

河內塔 Hanoi Tower

河內塔相傳是源自古印度神廟中的一段故事。傳說在古老的印度，有一座神廟，據說它是宇宙的中心。在廟宇中放置了一塊上面插有三根長木釘的木板，在其中的一根木釘上，從上至下被放置了 64 片直徑由小至大的圓環形金屬片。古印度教的天神指示祂的僧侶們將 64 片的金屬片移至三根木釘中的其中一根上。直到有一天，僧侶們能將 64 片的金屬片依規則從指定的木釘上全部移動至另一根木釘上，那麼，世界末日即隨之來到，世間的一切終將被毀滅。後來演變為一個著名的數學及程式問題。

簡單介紹一下河內塔問題是什麼，基本上就是 3 個塔 A、B、C，A 塔上有 N 個($N > 1$)圓盤，圓盤尺寸由下到上依次變小，要在每次只能移動一個圓盤，且大盤不能疊在小盤上的規則，將所有的圓盤移到 C 塔上。試問要如何移動，且最少移動次數為何？

承接上一段，最少移動次數為何呢？我們可以透過遞迴關係來思考，重新審視題目，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。那我們可以這樣想，當 $N=2$ 時，我們要先將小盤從 A 移到 B，再將大盤從 A 移到 C，最後將小盤從 B 移到 C 即可完成。那 $N=3$ 時呢？ $N=3$ 時，第一步要先將上面兩個小盤移到 B，再將最後一個大盤移到 C，最後一步將 B 的兩個小盤移到 C，在這裡我們可以注意到，第一步及最後一步都有「移動兩個小盤」的動作，也就是 $N=2$ 的那些步驟。所以我們就能整理出一個遞迴關係，也就是當有 N 個塔時，先把 A 塔頂部的 $N-1$ 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 $N-1$ 塊盤移到 C，如此遞迴下去便能得到結

果。

那我們回到一開始的傳說，搬動 64 個盤子到底需要多少個動作？有可能被搬完嗎？所以我們來估計一下河內塔遞迴演算法的複雜度，令搬動 N 個盤子，從塔 A 到塔 C 需要 S_n 個步驟，已知 $S_1 = 1, S_2 = 3, S_3 = 7$ ，且 $S_n = S_{n-1} + 1 + S_{n-1} = 2S_{n-1} + 1$ ，所以他滿足下列遞迴定義， $S_n = 2S_{n-1} + 1$ ，邊界條件為 $n = 1, S_n = 1$ ，可得 $S_n = 2^n - 1$ 。所以搬動 64 個盤子需要 $2^{64} - 1$ 約等於 $1.84e19$ 個動作，基本上來說，就是搬不完的 ouo。

接下來就來看看 C 是怎麼做河內塔的~

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int i = 0;
5  void hanoi(int n, char A, char B, char C) {
6      if(n == 1) {
7          i++;
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 start, end;
18
19     int n;
20     printf("input how many plates :");
21     scanf("%d", &n);
22
23     start = clock();
24
25     hanoi(n, 'A', 'B', 'C');
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 }
```

測試結果 CPU 為 AMD R5 2600



```
請輸入盤數:30
1073741823 4.320000 sec
-----
Process exited after 5.447 sec
請按任意鍵繼續...
```