

河內塔是一種擁有三種柱子(ABC 三柱)以及一些有大小之分的圓盤所組成的裝置，一開始，這些圓盤會已由小排到大的狀態擺在最左邊的柱子上(A 柱)，我們需要經過一系列的操作將這些圓盤永遠已由小排到大的狀態從 A 柱移到最右邊的柱子上(C 柱)。而在這一系列的操作中似乎又擁有著一種數學上的規律，使得我們可以經過簡單的邏輯計算將移動的次數計算出來。

Recursion，遞迴，在程式中是一種函式呼叫自己的一個動作，只要有結束遞迴的條件，就可以在幾行程式裡作出大量且有規律的計算。雖然會經過大量的計算，然而只要我們經過一些效率上的設計，遞迴的計算方式並不會對電腦造成負擔，甚至可以比一般的迴圈還要更有效率。

而我們需要怎麼樣才能利用遞迴的方式寫出河內塔的程式呢？我們先觀察一般動兩塊時的樣子，先把一個圓盤移到 B 桿上，再將第二個圓盤一動到 C 桿，接著就是將 B 桿上的圓盤移到 C 桿上，即完成。若是多一個盤子呢？我們需要先用兩個盤子的方式將圓盤移至 B 桿，再將最後的圓盤移到 C 桿，再把 B 桿的一個移到 A 桿，第二個移到 C 桿，A 桿上的再放到 C 桿上，即完成。也就是我們不管如何都是先把 A 桿上的 $(n - 1)$ 層移到 B 桿上，再將 A 桿剩下的第  $n$  層移至 C 桿，再把 B 桿的 $(n - 1)$ 移到 C 桿上。所以我們就先將 A 桿上 $(n - 1)$ 的圓盤移到 B 桿上，也就是  $\text{hanoi}(n - 1, A, C, B)$ 。再將第  $n$  層從 A 移到 C，也就是  $\text{hanoi}(1, A, B, C)$ ，最後將 B 桿上的  $n - 1$  層放到 C 層上，即為  $\text{hanoi}(n - 1, B, A, C)$ ，之後利用遞迴的呼叫把裡面  $n - 1$  層也以這樣的方式一個一個地移到 C

桿上。所以最後我們的 function 就是按照上面的思考方式寫出來的遞迴式。程

式碼如下：

```
#include<stdio.h>
```

```
void hanoi(int n, char A, char B, char C) {
```

```
    if (n == 1) {
```

```
        printf("Move disk %c to %c\n", A, C);
```

```
    } else {
```

```
        hanoi (n - 1, A, C, B);
```

```
        hanoi (1, A, B, C);
```

```
        hanoi (n - 1, B, A, C);
```

```
    }
```

```
}
```

```
int main() {
```

```
    int n = 16;
```

```
    hanoi (n, 'A', 'B', 'C');
```

```
    return 0;
```

```
}
```

而經過 time 函數計算，我的電腦需要執行約 9 秒才能將 16 個圓盤搬完。