

平面上有 A、B、C 三根木樁，其中的木樁 A 由上而下套著由小而大的圓環，假設我們想要將所有圓環由木樁 A 搬到木樁 C，而且搬動的過程受到以下三項限制：

- 1.一次只能搬動一個圓環。
- 2.每次搬動都須由某根木樁搬到另一根木樁，圓環不能被暫時放到其他地方。
- 3.對任何木樁上任意兩個相疊的圓環而言，上面的圓環一定要比下面的圓環小。請問：要完成此項工作最少須搬動圓環幾次？

以上就是著名的河內塔問題。

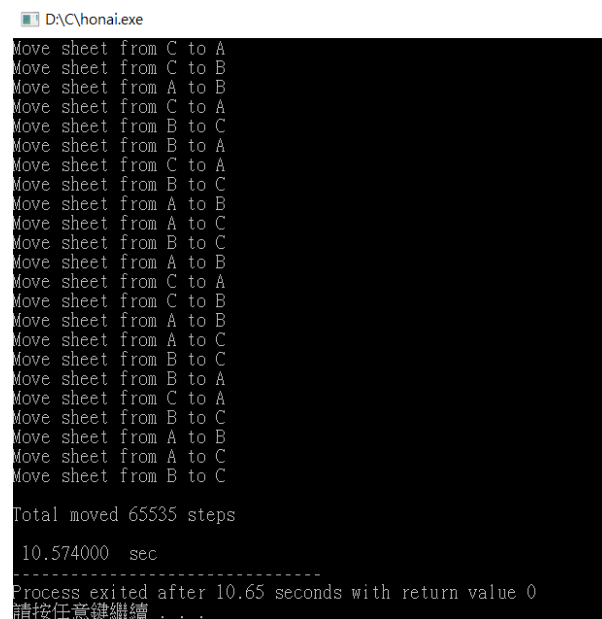
當任意  $n$  個盤子需搬移時，我們可以歸納出一套規則：

- 1.先將  $1 \sim n-1$  號盤子從 A 經由 B 搬至 C。
2. $n:A \rightarrow B$ (將  $n$  號盤子由 A 搬至 B)。
- 3.再將  $1 \sim n-1$  號盤子從 C 經由 A 搬至 B。

由前述規則可知，搬  $n$  個盤子可以分解成三大步，要利用遞迴解決河內塔問題，可以藉由上述特性重複呼叫這三大步，當  $n=1$  時為遞迴的終止條件。

程式碼及執行結果：

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 int moved=0;
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         moved++;
9     }
10    else {
11        hanoi(n-1, A, C, B);
12        hanoi(1, A, B, C);
13        hanoi(n-1, B, A, C);
14    }
15 }
16 int main() {
17     clock_t end, start;
18     start=clock();
19     int n;
20     printf("Please key-in number of sheet : ");
21     scanf("%d", &n);
22     hanoi(n, 'A', 'B', 'C');
23     printf("\nTotal moved %d steps\n\n", moved);
24     end=clock();
25     double diff = end - start; // ms
26     printf(" %f sec", diff / CLOCKS_PER_SEC );
27     return 0;
28 }
```



```
D:\C\hanoi.exe
Move sheet from C to A
Move sheet from C to B
Move sheet from A to B
Move sheet from C to A
Move sheet from B to C
Move sheet from B to A
Move sheet from C to A
Move sheet from B to C
Move sheet from A to B
Move sheet from A to C
Move sheet from B to C
Move sheet from A to B
Move sheet from C to A
Move sheet from C to B
Move sheet from A to B
Move sheet from A to C
Move sheet from B to C
Move sheet from B to A
Move sheet from C to A
Move sheet from B to C
Move sheet from A to B
Move sheet from A to C
Move sheet from B to C
Total moved 65535 steps
10.574000 sec
-----
Process exited after 10.65 seconds with return value 0
請按任意鍵繼續...
```

16 個盤共移動 65535( $2$  的  $16$  次方-1)次。

所需秒數：10.574 sec

CPU 型號：AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz