1. Suppose that the worst cost \( c(n) \) for some algorithm \( A \) satisfies the following recurrence relation:

\[
c(n) = \begin{cases} 
1, & \text{if } n = 1 \\
c(n-1) + 2^n, & \text{if } n > 1 
\end{cases}
\]

Prove by induction that \( c(n) = 2^{n+1} - 1 \).

2. For each of the following loops, give a formula for the exact number of times that the basic operation is performed.

(a) \[
\begin{align*}
p &\leftarrow 1 \\
\text{for } i &\leftarrow 1 \text{ to } n^2 \text{ do} \\
p &\leftarrow p \cdot i 
\end{align*}
\]

(b) \[
\begin{align*}
s &\leftarrow 0 \\
\text{for } i &\leftarrow 1 \text{ to } 2n \text{ do} \\
\text{for } j &\leftarrow 1 \text{ to } i \text{ do} \\
s &\leftarrow s + i 
\end{align*}
\]

(c) \[
\begin{align*}
s &\leftarrow 0 \\
\text{for } i &\leftarrow 1 \text{ to } n^2 \text{ do} \\
\text{for } j &\leftarrow 1 \text{ to } i \text{ do} \\
s &\leftarrow s + i 
\end{align*}
\]

3. Consider writing a recursive algorithm for finding the minimum value in an array \( A[0..n-1] \).

Algorithm \text{FindMin}(A, \text{hi})

\[
// \text{FindMin returns the minimum value} \\
// \text{in the subarray } A[0..\text{hi}] 
\]

Notice that to find the minimum value in the entire array, we call \text{FindMin}(A, n-1).

(a) Write the pseudocode for algorithm \text{FindMin}.

(b) Set up a recurrence relation for \( c(n) \), the worst cost of algorithm \text{FindMin}. \textbf{Remember to include a base case!} (Use the form of the recurrences in problems 1 and 2 as a model.) Note that \textit{comparison} is the basic operation in this algorithm.

(c) Solve the recurrence relation in part b.

4. Consider the problem of counting \textit{in binary} from the number 0 to \( 2^n - 1 \). For example, if \( n = 3 \), the count sequence would be

000, 001, 010, 011, 100, 101, 110, 111.

Suppose we have constructed an algorithm that outputs such a count sequence for any integer \( k \) of the form \( 2^n - 1 \). Furthermore, say that the basic operation is flipping a bit from 0 to 1 or 1 to 0. In the count sequence shown above, the number of bit flips is

\[ 1 + 2 + 1 + 3 + 1 + 2 + 1 = 11. \]

For \( n = 4 \), the number of bit flips is

\[ 1 + 2 + 1 + 3 + 1 + 2 + 1 + 4 + 1 + 2 + 1 + 3 + 1 + 2 + 1 = 26. \]

Study these two sums. You should see a relationship between them.

(a) Write a recurrence relation for \( c(2^n - 1) \). \textbf{Remember to include a base case!}

(b) ‘Unroll’ the recurrence and express \( c(2^n - 1) \) as a summation.

(c) \textbf{(Extra Credit)} Find the sum in part b. \textit{You might want to use the formula}

\[
\sum_{i=1}^{k} i2^i = (k-1)2^{k+1} + 2
\]

\textit{somewhere in your derivation}.