THE 5 Solutions

Answer 1

a)

(2 pts)

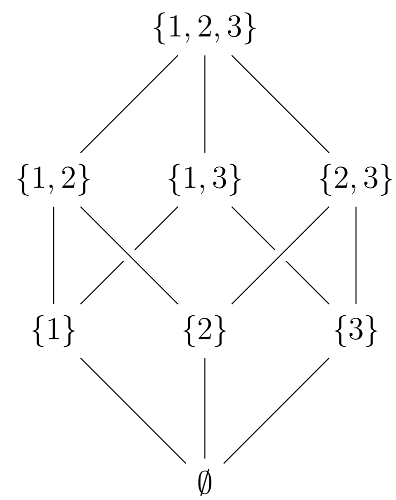

Figure 1: Hasse Diagram.

b)

(2 pts) Yes, for every pair of elements there is a least upper bound and a greatest lower bound.

c)

(2 pts) The only maximal element is $A = \{1, 2, 3\}$ which is the top element in the Hasse diagram.

d)

(2 pts) The only minimal element is $\emptyset$ which is the bottom element in the Hasse diagram.

e)

(2 pts) Yes, the greatest element is $A = \{1, 2, 3\}$. It is because $S \subseteq A$ whenever S is a subset of A.

f)

(2 pts) Yes, the least element is $\emptyset$. It is because $\emptyset \subseteq S$ for any subset S of A.

g)

(2 pts) The least upper bound of $\{1\}$ and $\{3\}$ is $\{1\} \cup \{3\}$ which is equal to $\{1, 3\}$.

Answer 2

a)

(2 pts) According to the Handshaking Theorem (p.653) the answer is $2m$ which is $2 \times 7 = 14$, where m is the number of edges.

b)

(2 pts) Since it is undirected, again the result is 14.

c)

(2 pts) Since it is undirected, again the result is 14.

d)

(2 pts) **Complete graph:** exactly one edge between each pair of distinct vertices. Such a subgraph with three vertices:

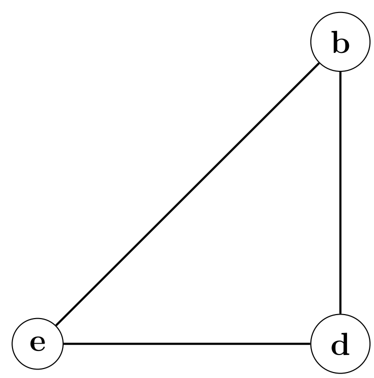

Figure 2: A complete subgraph of G.

e)

(2 pts) A bipartite graph is a graph that does not contain any odd-length cycles. Therefore, G is not a bipartite graph. For example, if we remove the edges bc and be we get a bipartite graph:

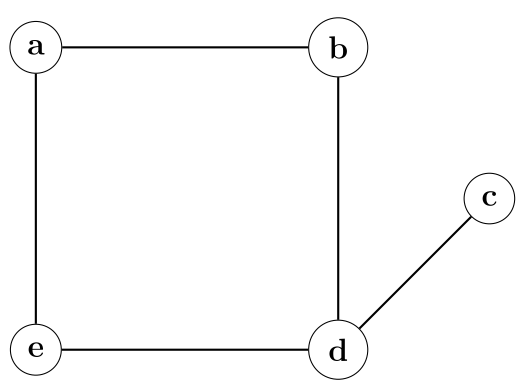

Figure 3: A bipartite subgraph of G.

f)

(2 pts) For every edge, there exist two possible direction options and we have 7 edges, so the answer is $2^7 = 128$.

g)

(2 pts) It shouldn't contain the same edge more than once. Such a path is e, a, b, e, d, b, c, d and its length is 7.

h)

(2 pts) There is 1 connected component which is G itself. The reason is that there is a simple path between every pair of distinct vertices of G. So it is already a connected component.

i)

(2 pts) An Euler circuit in a graph is a simple circuit containing every edge of the graph. In order to have an Euler circuit, every vertex must have even degree. We see that $deg(e) = 3$ so there is not an Euler circuit in G.

j)

(2 pts) An Euler path in a graph is a simple path containing every edge of the graph. There exists such a path in G: d, b, c, d, e, b, a, e.

k)

(2 pts) A simple circuit in a graph that passes through every vertex exactly once is called a Hamilton circuit. There exists such a circuit in G: a, b, c, d, e, a.

l)

(2 pts) A simple path in a graph that passes through every vertex exactly once is called a Hamilton path. There exists such a path in G: a, b, c, d, e.

Answer 3

(5 pts) First, let's check the invariants: the number of vertices, the number of edges and the number of vertices of each degree. They all must be the same.
G has 8 vertices, 16 edges and the degree of each vertex in G is 4. H also has 8 vertices, 16 edges and the degree of each vertex in H is 4, too. However, that is not enough to conclude that they are isomorphic.
(10 pts) We now will define a function i and then determine whether it is an isomorphism. By examining the cycles of length 3 and length 4, we can define the following one-to-one and onto function: $i(a) = a'$, $i(b) = c'$, $i(c) = e'$, $i(d) = g'$, $i(e) = b'$, $i(f) = h'$, $i(g) = d'$, $i(h) = f'$.
We now have a one-to-one correspondence between the vertex set of G and the vertex set of H. To see whether i preserves edges, we examine the adjacency matrix of G,

$$A_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} \begin{array}{c} \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array},$$

and the adjacency matrix of H with the rows and columns labeled by the images of the corresponding vertices in G,

$$A_H = \begin{array}{c} \\ a' \\ c' \\ e' \\ g' \\ b' \\ h' \\ d' \\ f' \end{array} \begin{array}{c} \begin{array}{cccccccc} a' & c' & e' & g' & b' & h' & d' & f' \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

Because $A_G = A_H$, it follows that i preserves edges. We conclude that i is an isomorphism, so G and H are isomorphic.

Answer 4

(10 pts) The iterations of Dijkstra's algorithm are described in the following table.

Path	a	b	c	d	e	f	g	h	i	j	k
-	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
a		$\boxed{3}$	∞	∞	5	∞	∞	4	∞	∞	∞
a			5	∞	5	10	∞	$\boxed{4}$	∞	∞	∞
a, b			$\boxed{5}$	∞	5	9	∞		6	∞	∞
a				8	$\boxed{5}$	7	11		6	∞	∞
a, h				8		7	11		$\boxed{6}$	∞	∞
a, b, c				8		$\boxed{7}$	11			12	∞
a, b, c				$\boxed{8}$			11			10	∞
a, b, c, f							11			$\boxed{10}$	10

(10 pts)

Step 1: Initialize the value for a as 0. The other lengths are ∞ for now.

Step 2: The closest vertex to a is b and the length is 3. Update the values for c and f.

Step 3: The second closest vertex to a is h and the length is 4. Update the values for f and i.

Step 4: Arbitrarily choose c as the third closest vertex to a and the length is 5. Update the values for d, f and g.

Step 5: The next one is e. There is no update.

Step 6: The next one is i. Update the value for j.

Step 7: The next one is f. Update the value for j.

Step 8: The next one is d. Update the value for k.

Step 9: Arbitrarily choose j as the next closest one. There is no update.

As we reached the target node, we now terminate the algorithm. We find that a shortest path from a to j is a, b, c, f, j, with length 10.

Answer 5

a)

(4 pts)
If we choose Prim's algorithm, we start by selecting an initial edge of minimum weight and continue by successively adding edges of minimum weight that are incident to a vertex in the tree and that do not form simple circuits. We stop when $n-1$ edges have been added.

Choice	**Edge**	**Cost**
1	$\{a, b\}$	1
2	$\{a, d\}$	3
3	$\{b, c\}$	4
4	$\{c, f\}$	2
5	$\{e, f\}$	2

According to the table above, the order in which the edges are added to the tree is $\{a, b\}$,$\{a, d\}$, $\{b, c\}$, $\{c, f\}$ and $\{e, f\}$.

If we choose Kruskal's algorithm, we start by choosing an edge in the graph with minimum weight and continue by successively adding edges with minimum weight that do not form a simple circuit with those edges already chosen. We stop after $n-1$ edges have been selected.

Choice	**Edge**	**Cost**
1	$\{a, b\}$	1
2	$\{c, e\}$	2
3	$\{c, f\}$	2
4	$\{a, d\}$	3
5	$\{b, c\}$	4

According to the table above, the order in which the edges are added to the tree is $\{a, b\}$,$\{c, e\}$, $\{c, f\}$, $\{a, d\}$ and $\{b, c\}$.

b)

(4 pts)
Prim's:

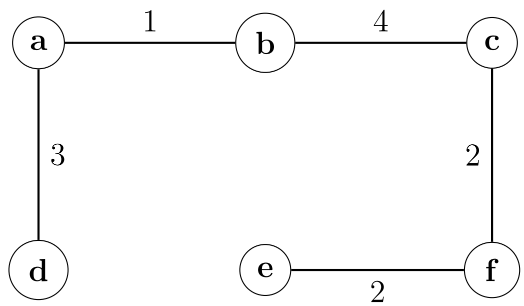

Figure 4: Minimum Spanning Tree for G.

Kruskal's:

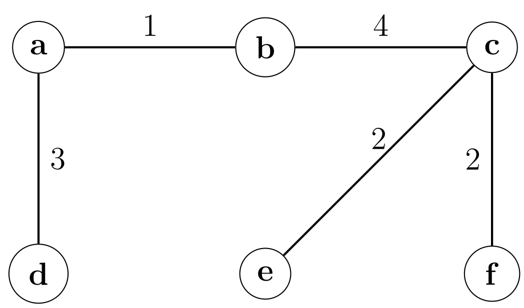

Figure 5: Minimum Spanning Tree for G.

c)

(4 pts)
Prim's:
No, it is not unique. We could choose another option at Choice 4, for example, edge $\{c, e\}$ because its cost is same with the edge we chose. Therefore, we would get another minimum spanning tree.
Kruskal's:
No, it is not unique. We could choose another option at Choice 3, for example, edge $\{e, f\}$ because its cost is same with the edge we chose. Therefore, we would get another minimum spanning tree.
As we can see, we found different minimum spanning trees with different algorithms because of the different choices we made during the execution of the algorithms.

Answer 6

a)

(3 pts) There is 13 vertices and 12 edges. Always $|v| - 1$ edges (p.752 Theorem 2). The height is 4, the length of the path p, r, u, y, n.

b)

(3 pts) **Postorder traversal:** Visit subtrees left to right, visit root.
w, s, m, t, q, x, n, y, u, z, v, r, p.

c)

(3 pts) **Inorder traversal:** Visit leftmost subtree, visit root, visit other subtrees left to right.
s, w, q, m, t, p, x, u, n, y, r, v, z.

d)

(3 pts) **Preorder traversal:** Visit root, visit subtrees left to right.
p, q, s, w, t, m, r, u, x, y, n, v, z.

e)

(3 pts) Vertices that have children are called internal vertices and the tree is called a full binary tree if every internal vertex has exactly 2 children (p.748 Definition 3). In our tree, for example, s is an internal vertex and it has only one child. Therefore, T is not a full binary tree.

Also we can use the theorem (p.753 Theorem 4) in this question. The first part of the theorem says that a full m-ary tree with n vertices has $i = (n-1)/m$ internal vertices. In tree T, $m = 2$ and $n = 13$. If it is a full binary tree, then it should have $12/2 = 6$ internal vertices but it has 8 internal vertices. So, T is not a full binary tree.