※ 考生請注意：本試題 □可 ☑不可 使用計算機

※ 本試題共計 4 頁。作答時可不抄題，但請務必將各題號標註清楚。

1. (15%) **True** or **False** with explanation: Each question is **3 points**. If your answer is **False**, please **give a brief explanation** to justify your answer. **Fail to do so will only get one point** if your answer is correct. However, if your answer is **True**, no need to provide any explanation.
   (1-1) The result of evaluating the postfix expression "8 4 3 - + 3 / 2 3 + * " is 12.
   (1-2) One of the major differences between an algorithm and a program is that the algorithm has to satisfy the finiteness requirement.
   (1-3) For any nonempty binary tree T, if n_0 is the number of leaf nodes and n_1 the number of nodes of degree 1, then $n_0 = n_1 + 1$.
   (1-4) A complete binary tree having n nodes have ($\log n$) levels.
   (1-5) The complexity of deleting the minimum from a min heap is $O(\log n)$.

2. (10%) Choices (**單選題**): For each of the 5 questions below, give the most suitable one answer to get the points.
   (2-1) In the specification of abstract data type, queue is implemented using
   (A) Array (B) Singly linked list (C) Doubly linked list
   (D) Circular list (E) None of the above

   (2-2) The time complexity of producing the first n Fibonacci numbers can be as efficient as
   (A) $O(\log_2 n)$ (B) $O(n)$ (C) $O(n \cdot \log_2 n)$ (D) $O(n^2)$ (E) $O(n^2 \cdot \log_2 n)$

   (2-3) The time complexity of sorting 2 n-element sequences can be as efficient as
   (A) $O(n)$ (B) $O(n \cdot \log_2 n)$ (C) $O(n^2)$ (D) $O(n^2 \cdot \log_2 n)$ (E) $O(n^3)$

   (2-4) The preorder traversal of a 15-node binary tree T produces the sequence

   A B C D E F G H I J K L ... ...

   Suppose the subtree with node **C** as root is a full binary tree and node **K** is the root of node **A**'s right subtree. If the number of possible distinct binary trees represented by the above sequence is x, then which of the following is true?
   (A) $151 \le x \le 300$ (B) $351 \le x \le 450$ (C) $501 \le x \le 600$
   (D) $601 \le x \le 750$ (E) $651 \le x \le 1000$ (F) None of the above

   (2-5) Which of the following is true?
   (A) Hashing technique enables to perform operations of *search*, *insert* and *delete* in the same expected time.
   (B) Hashing is an efficient technique in applications of both searching and sorting.
   (C) Implementing stack using circular list enables efficient handling of the "*stack full*" condition.
   (D) The number of spanning trees of a graph with 7 nodes is 63.
   (E) None of the above is true.

(背面仍有題目,請繼續作答)

※ 考生請注意：本試題 □可 ☑不可 使用計算機

3. (15%) Multiple Choices (複選題): For each question below, there may be more than one correct choice and you must give exactly all the correct choices to get the points.

(3-1) (5%) Which of the following statement(s) is (are) true?
(A) We can traverse a binary tree having n nodes with a time complexity of $O(\log n)$.
(B) The postorder and inorder sequences of a binary tree uniquely define the binary tree.
(C) A simple path in a graph can be a cycle.
(D) A forest has to at least contain one tree.
(E) The subtree of a binary search tree is also a binary search tree.

(3-2) (5%) Which of the following statement(s) is (are) true?
(A) A graph may not have an edge from a vertex back to itself.
(B) An n-vertex, undirected complete graph has n(n-1) edges.
(C) The complexity for performing depth first search and breath first search on a n-vertex undirected graph which is represented by an adjacency list is $O(n)$.
(D) The complexity for performing depth first search and breath first search on a n-vertex undirected graph which is represented by an adjacency list is $O(n)$.
(E) Dijkstra algorithm may not be able to find all the shortest paths from a vertex to all other vertices in a weighted directed graph with negative edge costs.

(3-3) (5%) Which of the following statement(s) is (are) true?
(A) Both heap sort and insertion sort are not stable sorting schemes.
(B) Heapsort can sort n unique numbers into an ascending or descending order in $O(\log n)$ time.
(C) Comparing the performance of the worst-case scenario, merge sort is better than quick sort.
(D) The worst case performance of heap sort is better than the average case performance of the insertion sort when the number of keys to be sorted is sufficient large.
(E) Radix sort and heap sort have the same space complexity.

4. (10%) Given a sequence of numbers below.

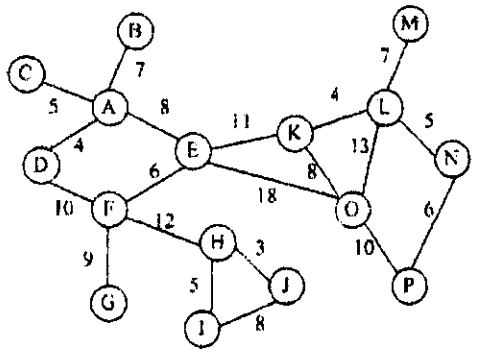
20 26 31 4 1 11 6 16 8 18 39 42

(4-1)(4%) Insert the numbers sequentially into an empty binary search tree. Please show the construction of the binary search tree step-by-step.

(4-2)(6%) Insert the numbers sequentially into an empty AVL tree. Please show the construction of the AVL tree step-by-step and indicate each rotation.

※ 考生請注意：本試題 □可 ☑不可 使用計算機

5. (10%) Given the following graph.



(5-1)(4%) How many binconnected components and articulation points in the graph?

(5-2)(3%) Find the minimum spanning tree using Kruskal's algorithm. Show your work step-by-step. In case of tie upon selecting an edge, an edge associated with a vertex that has lexicographic precedence should be selected.

(5-3)(3%) Find the minimum spanning tree starting from vertex K using Prim's algorithm. Show your work step-by-step.

6. (15%) For each of the terms below, please give its definition or explanation:

(6-1) Greedy method
(6-2) Priority queue
(6-3) Max tree
(6-4) Dynamic hashing
(6-5) Loading factor of a hash table

7. (10%) Given the following declarations in C programming language for a singly linked list:

```
typedef  struct  listNode *listPointer;
typedef  struct  listNode {
           char   data;
           listPointer  link;
   };
```

(7-1) Based on the above, develop a recursive function in C programming language which inverts a singly linked list L. (6%)

(7-2) Is your function in (7-1) more efficient than non-recursive ones? Please explain. (4%)

(背面仍有題目,請繼續作答)

※ 考生請注意：本試題 □可 ☑不可 使用計算機

8. (15%) Given an incomplete implementation of a function in C programming language as below, where M is an N×N matrix of integers, N > 2.

```
void matrix_sort (int *M, int N){
int T, I, J, K, L, W;
    ... ...
 }
```

The above function will sort the matrix M with minimal storage space such that

$\forall j, 0 \le j < N$, `M[i, j]` ≤ `M[(i+1), j]`, where $0 \le i \le (N-2)$,

and $\forall j, 0 \le j < (N-1)$, `M[(N-1), j]` ≤ `M[0, (j+1)]`.

That is, the elements of M are sorted in non-decreasing order by column-major.

(8-1) Please complete the implementation of function `matrix_sort`. (8%)

(8-2) What is the time complexity of your implementation? (2%)

(8-3) What is the space complexity of your implementation? (2%)

(8-4) Is the space complexity of your implementation the minimal? Please explain. (3%)