在之前法國數學家，愛德華·盧卡斯提出一個問題，這個問題大概是「有三根柱子，原先有 n 個圓盤套在同一根柱子，圓盤依大小由下而上，越上層則越小。如果完成移動 n 個圓盤套在其他的同一根柱子，需要移動的次數是多少那這個問題其實是可以透過簡單的實做完成的，但是盧卡斯開始著手研究如何用數學的方式計算解出此問題，從。從題意來剖析，要將河內塔移動到第三根柱子且跟原本有小到大往下排，就必須考慮著前後的移動次序，而且此問題隱含著需要最小移動次數的限制，要在最少的步數完成問題。從問題可得知我們是不可能多動一次等其他盤子，所以解決想法即是將盤子拿出實放置到對的位子，之後再一一回收就好。簡單來說，我的最後放置第 n 個盤子，不能卡到第 n-1 個盤子的放置，導致要延遲多動一步。從這樣的題目了解，我們知道需要前後關係的對應運算，因此我們將遞迴與此問題以及計算做連結。若我要擺好第 n 個，我必須先將 n-1 個擺置最底下，代表著會先從小的開始擺，然後再將大(n-1)的放置另一根柱子，再將放在其之上。

所以從三層河內塔來看，假設由上而下小到大排列，分別為 A、B、C 盤，我將 A 移至第三根柱子，留下第二根放 B 盤，之後在 B 盤上放上 A 盤，最後會留下第三根柱子的空位。此時我能使用最小的步數，不卡住其他盤子的情況下排完全部的盤子，在此發現一個規律，就是將第 n 個的盤子先放置第三根柱子，然後再將第 n-1 個的盤子，放置第二根柱子；之後再將此模式調整，將第三根柱子已放的盤子放置第二個，將最後一個盤子放置第三根柱子，此放置情形的步驟雛形已出現，分別為模式 1[將第 n 個的盤子先放置第三根柱子，然後再將第 n-1 個的盤子，放置第二根柱子]、模式二[第三根柱子已放的盤子放置第二個，將最後一個盤子放置第三根柱子]。之後發現若要考慮 n 層的變動，就必須在第一次移動模式後考慮(模式 1)，這樣才會是合理的考量，雖然在網路上可以透過數學歸納法得知層數的對應，但這也能用推理推出，如下列的程式碼形式。此外，數學家也間接計算出最佳步數為 2^N – 1 步，所以可由此關係驗證程式碼運作是否為正確。

最終上述做法可以用 recursion 函式寫出，如下列程式碼，我們可以在第一個動作呼叫自己，去考慮上一層要做甚麼，最後做到一時，在考慮原本層數的收尾情況。

程式碼

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>



int main() {



    clock_t start, end;



    start = clock();



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

    if(n == 1) {

    }

    else {

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

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

```c
    hanoi(n-1, B, A, C);
        }
    }
    int n;
    printf("請輸入盤數：");
    scanf("%d", &n);
    hanoi(n, 'A', 'B', 'C');

    end = clock();

    double diff = end - start;
    printf(" %f　秒", diff / CLOCKS_PER_SEC );
    return 0;
}
```

電腦反應所需時間 0.022000　秒

下為截圖

電腦 CPU　11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz　2.69 GHz vs2 > C main.c > scanf

```c
int main() {

    clock_t start, end;

    start = clock();

    void hanoi(int n, char A, char B, char C) {
    if(n == 1) {
    }
    else {
    hanoi(n-1, A, C, B);
    hanoi(1, A, B, C);
    hanoi(n-1, B, A, C);
        }
    }
    int n;
    printf("請輸入盤數：");
    scanf("%d", &n);
    hanoi(n, 'A', 'B', 'C');

    end = clock();

    double diff = end - start;
    printf(" %f  秒", diff / CLOCKS_PER_SEC );
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\88696\Desktop\vs2> cd "c:\Users\88696\Desktop\vs2\" ; if ($?) { gcc main.c -o main } ; if ($?) { .\main }
請輸入盤數：16
 1.001000  秒
PS C:\Users\88696\Desktop\vs2> cd "c:\Users\88696\Desktop\vs2\" ; if ($?) { gcc main.c -o main } ; if ($?) { .\main }
請輸入盤數：16
 0.805000  秒
PS C:\Users\88696\Desktop\vs2> cd "c:\Users\88696\Desktop\vs2\" ; if ($?) { gcc main.c -o main } ; if ($?) { .\main }
請輸入盤數：16
 0.022000  秒
```

處理器 (CPU)

11th Gen Intel(R) Core(TM)

i5-11400H @ 2.70GHz