中原大學 96 學年度碩士班入學考試

96/03/25 14:00~15:30 資訊工程學系

誠實是我們珍視的美德，
我們喜愛「拒絕作弊，堅守正直」的你！

科目：資料結構與演算法

□可使用計算機，惟僅限不具可程式及多重記憶者　■不可使用計算機

※ When giving answers to questions (e.g., Question 8), try to state your answers in Chinese. If you write in English, please be aware that the grader will not try to guess what you are trying to say (i.e., your English will be graded "as it is").

※ Please write your answers in the right order, and be sure to correctly number all your answers.

1. Algorithms are designed to solve many problems, but may be designed quite differently to solve the same problem. For example, given a sum of squares: $\sum_{i=1}^{n} k^2$, we would like to design algorithms to compute the sum given the input n ($n \geq 1$) using C/C++. Please determine the missing codes and answer the questions as follows: (**請勿在試題上作答**)

   a. The first algorithm is to simply use the mathematical formula for computing the sum of squares, which can be described by:

```
int SquareSum(int n) {
    return _____;  (2%)
};
```

   b. The second algorithm is to use the iterative approach, such as:

```
int SquareSum(int n) {
    int Sum = 0;
    for(i = 1; i ≤ n ; i++) {

        _____;  (2%)
    }
    return Sum;
}
```

   c. The third algorithm is to use the recursive approach, such as:

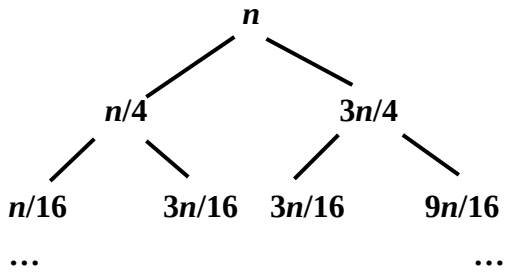
```
int SquareSum(int n) {
    if(n == 1) return 1;
    else    return _____;  (2%)
}
```

   d. Which of the three algorithms above is the least efficient? (1%)

   e. Which of the three algorithms requires the use of stacks? (1%)

2. A recurrence is an equation that describes a function in terms of its value on smaller inputs. The recurrence is generally defined as $T(n)$ in terms of n that can be solved to analyze the running time of a recursive algorithm. Please answer the following questions:

   a. Given the following recursion-tree, determine the corresponding recurrence and give a tight asymptotic bound for the recurrence. (2%)



   b. The master theorem provides a useful method for solving recurrences which can be described as follows:

   Let $a \geq 1$ and $b > 1$ be constants, and let $f(n)$ be a function:

$$T(n) = aT(n / b) + f(n)$$

   Then $T(n)$ can be bounded asymptotically as follows.

   (1) If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = ?$ (2%)

   (2) If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = ?$ (2%)

   (3) If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$ and if $af(n / b) \leq c f(n)$, then $T(n) = ?$ (2%)

   c. Use the master method to solve the following recurrence: (2%)

$$T(n) = 3T(n / 3) + n^2$$

中原大學 96 學年度碩士班入學考試

96/03/25 14:00~15:30 資訊工程學系

誠實是我們珍視的美德，
我們喜愛「拒絕作弊，堅守正直」的你！

科目：資料結構與演算法 （共 6 頁第 3 頁）

□可使用計算機，惟僅限不具可程式及多重記憶者 ■不可使用計算機

3. The 0-1 knapsack problem is an optimization problem that often appears in the discussion of computer algorithms. A thief robbing a store finds *n* items. If the thief wants to take as valuable a load as possible, but he can only carry at most *W* pounds in his knapsack. Each item must either be taken or left behind (i.e., $x_i = 0$ or 1, $i = 1 \ldots n$) and the *i*-th item has a value v_i and a weight w_i. Please answer the following questions:
   a. Mathematically, the 0-1 knapsack problem can be formulated as (please fill the empty): (2%) (**請勿在試題上作答**)

      Maximize: _____

      Subject to: _____

      $x_i = 0$ or 1, $i = 1 \ldots n$

   b. Suppose the brute-force approach is used, i.e., list all possible loads and then determine if the load is the most valuable and ≤ *W*, what's the running time in Θ-notation? (2%)
   c. Which of the following strategy is the most appropriate for designing an algorithm for solving such a problem, when both an optimal solution and efficiency are desired?

      (1) Brute-Force; (2) Recurrence; (3) Backtracking; (4) Dynamic Programming; (5) Greedy Algorithm. (2%)
   d. Suppose there are 8 items, and their weights and values are list below, which item(s) should the thief take if he makes the *greedy* choice and can carry at most 50 pounds ($W = 50$): (2%)

Item No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Value	20	5	15	10	10	15	10	50
Weight	10	20	10	15	10	5	20	20

Note: The unit of value is in US$, and the unit of weight is in pound.

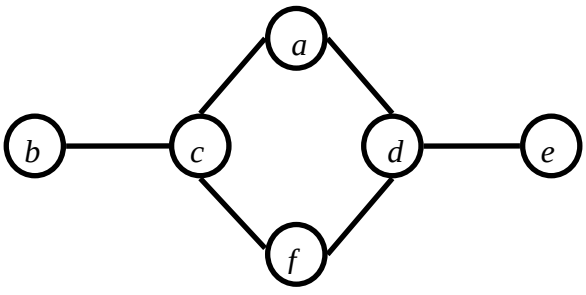
4. In graph theory, a complete graph is a graph in which every pairs of vertices are connected by an edge. The complete graph with n vertices can be denoted as K_n (Two examples of complete graphs are shown below).

K_3　　K_4

Please answer the following questions:

a. Which graph representation is more appropriate for representing a complete graph, adjacency lists or adjacency matrix? Briefly explain why. (2%)
b. Determine the total number of edges for K_{10}. (2%)
c. Determine the total number of spanning trees for K_6. (2%)
d. What's an Euler cycle for a graph? Is there an Euler cycle for K_{10}? (2%)
e. What's a Hamiltonian cycle for a graph? Is there a Hamiltonian cycle for K_{10}? (2%)
f. Determine the *clique* of the complete graph K_5. (2%)
g. Determine if K_4 is bipartite? (2%)

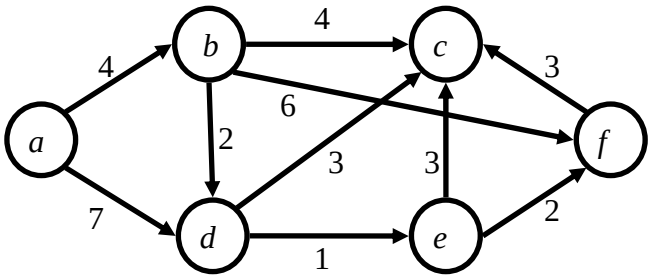
5. Given the undirected graph,



a. Find the Breadth-First Search (BFS) sequences (starting from the vertex *a*). (3%)
b. Find the Depth-First Search (DFS) sequences (starting from the vertex *a*). (3%)

(Note: At any moment, if more than one vertex can be visited next, always select the one in alphabetic order.)

6. Dijkstra's algorithm is often used to solve the single-source shortest-paths problem on a weighted, directed graph $G(V, E)$ with vertex set V and edge set E. Please answer the following questions:
   a. Which of the following data structure is the most appropriate for storing the distance from each vertex to the source vertex during implementation of the algorithm? (2%)
      (1) Stack; (2) Queue; (3) Priority Queue; (4) Binary Search Tree; (5) Hash Table.
   b. Dijkstra maintains a set S of vertices and add vertices in order to the set. Given the following directed graph, in what order are the vertices added if the source vertex is the vertex a? (2%)



7. Suppose a binary tree is such that each node (be it an internal node or a leaf node) contains a letter (e.g., 'a', 'b', 'c', etc.). Please answer the following questions. (15 %)
   a. Draw a binary tree. The binary tree must be such that when we in-order traverse this tree (and print the letters that are "contained" in the nodes), we get the following print-result:
      abcdefg
   b. Draw a binary tree. The binary tree must be such that when we pre-order traverse this tree (and print the letters that are "contained" in the nodes), we get the following print-result:
      abcdefg
   c. Draw a binary tree. The binary tree must be such that when we post-order traverse this tree (and print the letters that are "contained" in the nodes), we get the following print-result:
      abcdefg

中原大學 96 學年度碩士班入學考試

96/03/25 14:00~15:30 資訊工程學系

誠實是我們珍視的美德，
我們喜愛「拒絕作弊，堅守正直」的你！

科目：資料結構與演算法

□可使用計算機，惟僅限不具可程式及多重記憶者　■不可使用計算機

8. (a) Can we always perform a binary search on an array in order to find the information we want? (b) Whether your answer to the above question is 'yes' or 'no', please also state the reason why. (10%)

9. (a) Describe a programming task in which we should use one or more **stacks** when we write a program to solve the problem. Be sure to explain why we SHOULD use stacks. (8%) (b) Describe a programming task in which we should use one or more **queues** when we write a program to solve the problem. Be sure to explain why we SHOULD use queues. (7%)

10. It is often useful to read in data and put the information that we read in on a binary search tree (BST). The reason for doing so is that when later we want to search for the information we want, searching on a BST can be relatively fast. The problem, however, is that if we just read in a piece of data and directly add it to a BST, the BST may "grow" in ways that are not so desirable. Suppose we build BST tree in a "direct way" (by always directly adding new nodes onto the tree, without making any further changes to the existing "structure" of the tree). (a) What is the worst case? (b) Explain your answer (i.e., why is it the worst case). (10 %)