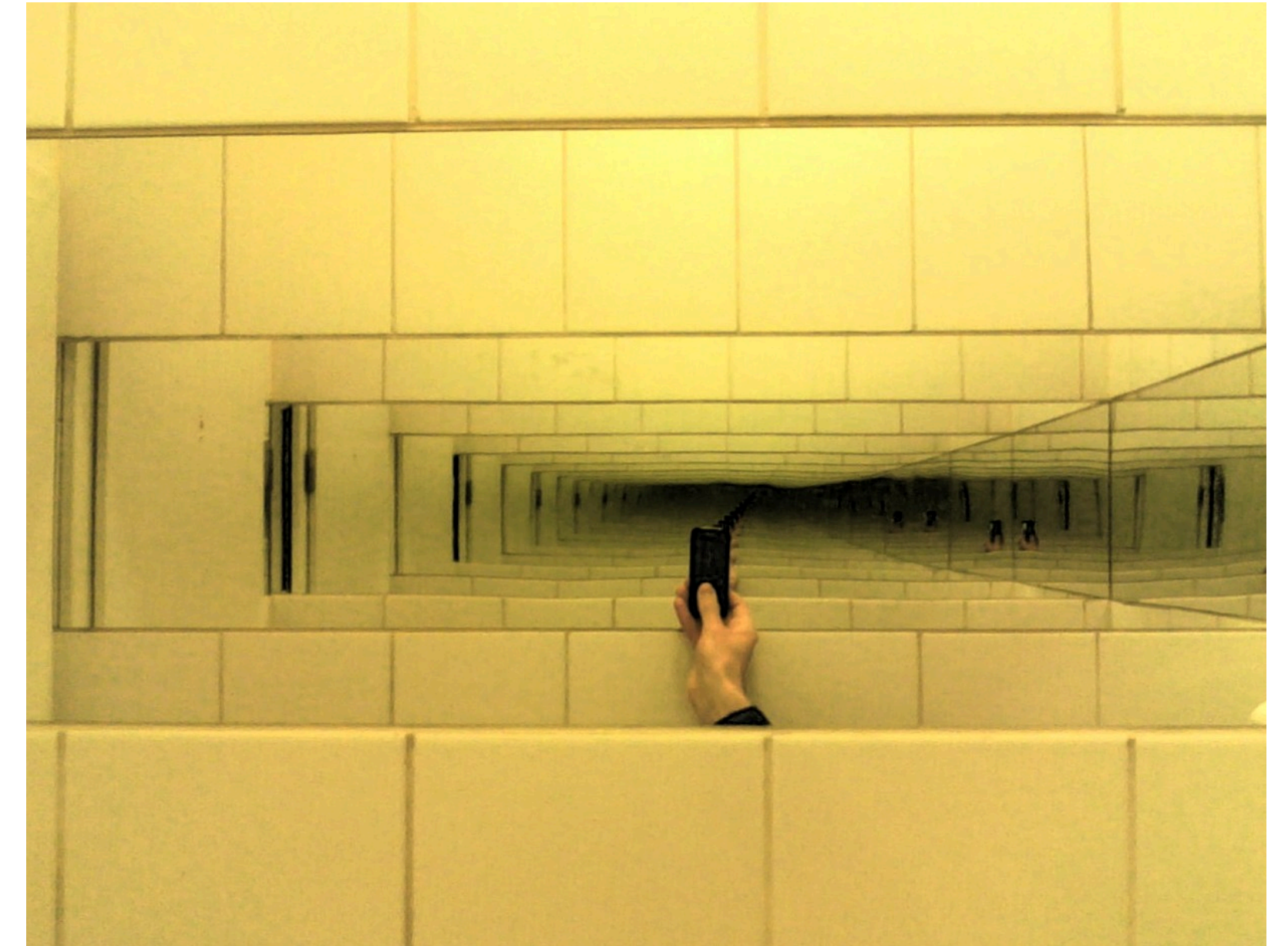


Recursion 遞迴

What is Recursion?

Something is defined by itself

- 例子：平面鏡互相反射
- 例子：Linux 是 Linux Is Not UniX 的簡寫
- 例子：從前有座山，山上有座廟，廟裡有個老和尚，還有一個小和尚。有一天小和尚對老和尚說：「給我講個故事吧，老和尚就說：從前有座山，山上有座廟，廟裡有個老和尚，還有一個小和尚。有一天小和尚對老和尚說：「給我講個故事吧，老和尚就說：從前有座山，山上有座廟，廟裡有個老和尚，還有一個小和尚。有一天小和尚對老和尚說：「給我講個故事吧，老和尚就說……



Recursion in Math

The definition of Fibonacci Sequence and Factorial

- 費氏數列（費波那契數列）的定義：
$$F(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & n \geq 2 \end{cases}$$
- 階乘的定義：
$$n! = \begin{cases} 1 & , \text{ if } n = 1 \\ n * (n-1)! & , \text{ if } n > 1 \end{cases}$$
- 自然數的定義：如果某個數 x 是自然數，那 $x + 1$ 也是自然數；1 是自然數。

Recursion in C++

How To Calculate Factorial?

- We define $\text{factorial}(n) = n * \text{factorial}(n - 1)$, $\text{factorial}(0) = 1$
- So that,
 - $f(0) = 1$
 - $f(1) = 1 * f(0) = 1 * 1 = 1$
 - $f(2) = 2 * f(1) = 2 * 1 = 2$
 - $f(3) = 3 * f(2) = 3 * 2 = 6$
 - $f(4) = 4 * f(3) = 4 * 3 = 12$

```
int factorial(int n){  
    if(n == 0) return 1;  
    return n * factorial(n - 1);  
}
```

Recursion in C++

How To Calculate Sum from 1 to N?

- We define $\text{sum}(n) = n + \text{sum}(n - 1)$, $\text{sum}(0) = 0$
- So that,
 - $f(0) = 0$
 - $f(1) = 1 + f(0) = 1 + 0 = 1$
 - $f(2) = 2 + f(1) = 2 + 1 = 3$
 - $f(3) = 3 + f(2) = 3 + 3 = 6$
 - $f(4) = 4 + f(3) = 4 + 6 = 10$

```
10 int sum(int n){
11     if(n == 0) return 0;
12     return n + sum(n - 1);
13 }
```

Recursion in C++

How To Calculate Fibonacci sequence?

- We define $\text{Fib}(n) = \text{Fib}(n - 1) + \text{Fib}(n - 2)$, $\text{Fib}(0) = 0$, $\text{Fib}(1) = 1$

- So that,

```
15 int fib(int n){
16     if(n <= 1) return n;
17     return fib(n - 1) + fib(n - 2);
18 }
```

- $\text{Fib}(0) = 0$

- $\text{Fib}(1) = 1$

- $\text{Fib}(2) = \text{Fib}(1) + \text{Fib}(0) = 1 + 0 = 1$

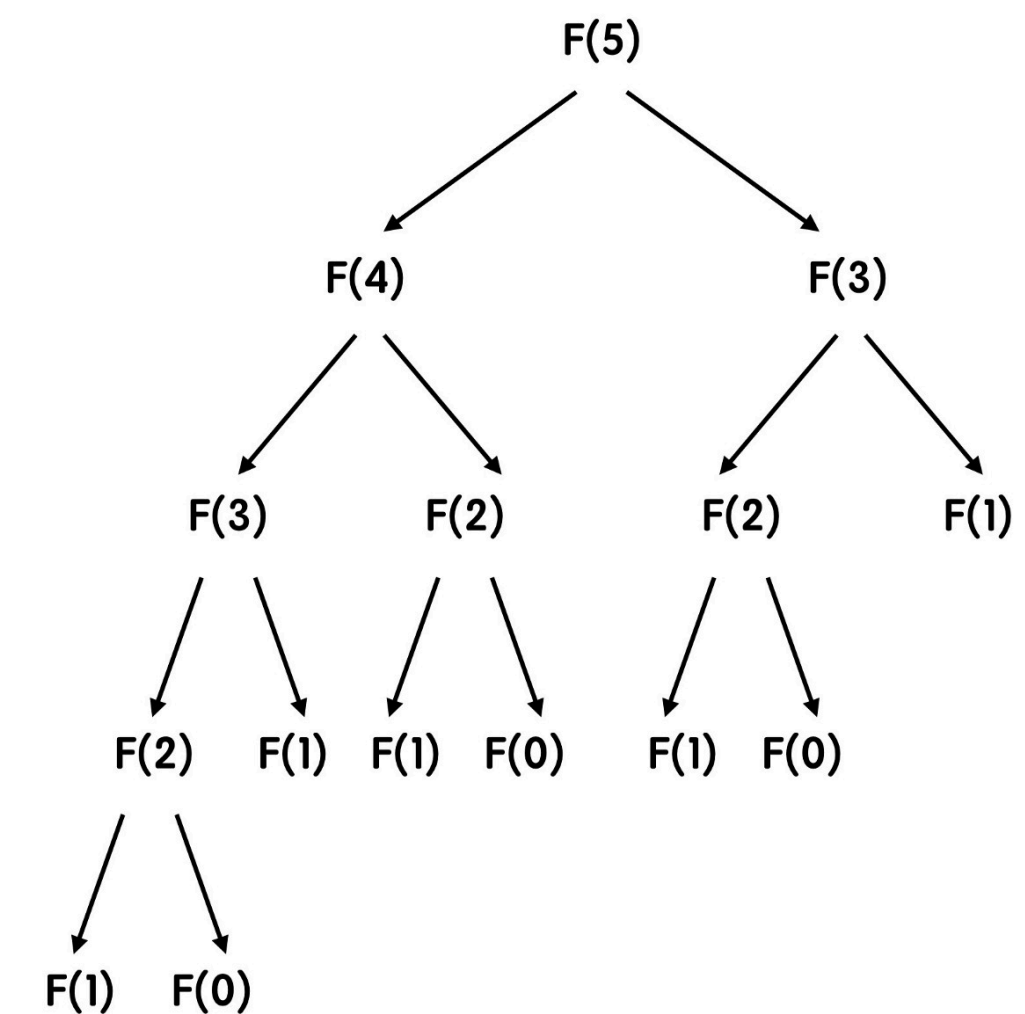
- $\text{Fib}(3) = \text{Fib}(2) + \text{Fib}(1) = 1 + 1 = 2$

- $\text{Fib}(4) = \text{Fib}(3) + \text{Fib}(2) = 2 + 1 = 3$

Practice

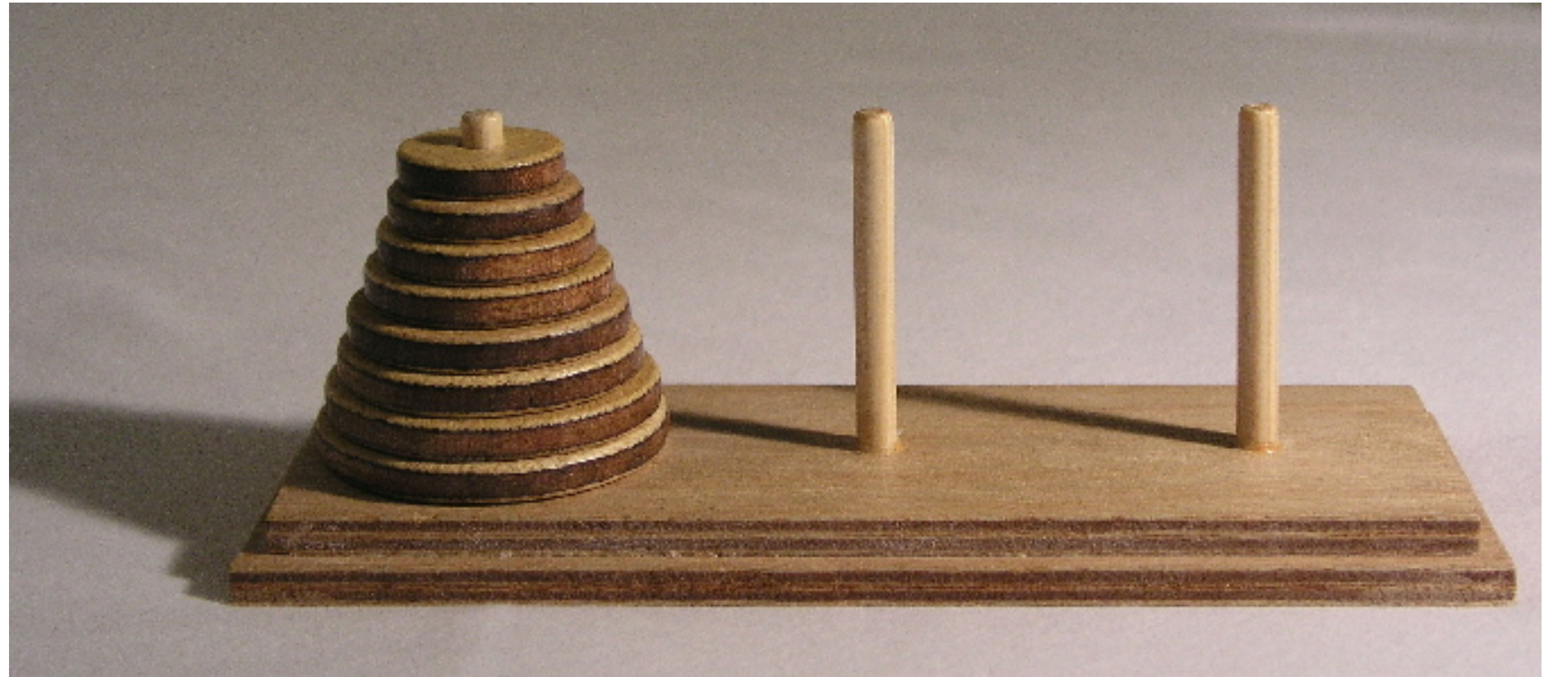
How can we use Recursion to solve problem? When to use?

- 當一個問題可以拆解成數個相似子問題的時候
 - 例如費波那契數，當你想要計算 $\text{Fib}(5)$ ，你需要 $\text{Fib}(4)$ 和 $\text{Fib}(3)$
 - 問題和問題具有相似行為



Recursion Instances

Hanoi Tower



有三根杆子A，B，C。A杆上有 N 個 ($N > 1$) 穿孔圓盤，盤的尺寸由下到上依次變小。要求按下列規則將所有圓盤移至 C 杆：

1. 每次只能移動一個圓盤；
2. 大盤不能疊在小盤上面。

提示：可將圓盤臨時置於 B 杆，也可將從 A 杆移出的圓盤重新移回 A 杆，但都必須遵循上述兩條規則。

問：如何移？最少要移動多少次？

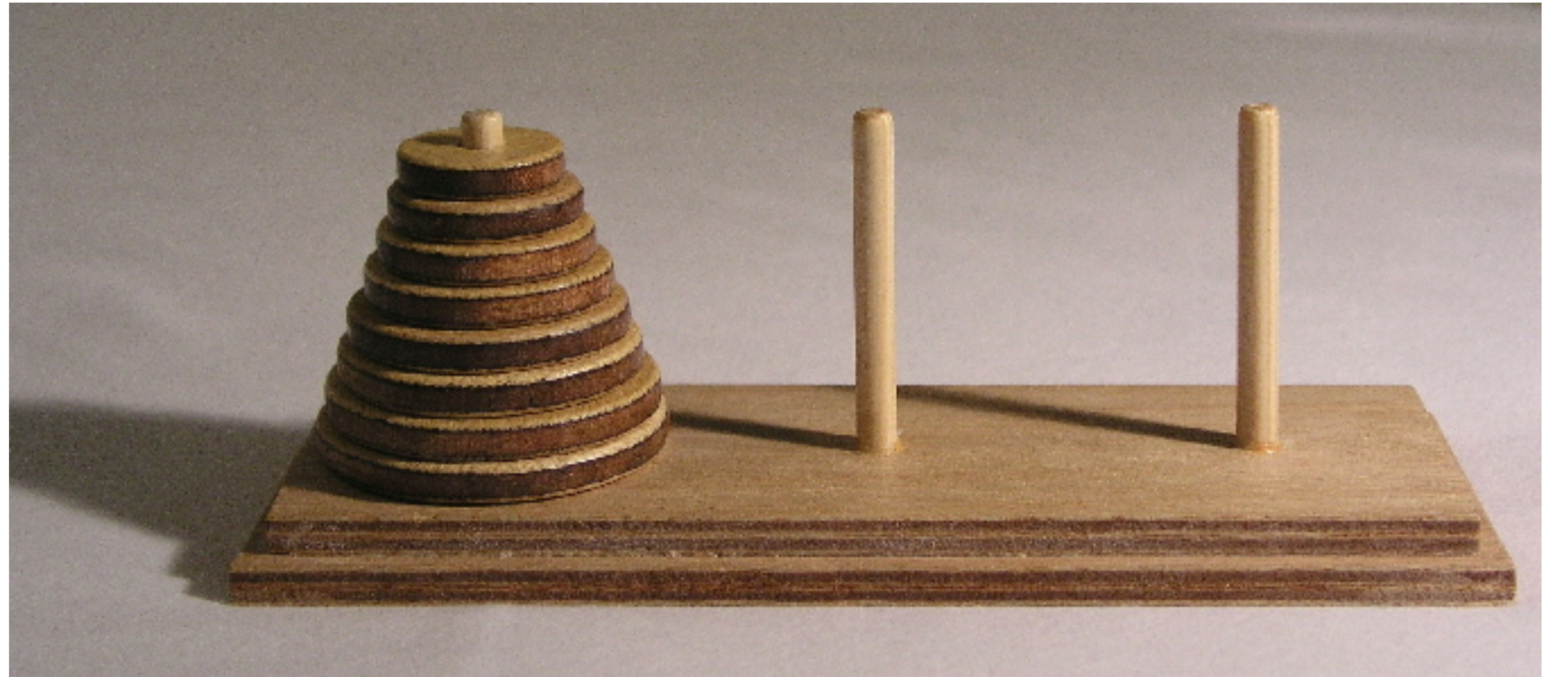
Recursion Instances

Hanoi Tower

要移動 n 個盤子，步驟很簡單：

1. 把 A 上面 $n - 1$ 個盤子從 A 移到 B
2. 把 A 最下面 1 個盤子從 A 移到 C
3. 把 B 的 $n - 1$ 個盤子從 B 移到 C

有沒有發現遞迴的想法？要移動 n 個盤子，必須先移動 $n - 1$ 個盤子！



Recursion Instances

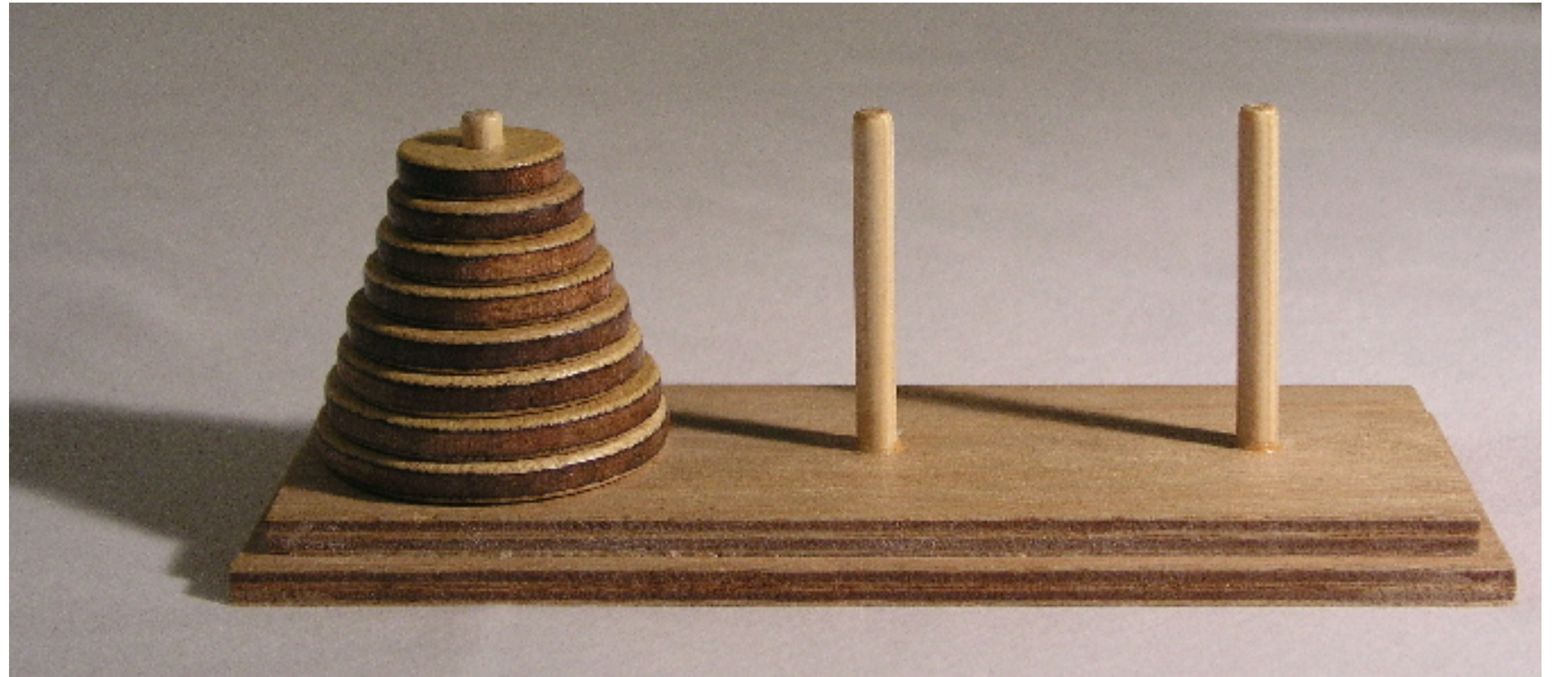
Hanoi Tower

遞迴式：

要移動 (N 個盤子，從 A，經由 B，移動到 C)

等價於三個步驟

1. 要移動 (N - 1 個盤子，從 A，經由 C，移動到 B)
2. 要移動 (1 個盤子，從 A，經由 B，移動到 C)
3. 要移動 (N - 1 個盤子，從 B，經由 A，移動到 C)



Recursion Instances

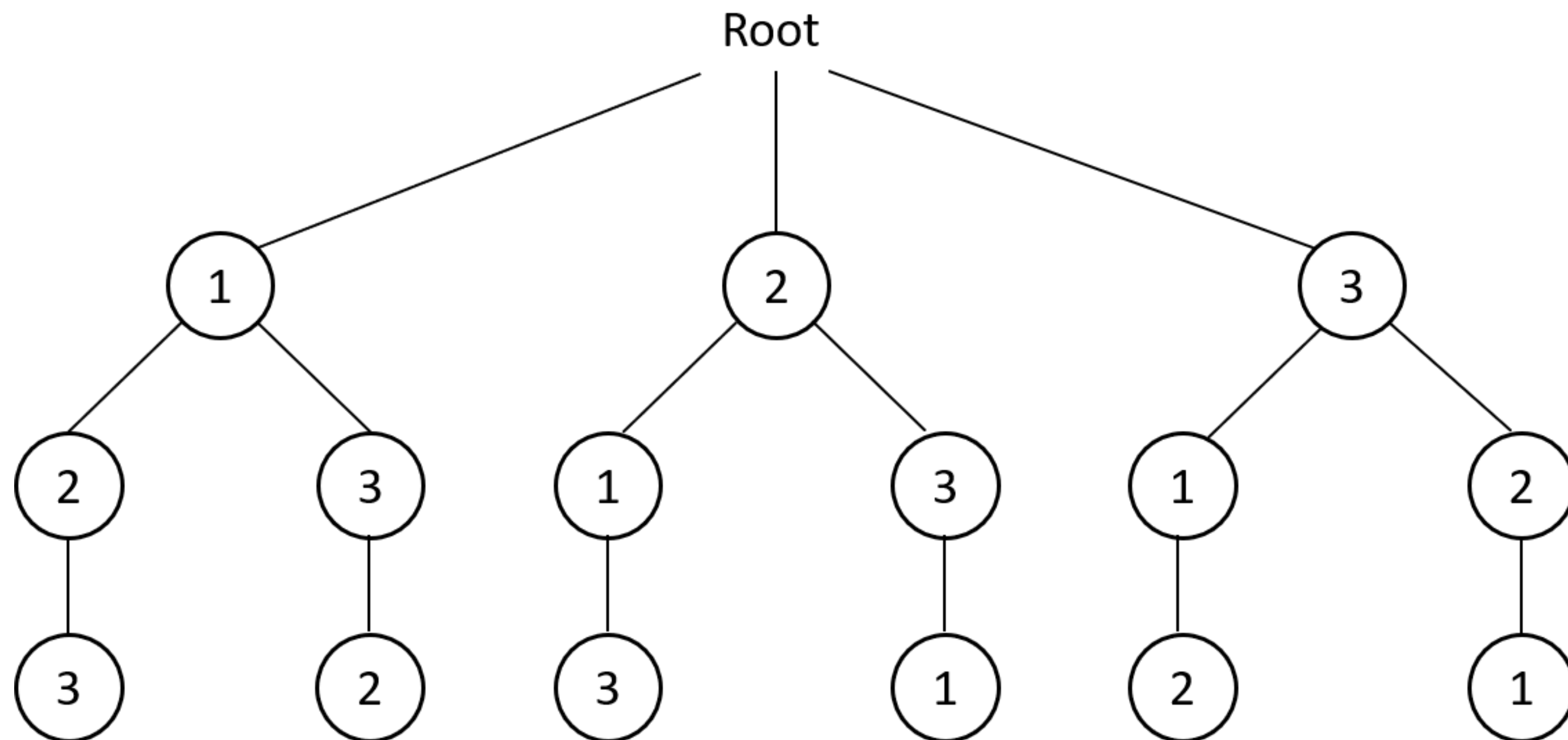
Hanoi Tower

```
1 #include<iostream>
2
3 using namespace std;
4
5 void hanoi(int n, char a, char b, char c){
6     cout << "Moving " << n << " Disc from " << a << " to " << c << " via " << b << endl;
7     if(n == 1) return; // no need to be recursive, just move it
8     hanoi(n - 1, a, c, b);
9     hanoi(1, a, b, c);
10    hanoi(n - 1, b, a, b);
11 }
12
13 int main(){
14     hanoi(4, 'A', 'B', 'C');
15 }
```

Recursion Instances

Permutation

- 要怎麼列出所有的 permutation ?



```
1 #include<iostream>
2
3 using namespace std;
4
5 const int N = 4;
6 int used[N] = {};
7 int now[N];
8
9 void permutation(int length){
10     if(length == N) { // use all numbers
11         for(int i = 0; i < N; i++)
12             cout << now[i] << " ";
13         cout << endl;
14         return;
15     }
16     for(int i = 0; i < N; i++){
17         if(used[i] == 0) { // if never us
18             used[i] = 1; // mark it use
19             now[length] = i; // record
20             permutation(length + 1); // do choose
21             used[i] = 0; // unmark, bec
22         }
23     }
24 }
25
26 int main(){
27     permutation(0);
28 }
```

Recursion Instances

Combination

- 要怎麼列出所有的 Combination ?
 - Harder than permutation -> 可能有重複 ?
 - 思考一下之前枚舉 pair 的時候是怎麼剔除重複的 (i, j) 和 (j, i)

Recursion Instances

N Queens Problems

- 如何能夠在 $N * N$ 的西洋棋棋盤上放置 N 個皇后，使得任何一個皇后都無法直接吃掉其他的皇后？有幾種擺法？
 - 皇后可以橫走、直走、斜走

