

計概作業:河內塔

B1029002 王琮翔

資工人一生必搬一次的塔



一座你搬完就世界末日的塔

什麼是河內塔問題？

河內塔問題是根據傳說形成的數學問題，有三根杆子 A、B、C。A 杆上有 N

個 ( $N > 1$ ) 穿孔圓盤，盤的尺寸由下到上依次變小。要求按下列規則將所有圓

盤移至 C 杆：

1. 每次只能移動一個圓盤；

2. 大盤不能疊在小盤上面。

提示：可將圓盤臨時置於 B 杆，也可將從 A 杆移出的圓盤重新移回 A 杆，

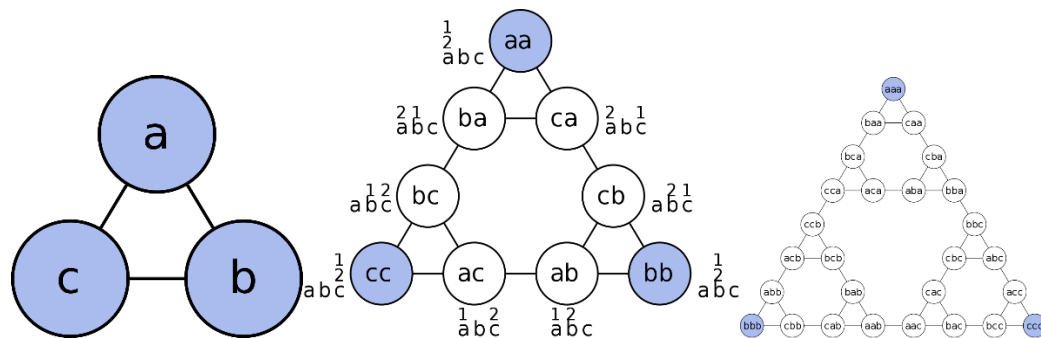
但都必須遵循上述兩條規則。

問：如何移？最少要移動多少次？

最早發明這個問題的人是法國數學家愛德華·盧卡斯(EDWARD LUCAS)。

傳說越南河內某間寺院有三根銀棒，上串 64 個金盤。寺院裡的僧侶依照一個古老的預言，以上述規則移動這些盤子；預言說當這些盤子移動完畢，世界就會滅亡。這個傳說叫做梵天寺之塔問題 ( Tower of Brahma puzzle )。但不知道是盧卡斯自創的這個傳說，還是他受他人啟發。

若傳說屬實，僧侶們需要  $2^{64}-1$  步才能完成這個任務；若他們每秒可完成一個盤子的移動，就需要 5849 億年才能完成。整個宇宙現在也不過 137 億年。



這個傳說有若干變體：寺院換成修道院、僧侶換成修士等等。寺院的地點眾說紛紜，其中一說是位於越南的河內，所以被命名為「河內塔」。另外亦有「金盤是創世時所造」、「僧侶們每天移動一盤」之類的背景設定。

和 recursion 的關係？

Recursion(遞迴)與河內塔的關係是因為河內塔問題的解法基本思想是遞迴。假

設有 A、B、C 三個塔，A 塔有  $n$  塊盤，目標是把這些盤全部移到 C 塔。那麼

先把 A 塔頂部的 塊盤移動到 B 塔，再把 A 塔剩下的大盤移到 C，最後把

B 塔的 塊盤移到 C。如此遞迴地使用下去，就可以求解。

怎麼用 recursion 解決此問題？

$$a_1=1$$

$$a_2=2 \times a_1 + 1$$

$$a_3=2 \times a_2 + 1 = 2 \times (2 \times a_1 + 1) + 1 = (2^2 a_1 + 2) + 1$$

$$a_4=2 \times a_3 + 1 = 2 \times ((2^2 a_1 + 2) + 1) + 1 = (2^3 a_1 + 2^2 a_1 + 2) + 1$$

$$a_n=2 \times a_{n-1} + 1 = (2^{n-1} a_1 + 2^{n-2} a_1 + \dots + 2^3 a_1 + 2^2 a_1 + 2) + 1 = 2^{n-1} a_1 + 2^{n-2} a_1 + \dots + 2^3 a_1 + 2^2 a_1 + 2^1 + 2^0$$

$$= 2^{n-1} + 2^{n-2} + \dots + 2^3 + 2^2 + 2^1 + 2^0$$

$$= 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{n-2} + 2^{n-1}$$

$$= 2^n - 1$$

另有想法，利用遞迴關係式  $a_n = 2a_{n-1} + 1$ ， $a_1 = 1$ ，求  $a_n$  的通式。

因為  $a_n = 2a_{n-1} + 1$ ，所以  $a_{n+1} = 2a_n + 1$ ，得  $a_{n+1} + 1 = 2(a_n + 1)$ 。因此

$$a_2 + 1 = 2(a_1 + 1) \quad \dots(1)$$

$$a_3+1=2(2a_2+1) \quad \dots(2)$$

$$a_4+1=2(2a_3+1) \quad \dots(3)$$

$$a_n+1=2(a_{n-1}+1) \quad \dots(n-1)$$

上列各式相乘 $(1) \times (2) \times (3) \times \dots \times (n-1)$ ，等號左右等量除、對消後，得

$$a_n+1=2(a_1+1) \times 2^{n-2}。$$

因為  $a_1=1$ ，所以  $a_n+1=2^2 \times 2^{n-2}$ ，得  $a_n=2^n-1$ 。

如果起始有  $n$  個圓盤在同一根柱子(A)，越上層的圓盤越小。您得將原先第二層到第  $n$  層的圓盤移到另外第二根柱子(B)，總共需移動  $2^{n-1} - 1$  次。之後，將原先的第一層圓盤(最大者)移至第三根空柱(C)上;再將柱子(B)的  $n-1$  個圓盤移到柱子(C)上，總共需移動 $(2^{n-1}) - 1$  次。完成整個過程需要移動的次數是 $(2^{n-1}) - 1)+1+(2^{(n-1)} - 1)=(2^n) - 1$ (次)。

「有三根柱子，原有  $n$  個圓盤套在同一根柱子，圓盤依大小由下而上，越上層則越小。欲完成移動  $n$  個圓盤套在其他的同一根柱子，則需要移動圓盤共  $2^n-1$  次。」

程式碼如下:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int i = 0;
5  void hanoi(int n, char L, char M, char R) {
6      if(n == 1) {
7          i++;
8      }
9      else {
10         hanoi(n-1, L, M, R);
11         hanoi(1, L, M, R);
12         hanoi(n-1, L, M, R);
13     }
14 }
15
16 int main() {
17     clock_t start, end;
18
19     int n;
20     printf("請輸入盤數：");
21     scanf("%d", &n);
22
23     start = clock();
24
25     hanoi(n, 'L', 'M', 'R');
26
27     end = clock();
28
29     double diff = end-start;
30     printf("%d %f sec", i, diff / CLOCKS_PER_SEC );
31
32     return 0;
33 }
34

```

C:\Users\王琮翔\Desktop\新文件1.exe

```

請輸入盤數：40
-1 3733.115000 sec
-----
Process exited after 3737 seconds with return value 0
請按任意鍵繼續 . . .

```

CPU: AMD Ryzen 7 4700U with Radeon Graphics 2.00 GHz