1. (8%) Solve the recurrence relation \( T(n) + T(n/4) = 2T(n/2) + \log_2 n \) for the \( \Theta \)-order of \( T(n) \), where \( T(1)=1 \).

2. This problem is about hashing.
   (a) (2%) What is a hashing function?
   (b) (2%) What is a collision?
   (c) (3%) In static hashing, what is the loading density?
   (d) (5%) Define a data structure for the hash table in which the chaining strategy will be implemented.
   (e) (4%) Write the chain\_insert() function of inserting an item into the hash table.

3. Assume that real matrices \( M_1, M_2, \ldots, M_n \) has dimensions \( r_1 \times c_1, r_2 \times c_2, \ldots, r_n \times c_n \), respectively where \( c_i = r_{i+1} \) for \( 1 \leq i \leq n-1 \). We want to compute \( M=M_1M_2\ldots M_n \).
   (a) (3%) What is the number of multiplication of real numbers in computing \( M = (((M_1M_2)M_3)\ldots M_n) \) by multiplying \( M_i \)'s sequentially? (assume that matrices are multiplied directly without employing special techniques).
   (b) (8%) We can reduce the multiplication cost by rearranging the multiplication order of \( M_i \)'s. For example, if \( M_1, M_2 \) and \( M_3 \) are \( 3 \times 5 \), \( 5 \times 2 \), and \( 2 \times 1 \)-dimensional, the multiplication cost of \( (M_1M_2)M_3 \) is higher than that of \( M_1(M_2M_3) \). Write an efficient algorithm to compute the minimum number of multiplications in computing \( M \).

4. (15%) Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Give three reasons for why these protocols invented a new abstract ID(port numbers), instead of using process IDs, which already existed when these protocols were designed.

5. (15%) As a network management protocol of the Internet, SNMP version1 (SNMPv1) is widely used. What are those possible limitations of SNMPv1?
6. Enterprises increasingly provide e-Services via the Internet to generate new revenue or create new efficiencies. A lot of e-Services are provided on the Internet and each e-Service such as Hotel reservation can be offered via several e-Service providers with different criteria (e.g. Five-star Hotel). A database is required to organize the information of e-Services and e-Service providers. Suppose that the data of e-Services is stored as an ordered file with 50000 records on a disk with block size = 1024 bytes. Each record stores the information of an e-Service. File records are of fixed size with record length = 100 bytes.

(a) (2%) What's the number of blocks needed for the data file of e-Services? How many block accesses would be needed to do a binary search on the data file?

(b) (5%) Assume that we have constructed a secondary index on a non-ordering key field of the file that is 15 bytes long and a block pointer that is 10 bytes long. What's the total number of index entries for the file? What's the total number of blocks needed for the index entries? How many block accesses would be needed to search for an e-Service using the secondary index?

7. Virtual stores are selling various product categories on the Internet. Suppose that virtual stores have been simplified and organized as a binary tree structure.

(i) Each node represents one virtual store. A root node H denotes the headquarters of all virtual stores. Each virtual store may sell more than one product category such as computer, CD-title, book and so on.

(ii) A XML document is used to record the information of a virtual store, including the store name, product categories and links to other virtual stores (XML documents).

(a) (4%) Design your XML document to record the information of virtual store.

(b) (6%) Write an algorithm to find those virtual stores selling books, by starting from H. You need to describe how you process the XML documents.
8. Let G represent a process of e-Services executed on the Internet.
   (i) G is organized by nodes and directed links. Each node represents an e-Service. Each directed link
       is a dependency. Dependencies are used to describe the execution order and relationship between
       e-Services within a process.
   (ii) A dependency \( d = <X, Y> \) connects two e-Services X and Y, in which X is the \textit{preceding}
       \textit{e-Service} and Y is the \textit{succeeding e-Service}. The dependency \( d \) is an outgoing dependency of X
       and an incoming dependency of Y. An e-Service may have more than one incoming dependency
       (predecessor) or outgoing dependency (successor).
   (iii) The length of a path is the number of dependencies on the path.
   (iv) E-Service X is said to have a higher order than Y if there is a path of length \( > 0 \) from X to Y, i.e.,
        X proceeds before Y, and their ordering relation is denoted by \( X > Y \).
   (v) G has one starting e-Service S, where \( S > V \), for all other e-Service V in G.
   (vi) A longest path is a path that contains the maximal number of dependencies.
   (vii) There is no cycle in G.

(a) (9%) Use adjacency matrix to represent G, the process of e-Services. Based on the adjacency
     matrix representation, write an algorithm to generate the ordering relations between all e-Services
     in G, i.e., \( \text{ordering}[X,Y] = 1 \), if \( X > Y \); otherwise, \( \text{ordering}[X,Y] = 0 \). Analyze the time
     complexity of your algorithm.

(b) (9%) Use adjacency lists to represent G, the process of e-Services. Based on the adjacency list
     representation, write an algorithm to find the length of the longest path in G. Analyze the time
     complexity of your algorithm.