

## 河內塔(Tower of Hanoi)

關於河內塔的起源要回溯到 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智力遊戲。這個遊戲名為河內塔，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棍的木板，在其中的一根木棍上，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棍中的其中一根上。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，也就是說不論在哪一根木棍上，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棍上全部移動至另一根木棍上，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。

以數學的角度來看，河內塔不過是一個基本的遞迴題目，先假設三根木棍分別為 A、B、C，由於移動 64 個金屬片太過繁瑣，先以移動一個金屬片( $a_1$ )為開頭，而移動次數為 1 次( $A \rightarrow C$ )，若要移動兩個金屬片( $a_2$ )則需要 3 次( $A \rightarrow B, A \rightarrow C, B \rightarrow C$ )，公式為  $a_2 = 1+1+1 = 2 * a_1 + 1$ ，若要移動三個金屬片( $a_3$ )則需要 7 次( $A \rightarrow C, A \rightarrow B, C \rightarrow B, A \rightarrow C, B \rightarrow A, B \rightarrow C, A \rightarrow C$ )，公式為  $a_3 = 2 * a_2 + 1$ ，依此類推下去，即可發現每次移動金屬片的次數會是 2 乘上上次移動的次數加 1，假設移動  $n$  個金屬片則移動次數為  $2 * a_{(n-1)} + 1 = 2^{(n)} - 1$ 。

由此可見，如果想得出這次的移動次數就必須先求出上一次的移動次數，這也就是遞迴的概念，而我設計的 Hanoi 程式的概念就是，如果總金屬片超過 2 個，那將第二個以下的金屬片假裝遮起來就簡單了，每次只處理兩個金屬片，也就是： $A \rightarrow B$ 、 $A \rightarrow C$ 、 $B \rightarrow C$  這三個步驟，而被遮住的部份，其實就是進入程式的遞迴處理即可，而最後跑完整個程式所需時間為 7.0000 秒。

處理器: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz    2.81 GHz

```
1  #include <stdio.h>
2  #include <time.h>
3
4  void hanoi(int n, char A, char B, char C) {
5      if(n == 1) {
6          printf("Move disk from %c to %c\n", A, C);
7      }
8      else {
9          hanoi(n-1, A, C, B);
10         hanoi(1, A, B, C);
11         hanoi(n-1, B, A, C);
12     }
13 }
14
15 int main() {
16     int n = 16;
17     clock_t start, end;
18     start = clock();
19     hanoi(n, 'A', 'B', 'C');
20     end = clock();
21     double diff = (end - start) / CLOCKS_PER_SEC;
22     printf("%f sec\n", diff);
23     return 0;
24 }
```

命令提示字元

```
Move disk from A to B
Move disk from C to A
Move disk from B to C
Move disk from B to A
Move disk from C to A
Move disk from B to C
Move disk from A to B
Move disk from A to C
Move disk from B to C
Move disk from A to B
Move disk from C to A
Move disk from C to B
Move disk from A to B
Move disk from A to C
Move disk from B to C
Move disk from B to A
Move disk from C to A
Move disk from B to C
Move disk from A to B
Move disk from A to C
Move disk from B to C
7.000000 sec
D:\homework\cs101\hw3>
```

為了節省程式時間，所以我把輸出改成第一個字母為原木棍第二個字母為移動後的木棍，在開始測試之前我會先以  $n=16$  的秒數為基準，再去推算當搬動的金屬片個數變多時，程式所需要花多少時間才能結束，公式為

$$\text{time} = (2^n - 1) / 65535 * 3 (\text{sec}) / 60(\text{min}) / 60(\text{hr})$$

備註: 65535 為  $n=16$  的搬移次數，3 為  $n=16$  的秒數

不過這都是預估的秒數，通常實際操作秒數會更久。

第一個是  $n=16$ ，因為輸出減少所以速度明顯變快，程式圖如下：

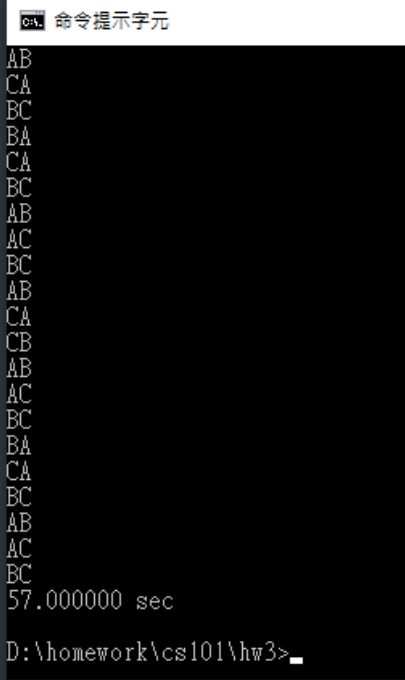
```
1  #include <stdio.h>
2  #include <time.h>
3
4  void hanoi(int n, char A, char B, char C) {
5      if(n == 1) {
6          printf("%c%c\n", A, C);
7      }
8      else {
9          hanoi(n-1, A, C, B);
10         hanoi(1, A, B, C);
11         hanoi(n-1, B, A, C);
12     }
13 }
14
15 int main() {
16     int n = 16;
17     clock_t start, end;
18     start = clock();
19     hanoi(n, 'A', 'B', 'C');
20     end = clock();
21     double diff = (end - start) / CLOCKS_PER_SEC;
22     printf("%f sec\n", diff);
23     return 0;
24 }
```

命令提示字元

AB  
CA  
BC  
BA  
CA  
BC  
AB  
AC  
BC  
AB  
CA  
CB  
AB  
AC  
BC  
BA  
CA  
BC  
AB  
AC  
BC  
3.000000 sec  
D:\homework\cs101\hw3>


第二個是  $n=20$ ，預估秒數為 48.0000 秒，程式圖如下：

```
1  #include <stdio.h>
2  #include <time.h>
3
4  void hanoi(int n, char A, char B, char C) {
5      if(n == 1) {
6          printf("%c%c\n", A, C);
7      }
8      else {
9          hanoi(n-1, A, C, B);
10         hanoi(1, A, B, C);
11         hanoi(n-1, B, A, C);
12     }
13 }
14
15 int main() {
16     int n = 20;
17     clock_t start, end;
18     start = clock();
19     hanoi(n, 'A', 'B', 'C');
20     end = clock();
21     double diff = (end - start) / CLOCKS_PER_SEC;
22     printf("%f sec\n", diff);
23     return 0;
24 }
```



第三個是  $n=27$ ，預估秒數為 6144.0000 秒，程式圖如下：

```
1  #include <stdio.h>
2  #include <time.h>
3
4  void hanoi(int n, char A, char B, char C) {
5      if(n == 1) {
6          printf("%c%c\n", A, C);
7      }
8      else {
9          hanoi(n-1, A, C, B);
10         hanoi(1, A, B, C);
11         hanoi(n-1, B, A, C);
12     }
13 }
14
15 int main() {
16     int n = 27;
17     clock_t start, end;
18     start = clock();
19     hanoi(n, 'A', 'B', 'C');
20     end = clock();
21     double diff = (end - start) / CLOCKS_PER_SEC;
22     printf("%f sec\n", diff);
23     return 0;
24 }
```



命令提示字元

BA  
CB  
AC  
AB  
CB  
AC  
BA  
BC  
AC  
BA  
CB  
CA  
BA  
BC  
AC  
AB  
CB  
AC  
BA  
BC  
AC  
AB  
CB  
AC  
BA  
BC  
AC

7783.000000 sec

D:\homework\cs101\hw3>

第四個是  $n=28$ ，預估秒數為 12288.0000 秒，程式圖如下：

```
1  #include <stdio.h>
2  #include <time.h>
3
4  void hanoi(int n, char A, char B, char C) {
5      if(n == 1) {
6          printf("%c%c\n", A, C);
7      }
8      else {
9          hanoi(n-1, A, C, B);
10         hanoi(1, A, B, C);
11         hanoi(n-1, B, A, C);
12     }
13 }
14
15 int main() {
16     int n = 28;
17     clock_t start, end;
18     start = clock();
19     hanoi(n, 'A', 'B', 'C');
20     end = clock();
21     double diff = (end - start) / CLOCKS_PER_SEC;
22     printf("%f sec\n", diff);
23     return 0;
24 }
```

命令提示字元

AB  
CA  
BC  
BA  
CA  
BC  
AB  
AC  
BC  
AB  
CA  
CB  
AB  
AC  
BC  
BA  
CA  
BC  
AB  
AC  
BC  
13376.000000 sec  
D:\homework\cs101\hw3>s