1. (a) Prove that \( P(n) = 5n^3 - 20n^2 - 100n - 50 \) is \( \Omega(n^3) \).

\[
\begin{array}{|c|}
\hline
\text{GRADE} & /20 \\
\hline
1 & \\
\hline
2 & \\
\hline
3 & \\
\hline
4 & \\
\hline
5 & \\
\hline
\text{SUM} & /100 \\
\hline
\end{array}
\]

(b) Consider the following pseudo-code. (The code does not do anything useful except to test you!)

\[
\text{for } j = 1 \text{ to } n \\
\quad \text{for } k = 1 \text{ to } j \\
\quad \quad x = x \times k
\]

Find the exact number of times the innermost statement (multiplication) is executed. Also, express the time complexity \( T(n) \) of this code in \( O() \) form.
2. (a) Find the exact solution of the following recurrence using any valid method. (Assume \( n = 2^k \).)
\[
T(1) = 1, \\
T(n) = 4T(n/2) + 2n, \quad n > 1.
\]

(b) Prove by induction that the solution of the following recurrence is \( T(n) = An + B \log n + C \), and find the constants \( A, B, C \). (Assume \( n = 2^k \).)
\[
T(1) = 1, \\
T(n) = 2T(n/2) + \log n, \quad n > 1.
\]
3. The following function finds the min and max of an array of \( n \) elements, where \( n \) is any integer.

```c
void MINMAX(dtype A[], int n, dtype& min, dtype& max){
    dtype min1,max1,min2,max2;
    if (n==1) min=max=A[0];
    else if (n==2){
        if (A[0] <= A[1]) {min=A[0]; max=A[1];}
        else {max=A[0]; min=A[1];} }
    else{  // case of \( n>2 \):
        MINMAX(A,n-2,min1,max1);
        else {min2=A[n-1]; max2=A[n-2];}
        if (min1 <= min2) min=min1; else min=min2;
        if (max1 <= max2) max=max2; else max=max1;
    }
}
```

(a) Write a recurrence for the number of *key-comparisons*, \( f(n) \).

(b) Use repeated substitution to find the exact solution.
4. Consider a $2^n \times 2^n$ board, with one of its four quadrants missing. That is, the board consists of only three quadrants, each of size $2^{n-1} \times 2^{n-1}$. Let’s call such a board a quad-deficient board. For $n = 1$, such a board becomes an L-shape 3-cell piece called a tromino.

(a) Use a divide-and-conquer technique to prove by induction that a quad-deficient board of size $2^n \times 2^n$, $n \geq 1$ can always be covered using some number of trominos. (By covering we mean that every cell of the board must be covered by a tromino piece, and the pieces must not overlap.) Use diagram to help describing your algorithm and proof.

(b) Illustrate the covering produced by the algorithm for $n = 3$ (that is, $2^3 \times 2^3$ board).
5. (a) Insert the following sequence of elements into a Binary-Search-Tree (BST), starting with an empty tree.

   50, 30, 80, 25, 18, 65, 69, 98

(b) Delete element 90 in the following BST. (First complete the picture by drawing a line from each node to its children. Be careful as you draw the lines to get a valid BST.)

```
    75
   /\   \\
  60   90
 /\    \\
55  70  80  120
 /\  \  \  \
52 58 65 78 82 110 150
 /\  \  \  \
63 67 87 100
 /\  \  \
66 68 85 105
 /\  \  \  \
83 86 102 107
```