

河內塔問題：河內塔是一座可以分為好多層的塔，而每層的大小都不相同，而搬動一層的條件為那層上方無其他層，且上層必須比下層小，而可以使用的空間只有三個空間，而河內塔問題為一座完整的河內塔(其他兩個空間皆是空的)最少需要幾步去讓這座河內塔完整的移到例外一個空間。而河內塔的由來是傳說中越南河內有一間寺廟裏面有三根柱子上面有64個金盤，當64個金盤依照上述規則移動完後世界就會毀滅。

recursion的功用在我看來是將一個大問題分成很多個相似或著相同的小問題，並且一直將分下去直到有辦法解為止所以我需要將搬動一整個河內塔簡化成一個簡單一點的問題，我要般的層數為16層，那我是不是可以先將問題變成搬動15層需要多少步，再加上搬動第16層需要多少步，在把搬動15層化簡成搬動14層以此類推下去直到變成搬動1層需要幾步。

```
#include<stdio.h>

#include<time.h>

int Hanoi(int n){

    if (n == 1){

        return 1;

    }else{

        return 2*Hanoi(n-1)+1;

    }

}
```

```

int main(){

    int ti = time(NULL);

    printf("%d\n",Hanoi(16));

    printf("%d",time(NULL)-ti);

    return 0;

}

```

上面是我所寫的Tower of Hanoi跑16層的程式碼使用的CPU為Intel i7-8550U跑的結果為65535以及0秒，0秒的原因純粹是跑得太快了，因為16層的河內塔重複執行的程式只有16次並且這16次都只需要跑兩行if跟return，這對電腦而言是一瞬間就跑完的。而我程式碼寫的想法是，先解設完整16層的河內塔再A柱上而B柱跟C柱是空的，並且移動第n層的到例外一根柱子上所需的步驟數為 a_n ，如果我要移動完整的河內塔到B柱上，那我就需要先將上面總計15層移動到C柱然後把第16層移到B柱最後再把C柱上的15層都移到B柱上，而這個步驟需要移動 $1+2*a_{15}$ (1為把第16層移到B住，而 $2*a_{15}$ 為兩次移動總計15層所需要的步驟)而移動整個河內塔的步驟也可以看做移動第16層所需要的步驟，所以移動第15層所需的步數就等於 $1+2*a_{14}$ ，以此類推最後就會到 $a_2 = 1 + 2*a_1$ ，而 a_1 就等於1，而這中間的過程就是使用 recursion解決。而從上面的想法可以得知其實要移動n層的河內塔只需 $2^n - 1$ 次就可以移動了，這樣就不需要使用recursion來解決了。其實我所寫的方法算是由後往前推，如果能理解上方的想法其實可以把算式換個方向想把 $a_n = 1 + 2*a_{n-1}$ 想成 $a_{n+1} = 1 + 2*a_n$ 這樣就可以換成從 $n = 1$ 去算直到 $n = 16$ 這算是兩種不同的做法，在河內塔上看似沒有區別但到了其他使用recursion去解決的問題的時候就有可能出現差異。