

HW3 : 河內塔

B0944109蔡尚融

河內塔起源與規則：

由法國數學家愛德華·盧卡斯發明, 構造簡單，僅由三根鐵棒 (起點、終點、暫存) 與數個由大而小向上堆疊的鐵盤構成。其主要目標為在一次只能搬動一個鐵盤、大鐵盤不可堆疊於小鐵盤之上的兩個規則下，用最少的步數將一疊鐵盤從初始鐵柱移動至目的鐵柱，雖然看似簡單，但一旦將鐵盤數提升，過程將變得極其繁瑣。

河內塔和 recursion 有什麼關係

recursion又稱遞迴，在程式中意指函式透過反覆呼叫自己來解決問題的結構，針對重複性的問題，透過遞迴能夠有效將一個複雜的問題進行簡化。

怎麼用用recursion 解決此問題

河內塔問題可以拆分成3個核心步驟：1. 將擋住最大鐵盤的鐵盤移到暫存柱、2.把要移動的鐵盤移至目標柱、3.把暫存柱的鐵盤移至目標柱。我們可以實作一個主要有一個判斷與三個步驟的函式move，其中的參數為(需移動總盤數D, 起始柱, 暫存柱, 目的柱)，判斷式判斷是否有盤子擋路。用來代表將鐵盤由起始柱移動至目標的過程。其中的三核心步驟就可以簡化為:

1.判斷是否有盤子擋路，若有：(若無則執行第3步驟:直接移動)

2. 將擋住最大鐵盤的鐵盤移到暫存柱 -> move(D-1, A, C, B);

3.把要移動的鐵盤移至目標柱 -> move(1, A, B, C);

4.把暫存柱的鐵盤移至目標柱 -> move(D-1, B, A, C);

(其中D-1指除了最大鐵盤外的其他盤)

(另外遞迴的停止點為D=1，代表沒東西擋路)

CPU型號: i7 8700

Process exited after 2.73 seconds

實際運行時間為2.73秒

程式碼:

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

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
void move(int D, char A, char B, char C){
```

```
if(D == 1){                #如果沒有鐵盤擋路則直接移動
```

```
    printf("move disk form %c to %c\n", A, C);
```

```
}else{
```

```
    move(D-1, A, C, B);    #將擋住最大鐵盤的鐵盤移到暫存柱
```

```
    move(1, A, B, C);      #把要移動的鐵盤移至目標柱
```

```
    move(D-1, B, A, C);    #把暫存柱的鐵盤移至目標柱
```

```
}
```

```
}
```

```
int main()
```

```
{
```

```
    char T[30];
```

```
    time_t cur;
```

```
    time(&cur);
```

```
    strftime(T, 100, "%H : %M : %S", localtime(&cur));
```

```
    printf("%s\n", T);
```

```
    move(16, 'A', 'B', 'C');
```

```
    strftime(T, 100, "%H : %M : %S", localtime(&cur));
```

```
    printf("%s\n", T);
```

```
    return 0;
```

```
}                (參考 : https://openhome.cc/zh-tw/algorithm/basics/hanoi-tower/)
```