Tower of Hanoi & Big O

Pseudocode:

FUNCTION MoveTower(n, source, dest, spare):
    IF n == 1, THEN:
        move disk 1 from source to dest //base case
    ELSE:
        MoveTower(n - 1, source, spare, dest)  // Step 1
        move disk n from source to dest        // Step 2
        MoveTower(n - 1, spare, dest, source)  // Step 3
    END IF
Tower of Hanoi

Tree of Recursive MoveTower Function Calls for n=3
Tower of Hanoi

Recursive `MoveTower` Function Calls for `n=3` makes 7 disk moves

```
1  2  3  4  5  6  7
```

```
MoveTower(1, A, B, C)
MoveTower(2, A, C, B)
MoveTower(3, A, B, C)
MoveTower(2, C, B, A)
MoveTower(1, C, A, B)
MoveTower(1, A, B, C)
```

Move Big Disk From A to B
The base case - when \( n = 1 \) - is easy: The monks just move the single disk directly.

\[ M(1) = 1 \quad \text{//number of disk moves} \]

In the other cases, the monks follow our three-step procedure.
1. First they move the \((n-1)\)-disk tower to the spare peg;
   - this takes \( M(n-1) \) moves.
2. Then the monks move the \( n \)th disk, taking 1 move.
3. And finally they move the \((n-1)\)-disk tower again, this time on top of the \( n \)th disk,
   - taking \( M(n-1) \) moves.

This gives us our recurrence relation,
\[ M(n) = 2 \, M(n-1) + 1 \]
A closed-form solution

Let's figure out values of $M$ for the first few numbers.

\[
\begin{align*}
M(1) &= 1 \\
M(2) &= 2M(1) + 1 = 3 \\
M(3) &= 2M(2) + 1 = 7 \\
M(4) &= 2M(3) + 1 = 15 \\
M(5) &= 2M(4) + 1 = 31 \\
\end{align*}
\]

By looking at this, we can see that

$M(n) = 2^n - 1$

We can verify this easily by plugging it into our recurrence.

\[
\begin{align*}
M(1) &= 1 = 2^1 - 1 \\
M(n) &= 2M(n-1) + 1 = 2(2^n - 1) + 1 = 2^n - 1
\end{align*}
\]

Big O = $O(2^n)$
Want more Big O?

- Check out Chapter 18 p1234-1242
Want more?

Things to think about for next semester
State Graph for Tower of Hanoi with 1 disk

Each state is represented by a labeled \textit{vertex}; legal moves are represented by \textit{edges}.

Solution

Move from state 1 to state 3
State Graph for Tower of Hanoi with 2 disks

Solution
State Graph: 3 disks