



Daffodil International University
Faculty of Science & Information Technology
Department of Computer Science and Engineering
Final Examination, Spring-2024
Course Code: CSE212 Course Title: Discrete Mathematics
Level: 2 Term: 1 Batch: 64

Exam Duration: 02 Hours

Marks: 40

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1. Consider the following adjacent matrix where a, b, c, d, e and f are the vertices of an undirected graph. CO3

	a	b	c	d	e	f
a	1	1	3	0	0	0
b	1	0	2	2	0	9
c	3	2	0	1	0	3
d	0	2	1	0	9	1
e	0	0	0	9	0	1
f	0	9	3	1	1	0

a) Construct a graph from the given matrix. [2]

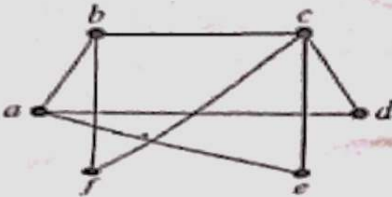
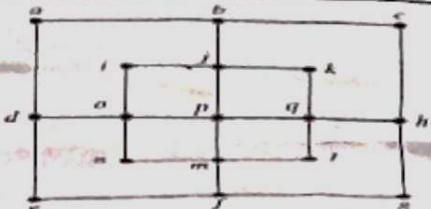
b) Prove handshaking theorem for the graph. [3]

c) Prove whether the following graphs G and H are isomorphic or not? [5]

2. a) Derive a relation from the graph and find whether the relation is reflexive, symmetric, antisymmetric and transitive. If not make at least one relation putting the pair which are needed. [5] CO2

b) Prove that $2 - 2.7 + 2.7^2 - \dots + 2.(-7)^n = (1 - (-7)^{n+1})/4$ whenever n is a nonnegative integer. [5] 5

$$1 - \frac{(-7)^{n+1}}{4} = \frac{49}{4}$$

3.	a) Draw the following graphs and also find out the degree of each vertices of the graphs. i. $K_{4,7}$ ii. W_5	[5]	CO3																																																																																	
b)	Apply coloring algorithm to show whether the graphs are bipartite or not.	[5]																																																																																		
	 Graph: A  Graph: B																																																																																			
4.	Suppose that the nodes of Graph G are $\{a, b, c, d, e, g, h\}$ and the weights of the edges in G are given in the following matrix. Here, the symbol (*) when there is no edge between a given pair nodes. <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>a</th> <th>b</th> <th>c</th> <th>d</th> <th>e</th> <th>f</th> <th>g</th> <th>h</th> </tr> </thead> <tbody> <tr> <th>a</th> <td>0</td> <td>4</td> <td>5</td> <td>*</td> <td>1</td> <td>2</td> <td>1</td> <td>5</td> </tr> <tr> <th>b</th> <td>4</td> <td>0</td> <td>1</td> <td>3</td> <td>*</td> <td>1</td> <td>8</td> <td>15</td> </tr> <tr> <th>c</th> <td>5</td> <td>1</td> <td>0</td> <td>9</td> <td>2</td> <td>*</td> <td>4</td> <td>7</td> </tr> <tr> <th>d</th> <td>*</td> <td>3</td> <td>9</td> <td>0</td> <td>8</td> <td>2</td> <td>4</td> <td>6</td> </tr> <tr> <th>e</th> <td>1</td> <td>*</td> <td>2</td> <td>8</td> <td>0</td> <td>*</td> <td>*</td> <td>10</td> </tr> <tr> <th>f</th> <td>2</td> <td>1</td> <td>*</td> <td>2</td> <td>*</td> <td>0</td> <td>5</td> <td>8</td> </tr> <tr> <th>g</th> <td>1</td> <td>8</td> <td>4</td> <td>4</td> <td>*</td> <td>5</td> <td>0</td> <td>*</td> </tr> <tr> <th>h</th> <td>5</td> <td>15</td> <td>7</td> <td>6</td> <td>10</td> <td>8</td> <td>*</td> <td>0</td> </tr> </tbody> </table>		a	b	c	d	e	f	g	h	a	0	4	5	*	1	2	1	5	b	4	0	1	3	*	1	8	15	c	5	1	0	9	2	*	4	7	d	*	3	9	0	8	2	4	6	e	1	*	2	8	0	*	*	10	f	2	1	*	2	*	0	5	8	g	1	8	4	4	*	5	0	*	h	5	15	7	6	10	8	*	0		CO3
	a	b	c	d	e	f	g	h																																																																												
a	0	4	5	*	1	2	1	5																																																																												
b	4	0	1	3	*	1	8	15																																																																												
c	5	1	0	9	2	*	4	7																																																																												
d	*	3	9	0	8	2	4	6																																																																												
e	1	*	2	8	0	*	*	10																																																																												
f	2	1	*	2	*	0	5	8																																																																												
g	1	8	4	4	*	5	0	*																																																																												
h	5	15	7	6	10	8	*	0																																																																												
a)	Now, Apply Prim's algorithm, starting at node a, to construct a minimum spanning tree and find the weight based on the given matrix.	[5]																																																																																		
b)	Apply Dijkstra on this weighted graph to find the shortest path from a to h.	[5]																																																																																		