Student Information

Full Name : Mehmet Rüçhan Yavuzdemir
Id Number : 2522159

Answer 1

a)

By the Handshaking Theorem, sum of degrees of all nodes equal to $2 \cdot |E| = 2 \cdot 7 = \mathbf{14}$. Alternatively, we can sum all the degrees up, $3+3+3+3+2 = \mathbf{14}$.

b)

Let R be adjacency list representation of G.

$$R = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Hence, there are **14** non-zero values.

c)

Let R be the incident matrix representation of G.

$$R = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{pmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Hence, there are **21** zeros.

d)

To have a complete graph with 4 vertices, we should have 4 vertices with all degrees 3. However, to have 4 vertices graph, we should remove vertex d. Although we have 4 vertices with degree 3, we still do not have a possible complete graph. If we select 5 vertices, there should be 5 vertices with degree 4, which is not the case. If we had c-e edge, we would have a complete graph with 4 vertices; a, b, c, e. But in this case, **there is no such a subgraph**.

e)

If we can color the graph with only two colors, with the constraint that adjacent vertices should have different colors, the graph is bipartite. Else, G is not bipartite. Let's assume we have white and black colors.

Let a be painted with white, then b, c, and e should be black since they are neighbors of a. However, there exists an edge between b and c, this contradicts our constraint. They cannot be in the same color. **Hence, G is not bipartite.**

f)

There are 7 undirected edges. If we convert this graph into a directed graph that has G as its underlying undirected graph, we would have 2 decisions for each edge. Hence, we have $2^7 = 128$ directed graphs.

g)

The longest simple path is a path that includes possible the edges. We can include at most 6 edges. For example, **a-b-c-a-e-d-c** is an example of the longest simple path. Hence, our answer is **6**.

h)

Let u and v be any of the vertices in G. There is always a path between them in since graph G is a connected graph. There are no separate other graphs and components left. Hence, G has only **1** connected component.

i)

An Euler circuit is a cycle that starts and ends at the same vertex and visits all the edges exactly once. The necessary condition for having an Euler circuit is that all vertices should have even degrees. If we find a vertex with an odd degree, by the definition of the Euler circuit, we can conclude that there is no Euler circuit in G.

Since $deg(v_a) = deg(v_b) = deg(v_c) = deg(v_e) = 3$, graph G does **not** have any Euler circuit.

j)

An Euler path is a trail that visits all the edges exactly once. The necessary condition for having an Euler path is that there should be zero or two vertices with odd degrees. If we find a contradiction, by the definition of the Euler path, we can conclude that there is no Euler path in G.

Since $deg(v_a) = deg(v_b) = deg(v_c) = deg(v_e) = 3$, $4 \neq 0$ and $4 \neq 2$, graph G does **not** have any Euler path.

k)

A Hamilton circuit is a cycle that starts and ends at the same vertex and visits all the vertices exactly once.

The cycle **b-c-d-e-a-b** is an example of a Hamilton circuit since it visits all the vertices exactly once.

l)

A Hamilton path is a trail that visits all the vertices exactly once.

The path **b-c-d-e-a** is an example of a Hamilton path since it visits all the vertices exactly once.

Answer 2

The function f with $f(a) = a'$, $f(b) = b'$, $f(c) = c'$, $f(d) = d'$ and $f(e) = e'$ is a one-to-one correspondence between the vertices of the graphs G and H according to their degrees. Hence, G and H are isomorphic.

Alternatively, we can see these edges in their adjacency matrix:

$$
G_E = \begin{array}{c c} & \begin{array}{c c c c c} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \left(\begin{array}{c c c c c} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{array} \right) \end{array} \tag{1}
$$

$$
H_E = \begin{array}{c c} & \begin{array}{c c c c c} a' & b' & c' & d' & e' \end{array} \\ \begin{array}{c} a' \\ b' \\ c' \\ d' \\ e' \end{array} & \left(\begin{array}{c c c c c} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{array} \right) \end{array} \tag{2}
$$

Since G_E and H_E are identical, G and H are isomorphic.

Answer 3

For Dijkstra's Algorithm, all we have to do is start with the source node.

At each step, we will choose the non-visited node that has minimum distance, and explore its non-visited neighbors. Update the distance if the new possible distance is smaller than the current distance.

We will denote a visited vertex by putting an apostrophe after it.

$$distance = \begin{array}{cccccccc} s & u & w & v & x & y & z & t \\ \left(\infty \right. & \infty & \infty & \infty & \infty & \infty & \infty & \left. \infty \right) \end{array} \text{ All distances are set to } \infty.$$

$$distance = \begin{array}{cccccccc} s & u & w & v & x & y & z & t \\ \left(0 \right. & \infty & \infty & \infty & \infty & \infty & \infty & \left. \infty \right) \end{array} \text{ s set to 0 since it is the source node.}$$

$$distance = \begin{array}{cccccccc} s' & u & w & v & x & y & z & t \\ \left(0 \right. & 4 & 3 & 5 & \infty & \infty & \infty & \left. \infty \right) \end{array} \text{ mark s as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u & w' & v & x & y & z & t \\ \left(0 \right. & 4 & 3 & 5 & 11 & \infty & 15 & \left. \infty \right) \end{array} \text{ mark w as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u' & w' & v & x & y & z & t \\ \left(0 \right. & 4 & 3 & 5 & 11 & 15 & 15 & \left. \infty \right) \end{array} \text{ mark u as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u' & w' & v' & x & y & z & t \\ \left(0 \right. & 4 & 3 & 5 & 7 & 11 & 15 & \left. \infty \right) \end{array} \text{ mark v as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u' & w' & v' & x' & y & z & t \\ \left(0 \right. & 4 & 3 & 5 & 7 & 8 & 13 & \left. \infty \right) \end{array} \text{ mark x as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u' & w' & v' & x' & y' & z & t \\ \left(0 \right. & 4 & 3 & 5 & 7 & 8 & 12 & \left. 17 \right) \end{array} \text{ mark y as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u' & w' & v' & x' & y' & z' & t \\ \left(0 \right. & 4 & 3 & 5 & 7 & 8 & 12 & \left. 15 \right) \end{array} \text{ mark z as visited and explore its neighbors}$$

$$distance = \begin{array}{cccccccc} s' & u' & w' & v' & x' & y' & z' & t' \\ \left(0 \right. & 4 & 3 & 5 & 7 & 8 & 12 & \left. 15 \right) \end{array} \text{ mark t as visited and explore its neighbors}$$

Hence, the shortest path from s to t is **s-v-x-y-z-t** with cost 15.

Answer 4

a)

For this question, we are going to make use of Kruskal's algorithm, which is a Greedy algorithm. At each step, we will choose the edge with the lowest weight. If the new edge would cause a cycle, we skip it. Let's sort the edge weights.
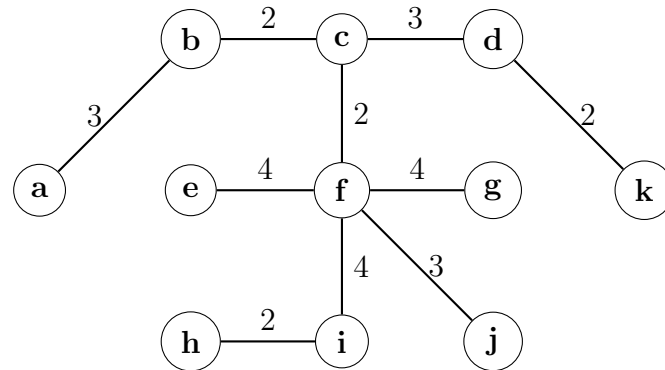2 2 2 2 3 3 3 4 4 4 4 5 5 5 5 ...

For the spanning tree, we should have $|V| - 1$ edges. There are 11 vertices, which means that to avoid a cycle, we should select 10 edges, and include all 11 vertices.

We can select the edges in this order:

(2)(b-c), (2)(c-f), (2)(h-i), (2)(d-k), (3)(a-b), (3)(c-d), (3)(f-j), (4)(e-f), (4)(f-i) (4)(f-g).

We selected 10 edges, and any of them cause a cycle, so we are done.

b)

c)

No, the spanning tree is not unique because there is still an unused edge that doesn't cause any cycle and can be used in another minimum spanning tree. For example, if we change the (4)(f-g) edge with the (4)(g-j) edge, we have again a minimum spanning tree. Notice that they have the same weight. Hence, the minimum spanning tree is not unique.