國立臺北科技大學

九十四學年度資訊工程系碩士班入學考試

離散數學與演算法試題

填准考證號碼

注意事項：

1. 本試題共 10 題，配分共 100 分。
2. 請按順序標明題號作答，不必抄題。
3. 全部答案均須答在答案卷之答案欄內，否則不予計分。

1. Let A and B be two problems. Suppose that we were able to establish the following fact: " if we could solve A in time O(T(n)), then we could solve B in time O(n log n +T(n))." Please mark by T (= True) or F (= False) each of the following statements: (10%)
   (1) If A has an $\Omega(n \log n)$ time lower bound, then B does too.
   (2) If B has an $\Omega(n \log n)$ time lower bound, then A does too.
   (3) If B has an $\Omega(n^2)$ time lower bound, then A does too.
   (4) If $T(n)=T(n/2)+cn$, then $T(n)=O(n \log n)$.
   (5) If $T(n)=T(n/2)+c\log n$, then $T(n)=O(\log^2 n)$.

2. Consider that a server wants to pass a binary tree T to a client by sending a traversal of T.
   (1) Suppose the server sends the tree T in post-order (As shown in Figure 1, the post-order traversal is CDBEA). Can the client re-construct the original tree T uniquely by following the received post-order traversal? In other words, whether the client can have only one tree representation by following the post-order traversal sent by the server? Show your reason. (5%)
   (2) Suppose the server now sends T in preorder. Does the conclusion in (1) still hold? If you think the client can have only one tree representation

by following the preorder traversal sent by the server, please prove it. Otherwise, provide a modified preorder traversal which allows the client to re-construct the original binary tree uniquely. (5%)

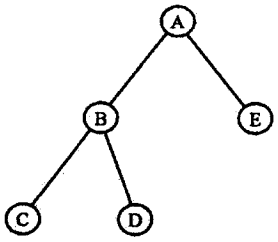

Figure 1: A binary tree T.

3.

(1) How many routes are there from the lower-left corner of an $n \times n$ square grid to the upper-right corner if we are restricted to traveling only to the right or upward and if we are allowed to touch but not go above a diagonal line from the lower-left corner to the upper-right corner? (5%)

(2) We denote the solution in (1) as C_n which is a function of n. Please show that C_n can be presented as the following recurrence relation:

$$(n+2)C_{n+1} = (4n+2)\,C_n, \quad n>0;$$

and $C_0=1$. (5%)

4. Simplify the expressions to equivalent statements that have as few symbols as possible.

(1) $(q \vee (\neg p \wedge q)) \wedge \neg p$ (2%)

(2) $\neg((p \wedge \neg q) \vee \neg(r \wedge q))$ (3%)

5. Show that the language $L = \{a^k \mid k = i^2, i \geq 1\}$ is not a finite state language. (10%)

6. Let $\boldsymbol{B}$ be the set of all positive integer divisors of 30: $\boldsymbol{B}$ = {1, 2, 3, 5, 6, 10, 15, 30}. For any $x, y \in \boldsymbol{B}$, define $x + y = [x, y]$, the l.c.m. of x, y; $x \cdot y = (x, y)$, the g.c.d. of x, y ; and, $\neg x = 30/x$. With 1 as the **0** element and 30 as the **1** element, please show that $(\boldsymbol{B}, +, \cdot, \neg, \mathbf{0}, \mathbf{1})$ is a Boolean algebra. (10%)

7. Solve the following recurrence relations by the method of generating functions:

$$a_{n+2} - 2a_{n+1} + a_n = n,\ n \geq 0,\ a_0 = 1,\ a_1 = 2. \qquad (5\%)$$

注意：背面尚有試題

8. The *Insertion Sort* sorts k numbers in two steps: sort the first $k-1$ numbers, and then insert the kth number. Determine, in the worst case, the *minimum number of comparisons* used by the Insertion Sort in sorting n numbers when
   (1) a *linked-list* is used to store the numbers. (5%)
   (2) an *array* is used to store the numbers. (5%)

9. If matrices A and B are of dimensions $p \times q$ and $q \times r$, respectively, then computing $A \times B$ in the straightforward manner takes $p \times q \times r$ multiplications. Further, matrix multiplication is associative, i.e.,

$$A \times B \times C = (A \times B) \times C = A \times (B \times C).$$

   Suppose we are given a sequence of matrices A_1, A_2, ..., A_n, where the dimensions of A_i are $d_{i-1} \times d_i$. Let $M(i, j)$ be the *minimum number of multiplications* needed to compute $A_i \times ... \times A_j$ for $1 \le i \le j \le n$. An idea for computing $M(i, j)$ is as follows: *Find the number k such that the number of multiplications in computing $(A_i \times ... \times A_k) \times (A_{k+1} ... \times A_j)$ is minimum.*
   (1) Write the recurrence relation for $M(i, j)$. (10%)
   (2) Compute $M(i, j), 1 \le i \le j \le 4$ for the multiplication $A_1 \times A_2 \times A_3 \times A_4$, where $d_0 = 30, d_1 = 1, d_2 = 40, d_3 = 10$, and $d_4 = 25$. (5%)

10. In object-oriented (OO) software development, program code for classes is written only after a *class diagram* is drawn (see Figure 2). In a class diagram, *classes* and *dependency relationships between classes* are two of the most important information. Class A depends on class B if, within one or more member functions of class A, a member function of class B is called. In a class diagram, this is shown by an arrow from class A to class B. For example, In Figure 2, class `ProductCatalog` depends on class `ProductSpecification`; Class `Sale` depends on Class `Payment`. A dependency is *circular* if it is possible to return to a class by following a sequence of dependency arrows.

    Note that if good OO design practice is followed, a class diagram will

contain no circular dependency between classes. This allows us to adopt the following ordering strategy in implementing the classes: *a class is implemented only after all the classes it is dependent are implemented.* In Figure 2, the circled numbers show a possible order for implementing the classes according to this strategy.

Assume that there is no circular dependency in a class diagram that contains M classes and N dependencies.

(1) Outline a linear time algorithm to determine the order in implementing the M classes that conforms to the above ordering strategy. (10%)

(2) Determine the time complexity of your algorithm. (5%)

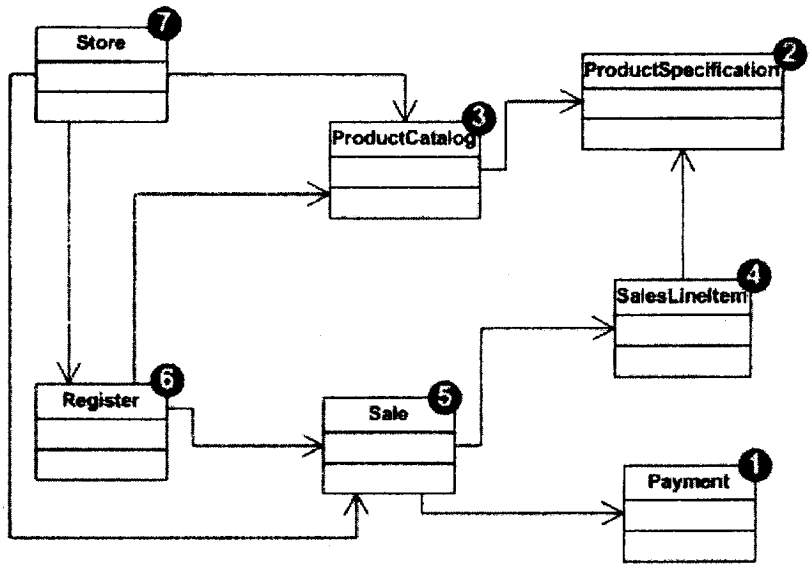

Figure 2: A class diagram