THE 5 Solutions

Answer 1

a)

(4 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. There exists such a circuit in G: a, c, b, g, k, l, m, j, f, e, j, i, l, h, i, d, h, g, c, d, e, a.

b)

(4 pts) An Euler path in a graph is a simple path containing every edge of the graph. Exactly two nodes must have odd degree for the starting/ending nodes to be different. There is no such a path in G.

c)

(4 pts) A simple circuit in a graph that passes through every vertex exactly once is called a Hamilton circuit. There is no such a circuit in G.

d)

(4 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, c, b, g, k, l, m, j, f, e, d, h, i.

e)

(4 pts) Answer is 3.

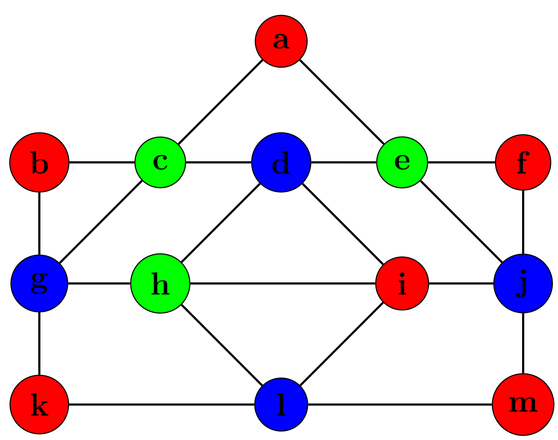

Figure 1: Graph G in Q1.

f)

(4 pts) No, it is not 2-colorable. 3 edges should be deleted: $\{b, c\}$, $\{e, f\}$, $\{h, i\}$.

g)

(4 pts) No, the edge $\{d, l\}$ should be added.

Answer 2

(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 3

a)

(8 pts) If n is even, 2 colors are enough, bipartite. If n is odd we need the third color, it is not bipartite.

- n : even , $\chi(C_n) = 2$, bipartite
- n : odd , $\chi(C_n) = 3$, not bipartite

b)

(7 pts) While constructing a cube graph we join the 2 cube graphs of previous dimension. We connect the same vertices by switching colors, which produces a cube graph colored by two colors. $\chi(Q_n) = 2$, bipartite.

Answer 4

a)

(7 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)

(7 pts) Prim's:

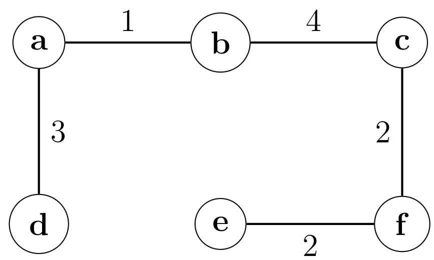

Figure 2: Minimum Spanning Tree for G.

Kruskal's:

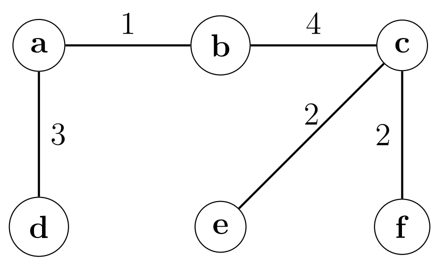

Figure 3: Minimum Spanning Tree for G.

c)

(7 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 5

a)

(7 pts) By Theorem (textbook p752) we know that a full binary tree with i internal vertices has $n = 2i + 1$ vertices. We also know that $n = l + i$ where l is the number of leaf vertices. Solving for i in $n = 2i + 1$ gives $i = (n - 1)/2$. Then inserting this expression for i into the equation $n = l + i$ shows that $l = n - i = n - (n - 1)/2 = (n + 1)/2$.

b)

(7 pts) It is 2, bipartite. Alternate the color at each level.

c)

(7 pts) Since we are looking for an upper bound on the height, we consider $n = m \times h + 1$. So $h \leq \frac{n-1}{m}$.