

國立清華大學  
電機工程學系

資料結構 Homework1

學號：103061223

姓名：李俊穎(Lee Junying)

My answer :

- (a) I use recursive
- (b) Move 31 times from tower 1 to 3  
when disks are 5
- (c) CPU time is 0.921sec when  $n=10$

## All of my source codes and my solution(想法)：

此次的作業河內塔是個很知名的數學問題，它的核心概念其實也就是遞迴函式的概念。在我呼叫 `function` 時，我把圓盤數量 `n`、以及代表塔的數字傳進來 (`a=1`，`b=2`，`c=3`)，雖然其龐大又複雜的計算很難處哩，但首先我們只要思考，當只有一個圓環時，可以直接將圓盤從 `tower 1` 移到 `3`，。當圓盤增加到 `n` 個的時候，其實我們可以把問題想成：把 `(n-1)` 個圓環透過 `tower 3` 移動到 `tower 2`，之後再將 `tower 1` 中最後一個圓環移動至 `tower 3`，再將 `tower 2` 中的 `(n-1)` 個圓盤透過 `tower 1` 移動到 `tower 3`，這樣一來就可以將 `n` 個圓盤從 `tower 1` 移動到 `tower 3`，問題也就解決了。

具體的寫法如下：

一開始有 `n` 個圓盤時，會執行 `function` 中 `else` 的部分，此時我就再次呼叫這個 `function`，這時候 `n` 就會一直減，減到 `n=1` 的時候(也就是最上層的圓盤)，然後就會將其從 `tower 1` 移動到 `tower 2`(因為我將 `b=2` 的直傳入當成 `c` 的值)，於是，`function move(1,a,c,b)` 做完了，就往回到上一層(`n=2` 的時候)繼續做，此時 `times` 已經=1，接下來將 `n=2` 移動到 `tower 3`，`times=2`，此時再呼叫一次 `move function`，並且傳入 `n-1(n=2)`，傳入後 `n=1`，將圓盤 1 從 `tower a` 移動到 `tower c`)，就會將圓盤 1 從 `tower 2` 移動到 `tower 3`(因為我將 `b=2` 的直傳入給 `a`，`c` 仍傳入 `c`)，這時就做完 `n=2` 的部分了，跳回 `n=3` 的部分由 `move(n-1, a, c, b)`繼續往下執行，這是由只有 2 個圓盤時推想，推展到 `n` 個也是同樣道理。

以下是我的程式碼：

```
void move(int n,int a,int b,int c)
{
    static int times=0;
    if(n==1)
    {
        times++;
        cout<<times<<" times: Move 1 from tower "<<a<<" to "<<c<<endl;
    }
    else
    {
        move(n-1,a,c,b);
        times++;
        cout<<times<<" times: Move "<<n<<" from tower "<<a<<" to "<<c<<endl;
        move(n-1,b,a,c);
    }
}
```

**The execution results of my program :**

此為 n=5 時的移動順序結果：

Enter the plates you want to play Towers of Hanoi:

```
5
1 times: Move 1 from tower 1 to 3
2 times: Move 2 from tower 1 to 2
3 times: Move 1 from tower 3 to 2
4 times: Move 3 from tower 1 to 3
5 times: Move 1 from tower 2 to 1
6 times: Move 2 from tower 2 to 3
7 times: Move 1 from tower 1 to 3
8 times: Move 4 from tower 1 to 2
9 times: Move 1 from tower 3 to 2
10 times: Move 2 from tower 3 to 1
11 times: Move 1 from tower 2 to 1
12 times: Move 3 from tower 3 to 2
13 times: Move 1 from tower 1 to 3
14 times: Move 2 from tower 1 to 2
15 times: Move 1 from tower 3 to 2
16 times: Move 5 from tower 1 to 3
17 times: Move 1 from tower 2 to 1
18 times: Move 2 from tower 2 to 3
19 times: Move 1 from tower 1 to 3
20 times: Move 3 from tower 2 to 1
21 times: Move 1 from tower 3 to 2
22 times: Move 2 from tower 3 to 1
23 times: Move 1 from tower 2 to 1
24 times: Move 4 from tower 2 to 3
25 times: Move 1 from tower 1 to 3
26 times: Move 2 from tower 1 to 2
27 times: Move 1 from tower 3 to 2
28 times: Move 3 from tower 1 to 3
29 times: Move 1 from tower 2 to 1
30 times: Move 2 from tower 2 to 3
31 times: Move 1 from tower 1 to 3
```

usetime= 0.232sec.

**CPU Time 花了 0.232 秒**

此為  $n=10$  的執行結果，需要 1023 次，花了 0.921 秒

```
1004 times: Move 3 from tower 3 to 1
1005 times: Move 1 from tower 2 to 3
1006 times: Move 2 from tower 2 to 1
1007 times: Move 1 from tower 3 to 1
1008 times: Move 5 from tower 2 to 3
1009 times: Move 1 from tower 1 to 2
1010 times: Move 2 from tower 1 to 3
1011 times: Move 1 from tower 2 to 3
1012 times: Move 3 from tower 1 to 2
1013 times: Move 1 from tower 3 to 1
1014 times: Move 2 from tower 3 to 2
1015 times: Move 1 from tower 1 to 2
1016 times: Move 4 from tower 1 to 3
1017 times: Move 1 from tower 2 to 3
1018 times: Move 2 from tower 2 to 1
1019 times: Move 1 from tower 3 to 1
1020 times: Move 3 from tower 2 to 3
1021 times: Move 1 from tower 1 to 2
1022 times: Move 2 from tower 1 to 3
1023 times: Move 1 from tower 2 to 3

usetime= 0.921sec.
Press <RETURN> to close this window...
```