1. (18%) Consider a “binary search tree”, T. Answer TRUE or FALSE to each of the following statements.
   (a) The root of T has the largest key.
   (b) T is a complete binary tree.
   (c) If there are \( n \) nodes in T, among those \( \lfloor n/2 \rfloor \) are leaves (terminal nodes).
   (d) Root is at level 0 (\( \text{level}(\text{root})=0 \)), and the level of node \( p \) is defined to be \( \text{level}(\text{parent}(p))+1 \) where \( \text{parent}(p) \) denotes the parent of \( p \). Number of nodes in level \( k \) is \( 2^k \).
   (e) The height of a tree is defined to be the largest level of any node in the tree.
       The height of a complete binary search tree is \( O(\log n) \).
   (f) The height of any binary search tree has height \( \Omega(\log n) \).

2. (7%) When storing data of a tree, we usually convert the tree to a binary tree (left child-right sibling) representation. Why do we prefer to have a binary tree representation of a tree that has arbitrary degree?

3. (10%) Array is a data structure that supports random access an entry. That means \( A[i] \) can be stored or retrieved in constant time. Explain why \( A[i] \) can be accessed in constant time. What if we have a 2-dimensional array, i.e., how can \( A[i][j] \) be accessed in constant time?

4. (15%) Divide and Conquer: We solve a problem by dividing the problem into two sub-problems, solve the two sub-problems, finally we obtain the solution to the original problem by merging the solutions to the sub-problems. Suppose that the time required by dividing and merging is \( cn \), where \( n \) is the problem size. Please give the recurrence of \( T(n) \) that describes the time required for solving the original problem. And then solve \( T(n) \).

5. (25%) Briefly answer the following questions:
   (a) Summarize the distinction between top-down and bottom-up strategies when designing a system.
   (b) What is an entity-relationship diagram? What is the difference between a entity-relationship diagram and a class diagram?
   (c) Explain the MVC design pattern. And give the benefits of the MVC structure.
   (d) What is eXtreme Programming?
   (e) Explain RPC, RMI, and CORBA technologies.

6. (15%) (a) Distinguish buffer and cache. (b) Explain how the digital signature works?
   (c) Explain dead lock with example. List the necessary conditions of dead lock.

7. (10%) Consider the programming languages and language processing.
   (a) List all type of statements supported by an imperative programming language.
   (b) What is a parsing tree? What is an expression tree? Explain them with examples.