• Do not turn this page until you have received the signal to start.
• This examination is closed book exam.
• Each question tells you how many points is worth.
• It is your responsibility to write clearly.
• Answer each question directly on the examination paper, in the space provided, and use a ‘blank’ page for rough work. If you need more space for one of your solutions, use one of the ‘blank’ pages and indicate clearly the part of your work that should be marked.
• There is an indication at the right-down corner with the number of the page that you are currently at and of the total number of pages

Student Number:

Student Name:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Your Mark</th>
<th>Total</th>
<th>Problem</th>
<th>Your Mark</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>15</td>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (10 points) Prove by simple induction the following statement for all natural numbers $n$:
\[ \sum_{t=0}^{n} 2^t = 2^{n+1} - 1 \]

Suggested Solution

Base Step: For $n = 0$, $\sum_{t=0}^{0} 2^t = 1 = 2^{0+1} - 1$.

Induction Step: Assume that $\sum_{t=0}^{n} 2^t = 2^{n+1} - 1$ is true. Then,
\[ \sum_{t=0}^{n+1} 2^t = (\sum_{t=0}^{n} 2^t) + 2^{n+1} = 2^{n+1} - 1 + 2^{n+1} = 2^{n+2} - 1 \]
2. (10 points) Prove by complete induction that \( a_n < 2^n \) for all the natural numbers \( n \geq 2 \). Define the sequence of integer \( a_i \) as follows:

\[
a_i = \begin{cases} 
2, & \text{if } 0 \leq i \leq 2 \\
 a_{i-1} + a_{i-2} + a_{i-3}, & \text{if } i > 2
\end{cases}
\]

Suggested Solution

Base Step: For \( n = 2 \), \( a_2 = 2 < 2^2 \), \( a_3 = 6 < 2^3 \), and \( a_4 = 10 < 2^4 \).

Induction Step: Assume that \( a_i < 2^i \) is true for all \( i, \, 4 < i \leq k - 1 \). We will prove that it is true for \( a_k < 2^k \). Then, we have that:

\[
\begin{align*}
 a_{k-3} &< 2^{k-3} \\
 a_{k-2} &< 2^{k-2} \\
 a_{k-1} &< 2^{k-1}
\end{align*}
\]

Summing the above we have that:

\[
a_k = a_{k-1} + a_{k-2} + a_{k-3} < 2^{k-1} + 2^{k-2} + 2^{k-3} < \\
< 2^{k-1} + 2^{k-2} + 2^{k-2} = 2^{k-1} + 2^{k-1} = 2^k
\]
3. (15 points) Answer the following three questions about induction:

a) (5 points) State the well-ordering principle.
b) (5 points) State the principle of simple induction.
c) (5 points) Prove that well-ordering implies simple induction.

Suggested Solution

a) Principle of well-ordering: Any nonempty subset $A$ of $\mathbb{N}$ contains a minimum element; i.e., for any $A \subseteq \mathbb{N}$ such that $A \neq \emptyset$, there is some $a \in A$ such that for all $a' \in A$, $a \leq a'$.
b) Principle of simple induction: Let $A$ be any subset of $\mathbb{N}$ that satisfies the following properties:

(i) $0$ is an element of $A$;
(ii) for any $i \in \mathbb{N}$, if $i$ is an element of $A$ then $i + 1$ is also an element of $A$.
Then $A = \mathbb{N}$.
c) Well-ordering implies induction: Assume that the principle of well-ordering holds. We will prove that the principle of induction is also true. Let $A$ be any set that satisfies the properties from simple induction. We need to prove that $A \supseteq \mathbb{N}$. We do so using a proof by contradiction.
Suppose, for contradiction, that $A$ is not a superset of $\mathbb{N}$. Then the set $\overline{A} = \mathbb{N} \setminus A$ must be nonempty. By the principle of well-ordering (which holds by assumption) $\overline{A}$ has a minimum element, say $i$. By $(i)$, $i \neq 0$. Thus, $i - 1 \in A$, then from $(ii)$, $i \in A$, contradiction.
4. (10 points) Define the set $S$ to be the smallest set such that:

- $(1, 0) \in S$
- if $(x, y) \in S$ then $(x + 1, x + y) \in S$

Prove that for every integer $n \geq 1$, the tuple $(n, \frac{n(n-1)}{2})$ is in $S$.

Suggested Solution

Base Step: For $n = 1$, $(1, \frac{1(0)}{2}) = (1, 0)$ is in $S$.

Induction Step: Assume that $(n, \frac{n(n-1)}{2})$ is in $S$. Then, by the definition of the set $S$, $(n + 1, n + \frac{n(n-1)}{2})$ is in $S$. For this tuple we have:

\[
(n + 1, n + \frac{n(n-1)}{2}) = (n + 1, \frac{n^2 - n + 2n}{2}) = \\
= (n + 1, \frac{n^2 + n}{2}) = (n + 1, \frac{n(n + 1)}{2})
\]
5. (15 points) Consider the following recursive linear search algorithm.

\[
\text{REC-LIN-SEARCH} (A, i, x): \\
\quad \text{if } i >= \text{len}(A): \\
\quad \quad \text{return False} \\
\quad \text{else:} \\
\quad \quad \text{return } A[i] == x \text{ or REC-LIN-SEARCH}(A, i + 1, x)
\]

a) (5 points) Define a recurrence \( T(n) \) for the worst-case runtime of \( \text{REC-LIN-SEARCH}(A, i, x) \), where \( n = \text{len}(A) - i \).

b) (5 points) Use unwinding to find a closed form for \( T(n) \). Based on the closed form, state a conjecture for \( f(n) \) such that \( T(n) \in \Theta(f(n)) \).

c) (5 points) Prove the bound on \( T(n) \) you gave in part (b) is correct.

Suggested Solution

a) 
\[
T(n) = \begin{cases} 
    c & \text{if } n = 0 \\
    T(n - 1) + d & \text{if } n > 0 
\end{cases}
\]

For some constants \( c, d \).

b) 
\[
T(n) = T(n - 1) + d \\
= T(n - 2) + 2d \\
= T(n - 3) + 3d \\
\ldots \\
= T(0) + d \cdot n = d \cdot n + c
\]

We conjecture that \( f(n) = n \), and \( T(n) \in \Theta(n) \).

c) In order to show that \( T(n) \in \Theta(n) \), we have to show that as \( n \) goes to infinity asymptotically \( T(n) \) and \( f(n) \) grow the same. Since,

\[
\lim_{n \to \infty} \frac{T(n)}{f(n)} = d
\]

the above limit is less than infinity and more than 0, we have that \( T(n) \in \Theta(n) \).
6. (20 points) Answer the following three questions about Divide-and-Conquer algorithms:

   a) (5 points) State the master theorem.

   b) (10 points) Informally describe a divide-and-conquer algorithm to find the index of a local maximum element in list of integers $A$. Your algorithm should have complexity strictly less than $O(n)$. An element of a list is a local maximum if it is greater or equal to its neighbor(s).
   
   Hint: The list $[1, 2, 3, 2, 4, 7, 5]$ has two local maximums, 3 and 7 with indexes 2 and 5 respectively. Finding either is a correct answer.

   c) (5 points) Use the master theorem to find the worst case complexity of your algorithm from (b).

   a) Let the function $T$ be defined as follows:

   $$ T(n) = \begin{cases} 
   k & \text{if } n = 0 \\
   aT(\lceil n/b \rceil) + f(n) & \text{if } n > 0 
   \end{cases} $$

   for $f \in \theta(n^d)$, then

   $$ T(n) \in \begin{cases} 
   \Theta(n^d) & \text{if } a < b^d \\
   \Theta(n^d \log n) & \text{if } a = b^d \\
   \Theta(n^{\log_b a}) & \text{if } a > b^d 
   \end{cases} $$

   b) \text{LOC-MAX}(A, i, j):

   if $i == j$:
   
   return $i$

   elif $i + 1 == j$:
   
   
   else: return $j$

   else:

   $m = \text{int}( (i+j)/2 )$

   if $A[m-1] <= A[m]$ and $A[m] >= A[m+1]$:
   
   return $m$

   elif $A[m-1] <= A[m]$ and $A[m] < A[m+1]$:
\textbf{return:} LOC-MAX(\(A, m, j\))
\textbf{elif} \(A[m-1] > A[m]\) and \(A[m] \geq A[m+1]\):
\textbf{return:} LOC-MAX(\(A, i, m\))
\textbf{else:}
\textbf{return:} LOC-MAX(\(A, m, j\))

c) We can define \(n = j - i\), then the recurrence relation giving the complexity of loc-max algorithm of (b) is given by:

\[
T(n) = \begin{cases} 
  c_1 & \text{if } n = 0 \\
  c_2 & \text{if } n = 1 \\
  T([n/2]) + d & \text{if } n > 1 
\end{cases}
\]

Using the master theorem the complexity of loc-max, 
\(T(n) \in \Theta(\log_2(n))\).