1. Prove the following polynomial is $\Theta(n^2)$.

$$P(n) = 5n^2 - 100n + 1000.$$ 

(a) Prove $O(n^2)$:

$$P(n) = 5n^2 - 100n + 1000$$

$\leq 5n^2 + 1000$, $n \geq 0$

$\leq n^2 \left( 5 + \frac{1000}{n^2} \right)$, $n \geq 100$

$\leq 5.1 \cdot n^2$, $n \geq 100$

(b) Prove $\Omega(n^2)$:

$$P(n) = 5n^2 - 100n + 1000$$

$\geq 5n^2 - 100n$

$\geq n^2 \left( 5 - \frac{100}{n} \right)$

$\geq n^2 \left( 5 - \frac{100}{100} \right)$, $n \geq 100$

$\geq 4n^2$, $n \geq 100$
2. Find the exact number of times (in terms of $n$) the innermost statement ($X = X + 1$) is executed in the following code. That is, find the final value of $X$. Then express the total running time in terms of $O()$.

\[
X = 0; \\
\text{for } i = 1 \text{ to } n - 1 \\
\text{for } j = i \text{ to } 2n + 1 \\
X = X + 1;
\]

\[
X = \sum_{i=1}^{n-1} \sum_{j=i}^{2n+1} (1)
\]

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

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

\[
= (n-1)(2n+2) - (n-1) \frac{1+(n-1)}{2} \quad \text{(Aritmtic sum)}
\]

\[
= (n-1)(2n+2) - (n-1) \frac{n}{2}
\]

\[
= (n-1)(1.5n+2)
\]

\[
X = \frac{3}{2}n^2 + \frac{1}{2}n - 2
\]

This is $\Theta(n^2)$.

So, total running time is $\Theta(n^2)$. 
3. Consider the following divide-and-conquer algorithm (recursive function). Input array $A$ is
Boolean, with each element either 0 or 1. Parameter $i$ is the starting index of the array, and
$n$ is the number of elements. The initial call is $\text{COMPUTE}(A, 0, n)$.

```java
int COMPUTE(Boolean A[], int i, int n) {
    if (n == 1) return A[i];
    n1 = [n/2];  //Length of first half of array
    n2 = [n/2];  //Length of second half of array
    C1 = COMPUTE(A, i, n1);
    C2 = COMPUTE(A, i + n1, n2);
    return (C1 + C2)
}
```

(a) Figure out what the function does. (What does it compute?) Explain briefly.

IT adds all elements. (Sum of all elements)
This is easy to see by induction.

(b) Let $f(n)$ be the number of times the arithmetic operation $(C1 + C2)$ is performed by
this algorithm. Assume that $n$ is a power of 2. Write a recurrence for $f(n)$. Find the
solution of the recurrence by repeated substitution.

\[
f(n) = \begin{cases} 
2f(n/2) + 1, & n \geq 2 \\
0, & n = 1 
\end{cases}
\]

\[
f(n) = 1 + 2f(n/2) \\
= 1 + 2 \left(1 + 2f(n/4)\right) \\
= 1 + 2 + 4f(n/4) \\
= 1 + 2 + 4 + \ldots + 2 + 2f(n/2^k) \\
\text{Geometric sum} \\
= 2^k - 1 \\
= n - 1
\]

$\text{f}(1) = 0$
4. (a) Use repeated substitution to find the solution of the following recurrence, where

\[ T(n) = \begin{cases} 
T(n/2) + n, & n \geq 2 \\
1, & n = 1 
\end{cases} \]

\[ n = 2^k. \]

\[ T(n) = n + T\left(\frac{m}{2}\right) \]
\[ = n + \left( \frac{m}{2} + T\left(\frac{m}{4}\right) \right) \]
\[ = n + \frac{m}{2} + T\left(\frac{m}{4}\right) \]
\[ = n + \frac{m}{2} + \frac{m}{4} + \cdots + 2 + T(1) \]
\[ = 2n - 1 \quad (\text{Geom Sum}) \]

(b) Prove by induction that the solution of the following recurrence is \( T(n) = An^2 + Bn \),
and determine the constants \( A \) and \( B \). Assume \( n = 2^k \).

\[ T(n) = \begin{cases} 
4T(n/2) + n, & n \geq 2 \\
1, & n = 1 
\end{cases} \]

Base, \( n = 1 \):
\[ T(1) = 1 \]
\[ = A + B \]
\[ \boxed{A + B = 1} \]

For any \( n \geq 2 \), Suppose
\[ T\left(\frac{m}{2}\right) = A\left(\frac{m}{2}\right)^2 + B\left(\frac{m}{2}\right) \]
Then,
\[ T(n) = 4T\left(\frac{m}{2}\right) + n \]
\[ = 4\left( A\frac{n^2}{4} + B\frac{n}{2} \right) + n \]
\[ = An^2 + (2B+1)n \]
\[ \text{Need} \]
\[ = An^2 + Bn \quad \text{Need} \]
\[ 2B + 1 = B \]

\[ \boxed{A + B = 1} \]
\[ \boxed{2B + 1 = B} \]
\[ \implies B = -1, \quad A = 2 \quad \implies T(n) = 2n^2 - n \]
5. We have four sorted lists, each with \( n/4 \) elements. (Elements are real-valued.) We want to merge these lists into a single sorted list of \( n \) elements.

(a) First consider the following naive approach.

- Merge the first and second list into a sorted list of \( 2n/4 \) elements,
- Merge the result with the third list to get a sorted list of \( 3n/4 \) elements,
- Merge the result with the fourth list.

Analyze the worst-case number of key comparisons. (Find the exact worst-case number, not order of it.)

\[
\text{Total} = (2 \frac{n}{4} - 1) + (3 \frac{n}{4} - 1) + (4 \frac{n}{4} - 1)
\]

\[
= \frac{9}{4} n - 3
\]

\[
= 2.25 \ n - 3
\]

(b) Describe a more efficient algorithm for this problem based on a divide-and-conquer technique. Use a diagram to help explain your algorithm. Analyze the worst-case number of key comparisons. (Again, find the exact worst-case number, not order of it.)

\[
\text{Total} = \left( \frac{n}{2} - 1 \right) + \left( \frac{n}{2} - 1 \right) + (n-1)
\]

\[
= 2n - 3
\]