

## 1.河內塔的起源

1883 年，一位法國的數學家 Edouard Lucas 教授在歐洲的一份雜誌上介紹了一個相當吸引人的難題——河內塔(Tower of Hanoi)，它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。規定在每次的移動中，只能搬移一片金屬片，並且在過程中必須保持金屬片由上至下是直徑由小至大的次序，也就是說不論在那一根木釘上，圓環形的金屬片都是直徑較小的被放在上層。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，萬物都將至極樂世界。

## 2.河內塔的數學原理

要推測 64 層的河內塔需要幾次搬動，我們可以先從簡單的下手，也就是層數較少的時候，設順序  $A - B - C$ ，要把塔從  $A$  搬到  $C(A \rightarrow C)$ ：

當層數 1 的時候：要 1 次( $A \rightarrow C$ )

當層數 2 的時候：要先做一次層數 1( $A \rightarrow B$ )，再把底層搬到  $C$ ，最後在一次 1 層( $A \rightarrow C$ )

當層數 3 的時候：要先做一次層數 2( $A \rightarrow B$ )，再把底層搬到  $C$ ，最後在一次 2

層(A→C).....

由此可以推敲，當第  $n$  層時，要先把上面  $n-1$  層搬到 B，然後把底層搬到 C，

最後再把 B 的  $n-1$  層搬回 C。由此可以推敲出一個數學的遞迴關係式：

$$\begin{cases} a_1 = 1 \\ a_n = 2a_{n-1} + 1 (n \geq 2) \end{cases}$$

其中  $a_1 = 1$ ，再藉由數列與級數的演算，原本的式子是可以再化簡成一般式：

$$a_n = 2^n - 1$$

### 3.與 recursion 的關係

而剛剛的數學有遞迴式，剛好可以運用 recursion 來解決此問題，因在遞迴式

中即看到每項都需呼叫前一項的質來進行運算，直到剩 1 層得時候，此本身即

是 recursion 的概念，所以河內塔便可輕易由 recursion 實踐。

### 4.如何實踐

實際的方法可直接使用上述數學推倒過程的方法去實踐，只需將之寫成城市去

執行

程式碼

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

```
void recursion(int num, char from, char mid, char dest) {
```

```
    if (num == 1) {
```

```
        printf("from %c move to %c\n", from, dest);
```

```

    } else {

        recursion(num - 1, from, dest, mid);

        recursion(1, from, mid, dest);

        recursion(num - 1, mid, from, dest);

    }

}

```

```

int main() {

    int n = 16;

    recursion(n, 'A', 'B', 'C');

}

```

執行結果

```

from A move to C
from B move to C
from B move to A
from C move to A
from C move to B
from A move to B
from C move to A
from B move to C
from B move to A
from C move to A
from B move to C
from A move to B
from A move to C
from B move to C
from A move to B
from C move to A
from C move to B
from A move to B
from A move to C
from B move to C
from B move to A
from C move to A
from B move to C
from A move to B
from A move to C
from B move to C
-----
Process exited after 3.916 seconds with return value 0
請按任意鍵繼續

```

Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz    2.90 GHz