國立臺灣師範大學 100 學年度碩士班招生考試試題

科目：**軟體基礎**　　　　適用系所：**資訊工程學系**

注意：1.本試題共 7 頁，請依序在答案卷上作答，並標明題號，不必抄題。2.答案必須寫在指定作答區內，否則不予計分。

1. 選擇題(單選題 1.1~1.13, 每題 2 分, 請將答案寫在答案紙上並標明題目編號)

1.1 What is the expected number of operations needed to loop through all the edges adjacent to a particular vertex given an adjacency matrix representation of the graph? (Assume m vertices are in the graph and n edges adjacent to the desired node.)

A) O(m)　　B) O(n)　　C) $O(m^2)$　　D) $O(n^2)$

1.2 Suppose you have a game with 5 coins in a row and each coin can be head or tail. You want to represent the states of the 5 coins with a graph. How many numbers of vertices might you expect to find in the state graph?

A) 7　　B) 10　　C) 25　　D) 32

1.3 Suppose we are sorting an array of eight integers using the quicksort algorithm, and we have just finished the first partitioning with the array looking like this:

3 5 1 9 8 12 11 10

Which of the following statement is correct?

A) The pivot could be either 8 or 9.
B) The pivot could be he 8, but it is not possible 9.
C) The pivot is not possible 8, but it could be 9.
D) Neither 8 nor 9 is the pivot.

1.4 What feature of heaps allows them to be efficiently implemented using an array?

A) Heaps are binary search trees.　　B) Heaps are complete binary trees.
C) Heaps are full binary trees.　　D) Heaps contain only integer data.

1.5 A queue is implemented with a linked list, where a front node and a rear node are kept track with two reference variables. Which of these reference variables will change during an insertion into an EMPTY queue?

A) Neither changes　　B) Only front changes.
C) Only rear changes.　　D) Both change.

1.6 What is the minimum number of pointers modified for deletion of any node from the circular doubly linked list?

A) 2 B) 3 C) 4 D) 5

1.7 Given an infix expression: 4+3*(6*3-12). Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. What is the maximum number of symbols that are in the stack at the same time during the conversion of this expression?

A) 2 B) 3 C) 4 D) 5

1.8 What is the worst-case time complexity of searching an element in a binary search tree with n nodes?

A) O(1) B) O(n) C) O(n log n) D) $O(n^2)$

1.9 What is the exact number of *nodes and edges in a spanning tree* for a graph with n nodes and e edges?

A) (n-1) nodes, e edges B) n nodes, (e-1) edges
C) n nodes, (n-1) edges D) n nodes, n edges

1.10 Give the following list of numbers: 30, 55, 36, 50, 33, 19, 61 which are stored in array tree[1], tree[2], …, tree[7]. In the heap sort algorithm, you have to perform an adjust procedure to construct a maximum heap. What is the number in tree[3] of the maximum heap?

A) 30 B) 33 C) 36 D) 50

1.11 According to the result of the constructed maximum heap in the previous question, tree[1] and tree[7] is interchanged and followed the readjust procedure to recreate a max heap for tree[1] to tree[6]. What will be the position of number 30 in the array after performing the readjust procedure?

A) Tree[2] B) Tree[3] C) Tree[4] D) Tree[5]

1.12 Let T be a binary tree with only leaf nodes and degree-2 nodes. Assume that the number of leaf nodes is 51. What is the number of degree-2 nodes?

A) 25 B) 26 C) 50 D) 51

1.13 Given the set of equivalence pairs: {(a,b), (e,f), (c,b), (x,a), (y,d), (d,f), (c, z)}. How many equivalent classes will be derived?

A) 2 B) 3 C) 4 D) 5

2. (4 分) The given figure uses an array to simulate a linked-list (first column stores data and second column stores the position of the next data). Based on the figure, please answer the following questions:

node	data	next
[0]	66	-1
[1]	25	-1
[2]	9	0
[3]	33	8
[4]	11	1
[5]	10	9
[6]	7	2
[7]	18	4
[8]	21	6
[9]	48	7

(a) The first element of the First list is stored in node[3].data. How many data elements are in the First list in total? (2 分)

(b) There is a second list also stored in the array. Where is the first element stored in the array? (2 分)

3. (4 分) Given an array of ten integers: 5 3 4 9 1 7 0 2 6 8

We would like to sort the integers from small to large.

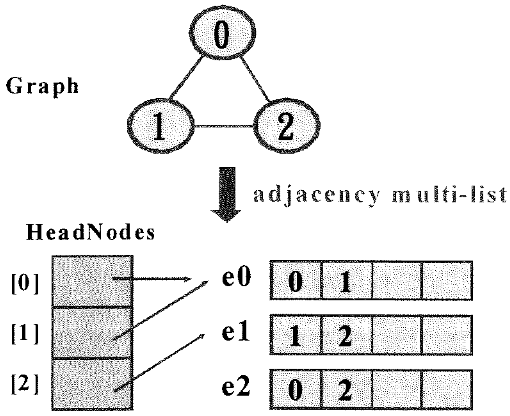
(a) What is the result in the array after executing the FIRST 3 iterations of the main loop in the **selection sort** algorithm. (2 分)

(b) What is the result in the array after executing the FIRST 3 iterations of the main loop in the **insertion sort** algorithm. (2 分)

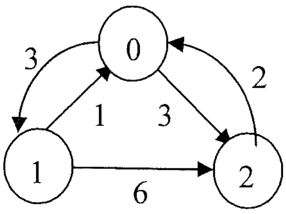
4. (5 分) There is an undirected graph G. The adjacency matrix of G is shown below. However, some entries in the matrix are missing. It is known that G is a connected graph and vertices 0, 2, and 3 are articulation points. Besides, "032154" is a result of DFS traversal on graph G. Please fill in the missing entries.

	0	1	2	3	4	5
0	0	0	0	$<a>$	1	0
1	0	0	1	0	0	0
2	0	$<b>$	0	1	0	0
3	$<c>$	0	1	0	0	1
4	1	0	0	0	0	$<d>$
5	0	0	0	1	$<e>$	0

5. (5 分) Please complete the following *adjacency multi-list* structure for the graph shown below. Each link field should be filled in by an edge identifier in {e0, e1, e2} or null.



6. (6 分) Consider all-pairs shortest-path problem for the graph shown below.
   (a) Please construct the adjacency matrix A^{-1}, in which A[i][j] represents a directed edge weight. (2 分)
   (b) Let A^k[i][j] be the shortest path from i to j without going through any intermediate vertex of index greater than *k*. Please derive the matrix A^1. (4 分)



7. (25 分) In 1969, Strassen published a divide-and-conquer algorithm for multiplying two n×n matrices A and B whose time complexity is better than that of "ordinary" method in terms of both multiplications and additions/subtractions. The following illustrates his method. Suppose we want the product C of two n×n matrices, A and B. Assuming that n is a power of 2, first we divide each of the matrices A, B and C into four submatrices as shown follows:

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Strassen determined that if we let

$$\begin{aligned}
m_1 &= (a_{11} + a_{22})(b_{11} + b_{22}) \\
m_2 &= (a_{21} + a_{22})\, b_{11} \\
m_3 &= a_{11}(b_{12} - b_{22}) \\
m_4 &= a_{22}(b_{21} - b_{11}) \\
m_5 &= (a_{11} + a_{12})\, b_{22} \\
m_6 &= (a_{21} - a_{11})(b_{11} + b_{12}) \\
m_7 &= (a_{12} - a_{22})(b_{21} + b_{22}),
\end{aligned}$$

the product C is given by

$$C = \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}.$$

(a) Suppose that n=4 and A, B matrices below. Please show how to compute the matrix m_1 in the first step of Strassen's method. (5 分)

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{bmatrix} \quad B = \begin{bmatrix} 8 & 9 & 1 & 2 \\ 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 \end{bmatrix}.$$

(b) How many multiplications would be performed in finding the product of two 4×4 matrices using Strassen's method? (5 分)

(c) Write a recurrence equation for the time complexity of Strassen's algorithm. (5 分)

(d) Solve the recurrence equation in (c). (5 分)

(e) Using the "ordinary" method for multiplying 4×4 matrices, how many multiplications of numbers will be needed? (5 分)

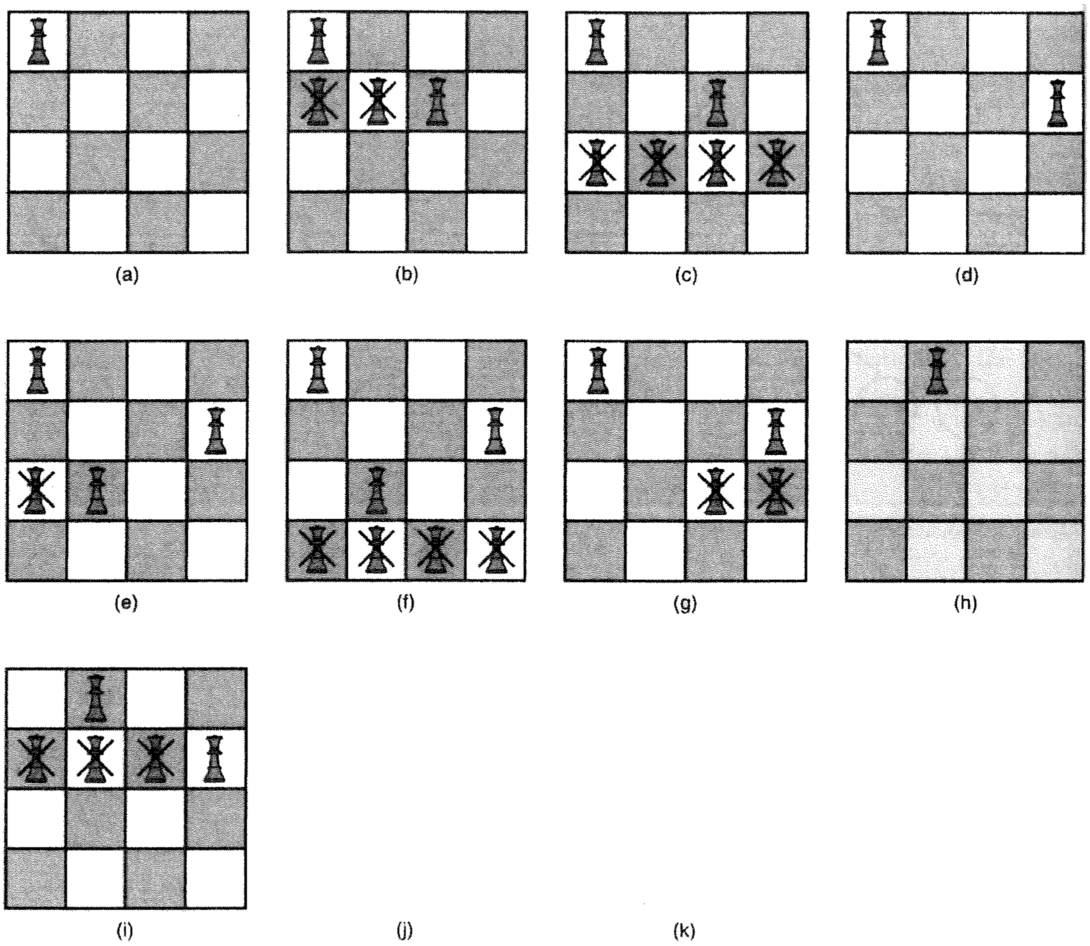
8. (10 分) Please design an O(|V|+|E|)-time algorithm that finds a longest simple path (no repeated vertices) in a weighted directed acyclic graph G=(V, E). Assume that the weight of a path is the sum of the weights of its edges.

9. (15 分) Backtracking is used to solve problems in which a sequence of objects is chosen from a specified set so that the sequence satisfies some criterion. The classic example of the use of backtracking is in the n-Queens problem. The goal in this problem is to position n queens on an n×n chessboard so that no two queens threaten each other. That is, no two queens may be in the same row, column, or diagonal. The sequence in this problem is the n positions in which the queens are placed, the set for each choice is the n2 possible positions on the chessboard, and the criterion is that no two queens can threaten each other.

(i) For the sake of brevity, we illustrate backtracking using the instance when n = 4. We can immediately simplify matters by realizing that no two queens can be in the same row. The instance can then be solved by sequentially assigning each queen a different row and checking which column combinations yield solutions. To summarize, backtracking consists of doing a search of each column for each row, checking whether any column is promising, and, if all columns are nonpromising, backtracking to the previous row. The following figure shows the actual chessboard positions that are tried when backtracking is used to solve the instance of the 4-Queens problem. Each nonpromising position is marked with a cross. A queen is marked with a cross in the figure if it is nonpromising. For example, if we have already placed queen 1 in column 1, then we cannot place queen 2 in either column 1 or column 2. In the figure, the last two steps are left blank. Please draw the two actual chessboard positions (j) and (k). (5 分)

(ii) According to the above process, finally we will get two solutions. Please show the two solutions. (5 分)

(iii) The above backtracking algorithm can be formulated as a depth-first search of the state space tree with backtracking. What is the state space tree of the 4-Queens problem? (5 分)


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)