CSE 250 Recitation
Feb 27 - Mar 3: Inductive Proofs
var towers = Array(new Stack(), new Stack(), new Stack())

def move(fromTower: Int, toTower: Int, numDisks: Int): Unit = {
    val otherTower = (Set(0, 1, 2) - fromTower - toTower).head

    if(numDisks == 1){
        moveOne(from = fromTower, to = toTower)
    } else {
        move(fromTower, otherTower, numDisks-1)
        moveOne(from = fromTower, to = toTower)
        move(otherTower, toTower, numDisks-1)
    }
}

1: Write down the recursive runtime
Runtime Growth Function

\[ T(n) = \begin{cases} 
\Theta(1) & \text{if } n \leq 1 \\
2 \cdot T(n - 1) + \Theta(1) & \text{otherwise}
\end{cases} \]
Hypothesis: Draw it out

\[
\begin{align*}
\text{layers?} \\
n \\
n-1 \\
n-2 \\
\vdots \\
1 \\
\end{align*}
\]

\[
\begin{align*}
\Theta(1) & \quad \Theta(1) \quad \Theta(1) \\
\Theta(1) & \quad \Theta(1) \quad \Theta(1) \\
\Theta(1) & \quad \Theta(1) \\
\end{align*}
\]

___ operations at level i
Sum Up

\[ \sum_{i=0}^{n-1} \sum_{j=1}^{2^i} \Theta(1) \]
Inductive Hypothesis

\[ T(n) \in \Theta(2^n - 1) \]
Bound from Above: Base Case

\[ T(1) \leq c \cdot (2^1 - 1) \]
Bound from Above: Inductive Case

Assume: \[ T(n - 1) \leq c \cdot (2^{n-1} - 1) \]

Show: \[ T(n) \leq c \cdot (2^n - 1) \]