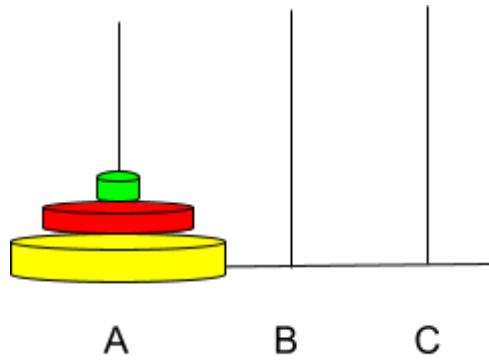


河內塔為一個古老的智力遊戲，起源於古印度神廟的故事，主要的內容是寺廟中間放置一塊上面插有三根木釘的木板，而其中一根木釘上，由上到下，由小到大，放置64片圓環金屬片，而神指示僧侶要將64片的金屬片全部移至另一根，但條件是每次移動時只能搬動一片金屬片，並且還是要遵守金屬片半徑較大的在最下面，半徑小的在最上面，直到僧侶將64片金屬片按照規定全部移至另一根木釘上，那麼，世界末日就隨即到來，而現在河內塔也是按照這個規則來操作。以下簡略介紹河內塔遊戲：一樣也是一塊板子上有三根柱子，假設今天需要搬三個塔，我們先將這三個塔按照半徑大小，由下往上全部放置在A柱子上，並且將全部塔搬到C柱子上，最快的方法是，先將綠色搬到C柱子，再將紅色塔搬到B柱子，綠色在搬回B柱子，將黃色搬到C柱子，綠色搬回A柱子，將紅色搬



到B柱子，最後將綠色搬到C柱子，這樣就完成搬塔了，雖然這看起來有點複雜，但解法相當簡單。按照這個解法，以最少次數的搬動河內塔，會發現移動盤子得順序是有規律的，先設定起始柱和目標柱，而剩下的

就是暫存柱，假設盤數為n，先把n-1個盤全部移到暫存柱，再把第n個盤一道目標柱，最後將放在暫存柱的所有盤全部移到目標柱，會發現一件很有趣的事，n層盤數的最小移動次數是 $2^n - 1$ ，所以跟recursion有關係。像上面舉的移動三個塔的例子一樣，需要移動七次才能完成，不管用幾個盤，最後得出來得遞迴公式為 $2^n - 1$ 。下圖為遞迴公式的解釋：

$$\begin{aligned}
 a_1 &= 1 \\
 a_2 &= 2 \times a_1 + 1 \\
 a_3 &= 2 \times a_2 + 1 = 2 \times (2 \times a_1 + 1) + 1 = (2^2 a_1 + 2) + 1 \\
 a_4 &= 2 \times a_3 + 1 = 2 \times ((2^2 a_1 + 2) + 1) + 1 = (2^3 a_1 + 2^2 a_1 + 2) + 1 \\
 &\vdots
 \end{aligned}$$

$$\begin{aligned}
 a_n &= 2 \times a_{n-1} + 1 = (2^{n-1} a_1 + 2^{n-2} a_1 + \dots + 2^3 a_1 + 2^2 a_1 + 2) + 1 = 2^{n-1} a_1 + 2^{n-2} a_1 + \dots + 2^3 a_1 + 2^2 a_1 + 2^1 + 2^0 \\
 &= 2^{n-1} + 2^{n-2} + \dots + 2^3 + 2^2 + 2^1 + 2^0 \\
 &= 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{n-2} + 2^{n-1} \\
 &= \frac{2^n - 1}{2 - 1} = 2^n - 1
 \end{aligned}$$

假設起始為 n 個圓盤在同一根柱子，越上層的圓盤小，就得將原先第二層到 n 層的盤搬到另一根柱子，算起來總共需要搬動 2^{n-1} 次，之後將原先的第一層盤移到目標柱，最後將暫存柱的盤全部移到目標柱上，總共需要移動 2^{n-1} 次。完整過程為：

$(2^{n-1} + 1) + (2^{n-1}) = 2^n - 1$ 次。

下圖為河內塔的 code：

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int moved=0;

void hanoi(int n, char A, char B, char C) {
    if(n == 1) {
        printf("Move sheet from %c to %c\n", A, C);
        moved++;
    }
}

```

```

else {

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

    hanoi(1, A, B, C);

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

}

}

int main() {

    int n;

    clock_t begin,end;

    double cost;

    begin =clock();

    printf("Please key-in number of sheet:");

    scanf("%d", &n);

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

    printf("\nTotal moved %d steps\n\n",moved);

    end=clock();

    cost = (double)(end - begin)/CLOCKS_PER_SEC;

    printf("constant CLOCKS_PER_SEC is: %ld, time coast is %lf

secs",CLOCKS_PER_SEC,cost);
}

```

```
return 0;  
}  
  
}
```

```
1 #include <stdio.h>  
2 #include <stdlib.h>  
3 #include <time.h>  
4 int moved=0;  
5  
6 void hanoi(int n, char A, char B, char C) {  
7     if(n == 1) {  
8         printf("Move sheet from %c to %c\n", A, C);  
9         moved++;  
10    }  
11    else {  
12        hanoi(n-1, A, C, B);  
13        hanoi(1, A, B, C);  
14        hanoi(n-1, B, A, C);  
15    }  
16}  
17  
18 int main() {  
19     int n;  
20     clock_t begin,end;  
21  
22     printf("Please key-in number of sheet : ");  
23     scanf("%d", &n);  
24     double cost;  
25     begin =clock();  
26     hanoi(n, 'A', 'B', 'C');  
27     printf("\nTotal moved %d steps\n\n",moved);  
28     end=clock();  
29     cost = (double)(end - begin)/CLOCKS_PER_SEC;  
30     printf("constant CLOCKS_PER_SEC is: %ld, time coast is %lf secs",CLOCKS_PER_SEC,cost);  
31  
32     return 0;  
33 }
```

```
constant CLOCKS_PER_SEC is: 1000000, time coast is 0.005110 secs
```

經過程式計算後，16個盤所需要的時間為0.00511secs，此台電腦的cpu為AMD Ryzen

5 4500U with Radeon Graphics 2.38 GHz。