1. Prove the following polynomial is $\Theta(n^3)$.

$$P(n) = 2n^3 - 5n^2 + 10n - 20$$

(a) Prove $O(n^3)$:

(b) Prove $\Omega(n^3)$:
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 $O(\ )$.

```plaintext
X = 0;
for i = 1 to 2n - 1
  for j = i to 5n - i
    X = X + 1;
```
3. Consider the following divide-and-conquer algorithm (recursive function). Parameter $i$ is the starting index of the array, and $n$ is the number of elements. The initial call is $\text{COMPUTE}(A, 0, n)$.

```c
int COMPUTE (int A[], int i, int n) {
    if ($n == 1$) return $A[i]$;
    $n1 = \lfloor n/2 \rfloor$;  //Length of first half of array
    $n2 = n - n1$;       //Length of second half of array
    $C1 = \text{COMPUTE} (A, i, n1)$;
    $C2 = \text{COMPUTE} (A, i + n1, n2)$;
    return ($C1 * C2$)
}
```

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

(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.

(c) Now consider the general case where $n$ is any integer. Write a recurrence for $f(n)$. Guess the solution and prove it correct by induction.
4. (a) Use Master Theorem to obtain the solution form for the following recurrence. Then find the exact solution. (Assume \( n \) is a power of 2.)

\[
T(n) = \begin{cases} 
8T(n/2) + n, & n \geq 2 \\
1, & n = 1.
\end{cases}
\]

(b) Use repeated substitution to find the solution of the following recurrence. (Assume \( n \) is a power of 2.)

\[
T(n) = \begin{cases} 
8T(n/2) + n^2, & n \geq 2 \\
1, & n = 1.
\end{cases}
\]
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.)

(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.)