1. (a) (10 pts.) Prove the following function is $\Theta(n^4)$.

$$P(n) = 10n^4 - 50n^3 + 200n^2 - 1000n$$

(b) (10 pts.) Consider the following pseudo-code.

```plaintext
x = 0;
for k = 1 to n
    for j = k to 2n
        x = x + 1;
```

i. Find the exact number of times the innermost statement (increment $x$) is executed. That is, what is the final value of $x$?

ii. Express the time complexity $T(n)$ of this code in $O()$ form.
2. Consider the sequence \( F_0, F_1, F_2, \cdots \) defined recursively as follows. (This recursive definition is called a recurrence relation.)

\[
F_n = \begin{cases} 
0, & n = 0 \\
F_{n-1} + n, & n \geq 1.
\end{cases}
\]

(a) (5pts) Compute and tabulate \( F_n \) for \( n \) values 0 to 10.

(b) (15 pts) Prove by induction that for all \( n \geq 0 \), the solution is

\[
F_n = \frac{n^2 + n}{2}.
\]
3. (a) (10 pts.) Find the exact solution of the following recurrence. (Assume $n = 2^k$.) You may use either the repeated substitution method, or the master-theorem.

$T(1) = 1,$
$T(n) = 8T(n/2) + n^2, \quad n > 1.$

(b) (10 pts.) Consider the Fibonacci sequence: $F_1 = 1, F_2 = 1,$ and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 3.$

Use either repeated substitution or induction to prove that for all $n \geq 2,$

$$F_n \geq 2^{(n-2)/2}. \quad (1)$$
4. (a) Write a recursive version of the binary search algorithm modified so that it locates the left-most occurrence of the search key $X$ found in the array (in the event that there are several occurrences). For example, for the following array:

<table>
<thead>
<tr>
<th>$i$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A[i]$</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>20</td>
<td>25</td>
</tr>
</tbody>
</table>

a search on key 5 should return index 2.

(b) Illustrate your algorithm on the above example array.
5. (a) (8 pts.) Insert the following sequence of elements into a Binary-Search-Tree (BST), starting with an empty tree.

70, 35, 40, 90, 20, 50, 45, 60, 80, 95, 85

(b) (4 pts) A binary tree is called *height-balanced* if for every node in the tree, the heights of its left and right subtrees differ by at most 1. Is the final BST produced in part (a) height-balanced? Explain.

(c) (8 pts.) Delete element 80 in the following BST. Mark the operations on the given BST. Then, redraw the final BST after deletion.