1. Function \( f(n) = n^2 + 5n - 6 \) when \( n \leq 50 \) and \( f(n) = n\log n - 7n \) for \( n > 50 \). Write \( f(n) \) in asymptotic notation in the simplest possible form.

2. Prove or disprove carefully: \( O(n) + \Omega(n) = O(n) \) and \( O(n) + \Omega(n) = \Omega(n) \).

3. Draw the recursion tree for \( T(n) \) when \( n = 6 \) where \( T(n) = T(n/2) + T(n - 1) \) if \( n \) is even and \( T(n) = T(n - 1) + T(n - 2) \) if \( n \) is odd. Assume \( T(1) = 1 \).

4. Solve the recursion: \( T(n) = T(n/3) + T(2n/3) + O(n) \) for \( n > 3 \). You may assume \( n \) is a power of 3 and that \( T(n) = \Theta(1) \) for \( n \leq 3 \).

5. Derive best possible upper and lower bounds for \( \sum_{i=1}^{n} 2^i / \sqrt{i} \).

6. Analyze the worst-case time complexity of the following algorithm in detail: Input is an array of some size, say \( n \), containing numbers. You may assume \( n \) is a power of 4.

Algorithm Weird(Array A)
If A is of size less than 5, then call Merge-sort(A)
otherwise {
    Divide A into four equal parts B, C, D and E.
    Weird(B)
    Weird(C)
    Weird(D)
    Weird(E)
    F = Merge(B,C)
    G = Merge(D,E)
    A = Merge(F,G)
}
Output A