1: [10 points] Briefly describe when it is not efficient to apply the divide and conquer approach. Just describe one case.

2: [10 points] True/False Questions. Just say T(ue) or F(alse) for each of the following questions.

(a) [2 points] $9,999,999n^3 \in \Theta(n^3)$.
(b) [2 points] $n^2 \in o(n^3)$.
(c) [2 points] $n^4 \in \omega(n^4)$.
(d) [2 points] $n \lg n \in o(n)$.
(e) [2 points] $n^2 \in O(n^3)$.

3: [10 points] Write the recurrence equations for merge sort and solve the equations to compute the time complexity. Assume that $n = 2^k$ where $k$ is a positive integer.
4: [20 points] Prove that \( a_1 n + a_2 n^2 + a_3 n^3 + \cdots + a_{k-1} n^{k-1} + a_k n^k = \Theta(n^k) \) where \( a_i > 0 \) (1 \( \leq \) \( i \leq \) \( k \)).

**Hint:** Consider \( \max(a_1, a_2, \cdots, a_k) \) for \( O \) (big Oh) and \( \min(a_1, a_2, \cdots, a_k) \) for \( \Omega \).

5: [10 points] Sort the following list in increasing order of asymptotic time complexity:

- \( 2^n \)
- \( 10^7 n \log n \)
- \( n^2 - 3n \)
- \( 5n^3 \)
- \( n! \)
- \( n^6 - 10^9 n^4 \)

6: [10 points] Develop a divide-and-conquer algorithm that searches a sorted list of \( n \) integers to find an arbitrary integer \( x \) in the list. Especially, write a ternary search algorithm that divides a list into three smaller sublists of equal size where \( n = 3^k \) for \( k > 0 \).
7: [20 points] Answer the following questions.

(a) [10 points] Define the principle of optimality. To solve a problem via dynamic programming, the principle of optimality should hold for the problem. Why?

(b) [10 points] Briefly show that the principle of optimality holds for the minimum spanning tree problem.

8: [10 points] Briefly describe how dynamic programming algorithms can reduce the time complexity to solve certain problems compared to recursive divide conquer algorithms.
9: Extra Credit [10 points]. Construct the optimal binary search tree for three keys A, B, and C with search probabilities 0.2, 0.3, and 0.5.