國立臺灣師範大學107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 5 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，否則依規定扣分。

1. Given an array of eight integers: 6, 3, 8, 1, 4, 5, 9, 2.

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

(a) What is the result in the array after executing the FIRST 5 iterations of the main loop in the selection sort algorithm. (3 分)

(b) What is the result in the array after executing the FIRST 5 iterations of the main loop in the insertion sort algorithm. (3 分)

(c) Please complete the following recursive implementation of the binary search algorithm in C/C++ code. (3 分)

```
int BinarySearch (int *a, const int x, const int left, cons int right)
{// Search the sorted array a[left], …, a[right] for x
   if (left <= right){
      int middle = _____<1>_____;
      if (x <= a[middle] return BinarySearch(a, x, _____<2>_____);
         else if (x > a[middle] return BinarySearch(a, x, _____<3>_____);
         return middle;
    }
    return -1;
}
```

(d) Please complete the sequential search algorithm in C/C++ code. (3 分)

```
int SequentialSearch (int *a, const n, const int x)
{// Search the sorted array a[0], …, a[n-1] for x
…
}
```

2.(a) Suppose you are given a sparse matrix shown below:

$$\mathbf{M}=\begin{bmatrix} 0 & 2 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 6 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 \end{bmatrix}$$

Please show the linked representation of the sparse matrix such that a program can access the non-zero terms in the same row or in the same column efficiently. (4 分)

(b) Suppose the above sparse matrix is an adjacency matrix of a digraph. In the graph, there are 5 nodes (denoted as N0, N1, …, and N4). Besides, M[i, j] = 0 implies that there is no edge between node Ni and Nj. Otherwise, M[i, j] denote the edge weight between node Ni and Nj. Please show the DFS and BFS visiting order of the nodes on the digraph starting from N0, respectively. The visiting order should follow the direction of the directed edges. (4 分)

3. The array representation of a binary tree is shown below:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-	A	B	C	-	F	D	E	-	-	-	-	-	G	-	H

where "-" represents a null value.

For any node with index i, $1 \le i \le n$, we have

- *Parent*(i) is at $\lfloor i/2 \rfloor$ if $i \neq 1$. If $i = 1$, i is at the root and has no parent.
- If i has left child, *LeftChild*(i) is at $2i$.
- If i has right child, *RightChild*(i) is at $2i+1$.

(a) Please show the results of inorder traversal and postorder traversal of the binary tree, respectively. (4 分)

(b) Suppose that we have the following key values: 4, 10, 20, 40, 2, 15, 3, 1, 18. Please write out the max heap after each value is inserted into the heap one by one. (6 分)

4. Given an adjacency matrix of a digraph consisting of 3 nodes as the following.

	0	1	2
0	0	6	3
1	5	0	4
2	2	8	0

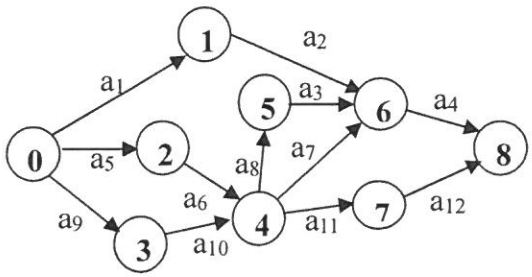
(a) Please write a pseudo code to find the all-pairs shortest-paths. (6 分)

(b) Please show the result of the all-pairs shortest-paths for the given digraph. (4 分)

5. Given the following AOE network.

Start vertex:: V0

Finish vertex: V8



The cost of each activity is shown in the following table:

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12
2	5	2	6	1	4	4	1	2	2	7	3

(a) Please compute the earliest and the latest starting times for each activity. Please explain the computing process and show the result of computation. (6 分)

(b) Please find the critical activities which are on all the critical paths. (4 分)

6. The pseudo code of the Dijkstra's algorithm is given as follows.

```
DIJKSTRA(G, w, s)
// G: input graph, w: edge weights, s: source vertex
1   INITIALLIZE(G, s)
2   S = {}
3   Q = G.V
4   while Q ≠ {}
5       u = EXTRACT-MIN(Q)
6       S = S ∪ {u}
7       for each vertex v ∈ G.Adj[u]
8           if v.d > u.d + w(u, v)
9               v.d = u.d + w(u, v)
```

(a) What does the algorithm compute? (5 分)

(b) Please briefly explain the reason why the Dijkstra's algorithm can be considered as a greedy algorithm. (5 分)

(c) Give a condition where the Dijkstra's algorithm is not applicable. Describe the condition with an example. (5 分)

7. The maximum-subarray problem is the task of finding the contiguous subarray within a one-dimensional array of numbers which has the largest sum. For example, for the array of values −9, 10, −9, 10, −5, 4; the contiguous subarray with the largest sum is 10, −9, 10 with sum 11.

(a) Please briefly describe a brute-force solution to this problem and justify its time complexity. (5 分)

(b) Let's think about a solution using divide-and-conquer. We find the midpoint, say *mid*, of the array $A[low \ldots high]$, and consider the subarrays $A[low \ldots mid]$ and $A[mid+1 \ldots high]$. A maximum subarray must lie in exactly one of the

following places:

i. entirely in the subarray $A[low...mid]$

ii. entirely in the subarray $A[mid+1...high]$

iii. crossing the midpoint

Please briefly describe an algorithm that finds a maximum subarray crossing the midpoint in time *linear* in the size of $A[low...high]$. (8 分)

(c) As it appears above, the divide-and-conquer solution finds maximum subarrays of $A[low...mid]$ and $A[mid+1...high]$ recursively, with a linear-time procedure that finds a maximum subarray crossing the midpoint. Please write a recurrence for the running time of the divide-and-conquer approach, and solve the recurrence. (5 分)

(d) This problem can also be solved by dynamic programming. The idea is to start at the left end of the array, process toward the right, and keep track of the maximum subarray seen so far. Please use the above idea to develop a non-recursive, linear-time algorithm for the maximum-subarray problem. Your algorithm should return the sum value (e.g. 11 in the above example). (7 分)

8. Suppose P ≠ NP, please draw the Venn diagram of the complexity classes P, NP, NP-complete and NP-hard. (10 分)