

河內塔

資工三甲 潘廷相 B0829065

什麼是河內塔問題，這個問題中 64 個金盤的它每個的大小都是不一樣的，並在一開始的時候會依序從底部最大排到頂部最小，而搬移的規則有三個：一次只能搬移一個金盤 盤子只能在三根柱子中被搬動（你不能拿起來放在旁邊，必須挑一根柱子放下。）尺寸較小的盤子需永遠保持在最上方，最後數學家用公式證明出，解決河內塔問題的最佳步驟為 $2^N - 1$ 次，其中的 N 為金盤的數量，一般化的河內塔問題中圓環的個數為 8。

和 recursion(遞歸)有什麼關係，我們經由簡單的測試，可以發現當 $N=1$ 時，只需要搬 1 次， $N=2$ 時，需要搬 3 次， $N=3$ 時，需要搬 7 次， $N=4$ 時，需要搬移 15 次，便可以知道最佳步驟似乎是必然可以等於 $2^N - 1$ ，由此得知，此河內塔問題是有規律可進行運算的，當知道問題具有固定模式或重覆規律的問題時，"遞迴方法"便是我們可以使用的一個很有效的思考策略。(費氏、等差、等比皆是相同概念的例子)

怎麼用 recursion(遞歸)解決此問題，我們需要知道遞迴的關係式、一般項通式，我們可以由上述的關係得知，A、B、C 三個塔，有 N 個盤，目標是把這些盤全部移到 C 塔。那麼我們要先把 $N-1$ 塊盤移到 B 塔，再把最大的盤移到 C，再把 B 塔的 $N-1$ 塊盤移到 C。以四盤為例，我們必須將圓盤 1-3 由 A 移到 B，才能移動圓盤 4 由 A 移到 C。為了完成圓盤 1-3 由 A 移到 B，必需將圓盤 1-2

由 A 移到 C，才能移動圓盤 3 由 A 移到 B。為了完成圓盤 1-2 由 A 移到 C，必需將圓盤 1 由 A 移到 B，才能移動圓盤 2 由 A 移到 C。

由此我們便可以利用此方式，用 recursion(遞歸)解決此問題。

C 語言程式碼，16 個盤，紀錄時間，精確到秒

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void hanoi(int n, char A, char B, char C) {
6      if(n == 1) {
7          printf("Move sheet from %c to %c\n", A, C);
8      }
9      else {
10         hanoi(n-1, A, C, B);
11         hanoi(1, A, B, C);
12         hanoi(n-1, B, A, C);
13     }
14 }
15
16 int main() {
17     clock_t begin, end;
18     double cost;
19     //開始記錄
20     begin = clock();
21     /*待測試程式段*/
22     int n = 16;
23     hanoi(n, 'A', 'B', 'C');
24     //結束記錄
25     end = clock();
26     cost = (double)(end - begin)/CLOCKS_PER_SEC;
27     printf("constant CLOCKS_PER_SEC is: %ld, time cost is: %lf secs", CLOCKS_PER_SEC, cost);
28     return 0;
29 }
30
```

執行結果：

```
Move sheet from A to B
Move sheet from A to C
Move sheet from B to C
constant CLOCKS_PER_SEC is: 1000, time cost is: 8.358000 secs
-----
Process exited after 8.55 seconds with return value 0
請按任意鍵繼續 . . .
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Dez\Desktop\main.exe
- Output Size: 129.119140625 KiB
- Compilation Time: 2.63s
```

CPU 型號 - 處理器 Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz，2501

Mhz，4 個核心，4 個邏輯處理器