Ch6: Recursion

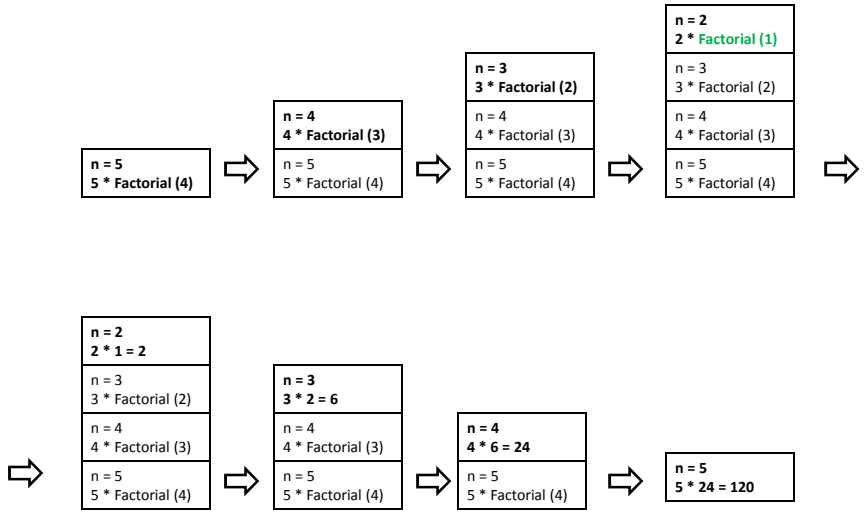
- Recursion is the process of reducing the problem into smaller versions of itself.
- For a given problem to be solved recursively
  - Every recursive definition must have one (or more) base case.
  - The general case must eventually be reduced to a base case.
  - The base case stops the recursion.

Examples: Factorial

$$n! = \begin{cases} n & n = 1 \\ n * (n-1)! & n > 1 \end{cases}$$

```
int Factorial (int n)
{
    if (n == 1)    // ground case
        return n;
    else           // recursive case
        return n * Factorial (n - 1);
}
```

Factorial (5)



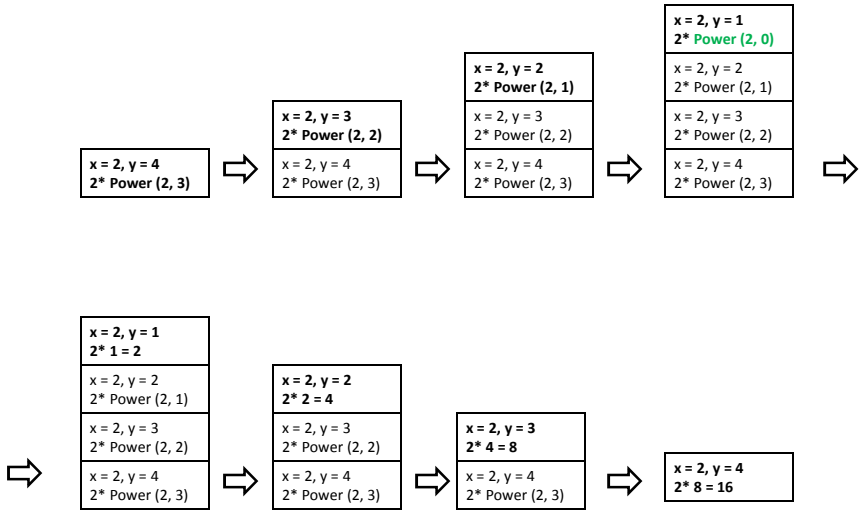
Factorial (5) = 120

Examples: Power x^y

$$x^y = \begin{cases} 1 & y = 0 \\ x * x^{y-1} & y > 0 \end{cases}$$

```
int Power (int x, int y)
{
    if (y == 0)    // ground case
        return 1;
    else           // recursive case
        return x * Power (x, y - 1);
}
```

Power (2, 4)



Power (2, 4) = 16

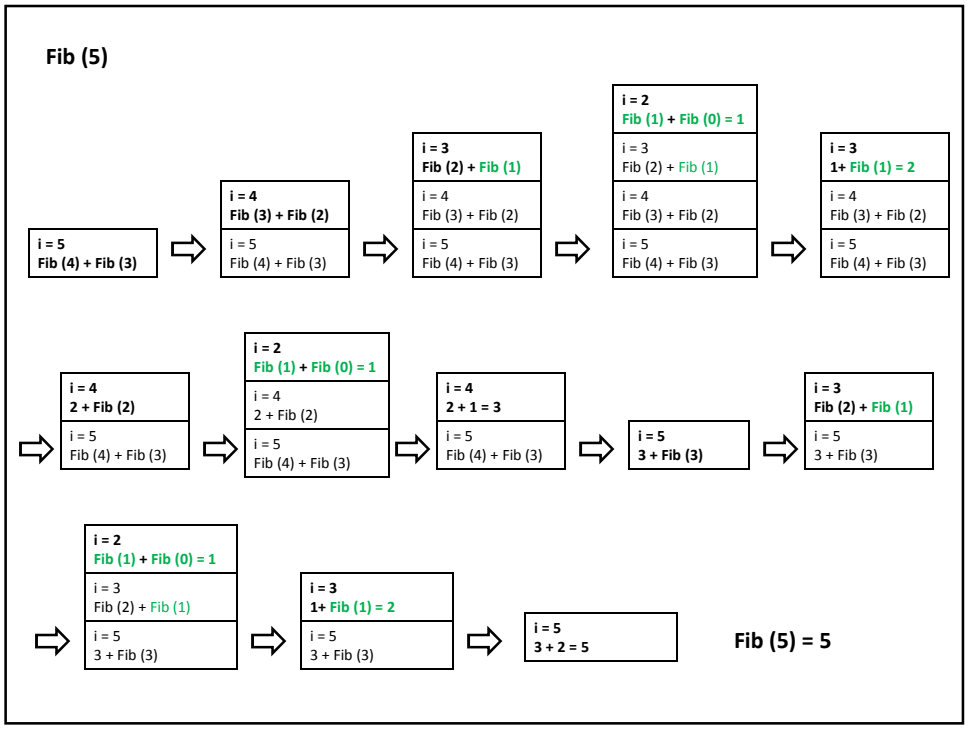
Examples: Fibonacci Series

Fibonacci Series:

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**…**
0	1	1	2	3	5	8	13	21	34	…

$$Fib\ (i) = \begin{cases} i & i < 2 \\ Fib\ (i-1) + Fib\ (i-2) & i \geq 2 \end{cases}$$

```
int Fib (int i)
{
    if (i > 2)    // ground case
        return i;
    else          // recursive case
        return Fib (i – 1) + Fib (i – 2);
}
```



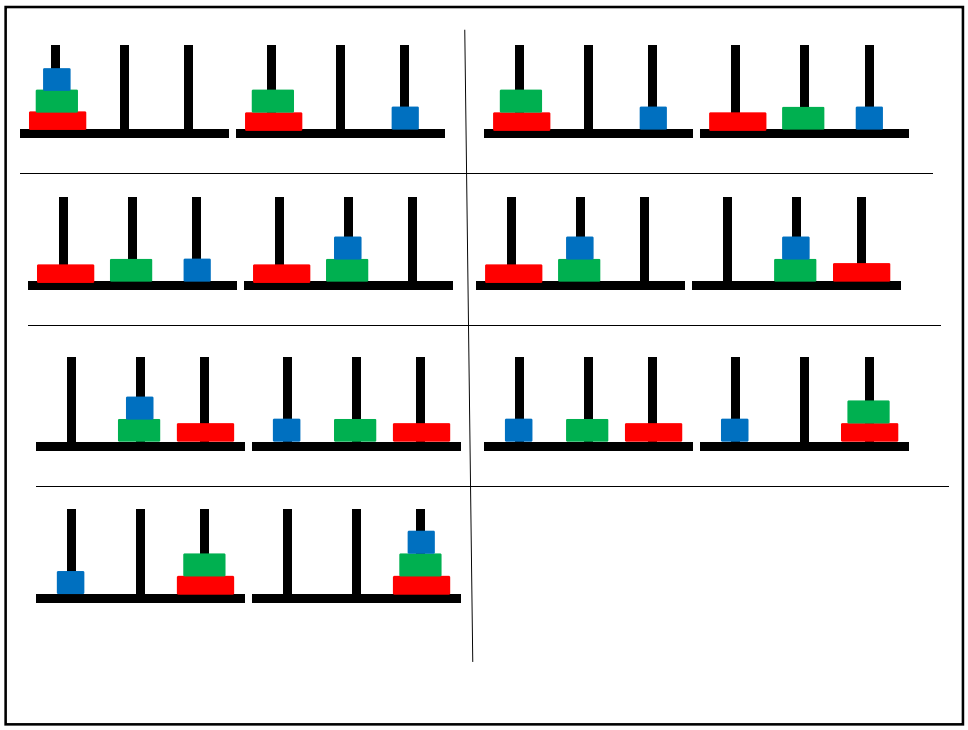
Examples: Tower of Hanoi

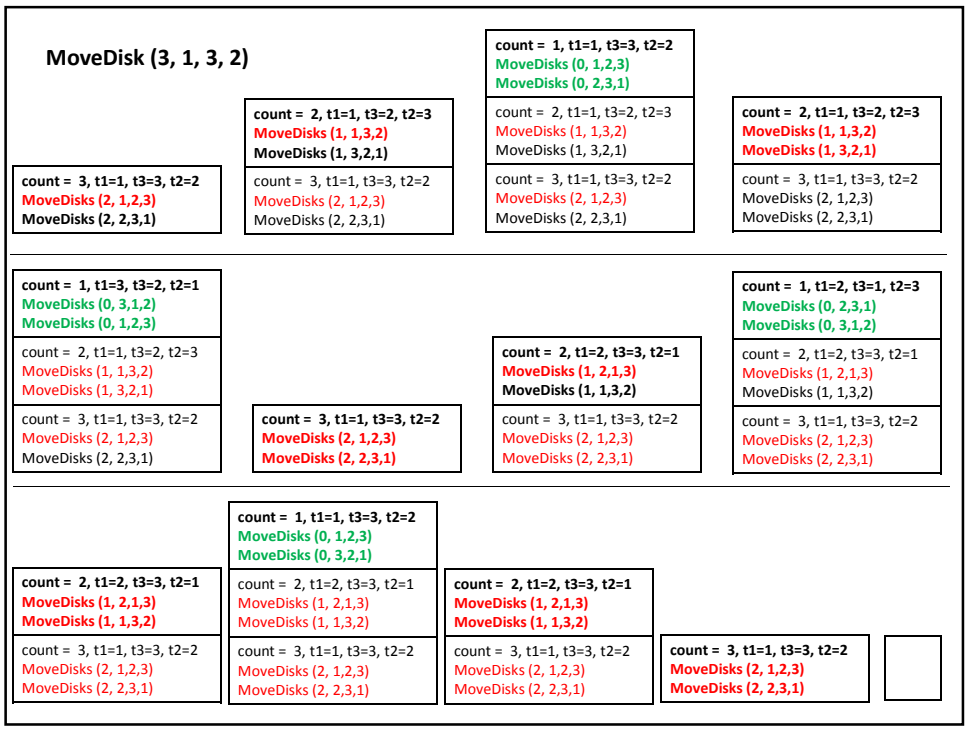
❖ Move disks from tower 1 to tower 3

Rules:

1. Only one disk can be moved at a time
2. The removed disk must be placed on one of the towers
3. A larger disk cannot be placed on top of smaller disk

```
void MoveDisks (int count, int t1, int t3, int t2)
{
   if (count > 0) {    // ground case
         MoveDisks (count - 1, t1, t2, t3);
         cout<<“Move disk ”<<count<< “Form ”<<t1<<“to ”<<t3;
         MoveDisks (count - 1, t2, t3, t1);
}
```




MoveDisk (3, 1, 3, 2)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 2, t1=1, t3=2, t2=3
MoveDisks (1, 1,3,2)
MoveDisks (1, 3,2,1)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 1, t1=1, t3=3, t2=2
MoveDisks (0, 1,2,3)
MoveDisks (0, 2,3,1)
count = 2, t1=1, t3=2, t2=3
MoveDisks (1, 1,3,2)
MoveDisks (1, 3,2,1)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 2, t1=1, t3=2, t2=3
MoveDisks (1, 1,3,2)
MoveDisks (1, 3,2,1)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 1, t1=3, t3=2, t2=1
MoveDisks (0, 3,1,2)
MoveDisks (0, 1,2,3)
count = 2, t1=1, t3=2, t2=3
MoveDisks (1, 1,3,2)
MoveDisks (1, 3,2,1)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 2, t1=2, t3=3, t2=1
MoveDisks (1, 2,1,3)
MoveDisks (1, 1,3,2)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 1, t1=2, t3=1, t2=3
MoveDisks (0, 2,3,1)
MoveDisks (0, 3,1,2)
count = 2, t1=2, t3=3, t2=1
MoveDisks (1, 2,1,3)
MoveDisks (1, 1,3,2)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 1, t1=1, t3=3, t2=2
MoveDisks (0, 1,2,3)
MoveDisks (0, 3,2,1)
count = 2, t1=2, t3=3, t2=1
MoveDisks (1, 2,1,3)
MoveDisks (1, 1,3,2)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 2, t1=2, t3=3, t2=1
MoveDisks (1, 2,1,3)
MoveDisks (1, 1,3,2)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)
count = 3, t1=1, t3=3, t2=2
MoveDisks (2, 1,2,3)
MoveDisks (2, 2,3,1)