閱讀書名:

解題背後的心理學

Psychology of Problem Solving: The Background to Successful Mathematics Thinking

班級: 117

座號: 35

作者: 謝育声

前言:

人類每天都在解決問題。從攸關生命、影響一生的人生轉捩點，以至於每天雞毛蒜皮、無關緊要的小事，都在解決問題的範疇裡。人類對問題探索、比賽與謎題的興致，自古至今對數學與邏輯學的發展貢獻甚大。而解題心理學也在這蓬勃發展的數百年間，終於開出茂盛的花果。本書旨在探討「解題」當下與處理問題時產生的心理狀態，與產生的心理狀態對於解題本身的影響。對在解決問題漫漫長路上初出茅廬的我，更顯其實用性。

圖書作者與資料:

書籍細節:

初版日期: 2021/3

出版商: 八旗文化/ 遠足文化

ISBN: 978-986-5524-43-2

作者:

Alfred S. Posamentier, 紐約市科技學院傑出講師

Gary Kose, 長島大學布魯克林校區心理學正教授

Danielle Sauro Virgadamo, 臨床心理學家

Kathleen Keefe-Cooperman, 長島大學諮商及發展系助理教授

譯者: 謝雯伃

內容摘錄 1: 河內塔問題

出處: *解題背後的心理學*，p.33~34

作者: 愛德華·盧卡斯(法)，1842~1891

謎題內容: 一間位於河內的寺廟裡有三根釘子，其中一根釘子上面放了 64 個由上而下依小到大放置的金盤。最大的盤子在底層，最小的則在最上層。僧侶奉神之命令，要將所有金盤依照由上到下、由小到大的順序，從第一根釘子移動到第三根釘子上。移動過程中，三根釘子全部都可以用到。當僧侶移動最後一個盤子時，世界就毀滅了，為什麼?

答案: 因為解開問題最簡潔的方法需要$2^{64}-1$次移動。就算每搬動一次盤子只需花一秒，也要花上 18446744073709551615 秒 = 307445734561825860 分鐘 = 約 584942417355 年!

若要實作: 用撲克牌當作金盤子，在三片瓷磚上就能夠實作了。

公式: 若有 n 個盤子，所需要最簡潔的解法就需要$2^{n}-1$次移動。

程式計算(Python): 假設每搬動一次盤子需要 1 秒

```
test.py > ...
n=int(input("please input the layers of your Hanoi tower:"))
print("Assuming moving a plate consumes a second.")
print("The time used is", 2**n -1, "s.")
print("Which is",(2**n -1)/60, "min.")
print("Which also is",(2**n -1)/3600, "h.")
print("Which also is",(2**n -1)/3600/360, "year")
```

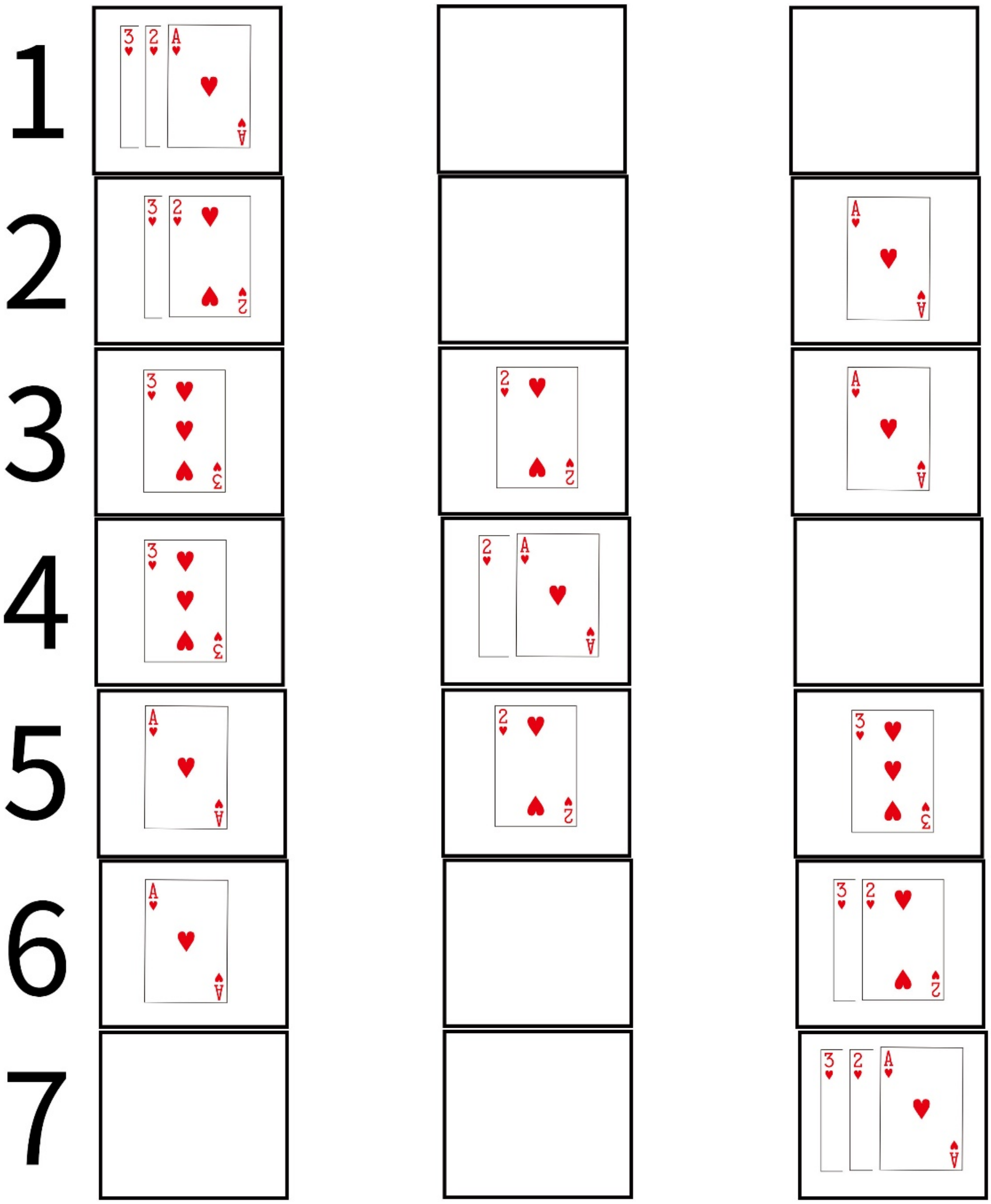
```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

PS D:\USER\Documents\GitHub> & C:/Users/USER/anaconda3/python.exe d:/USER/Documents/GitHub/test.py
please input the layers of your Hanoi tower:5
Assuming moving a plate consumes a second.
The time used is 31 s.
Which is 0.5166666666666667 min.
Which also is 0.008611111111111111 h.
Which also is 2.3919753086419752e-05 year
```

我的觀點 1:「找出規律(自創公式)」

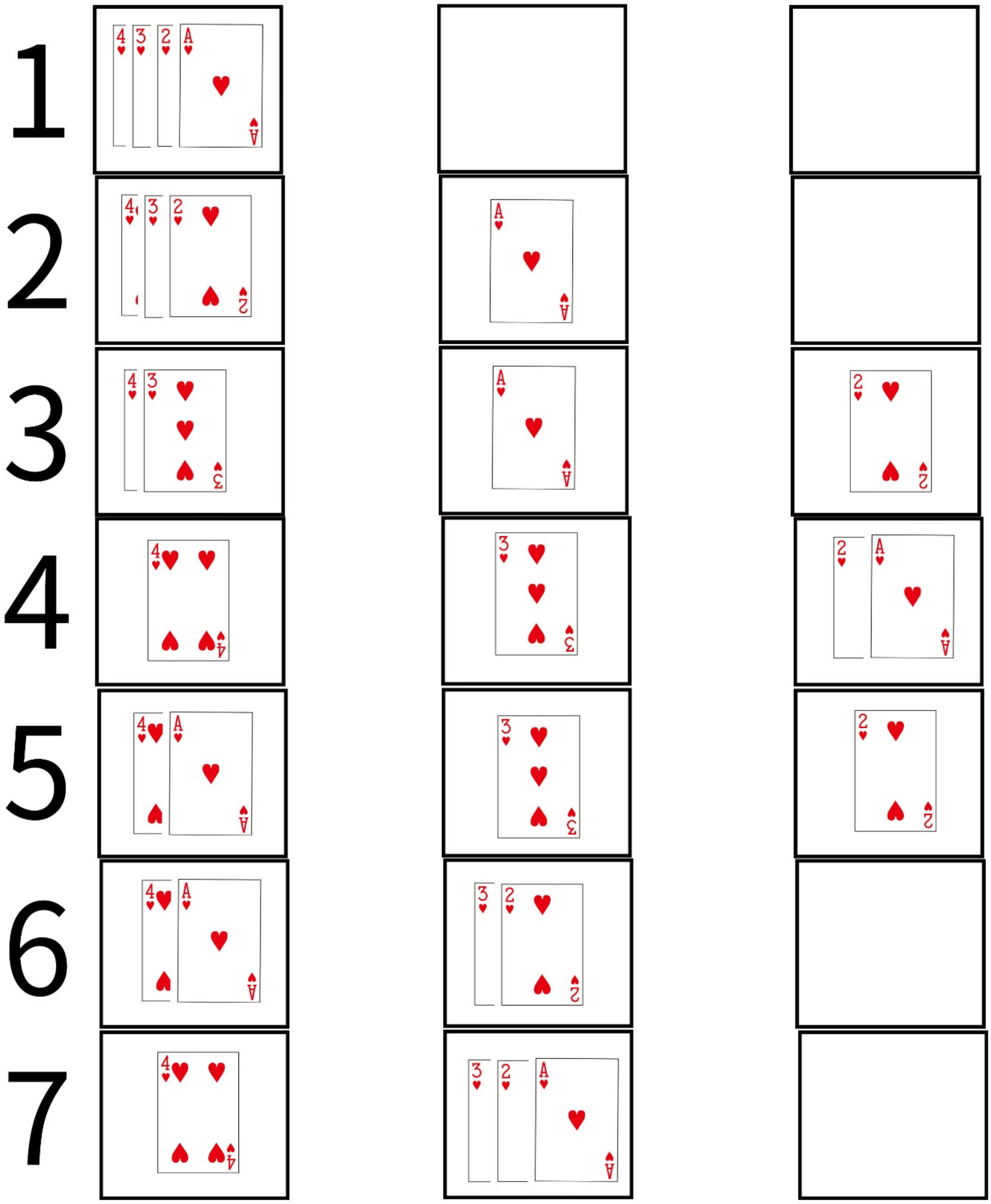
河內塔問題無論總共有幾個盤子，都有既定的移動方式。將相同的既定移動方式串連起來，便能形成迴圈(loop)。例如: 先移開 1 號盤(最小)，再移 2 號(次小)，再把 1 移到 2 上，就能空出一根釘子以放置更大的盤子。以此類推...

面對初次見到的題目，遇到時不應一頭栽進埋頭苦寫，而是要找出題目內含的規律(尤其是面對計算繁瑣的題目時)。就算前面多花了三五分鐘整理思緒、找出規律，短少的時間也能在解題時利用使用規律節省的時間迅速追回。

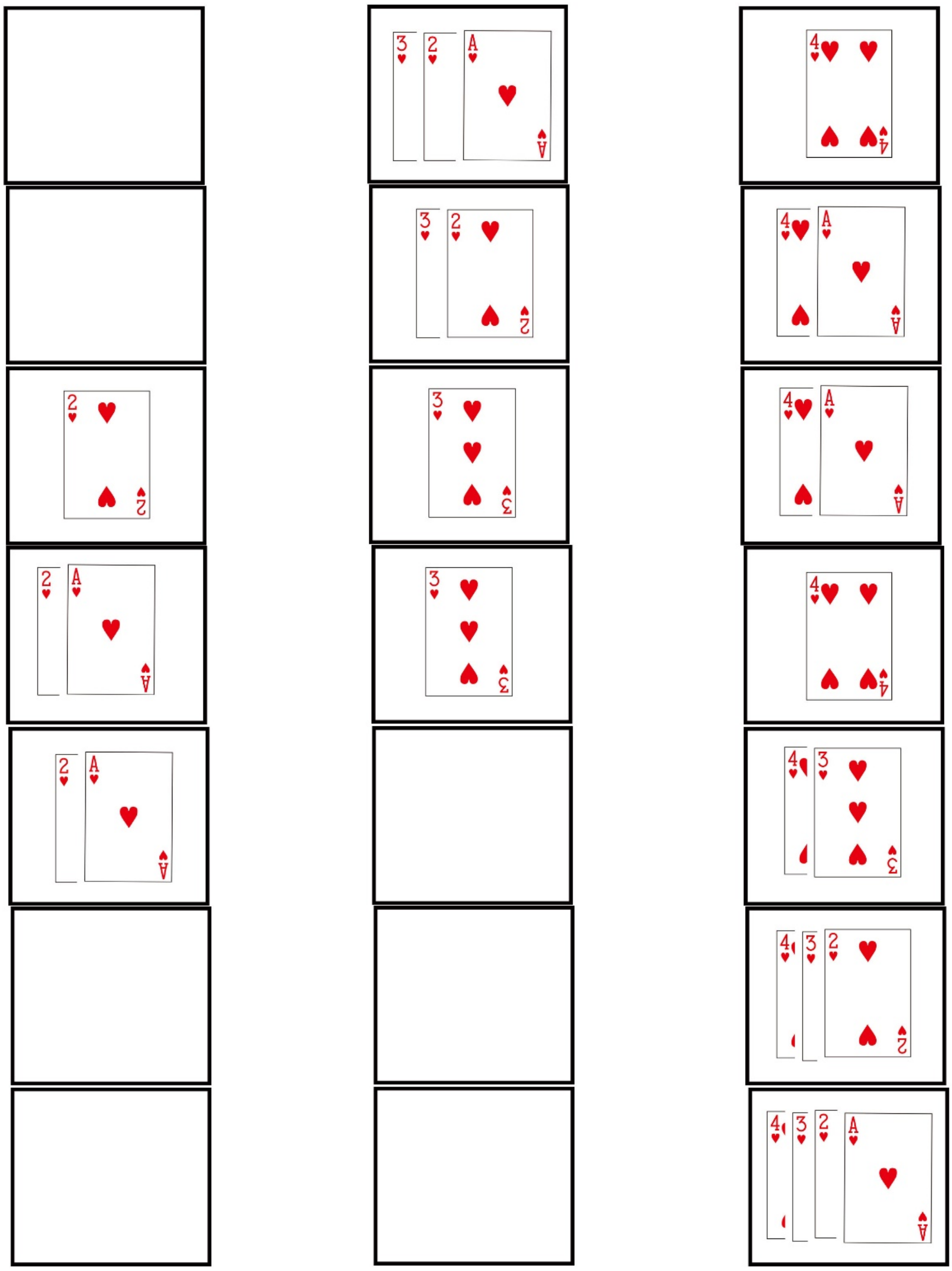
以撲克牌作範例 (一階三張牌，二階四張牌，以此類推)



一階河內塔謎題:2 -1=7次



二階河內塔謎題:$2^4-1=15$次



二階河內塔謎題:2^4-1=15次

內容摘錄 2: 心理定勢

出處: *解題背後的心理學*，p.35~37

學說提倡者: 奧斯瓦德·居爾佩(德), 1862~1915

學說內容: 居爾佩認為高等的心智程序是不倚賴感官形象的，而是可以透過觀察解題程序，以系統性、科學化的方法呈現出來。尤其大腦在經過多次重複性高的程序後，極容易受到框架的限制。甚至面對有更明確、簡潔解法的題目時，也會受到心理定勢極強的影響。

我的觀點 2:「試卷越後面越要注意定勢」

在解題時，越後面做的題目越容易受到前面試題解法的桎梏。小則解題步驟變長、浪費寶貴的考試時間; 大則完全無法解出題目。長時間練習同一類型的試題，充其量只能稱作「練習」，而無法稱為「解題」。甚至會導致更強的機械化解題定勢出現!

對於越高等精深的學術內容，定勢解題法便越吃虧。破解定勢魔咒的最佳辦法即是: 做試卷上每一題前都要先進行完整思考(就算似曾相識)，刺激自己大腦不要僵化，做出判斷再開始處理。

內容摘要 3: 框架效應

出處: *解題背後的心理學*，p.106~109

框架 1: 人類傾向換取更多利益，因此雖然表達內容相同，當敘述是正面闡述時，我們便會偏向該敘述。相反的，若題目採用負面敘述時，我們就會排除含有負面敘述的選項。

框架 2: 倘若題目的敘述包含「損失」時，我們會傾向「尋求風險」，賭一把試試看是否能減少損失; 如果題目敘述著重在「獲得」時，我們便會趨向保守，認為拿到「可確切到手的獎勵」就好。

我的觀點 3:

「一體兩面的敘述，

在做決定時會有截然不同的結果。」

如何解決: 客觀思考。常言: 「不見廬山真面目，只因身在此山中」、「當局者迷，旁觀者清」。跳脫出自己主觀思考的框架，就能看見問題更全面的解方，也能更有效率的解決問題。一件事情總有兩個面向，跳脫自己慣用的思考方向，才能看見問題全部的內容，進而以最全面的方法解決該問題。

內容摘要 4:

出處: *解題背後的心理學*，p.94~98

我的觀點 4:「共變不等於因果關係」

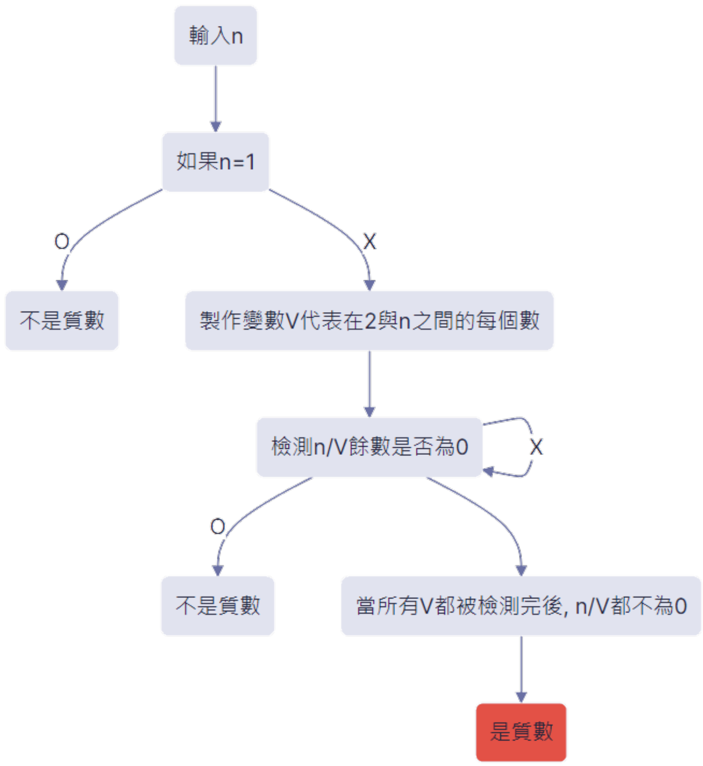
我們認為事物 A 與事物 B 有關係，多半都是因為其有共變關係(若 A 存 B 存，若 A 不存 B 不存…)。然而，絕大多數有共變的相關事物都沒有明確的因果關係。就如「成正比」與「有正向關係」不相同一般 。誤用共變關係，就會導致解決問題時被許多自己認定、浮雲飄渺的「線索」誤導，導致出錯。若要解決共變誤導，最佳方法即只相信有依據、可被多次證實的資訊來源，雖要大膽的假設，但更要小心的求證!

討論議題：演算法與捷思法

討論 1:演算法與捷思法在不同情況下的效率

示範例題：找出一正整數 n 是否為質數

電腦演算法處理(Python):




```python
def is_prime(number):
    for i in range(2,number):
        if number % i == 0:
            return False
    return True

n = int(input('輸入一個正整數:'))

if n ==1:
    print('不是質數')
elif is_prime(n):
    print('是質數')
else:
    print('不是質數')
```

人腦捷思法處理：經驗法則(曾經背過/用小質數找因數*)

設 n 為 5:

捷思法：幾乎瞬間(設為 0.0s)

演算法：約 0.0009965896606445312s

此時捷思法效率高於演算法

*拿到題目時，將 2、3、5、7、11、13...等小質數「代進去算」，若題目給出之數是某一小質數的倍數，該數便不是質數。

```python
import time

time_start = time.time() #開始計時

#要執行的程式碼，或函式
#要執行的程式碼，或函式
def is_prime(number):
    for i in range(2,number):
        if number % i == 0:
            return False
    return True

n = 5

if n ==1:
    print('不是質數')
elif is_prime(n):
    print('是質數')
else:
    print('不是質數')


time_end = time.time()    #結束計時

time_c= time_end - time_start   #執行所花時間
print('time cost', time_c, 's')
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS D:\USER\Documents\GitHub> & C:/Users/USER/anaconda3/python.exe d:/USER/Documents/GitHub/Python/計時.py
是質數
time cost 0.0009965896606445312 s
PS D:\USER\Documents\GitHub>
```

設 n 為

99999999999999999999999999999999999999999999

99999999999999999999999999999999999999999999

9999999999999999999999999999999999999997:

捷思法：算不出來(時間無限)，因為完全沒有處理超大數的經驗

演算法：約 0.5202786922454834 s

此時演算法效率遠高於捷思法

```python
import time

time_start = time.time() #開始計時

#要執行的程式碼，或函式
#要執行的程式碼，或函式
def is_prime(number):
    for i in range(2,number):
        if number % i == 0:
            return False
    return True

n = 99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999997

if n ==1:
    print('不是質數')
elif is_prime(n):
    print('是質數')
else:
    print('不是質數')


time_end = time.time()    #結束計時

time_c= time_end - time_start   #執行所花時間
print('time cost', time_c, 's')
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS D:\USER\Documents\GitHub> & C:/Users/USER/anaconda3/python.exe d:/USER/Documents/GitHub/Python/計時.py
不是質數
time cost 0.5202786922454834 s
```

討論:

由上面兩個實例可證出: 在處理有經歷過，我們認為較簡單的事物時，捷思法在效率上能有較好的成績(因為有經驗法則加持)。但在處理較複雜的題目時，演算法便會因為其「一定得的出解」的特性而遙遙領先(演算法使用時間雖長，但必小於捷思法無法解出問題的無限時間)。

延伸: 在打這段心得的時候，我理解到「人類認為一件事情簡單或艱難，很大一部分取決於『是否有處理過該類事務的經驗』。」因此，在做每一件「新」事情的時候，覺得艱難是人之常情，合乎邏輯的事情。然而，考量到未來面對該類問題能有迎刃而解的效率，探索新事物、踏出自己舒適圈的確有其必要性。對解決問題上如此，生活上亦如此。

討論 2: 演算法與捷思法的比較

	演算法	捷思法
是否必定有解	是	否
適用情境	無經驗的事物	有經驗的事物
常用於	電腦	人類
類似名詞	按照步驟	經驗法則
主動性(在人類上)	自動	需有意識使用

討論 3: 捷思法的主要類型

類型 1: 試錯法: 嘗試各種可想像的解，並找出符合題目敘述者。

缺點: 極度枯燥，且需要耗費大量時間。但基於電腦、電子計算機的特性，它們非常適合使用嘗試錯誤來解決問題。

類型 2: 窮舉搜索法(暴力法): 把符合問題的每一個解都進行驗證以得出答案。

缺點: 無法有效證明未來情況(無法預測未來，形成定理)。舉例: 試證明每一屆夏季奧運舉行年份都能被 4 整除(排除 1916, 1940, 1944, 2021 四年)。

公式證明法: 第一屆奧運為 1896 年雅典奧運，而 $1896 = 474 \times 4$ ，所以下 n 屆奧運為 $474 \times 4 + 1 = (474 + 1) \times 4$ 年，故無論是第幾屆夏季奧運，其舉行年分都會是 4 的倍數，故得證。

窮舉法: 1896, 1900, 1904, 1908, …, 2004, 2008, 2012, 2016 皆為 4 的倍數，故到目前為止，夏季奧運的舉行年分都是 4 的倍數，故得證。

類型 3: 代表性捷思法: 以該事物代表性的例證以推論未來情況。

常見謬誤: 以偏概全。舉例: 若拋擲一枚硬幣前五次都是反面，大部分人都會認為第六次拋擲的結果會是正面，以使正反面總比例接近 1:1。實際上，若硬幣質點在正中心，質量分布平均，第六次拋擲出現反面的機率仍是 50%。

類型 4: 目標分析法:

找出最終目標與目前處境之間的次目標，一步步完成小目標以接近最終目的。

雖說「不要急著吃棉花糖」，但人類天性仍傾向於能有立即性享受的事物(這就是為什麼比起好好看完一本小說，我們更想馬上去打一場傳說)。將大目標分割成易於完成的小目標，藉由達成小目標所獲得的成就感，按部就班的推進至最終宏大的目標。

結語:

數學，不只存在於死刻呆板的課本、講義、考卷上。生活中無時無刻做出的決定，都是數學邏輯判斷與心理狀態交織成的妥協方案。學習如何做出在當下情況中能產生最多得利者的決定，對於生活至關重要。而其中對解題思維影響甚巨的，便是框架、定勢、與共變所造成的誤導。藉由客觀思考、跳脫原始思維、與小心求證三種解題方式，便能一一將誤導化解。如此一來，無論是面對何種問題，皆能迎刃而解!

資料來源:

1. Posamentier, A.S. ; Kose, G. ; Virgadamo, D.S. ; Keefe-Cooperman , K. (2021)。《解題背後的心理學》。新北：八旗文化/遠足文化。
2. 撲克牌圖源: https://tw.pixtastock.com/illustration/64081811
3. 撲克牌流程圖: 謝育声製圖，Adobe Illustrator
4. 因果/共變圖: 謝育声製圖，Adobe Illustrator
5. 窮舉法: https://www.wikiwand.com/en/proof_by_exhaustion
6. 嘗試錯誤法: https://www.wikiwand.com/zh_TW/嘗試錯誤法
7. 奧運舉辦次數: https://ctee.com.tw/news/global/241996.html
8. 代表性捷思法: https://www.wikiwand.com/zh_TW/代表性捷思法