Example 6 -- Towers of Hanoi
(see https://en.wikipedia.org/wiki/Tower_of_Hanoi)

```
// Prints instructions for moving disks from one
// pole to another, where the three poles are
// labeled with integers "from", "to", and "other".
// Code from rosettacode.org
public void move(int n, int from, int to, int other) {
    if (n == 1) {
        System.out.println("Move disk from pole "+ from + " to pole "+ to);
    } else {
        move(n - 1, from, other, to);
        move(1, from, to, other);
        move(n - 1, other, to, from);
    }
}
```

Recursive function!
Let's use recurrence relations.
Let $H(n) = \#$ executions to run alg. on $n$. 

Friday, June 23, 2017
12:40 PM
Base case \( \@ n=1 \)

```java
if (n == 1) {
    System.out.println("Move disk from pole " + from + " to pole " + to);
} else {
    move(n - 1, from, other, to);
    move(1, from, to, other);
    move(n - 1, other, to, from);
}
```

All other \( H(n) \):

\[
H(n) = H(n-1) + 1 + H(n-1)
= 1 + 2H(n-1)
\]

Expanding (plug in for \( H(n) \)):

1st \( H(n) = 1 + 2H(n-1) \)

2nd \( = 1 + 2 + 4H(n-2) \)

3rd \( = 1 + 2 + 4 + 8H(n-3) \)

\( \vdots \)

\( k \)th \( = 2^0 + 2^1 + 2^2 + \cdots + 2^{k-1} + 2^k H(n-k) \)

Plugging in \( n-1 \) into \( H(n) \)

```javascript
so \( H(n-1) = 1 + 2[1 + 2H(n-1)] \)
= 1 + 2 + 4H(n-2)```

The expansions uncover this pattern!
\[ H(n) = 2^k - 1 + 2^k H(n-k) \]

**Base case is at** \( H(1) \), so

**let's solve** \( n-k = 1 \)

\[ \Rightarrow k = n-1 \]

**Plug it in:**

\[ H(n) = 2^k - 1 + 2^k H(n-k) \]
\[ = 2^{n-1} - 1 + 2^{n-1} H(n-[n-1]) \]
\[ = 2^{n-1} - 1 + 2^{n-1} H(1) \]
\[ = 2^{n-1} - 1 + 2^{n-1} (1) \]
\[ = 2 (2^{n-1}) - 1 \]

\[ H(n) = 2^n - 1 \]

\[ \Rightarrow O(2^n) \]