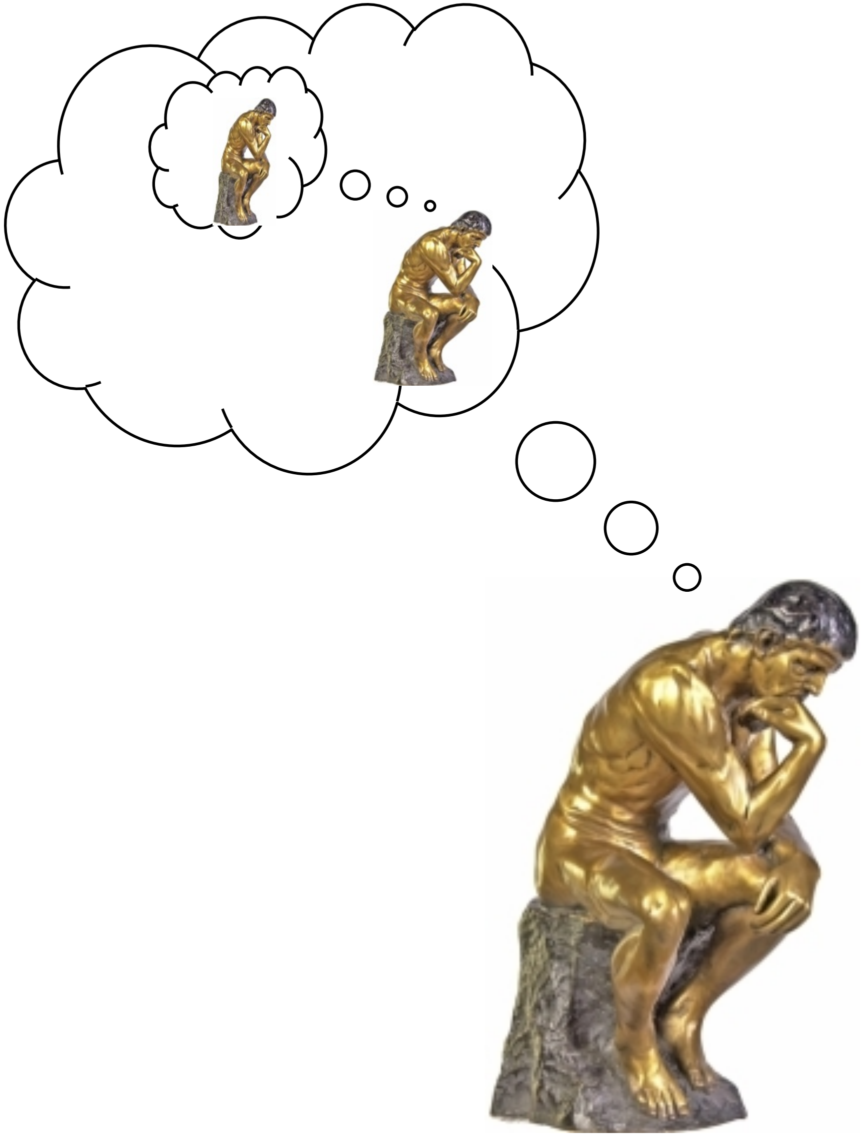
Recursive Programming

To understand recursion, first one must understand recursion



A recursive algorithm is one that solves a problem by using the same algorithm to solve a smaller part of the problem.

For example:

List all the presidents of the US.

Can be solved by

Identifying the *first* president, then
List the *rest* of the presidents.

```
To List all the presidents
    If there are any president to list,
        Identify the first president, then
        List the rest of the presidents.
```

can be done with no recursion

Recursive call but smaller problem

List:

Washington
Adams, J
Jefferson
Madison
Monroe
Adams, JQ
Jackson
Van Buren

Identify:
Washington

List:

Adams, J
Jefferson
Madison
Monroe
Adams, JQ
Jackson
Van Buren

Identify:
Adams,J

List:

Jefferson
Madison
Monroe
Adams, JQ
Jackson
Van Buren

Problems that are recursive by definition

Factorial

$$5! = \underbrace{1 * 2 * 3 * 4}_{} * 5$$

$$4! = \underbrace{1 * 2 * 3}_{} * 4$$

$$3! = \underbrace{1 * 2}_{} * 3$$

$$2! = \underbrace{1}_{} * 2$$

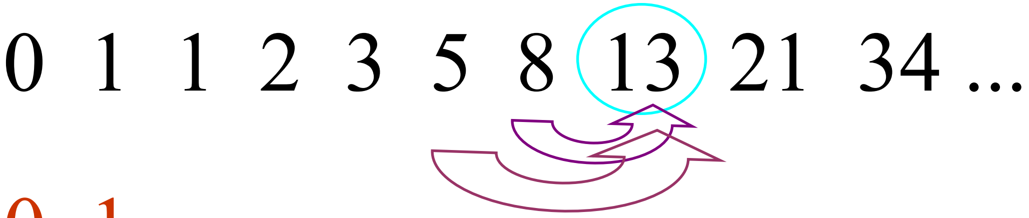
$$1! = 1$$

$$0! = 1$$

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

Problems that are recursive by definition

Fibonacci Numbers



0 1 1 2 3 5 8 13 21 34 ...

0 1

The n*th* Fibonacci number is the sum of the previous two.

Fibonacci(n) = Fibonacci(n-1)+Fibonacci(n-2)

```
private static int fibonacci (int n) {
     if (n==0)
        return 0;
     if
        (n==1)
        return 1;
     return fibonacci(n-1)+fibonacci(n-2);
  }
```

Problems that are recursive by definition

Greatest Common Divisor

Euclidian Algorithm:

Given two natural numbers *a* and *b*: check if *b* is zero; if yes, *a* is the gcd. If not, repeat the process using (respectively) *b*, and the remainder after dividing *a* by *b*.

```
private static int gcd (int n, int d) {
        if (d == 0) return n;
        else return gcd(d, n % d);
  }
```

Problems that are recursive by definition

Binomial Coefficient

Given n items, how many ways can you choose r of them?

The Binomial Coefficient is given by

$$\begin{matrix} n \\ r \end{matrix} = \frac{n!}{(n \quad r)!r!}$$

Example: How many ways can you choose two socks from a collection of five socks (A,B,C,D,E)?

$$\begin{matrix} 5 \\ 2 \end{matrix} = \frac{5!}{(5 \quad 2)!2!} = \frac{5*4*3*2*1}{3*2*1*2*1} = 10$$

(AB,AC,AD,AE,BC,BD,BE,CD,CE,DE)

Binomial Coefficients also give us the coefficients of (a+b)^n

$$(a+b)^n = {n \atop 0} a^n b^0 + {n \atop 1} a^{n\;1} b^1 + {n \atop 2} a^{n\;2} b^2 + {n \atop 3} a^{n\;1} b^3 + \ldots + {n \atop n} a^0 b^n$$

$$(a+b)^4 = {4 \atop 0} a^4 b^0 + {4 \atop 1} a^3 b^1 + {4 \atop 2} a^2 b^2 + {4 \atop 3} a^1 b^3 + {4 \atop 4} a^0 b^4$$

$$(a+b)^4 = 1a^4 b^0 + 4a^3 b^1 + 6a^2 b^2 + 4a^1 b^3 + 1a^0 b^4$$

$$(a+b)^4 = a^4 + 4a^3 b^1 + 6a^2 b^2 + 4a^1 b^3 + b^4$$

Binomial Coefficients also give us
Pascal's Triangle which gives us the coefficients of (a+b)^n

r

n	0	1	2	3	4	5	6
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
6	1	6	15	20	15	6	1

$$\begin{matrix} n \\ r \end{matrix} = \frac{n!}{(n \quad r)!r!} = \begin{matrix} n & 1 \\ r & 1 \end{matrix} + \begin{matrix} n & 1 \\ r & \end{matrix}$$

$$(a+b)^4 = a^4 + 4a^3b^1 + 6a^2b^2 + 4a^1b^3 + b^4$$

$$\begin{matrix} n \\ r \end{matrix} = \frac{n!}{(n \quad r)!r!} = \begin{matrix} n & 1 \\ r & 1 \end{matrix} + \begin{matrix} n & 1 \\ r & \end{matrix}$$

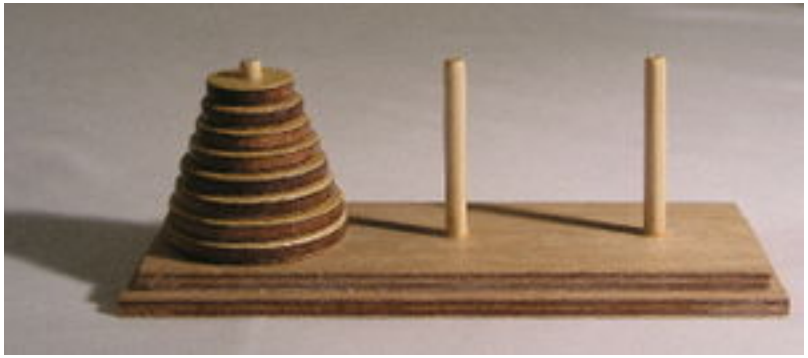
```
private static int bc (int n, int r) {
   if (n==0 || r==0 ||n==r) {
       return 1;
   }
   else
      return bc(n-1,r)+bc(n-1,r-1);
   }
```

The Towers of Hanoi

The puzzle was invented by the French mathematician Édouard Lucas in 1883.

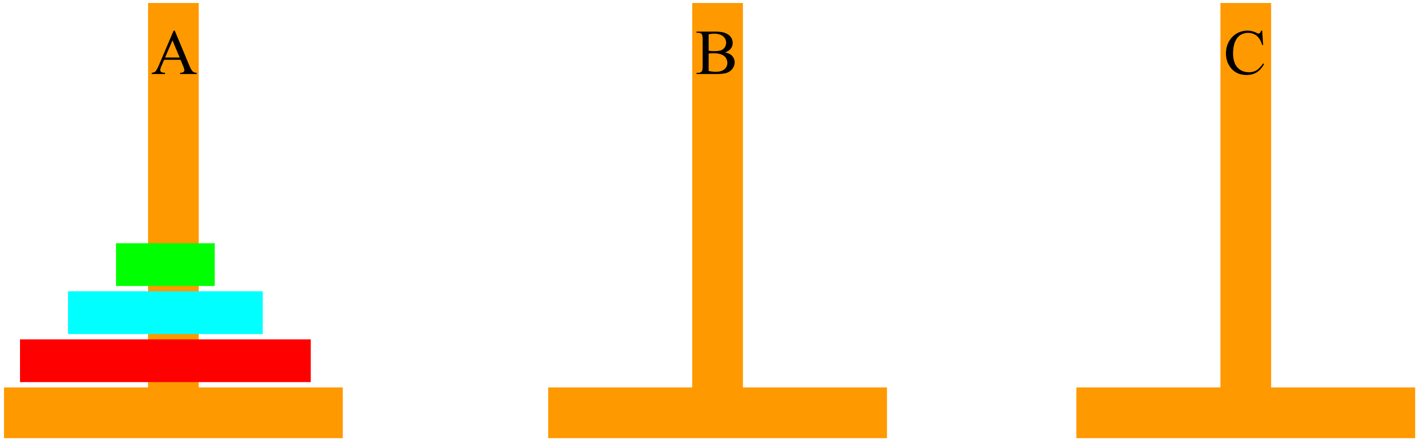
There is a legend about an Indian temple which contains a large room with three time-worn posts in it surrounded by 64 golden disks. The priests of Brahma, acting out the command of an ancient prophecy, have been moving these disks, in accordance with the rules of the puzzle.

According to the legend, when the last move of the puzzle is completed, the world will end. The puzzle is therefore also known as the Tower of Brahma puzzle. It is not clear whether Lucas invented this legend or was inspired by it.

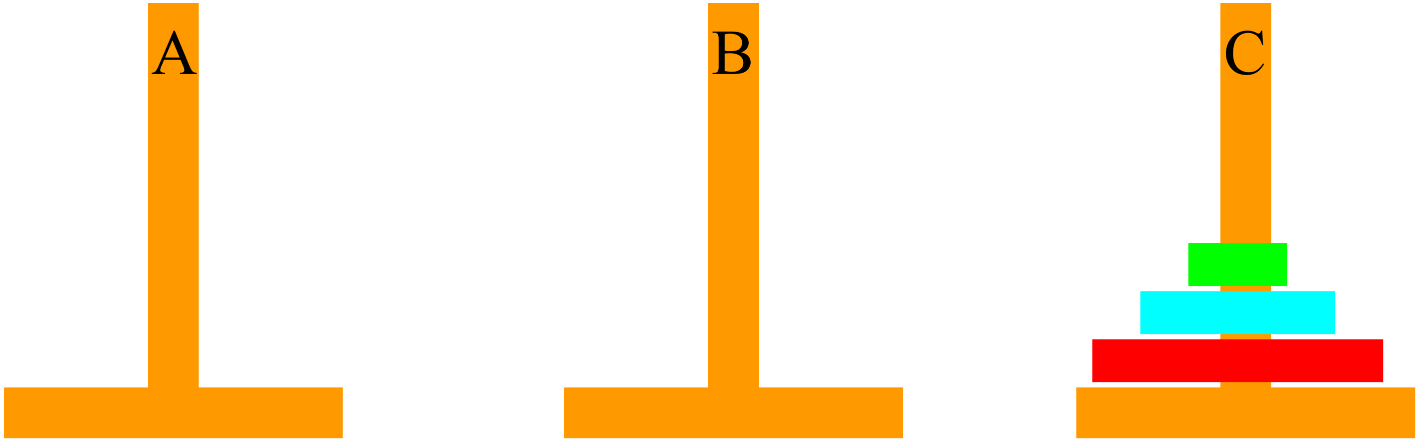


If the legend were true and if the priests were able to move the 64 disks at a rate of 1 per second, using the smallest number of moves, it would take them $2^{64}-1$ seconds or roughly 585.442 billion years. The universe is currently about 13.7 billion years old.

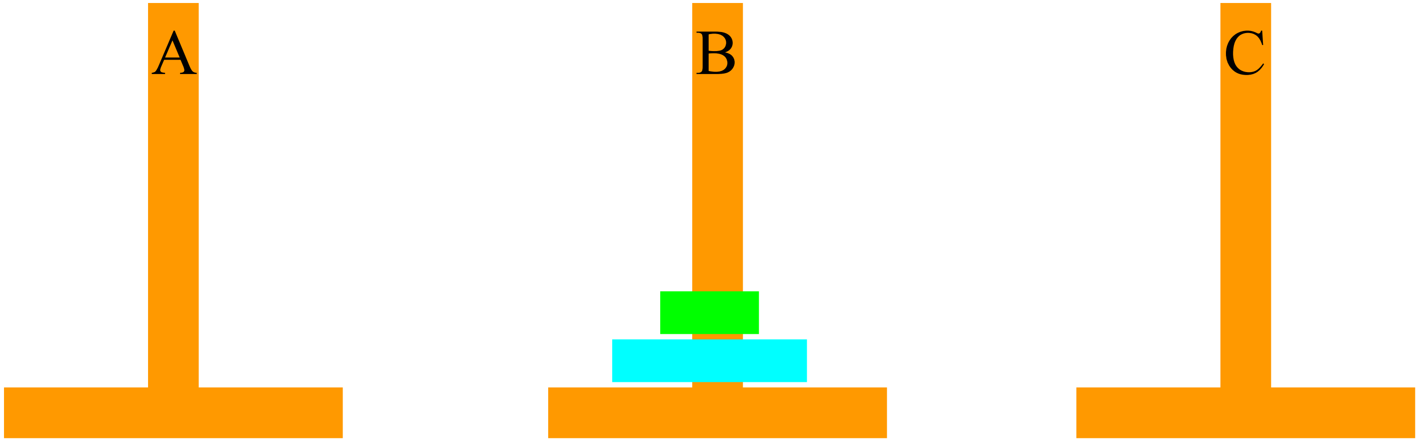
Move the rings from pole A to pole C



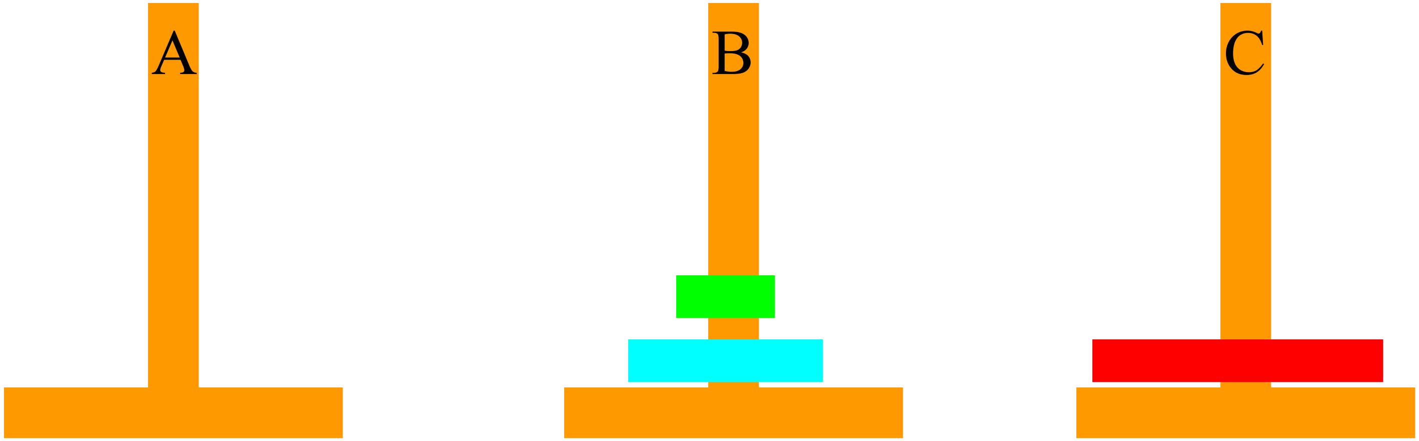
You may move one ring at a time, and never put a bigger ring on top of a smaller ring.

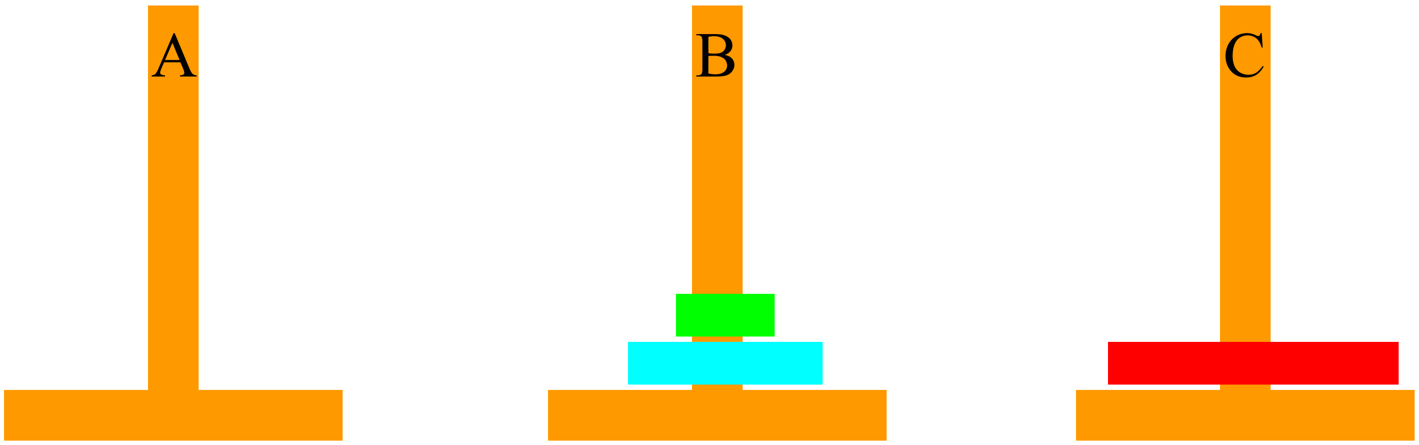


First, move the top *n-1* rings from A to B

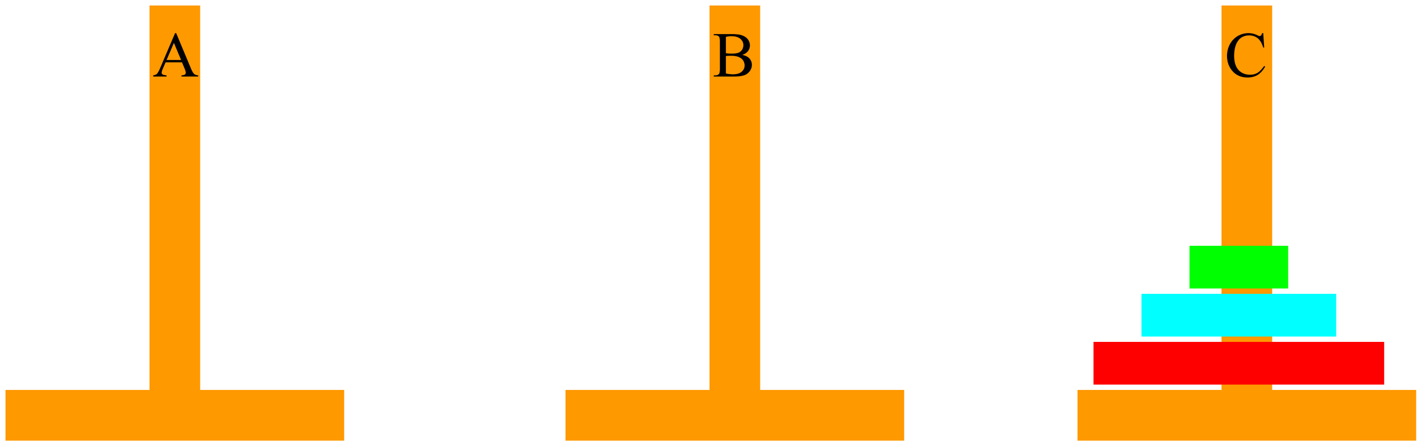


Then move the final ring from A to C





Now, move *n-1* rings from B to C



But we can't move more than one ring at a time!

Yes, but that's the problem we're trying to solve...
move *n* rings from A to C (with the help of B)

So,
We can move *n-1* rings from A to B (with the help of C)
Move the last ring from A to C
Then move *n-1* rings from B to C (with the help of A)

```
private static void moveRings(int numberOfRings,
                              String fromTower,
                              String toTower,
                              String tempTower) {

   if (numberOfRings == 0) return;
   moveRings(numberOfRings-1, fromTower, tempTower, toTower);
   System.out.println("Move a ring from tower "+
                       fromTower+" to tower "+toTower);
   moveRings(numberOfRings-1, tempTower, toTower, fromTower);
}
```