

河內塔是一個和數學邏輯相關的遊戲，大致內容為將不同大小的圓盤從一根桿子移到另一根桿子上，具體規則如下：有 A、B、C 三根桿子，A 桿上有 N 個圓盤，盤子的尺寸由上到下遞增，要按照以下兩項規則將所有圓盤從 A 桿移到 C 桿，1.每次只能移動一個圓盤 2.大盤不能疊在小盤上面。關於河內塔最有效率的解法，可以用遞迴的方式去逆推，從 A 桿移動 X 層的河內塔到 C 桿的方式，可以轉換成以下三個步驟 1.移動(X-1)層的河內塔至 B 桿 2.將第 X 個圓盤移到 C 桿 3.移動(X-1)層的河內塔至 C 桿。而移動(X-1)層的方式又可以轉換成 1.移動(X-2)層的河內塔至 C 桿 2.將第(X-1)個圓盤移到 B 桿 3.移動(X-2)層的河內塔至 B 桿。以此類推，就可以得出河內塔的所有步驟。除此之外，我用撲克牌模擬河內塔的實際情形時觀察出一些隱藏的規律(僅適用於最有效率的解法)，首先上一步移動過的不會馬上繼續移動是基本(無論是移回去或是移動到第三根桿子上)，接下來奇數項(第 1,3,5,7.....個圓盤)不會和偶數項相連，然後在符合上述條件的情況下，如果出現空的桿子，且將圓盤移動到空桿子上非必要，則優先移動到空桿子以外的地方。用撲克牌以上兩個規則的實際情況，以五層的河內塔為例，A 桿上有 1(A),2,5 共三個圓盤，B 桿上有 3,4 共兩個圓盤，此時將要移動第 13 步，C 桿上沒有圓盤，如果跳過奇數偶數項不能相連的條件，就會直接把 1 從 A 桿移到 B 桿，但按照規律來的話，1 就只能移動到 C 桿。實際操作時我會把奇數項用黑色的花色(黑桃和梅花)、偶數項用紅色的花色表示(紅心和方塊)，這樣什麼牌不能移到什麼牌上就一目瞭然了。另外我還注意到一個現象，在移動 1~3 的過程中不管往哪移動，順序始終都是 1213121，再進一步拓展，發現在這之前或之後往往會接一個 4，找到的排列情況是 2 次 1213121 中間夾著一個 4，由此發現從和移動位置不同的另外一種方面的規律，即在完成 N 層河內塔的完整過程中，第 X 個圓盤會移動「2 的(N-X)次方」次。

程式碼：

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

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

```
    if(n == 1) {  
        printf("Move sheet from %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;  
    printf("請輸入盤數：");  
    scanf("%d", &n);  
    hanoi(n, 'A', 'B', 'C');  
    return 0;  
}
```

所需時間：0 秒

CPU：AMD Ryzen 9 5900HX with Radeon Graphics