國立雲林科技大學資訊管理系

113 學年度第 1 學期博士班資格考 (Ph.D. Qualify Exam)

考試科目 (Subject)：演算法 (Algorithm)

時數與方式 (Hour and type)：四小時 (4 hours)

Close book, 不可使用計算器 (Calculators not allowed)

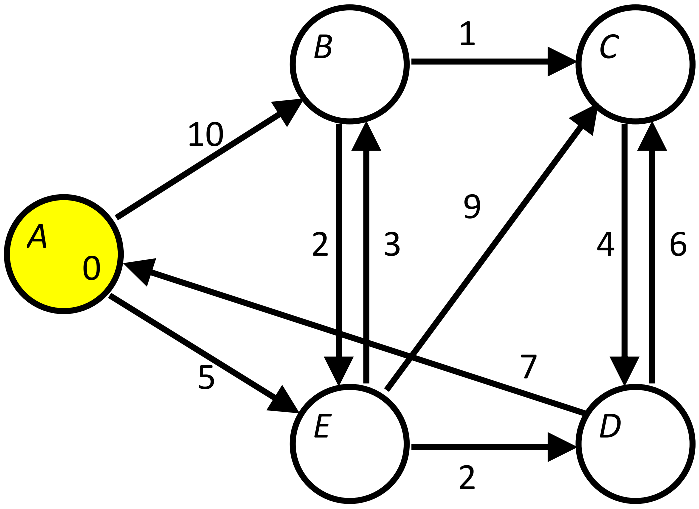
Part 1: 30 points (Optional Questions: Answer 3 out of 4, 10 points each; if you answer more than 3 questions, only the first 3 answered questions will be graded.)

1. There is a binary tree with the preorder traversal sequence BADGEICFH and the inorder traversal sequence GDAEIBCHF.
   (a) Please draw this binary tree. (7%)
   (b) What is the postorder traversal sequence. (3%)

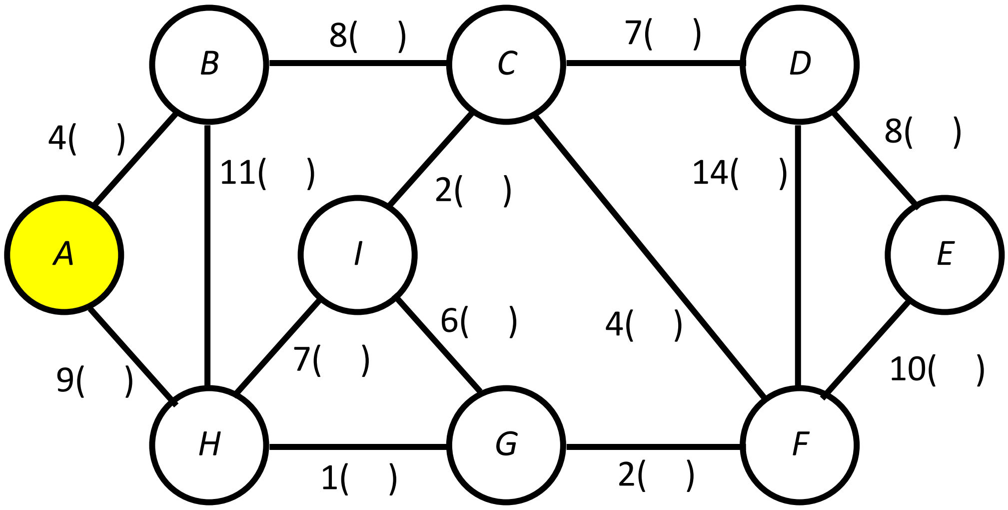
2. Find the longest common subsequence (LCS) of two sequences X="ABCBDAB" and Y="BDCABA" by using the dynamic programming technique.
   (a) Fill the table with C_{ij} (the length of the LCS of X_i and Y_j) and the trace directions with symbols like ←↑↖. (7%)
   (b) Show the LCS of X and Y. (3%)

		Y_j	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
			B	D	C	A	B	A
X_i		C_{ij}	0	0	0	0	0	0
X_1	A	0	↑0					
X_2	B	0						
X_3	C	0						
X_4	B	0						
X_5	D	0						
X_6	A	0						
X_7	B	0						

3. Given the following graph, use Dijkstra's Algorithm to find the shortest paths from the vertex A to all other vertices.
   (a) Fill in the circles with the shortest distances obtained. (7%)
   (b) Show the shortest path from vertex A to vertex C. (3%)



4. Given the following graph, to start from vertex A, find the minimum spanning tree (MST) of the graph using the Prim's algorithm.
   (a) Fill in the parentheses with the order in which each edge was added to generate the MST, using 1 to indicate the first edge added, 2 for the second, and so on. (7%)
   (b) Draw the generated MST in a tree structure. (3%)
   (Hint: Prim's algorithm uses a min-priority queue Q to select the next edge)



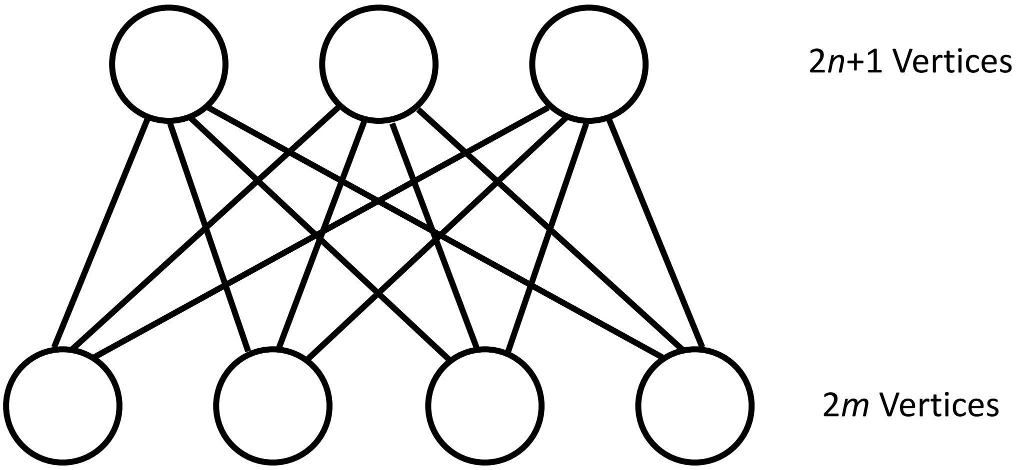
Part 2: 40 points (Optional Questions: Answer 4 out of 5, 10 points each; if you answer more than 4 questions, only the first 4 answered questions will be graded.)

5. Please explain the differences between O(), Ω(), and Θ() in terms of **asymptotic upper, lower, and tight bounds**, as well as **asymptotically non-negative and positive functions**. (10%)

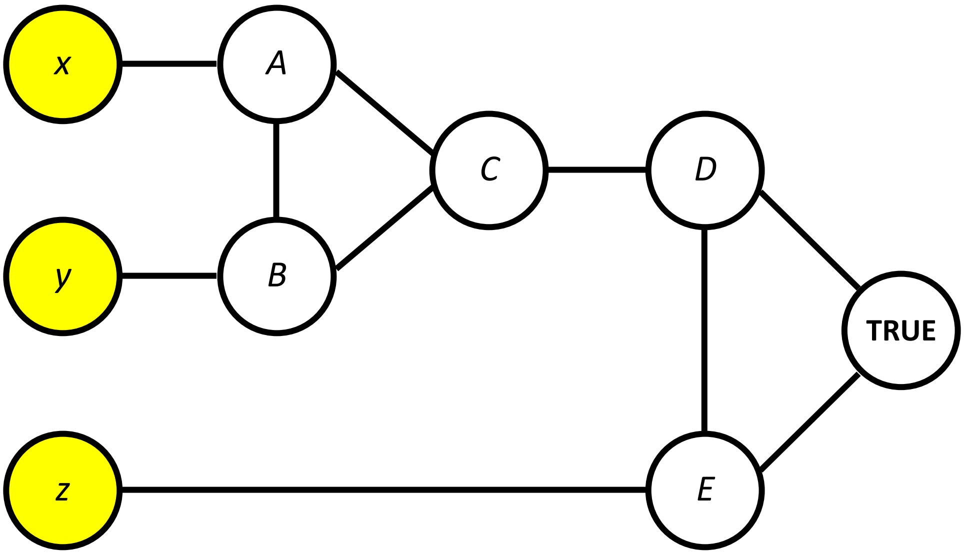
6. Please explain the differences between
   (a) P (Polynomial),
   (b) NP (Nondeterministic Polynomial),
   (c) NPC (NP-complete), and
   (d) NPH (NP-hard)
   in terms of problems that can be **decided**, **verified** or **reduced** in polynomial-time. (3% each, max 10%)

7. Prove that $\boldsymbol{if}\ P \cap NPC \neq \emptyset, \boldsymbol{then}\ P = NP.$ (10%)

8. Prove that if G is an undirected bipartite graph with an odd number of vertices, as illustrated in the following graph, then G is non-Hamiltonian. (10%)



9. To reduce the 3-CNF-SAT problem to a 3-COLOR (**TRUE**, **FALSE**, or **RED**) problem, the clause $(x \vee y \vee z)$ is represented as the following graph. Demonstrate that if each of x, y and z is colored either c(**TRUE**) or c(**FALSE**), then the widget is 3-colorable if and only if at least one of x, y, or z is colored c(**TRUE**). (10%) (Hint: $c(u) \neq c(v)$ for every edge $(u, v) \in E$)



Part 3: 30 points (Optional Questions: Answer 2 out of 4, 15 points each; if you answer more than 2 questions, only the first 2 answered questions will be graded.)

10. Justify that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$ by:
    (a) Explaining the meanings of n_0, c_1, and c_2. (5%)
    (b) Providing a valid set of values for n_0, c_1, and c_2. (5%)
    (c) Demonstrating how you derived these values. (5%)

11. Please explain the specific problems that
    (a) Euler Tour,
    (b) Hamiltonian Cycle, and
    (c) Traveling Salesman Problem (TSP)
    each aim to solve. (5% each, total 15%)

12. Reduce the SAT problem $\phi = (x \vee y) \wedge \bar{z}$ to a 3-CNF-SAT problem:
    (a) Show the parse tree of ϕ. (3%)
    (b) Rewrite ϕ as the AND of the root variable and a conjunction of clauses describing the operation of each node. (4%)
    (c) Convert each clause into conjunctive normal form with truth-table. (6%)
    (d) Construct the 3-CNF formula. (2%)
    (Hint: $\bar{z} = \mathrm{NOT}(z)$, which is the complement of z)

13. Given a binary search tree with N nodes, design an algorithm to find the k-th largest element in the tree. Assume that the value of k is valid, i.e., $1 \le k \le N$. The algorithm must run in $O(h)$ time, where h is the height of the tree. Provide a detailed step-by-step explanation of the algorithm using pseudocode. Prove the correctness of the algorithm and analyze its time complexity. (15%)