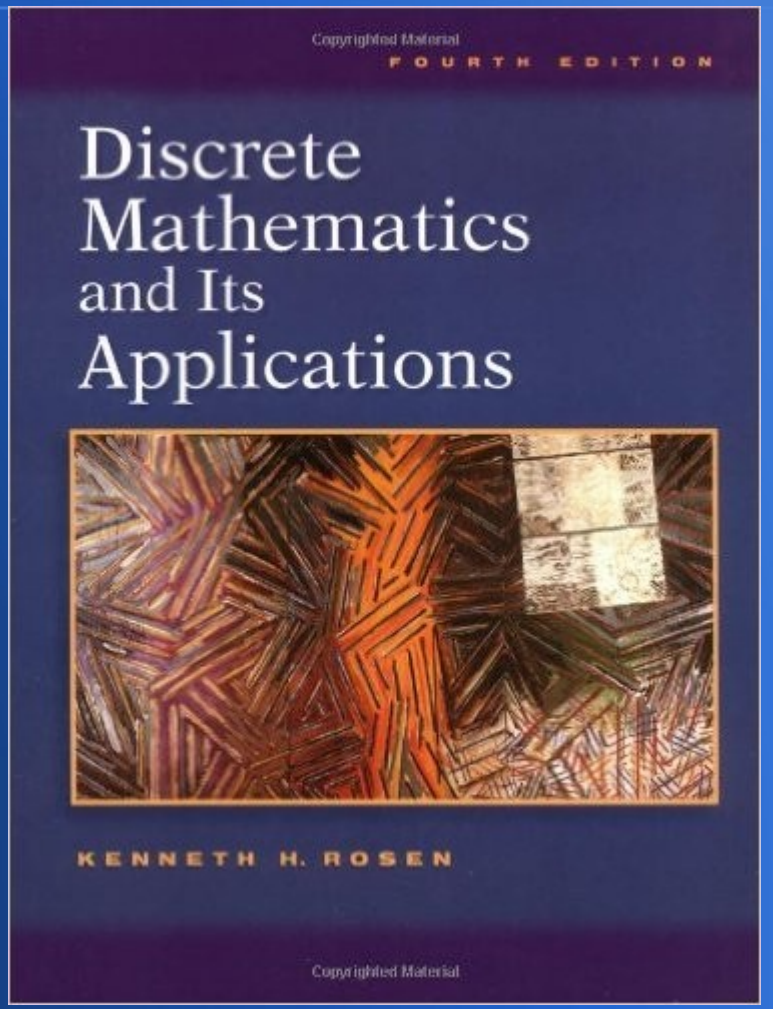


Kolman, Busby, Ross 的離散數學

- Ch01　基礎　Ch02　邏輯
- Ch03　計數　Ch04　關係與有向圖
- Ch05　函數　Ch06　有序關係與結構
- Ch07　樹　Ch08　圖論的主題
- Ch09　半群與群
- Ch10　語言與有限狀態機
- Ch11　群與編碼

Kenneth H. Rosen 的離散數學

- 第 1 章 基礎：邏輯與證明
- 第 2 章 集合、函數、序列與總和
- 第 3 章 基礎工具：演算法、整數與矩陣
- 第 4 章 歸納與遞迴 第 5 章 計數
- 第 6 章 進階計數技巧
- 第 7 章 關係 第 8 章 圖形 第 9 章 樹
- 第 10 章 布爾代數



您會發現

- 雖然內容都有不少重疊，

  但也有不少差異。

所以

- 離散數學真的是因為《離離散散、東一塊西一塊的》，所以才叫《離散數學》嗎？

這 …

當然不是囉！

那為甚麼叫離散數學呢？

喔！

那是因為

它不是連續數學

或者說

- 它就是《不連續數學》的意思！

學生 ...

那還不是一樣

不連續

- 所以就《斷斷續續》的 …
- 於是就東一塊西一塊囉！

老師

- 非也非也 . . .

所謂的不連續

- 是指《不是連續函數》的數學！

還記得剛進大學時

- 您曾經學過微積分吧？
- 微積分老師一直在說，假如一個函數是《連續且可微》的話，那麼...

相反的

- 離散數學就是研究《不是連續函數》的那種數學。
- 也就是
  - Not Continuous => Discrete

學程式設計的時候

- 您一定知道，電腦的數值通常有兩種，一種是《整數》 int ，另一種稱為《浮點數》 float
- float 是用來表達實數的！

連續數學的對象

通常探討的就是《實數空間》裏的問題

- 而離散數學，則是討論《整數》或者其他可以用電腦直接精確表達的結構，像是：
  - 0 與 1 – 布林代數
  - 一堆元素 – 集合
  - 一堆元素連起來 – 圖形
  - 一堆元素之間的運算關係 – 代數（群體環）

大致說來

- 《電腦的數學計算》可分為兩類，一種是《離散數學》，另一種是《數值分析》。

- 《數值分析》處理的對象，是《連續數學》經過《浮點數化》之後的結果。而離散數學則通常不討論《浮點數》。

以下是《數值分析》討論的主題

複數運算
多項式計算
矩陣計算
線性方程組的求解
非線性方程組的求解
代數插值法
數值積分法
常微分方程組的求解
擬合與近似
特殊函數
極值問題
亂數產生與統計描述
數學變換與濾波

您可以看到基本上就是

1. 微積分
2. 工程數學
3. 線性代數
4. 機率統計

該如何用電腦算的問題？

數值分析討論的

- 主要是《連續函數》的數學，該如何用電腦計算的問題。
- 而離散數學，則是討論《非連續函數》的電腦計算問題。

問題是

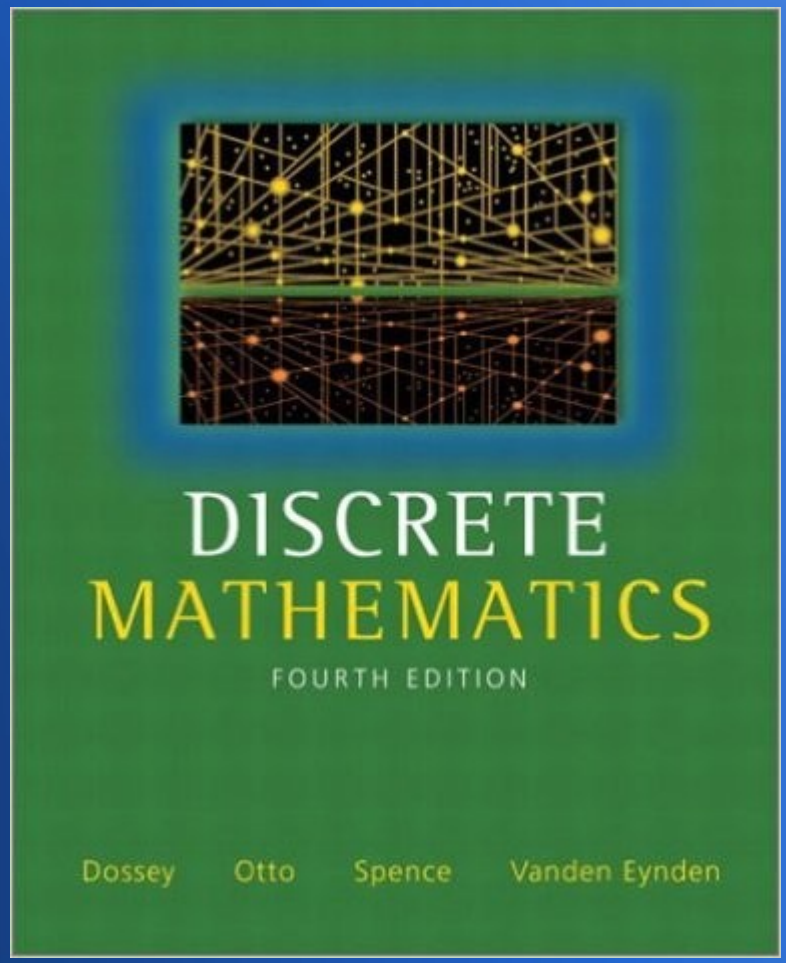
- 電腦到底會處理那些《非連續函數》的數學問題呢？

讓我們再看一下

- 前面那幾本書的目錄，應該就會瞭解了！

John A. Dossey 的離散數學

- 第一章 組合問題與方法入門
- 第二章 集合、關係與函數
- 第三章 圖論 第四章 樹論
- 第五章 配對 第六章 網路流量分析
- 第七章 計數技術
- 第八章 遞迴關係與生成函數
- 第九章 組合電路與有限狀態機

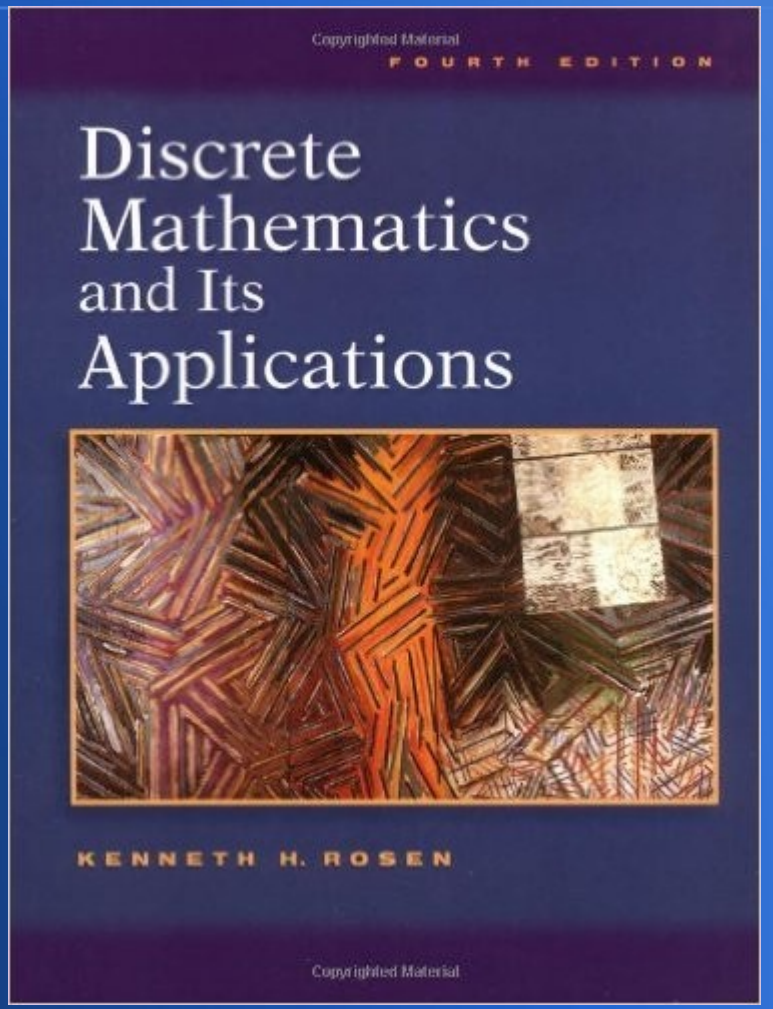


Kolman, Busby, Ross 的離散數學

- Ch01　基礎　Ch02　邏輯
- Ch03　計數　Ch04　關係與有向圖
- Ch05　函數　Ch06　有序關係與結構
- Ch07　樹　Ch08　圖論的主題
- Ch09　半群與群
- Ch10　語言與有限狀態機
- Ch11　群與編碼

Kenneth H. Rosen 的離散數學

- 第 1 章 基礎：邏輯與證明
- 第 2 章 集合、函數、序列與總和
- 第 3 章 基礎工具：演算法、整數與矩陣
- 第 4 章 歸納與遞迴 第 5 章 計數
- 第 6 章 進階計數技巧
- 第 7 章 關係 第 8 章 圖形 第 9 章 樹
- 第 10 章 布爾代數



讓我們稍微整理一下

- 大致上會整理出下列主題
  - 集合論
  - 布林代數
  - 排列組合計數
  - 狀態機
  - 圖論
  - 遞歸關係

其中有些書不只討論《布林代數》，而會擴大到《代數結構》上，像是《群、體、環》的主題就是《代數》

有些書會把《圖論》的《最大網路流問題》，獨立成一個章節。
如果《最大網路流》再進一步加上《線性規劃、整數規劃、二次規劃》等主題，差不多就是《管理數學》中《作業研究》的課程了

（作業研究是探討如何在有限制條件下最大或最小化的數學，這對工業生產管理很重要，是工業管理科系最重要的數學課程之一）

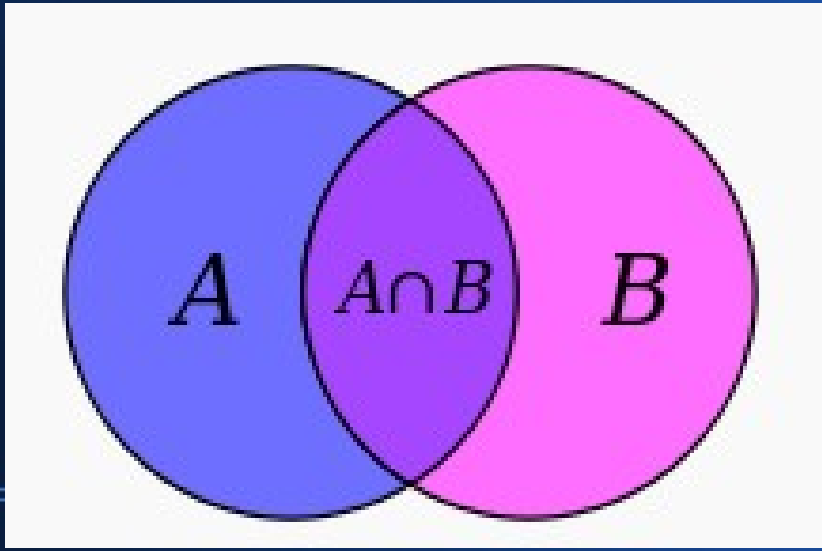
首先讓我們看看布林代數

- 布林代數可以說是電腦 0 與 1 的數學基礎。
- 您或許還記得《數位邏輯》中的 and, or, not，還有笛摩根定律和卡諾圖。

$a \vee (b \vee c) = (a \vee b) \vee c$	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$	結合律
$a \vee b = b \vee a$	$a \wedge b = b \wedge a$	交換律
$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$	吸收律
$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	分配律
$a \vee \neg a = 1$	$a \wedge \neg a = 0$	互補律

如果把布林代數稍微擴大

- 不要只有 0 與 1
- 那就會出現像 {0, 1, 2, 3, 4 ..} 或 {a, c, j, k … } 之類的內容，這就變成《集合論》了。



然後若進一步

- 討論《集合元素》形成的代數關係，就會探討
  - 群 (Group)、體(Field)、環(Ring)

  這些代數結構
- 《布林代數》就擴大為《代數學》了！
- 這裡還會延伸到《Galois Field》這個和密碼學有關的主題，也就是RSA這種公開金鑰密碼背後的數學。

讓我們看看一些重要的代數結構

- 特別是《群和體》

群（Group）

- 一個具有《封閉性、結合性，單位元素、反元素》的《集合與運算》所形成的數學結構

1. 封閉性： 對於所有G中a, b，運算$a \cdot b$的結果也在G中。[b]
2. 結合性： 對於所有G中的a, b和c，等式 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$成立。
3. 單位元素： 存在G中的一個元素e，使得對於所有G中的元素a，等式$e \cdot a = a \cdot e = a$成立。
4. 反元素： 對於每個G中的a，存在G中的一個元素b使得$a \cdot b = b \cdot a = e$，這裏的$e$是單位元素。

體（Field）

- 一種可進行《加、減、乘、除》運算的代數結構

體明確的滿足如下性質：

在加法和乘法上封閉

對所有屬於F的a, b，$a + b$和$a * b$屬於F（另一種說法：加法和乘法是F上的二元運算）。

加法和乘法符合結合律

對所有屬於F的a, b, c，$(a + b) + c = a + (b + c)$，$(a * b) * c = a * (b * c)$

加法和乘法符合交換律

對所有屬於F的a, b，$a + b = b + a$，$a * b = b * a$

符合乘法對加法的分配律

對所有屬於F的a, b, c，$a * (b + c) = (a * b) + (a * c)$

存在加法單位

在F中有元素0，使得所有屬於F的a，$a + 0 = a$

存在乘法單位

在F中有不同於0的元素1，使得所有屬於F的a，$a * 1 = a$

存在加法反元素

對所有屬於F的a，存在$-a$使得$a + (-a) = 0$

存在乘法反元素

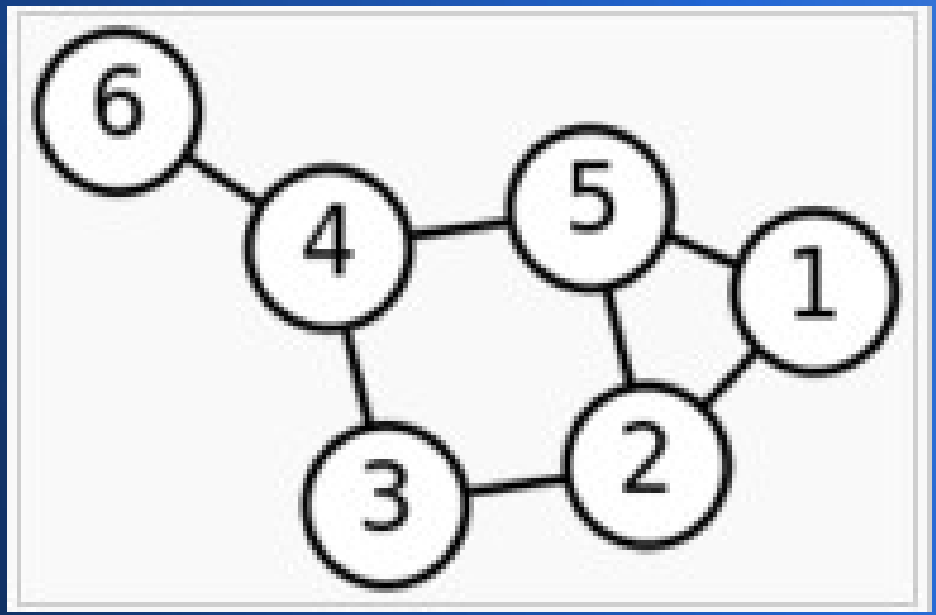
對所有$a \neq 0$，存在元素a^{-1}使得$a * a^{-1} = 1$

《代數》是很有系統的結構

- 但是《圖論》則是比較零散的結構

為了描述《集合元素》之間的關係

- 於是有了《圖論》和《關係》之類的結構。



有了集合、圖形與代數之後

- 為了要算算到底有多少可能的情況，就會出現《排列組合和計數》
- 有些資工系還會開一門進階的《組合數學課》，更深入的討論《離散數學的進階議題》，像是《Polya 計數定理》等等。

這樣、我們差不多串連起

- 離散數學從《布林、集合、圖論、代數》等這些主題了。
- 最後只剩下《遞歸關係》和《狀態機》還沒談到。

遞歸關係和狀態機

- 與演算法和電路有關
  - 遞歸關係 => 遞迴算法
  - 狀態機 => 有狀態的電路

遞歸關係的用途

- 是用來計算《遞迴算法》到底要執行多久用的。
- 舉例而言，您可以算出下列遞歸關係的 f(n) 是多少嗎？
  - f(n) = 2*f(n/2) + n
  - f(1) = 1

答案是 n log(n) ，這就是合併排序法的複雜度，代表了合併排序要執行這麼久。

- 描述遞歸關係的《差分方程式》，除了代入後用左右消去法之外，也可以用《生成函數》來算，這是更有系統的方法。

而狀態機呢？

- 則是《集合間轉移》的電路或演算法模型。
- 當您的程式或電路，允許有《狀態》的時候，就可以套用狀態機。

狀態機與電路的關係

- 沒有辦法儲存《狀態》的電路，稱為《組合電路》，而可以儲存狀態的電路，稱為《循序電路》。
- 您或許還記得，數位邏輯中從《正反器》開始的電路，是有倒勾迴路結構的，那種就稱為《循序電路》。
- 《正反器》可構成暫存器，《暫存器》可用來《儲存狀態》，於是電路就可以變成一種狀態機。

狀態機與演算法的關係

- 同樣的，如果不是用《電路》，也可以直接用《程式》寫出狀態機，這樣只要用簡單的幾個狀態就能完成系統，對記憶體的要求會非常的低。
- 像是《淘寶網》之類的大型網站，聽說為了降低伺服器的負擔，大量的採用狀態機架構來處理請求。
- 而且狀態機的觀念在編譯器的《LR 語法剖析器》當中也有用到。

另外《狀態機》若與《機率模型》結合

- 還會產生像《馬可夫鏈》這樣的《隨機過程》，對《語音處理、自然語言》等領域都有很強大的應用。

等到講完狀態機之後

- 可能會接著講《堆疊機》和《圖靈機》，這是《電路機器》角度的看法。
- 如果用《正規語言》的角度來看：《正規表達式 = 狀態機》、《Context-Free 語法 = 堆疊機》、《無限制生成語法 = 圖靈機》。
- 《正規語言》和《計算理論》只是用兩種不同角度來看計算的問題而已。

而且計算理論中

- 探討《可計算性》問題的部分，基本上是布林邏輯的延伸
- 而探討 NP-Complete 的部分，則是《演算法複雜度》的延伸。

於是就可以把

- 《數位電路、演算法、計算理論》連成一氣
- 形成資工系的理論體系之基礎。

大致上來說

- 離散數學就是《演算法》與《數位電路》所需要用到的數學。
- 而這也是為何《資訊科系》必須要學離散數學的原因。

希望看完本次的十分鐘系列之後

- 您會更清楚為何《資工系需要學離散數學》的原因。

離散數學

- 絕對不是東一塊西一塊，離離散散的數學喔！

希望您會喜歡這次的十分鐘系列

- 我們下次見

Good Bye !