Data Structure (50%)

1. (6%) Derive the following functions \( f(n) \) (in the simplest form).
   
   (a) \( a_m n^m + \ldots + a_1 n + a_0 = \Theta(f(n)) \), (Note: all \( a_i \) are positive constants.)
   
   (b) \( \sum_{i=2}^{n} \log n = \Theta(f(n)) \),
   
   (c) \( \sum_{i=1}^{n} (n-i) \log n = \Theta(f(n)) \).

2. (7%) Insert a sequence of keys (61 23 92 33 54 77 20 11 31 47) into a data structure which has no keys initially. Depict the data structure after these insertions, if it is an AVL tree.

3. (12%) Design a data structure, called vector, which requires the following operations:
   
   (1) ElementAt(i): returns the element at the position \( i \);
   
   (2) AddElement(e): adds the element \( e \) at the end of the vector.

   Please design the data structure, vector, such that
   
   (a) the time complexity for the first operation is always \( O(1) \),
   
   (b) the amortized time complexity for the second operation is \( O(1) \),

   (assume each memory allocation (e.g., malloc) function call takes \( O(1) \))
   
   (c) the space time complexity is \( O(n) \) if there are \( n \) elements in the vector.

   Please also explain why (a)-(c) are satisfied in your data structure. (Hint: consider the data structure java.util.Vector in Java’s utility library.)

4. (8%) Write a program that performs the search of a 2-3 tree. (you can choose any well known programming language)

5. (9%) A hash table using a uniform hashing function \( h \) has a loading density \( d = n/h \), where \( n \) is the number of records to be hashed and \( h \) is the number of buckets (number of head nodes). Let \( S_n \) denote the expected number of identifier comparisons needed to locate a randomly chosen \( X_i \), \( 1 \leq i \leq n \). Assuming chaining is used, show that \( S_n \approx 1 + d/2 \).

6. (8%) Show that the number of spanning trees is larger than \( 2^m \) for a complete graph with \( m \) nodes.
Algorithm (50%)

7. The following recurrence relations describe the running times of some recursive algorithms. What are the asymptotic running times of these algorithms? (No proofs needed) (8 points)

1. \( T(n) = 9 \ T(n/2) + O(n^3), \ T(1) = 1 \)

2. \( T(n) = 9 \ T(n/3) + O(n^2), \ T(1) = 1 \)

3. \( T(n) = 5 \ T(n-1) - 6 \ T(n-2), \ T(1) = 5, \ T(2) = 13 \)

8. Use dynamic programming approach to find the minimal number of edit steps (insertion, deletion, or replacement) used to change the string \( A = acbabca \) (7 chars) to the string \( B = babcbac \) (7 chars). (8 points)

1. How many edit steps are used? (You should attach a tableau in which you get the number.)

2. Describe how those edit steps are applied in changing \( A \) to \( B \).

9. The diameter of a tree \( T = (V, E) \) is the largest of all shortest-path distance in the tree. Design a linear time algorithm to compute the diameter of a tree. (Don't write the detailed code of the algorithm. Instead, you just describe your design idea and explain why your algorithm is linear.) (8 points)

10. What is the clique problem? Prove that the clique problem is \( NP \)-complete. (10 points)

11. Find the product \( P(x) \cdot Q(x) \), by hand, using the divide-and-conquer polynomial multiplication algorithm.

\[
P(X) = x + 2x^2 + 3x^3 + 4x^4
\]

\[
Q(X) = 4x + 3x^2 + 2x^3 + x^4
\]

How many operations are required overall? (8 points)

12. State the algorithm of the Graham's scan used for convex hull construction. (8 points)