Test #1

Instructions. This is a 75-minute test. There are 5 questions worth a total of 60 points.

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>60</td>
</tr>
</tbody>
</table>

Name: ______________________________
1. (15 points) In each of the following parts, suppose we wish to prove that a given predicate $P(n)$ holds for all nonnegative integers $n$.

   (a) (5 points) Describe the structure of a proof by “weak” induction.

   (b) (5 points) Describe the structure of a proof by “strong” induction.

   (c) (5 points) State the well-ordering property of the set of nonnegative integers, and use this property to establish the validity of the weak induction technique described in part (a).
2. **(10 points)** Prove by induction that

\[
\sum_{0 \leq i < n} i \cdot 2^i = n \cdot 2^n - 2^{n+1} + 2
\]

for all \( n \geq 0 \).
3. **(10 points)** Find the error(s) in the following incorrect proof that $2^n = 1$ for all nonnegative integers $n$. For the base case, $n = 0$, note that $2^0 = 1$. For the induction step, let $n$ be a nonnegative integer and assume that $2^k = 1$ for all integers $k$ such that $0 \leq k \leq n$; it remains to prove that $2^{n+1} = 1$. Note that

$$2^{n+1} = \frac{2^n \cdot 2^n}{2^{n-1}} = \frac{1 \cdot 1}{1} = 1,$$

where the second equation follows from the induction hypothesis (which implies that $2^n = 1$ and $2^{n-1} = 1$).

4. **(10 points)** Recursively define the set of bit strings that have more ones than zeros, briefly justifying your answer. Note: Your definition is not allowed to make use of a function that counts the number of zeros or ones (or the difference between the number of zeros and ones) in a bit string.
5. (15 points total) Each of the following parts is concerned with the recursive program foo(n) defined below.

\[
\text{foo}(n) \\
\quad \text{if } n = 0 \text{ then} \\
\quad \quad \text{return } 1 \\
\quad \text{else} \\
\quad \quad \text{return } 1 + \min\{2 \cdot \text{foo}\left(\left\lfloor \frac{n-1}{2} \right\rfloor \right), 3 \cdot \text{foo}\left(\left\lfloor \frac{n-1}{3} \right\rfloor \right)\}
\]

(a) (4 points) Prove by (strong) induction on \( n \) that the recursive program foo(\( n \)) terminates for all \( n \geq 0 \).

(b) (8 points) Prove by (strong) induction on \( n \) that for all \( n \geq 0 \), foo(\( n \)) returns a value greater than or equal to \( n \). Hint: Note that foo(0) = 1, foo(1) = 3, foo(2) = 3, foo(3) = 4, foo(4) = 7, and so on; we’ve seen this sequence before!

(c) (3 points) For any nonnegative integer \( n \), let \( a_n \) denote the total number of additions performed during the execution of foo(\( n \)). Give a recursive definition for \( a_n \) for all \( n \geq 0 \).