九十二學年度 資訊工程學系 （所） ________組碩士班研究生招生考試
科目 基礎計算機科學 科號 2701 共 4 頁第 1 頁 *請在試卷【答案卷】內作答

1. (8%) Assume a tree T has E edges where E is an even integer number. Removing one edge from T yields two disjoint trees T1 and T2. Given that the number of vertices in T1 is equal to the number of edges in T2. Determine the number of vertices in T2 and the number of edges in T1.

2. (8%) Let M be a finite state machine (FSM) as shown in Fig. 1. Each state in M, except for S1 and S9, has certain delay units as shown in Fig. 2. Find the maximum-delay input sequence that takes M from S1 to S9. Note that each state can be only traversed once at most.

Present State	Next State	
	Input 0	1
S1	S2	S3
S2	S4	S2
S3	S5	S4
S4	S6	S8
S5	S7	S2
S6	S8	S7
S7	S9	S3
S8	S2	S9
S9	S5	S3

Fig. 1

State	S2	S3	S4	S5	S6	S7	S8
Delay	2	1	3	3	1	3	2

Fig. 2

3. (8%) For $n \geq 1$, show that if $n \geq 64$, then n can be written as a sum of 5's and /or 17's.

4. (5%) Let $A = \{1, 2, 3, 4, 5\}$, $B = \{w, x, y, z\}$, $A_1 = \{2, 3, 5\} \subseteq A$ and $g : A_1 \to B$. In how many ways can g be extended to a function $f: A \to B$.

5. (5%) Let A be a set with $|A| = n$, and let R be a relation on A that is antisymmetric. What is the maximum value for $|R|$?

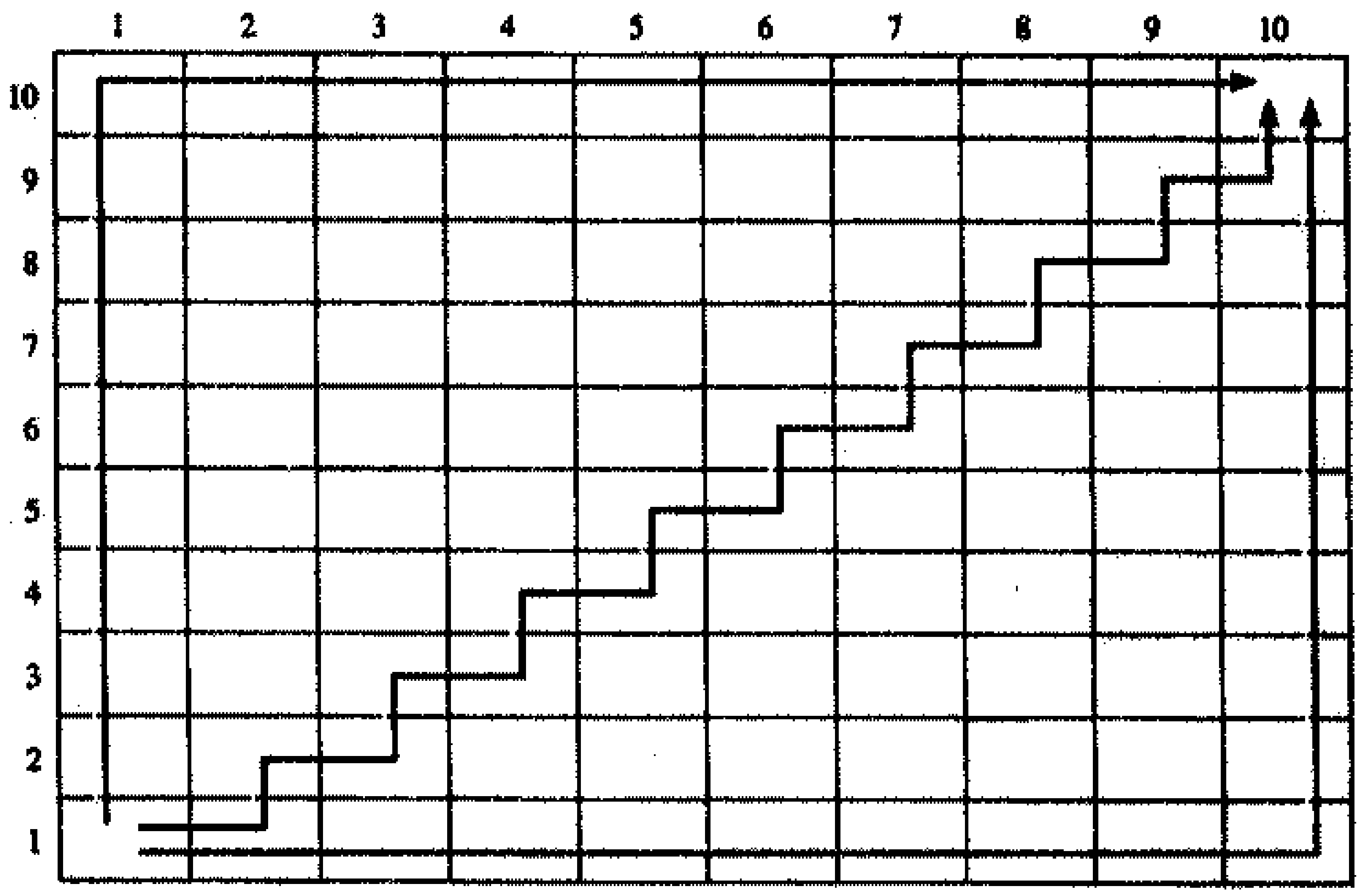
6. (12%) Let a_r denote the number of edges in a complete graph on r vertices.

(a) (4%) Derive a recurrence relation for a_r in terms of a_{r-1}.

(b) (8%) Use the generating functions to solve the recurrence relation.

7. (4%) S is a finite alphabet. Let A denote the set all nonempty strings of letters from S. Let $+$ be a binary operation on A such that for any two strings a and b in A, $a+b$ yields a string which is the concatenation of strings a and b. Show that $(A,+)$ is a semigroup.

8. (8%) We want to count the number of possible paths moving from row 1, column 1 to row N, column N in a two-dimensional grid. Steps are restricted to going up or to the right, but not diagonally. The following illustration shows three of such paths, when $N = 10$:



(a) (4%) The following function, `NumPaths`, is supposed to count the number of paths, but it has some bugs. Debug the function. (**Note:** The LOC (line of code) of the corrected version should <10. LOC >10 is not accepted).

```
Int NumPaths(int row, int col, int n)
{
   if (row==n)
       return 1;
   else
       if (col == n)
          return NumPaths + 1;
        else
          return NumPaths(row + 1, col) * NumPaths(row, col + 1);
 }
```

(b) (4%) After you have corrected the function, trace the execution of `NumPaths` with n=4 by hand. Is this algorithm efficient? If so, briefly explain why. If not, how the efficiency of this operation can be improved?

9. (3%) A very large array of elements is to be sorted. The program is to be run on a general personal computer (PC) with limited memory. Which sorting algorithm would be a better choice: Heap Sort or Merge Sort? Why?

10. (6%) Given an initial sequence of numbers as shown in the following array.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
26	24	3	17	25	24	13	60	47	1

Select sorting algorithms that would produce the following results after four iterations. Possible sorting algorithms are heap sort, quick sort, merge sort, insertion sort, selection sort, bubble sort. (**Note:** Explanation is needed)

(a) (2%)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	3	13	17	26	24	24	25	47	60

(b) (2%)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	3	13	17	25	24	24	60	47	26

(c) (2%)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
3	17	24	26	25	24	13	60	47	1

11. (5%) (a) Describe a model for the following scheduling problem. Given tasks T1, T2, … , Tn and a set of constraints, each of the form 'Ti must be completed prior to the start of Tj', find a schedule of tasks satisfying all the constraints. (b) For the above model, describe briefly how to find a legal schedule of tasks.

12. (9%) Exhibit a data structure to represent directed graphs with positive weights associated with edges. The vertices are numbered from 1 to n and the edges from 1 to m (n and m are given). Your structure should support the following operations:

op1: Adj (v): given vertex v, return a list of all vertices adjacent to v, i.e., all w such that <v, w> is an edge of the graph.

op2: End (e, i): given edge e=<v, w>, return v if i=0 and w if i =1

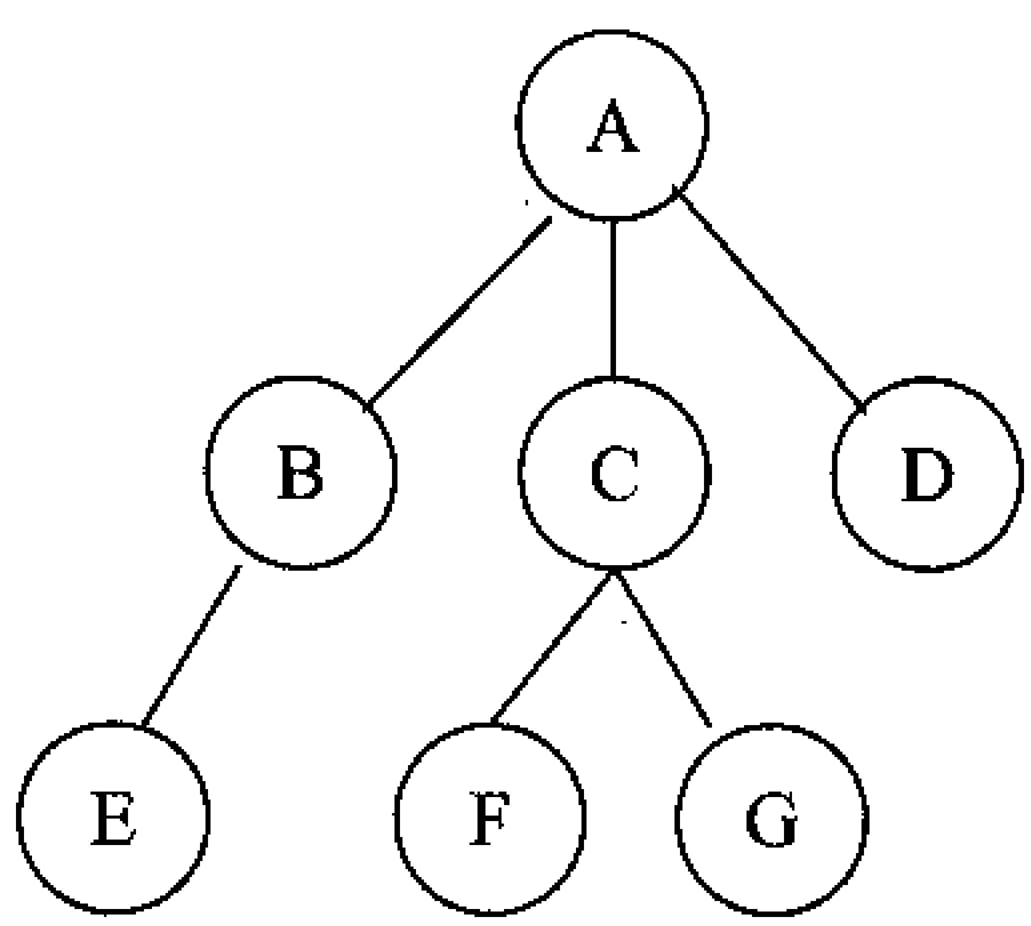
op3: Weight (e): given edge e, return the weight of e

op4: Oppweight (e): given edge e=<v, w>, return the weight of the edge <w, v> (return –1 if <w, v> is not an edge).

Operations **op2, op3** and **op4** should take O(1) time; operation **op1** should take O(k) time, where k is the number of adjacent vertices. The representation of the graph should use at most O(m+n) memory. (a) (4%) give the data structure (b) (5%) explain why your representation meets the space constraint and time constraints of the operations. Do not worry about initialization: the graph is given in the representation chosen before your algorithms are called.

13. (4%) Let a graph be dented as G=(V, E). Discuss how to test if the graph is connected in O(|V| + |E|).

14. (15%) A clock tree is a general tree in which each node n_i is associated with a size size(n_i) and delay delay(n_i). Delay(n_i) is in turn a function of the sizes of every immediate child of node n_i and n_i itself. That is, delay(n_i) = size(n_i) + Σsize(n_j)/ size(n_i), where n_j is a child of n_i. Assume both size and delay are unit-less. The path delay from the root to a node is defined as the summation of delays of all nodes along the path. The skew of the clock tree is defined as the difference between the longest path delay and the shortest path delay among all root-to-leaf paths. Write a recursive procedure (in pseudo code) that computes the skew of the clock tree. Depicted below is an example clock tree. The table followed shows the computation that gives skew 16 – 10 = 6.



Node	Size	Node Delay	Path Delay	Comment
A	1	1+(2+1+3)/1=7	7	
B	2	2+4/2=4	7+4=11	
C	1	1+(2+3)/1=6	7+6=13	
D	3	3+0/3=3	7+3=10	Shortest root-to-leaf path
E	4	4+0/4=4	7+4+4=15	
F	2	2+0/2=2	7+6+2=15	
G	3	3+0/3=3	7+6+3=16	Longest root-to-leaf path