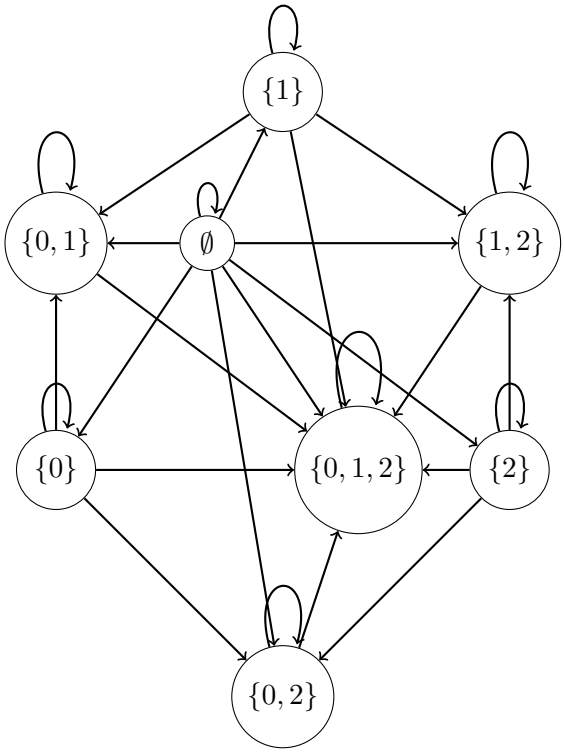
Student Information

Full Name : Onat ÖZDEMİR
Id Number : 2310399

Answer 1

a)



b)

In order to prove (S,R) is a poset, firstly we must prove that R is a reflexive, antisymmetric and transitive relation on S.
Proof of R is Reflexive: In order to prove R is a reflexive relation, we have to prove that $\forall x \in S\ (xRx)$. According to the theorem placed in Rosen's Discrete Mathematics and Its Applications book, page 120, THEOREM1 for every set A, $A \subseteq A$ satisfies. Then, according to the definition of S, since each $x \in S$ is a set,

$$\forall x \in S\ (x \subseteq x)$$

Then,

$$\forall x \in S\ (xRx)$$

Thus, by the definition, we have proved that R is a reflexive relation on S.
Proof of R is Antisymmetric: In order to prove R is an antisymmetric relation, we have to prove that $\forall x, y \in S\ ((xRy \wedge yRx) \rightarrow (x = y))$. Let x and y be arbitrarily chosen elements of S, if $x \subseteq y$ and $y \subseteq x$ then by the Axiom of Extensionality we can conclude that $x = y$. Since we choose x and y

arbitrarily then, we can conclude that $\forall x, y \in S\ ((x \subseteq y \wedge y \subseteq x) \rightarrow (x = y))$. Hence, we have proved that $\forall x, y \in S\ ((xRy \wedge yRx) \rightarrow (x = y))$. Thus, by the definition, R is an antisymmetric relation.
Proof of R is Transitive: In order to prove R is a transitive relation, we need to prove that $\forall x, y, z \in S\ ((xRy \wedge yRz) \rightarrow xRz)$. Let $x, y, z \in S$ be arbitrarily chosen sets such that $x \subseteq y$ and $y \subseteq z$. Assume $k \in x$ where k is arbitrarily chosen. Since $x \subseteq y$, by the definition of subset we know that $k \in x \rightarrow k \in y$. Then, from $k \in x$ and $k \in x \rightarrow k \in y$ we can conclude that $k \in y$ by modus ponens. Since $y \subseteq z$, by the definition of subset we know that $k \in y \rightarrow k \in z$. Then, from $k \in y$ and $k \in y \rightarrow k \in z$ we can conclude that $k \in z$ by modus ponens. We assumed $k \in x$ and proved that $k \in z$ hence we can conclude that $k \in x \rightarrow k \in z$. Since k is arbitrarily chosen, we can conclude that $\forall k(k \in x \rightarrow k \in z)$. Hence, by the definition of subset, $x \subseteq z$. We assumed $x \subseteq y$ and $y \subseteq z$, then proved that $x \subseteq z$. Hence, since x, y, z are arbitrarily chosen, we have proved $\forall x, y, z \in S\ ((xRy \wedge yRz) \rightarrow xRz)$. Thus, R is a transitive relation.

Since we have proved R is a reflexive, antisymmetric and transitive relation on S, by the definition of partial ordering, we have proved that R is a partial order.
Thus, we have proved that (S,R) is a poset.

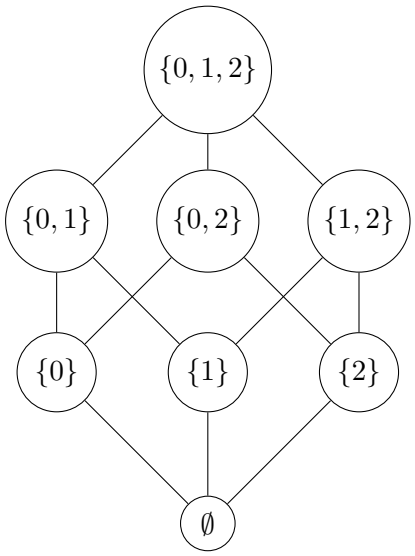
c)

By the definition, if (S,R) is a total order then (S,R) must be a poset and every two elements of S must be comparable under R. We have already proved that (S,R) is a poset in 1-b part. So we need to check whether every two elements of S is comparable or not. By the definition, if every two elements of S is comparable then,

$$\forall x, y \in S\ (xRy \vee yRx) \tag{1}$$

must be satisfied. Let x be $\{0\}$ and y be $\{1\}$ where $x, y \in S$. Since $\{0\} \not\subseteq \{1\}$ and $\{1\} \not\subseteq \{0\}$, we can conclude that $\{0\}R\{1\} \vee \{1\}R\{0\}$ is false. So, we have showed that there exist at least one pair of elements of S such that those elements are not comparable under R relation. Hence, (1) is not satisfied. Since, (1) is not satisfied, by the definition of total ordering, (S,R) is not a total order.

d)



Maximal elements: $\{0, 1, 2\}$
Minimal elements: $\emptyset$

e)

Since we have proved that (S,R) is a poset in the b part, inorder to show (S,R) is lattice it is enough to show that every pair of elements of S has both a least upper bound and greatest lower bound. Pair of elements and their least upper bounds and greatest lower bounds can be seen below;

Pair	L.U.B	G.L.B
$\emptyset$ and $\{0\}$	$\{0\}$	$\emptyset$
$\emptyset$ and $\{1\}$	$\{1\}$	$\emptyset$
$\emptyset$ and $\{2\}$	$\{2\}$	$\emptyset$
$\emptyset$ and $\{0,1\}$	$\{0,1\}$	$\emptyset$
$\emptyset$ and $\{1,2\}$	$\{1,2\}$	$\emptyset$
$\emptyset$ and $\{0,2\}$	$\{0,2\}$	$\emptyset$
$\emptyset$ and $\{0,1,2\}$	$\{0,1,2\}$	$\emptyset$
$\{0\}$and $\{1\}$	$\{0,1\}$	$\emptyset$
$\{0\}$and $\{2\}$	$\{0,2\}$	$\emptyset$
$\{0\}$and$\{0,1\}$	$\{0,1\}$	$\{0\}$
$\{0\}$and$\{1,2\}$	$\{0,1,2\}$	$\emptyset$
$\{0\}$and$\{0,2\}$	$\{0,2\}$	$\{0\}$
$\{0\}$and$\{0,1,2\}$	$\{0,1,2\}$	$\{0\}$
$\{1\}$and $\{2\}$	$\{1,2\}$	$\emptyset$
$\{1\}$and $\{0,1\}$	$\{0,1\}$	$\{1\}$
$\{1\}$and $\{0,2\}$	$\{0,1,2\}$	$\emptyset$
$\{1\}$and $\{1,2\}$	$\{1,2\}$	$\{1\}$
$\{1\}$and $\{0,1,2\}$	$\{0,1,2\}$	$\{1\}$
$\{2\}$and$\{0,1\}$	$\{0,1,2\}$	$\emptyset$
$\{2\}$and$\{1,2\}$	$\{1,2\}$	$\{2\}$
$\{2\}$and$\{0,2\}$	$\{0,2\}$	$\{2\}$
$\{2\}$and$\{0,1,2\}$	$\{0,1,2\}$	$\{2\}$
$\{0,1\}$and$\{1,2\}$	$\{0,1,2\}$	$\{1\}$
$\{0,1\}$and$\{0,2\}$	$\{0,1,2\}$	$\{0\}$
$\{0,1\}$and$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1\}$
$\{0,2\}$and$\{1,2\}$	$\{0,1,2\}$	$\{2\}$
$\{0,2\}$and$\{0,1,2\}$	$\{0,1,2\}$	$\{0,2\}$
$\{1,2\}$and$\{0,1,2\}$	$\{0,1,2\}$	$\{1,2\}$
$\emptyset$and$\emptyset$	$\emptyset$	$\emptyset$
$\{0\}$and$\{0\}$	$\{0\}$	$\{0\}$
$\{1\}$and$\{1\}$	$\{1\}$	$\{1\}$
$\{2\}$and$\{2\}$	$\{2\}$	$\{2\}$
$\{0,1\}$and$\{0,1\}$	$\{0,1\}$	$\{0,1\}$
$\{1,2\}$and$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
$\{0,2\}$and$\{0,2\}$	$\{0,2\}$	$\{0,2\}$
$\{0,1,2\}$and$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$

By looking the table above, we can easily seen that every pair of elements of S has both least upper bound and greatest lower bound. Additionally, by looking the table above, it can be easily seen that for arbitrarily chosen $x, y \in S$ least upper bound of x and y equal to $x \cup y$ while the greatest lower bound of x and y equal to $x \cap y$. Since we are able to find least upper bound and greatest lower bound for every pair of elements of S, we have showed that (S,R) is a lattice.

Answer 2

a)

Initial Vertex	Terminal Vertices
a	
b	a, c
c	f
d	a, c, d, e, g
e	c, f, g
f	b
g	d

b)

Rows and columns are ordered as a,b,c,d,e,f,g

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

c)

$deg^-(a) = 2$
$deg^+(a) = 0$
$deg^-(b) = 1$
$deg^+(b) = 2$
$deg^-(c) = 3$
$deg^+(c) = 1$
$deg^-(d) = 2$
$deg^+(d) = 5$
$deg^-(e) = 1$
$deg^+(e) = 3$
$deg^-(f) = 2$
$deg^+(f) = 1$
$deg^-(g) = 2$
$deg^+(g) = 1$

d)

e,c,f,b,a
d,e,f,b,a
d,c,f,b,a
e,g,d,c,f
d,e,f,b,c
g,d,c,f,b

e)

c,f,b,c
d,e,g,d
d,g,d,d
f,b,c,f
b,c,f,b
e,g,d,e
g,d,e,g
g,d,d,g
d,d,g,d

f)

To prove that G is weakly connected, we need to show that there is a path between every two distinct vertices in the underlying undirected graph. All possible distinct vertex pairs of G and paths between those vertex pairs can be seen from the table below;

Vertex Pairs	Paths
a and b	a, b
a and c	a, b, c
a and d	a, d
a and e	a, d, e
a and f	a, d, e, f
a and g	a, d, e, g
b and c	b, c
b and d	b, c, d
b and e	b, c, e
b and f	b, f
b and g	b, c, e, g
c and d	c, d
c and e	c, e
c and f	c, f
c and g	c, e, g
d and e	d, e
d and f	d, e, f
d and g	d, e, g
e and f	e, f
e and g	e, g
f and g	f, e, g

Hence, by observing the table given above, we have proved that there exists a path between every two distinct vertices in the underlying undirected graph. Thus, by the definition G is weakly connected.

g)

$\{a\}, \{b, c, f\}, \{d, e, g\}$

h)

Let adjacency matrix of H be A where A is,
(Rows and columns are ordered as d,e,f,g)

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

To find number of different paths of length 3 from d to g in H, we can use the THEOREM2 placed in Rosen's Discrete Mathematics and Its Applications book page 688. Then by using matrix multiplication we should find the A^3,

$$A^3 = \begin{bmatrix} 4 & 2 & 1 & 3 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 2 \end{bmatrix}$$

Then, by using to theorem mentioned above, number of different paths of length 3 from d to g in H which is represented as N,

$$N = \ (1,4)th \ entry \ of \ A^3 \ = 3$$

Thus, number of different paths of length 3 from d to g in H is 3.

Answer 3

Firstly, let we entitle each edge of G as,

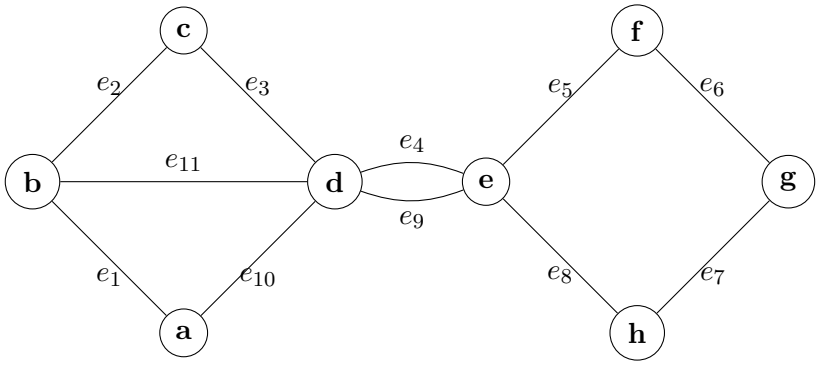


Figure 1: G

a)

Since the path which can be represented as e_1, e_{10}, e_{11}, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 doesn't contain the same edge more than once, by the definition given path is a simple path. Moreover, since the given path is simple and containing every edge of G then by the definition, the given path is an Euler Path. Thus, we have proved that G has an Euler Path.

b)

Since G is an undirected graph and we can find a path between every pair of distinct vertices of G, then by the definition G is a connected graph. Since G is a connected multigraph then we can use the

THEOREM 1 placed in the Rosen's Discrete Mathematics and Its Applications book page 696. Assume G has a Euler circuit then by the given theorem we can conclude that each vertices of G must have even degree. However, degree of b is 3 which is not an even degree. Since this fact contradicts with our assumption, by using proof by contradiction we have proved that G doesn't have any Euler circuit.

c)

Since the path which can be represented as e_1, e_2, e_3, e_4, e_5, e_6, e_7 doesn't contain the same edge more than once, by the definition given path is a simple path. Moreover, since the given path is simple and passes through every vertex of G exactly once then by the definition given path is a Hamiltonian Path. Thus, we have proved that G has a Hamiltonian Path.

d)

Since the degrees of a, c, h, f are equal to 2, every possible circuits that include all vertices of G must have $e_2, e_3, e_1, e_{10}, e_5, e_6, e_7, e_8$ edges. Let's assume that G has a Hamiltonian Circuit, then by the definition, the hamiltonian circuit of G must pass through every vertex exactly once. Since every possible circuits that include all vertices of G must have $e_2, e_3, e_1, e_{10}, e_5, e_6, e_7, e_8$ edges, then the Hamiltonian Circuit of G must also have $e_2, e_3, e_1, e_{10}, e_5, e_6, e_7, e_8$ edges. However, since one of the endpoints of e_3, e_{10}, e_5 and e_8 is d, there isn't any circuit that include all vertices of G and pass through d exactly once. Since this fact contradicts with our assumption, by using the proof by contradiction, G doesn't have any Hamiltonian Circuit.

Answer 4

Firstly, let $V = \{a, b, c, d, e\}$ for $G = (V, E)$ and $V' = \{a', b', c', d', e'\}$ for $G' = (V', E')$ also we know that both G and G' are simple graphs.If we can find a one to one and onto f function where $f : V \to V'$ such that x and y vertices are adjacent in G if and only if $f(x)$ and $f(y)$ are adjacent in G', then by the definition we can prove that G and G' are isomorphic.
Let $f : V \to V'$ be a one to one and onto function such that,

$$f(a) = a', \qquad f(b) = b', \qquad f(c) = c', \qquad f(d) = d', \qquad f(e) = e'$$

Then, we should check that x and y are adjacent in G if and only if $f(x)$ and $f(y)$ are adjacent in G',

Table 1: Adjacency Comparison

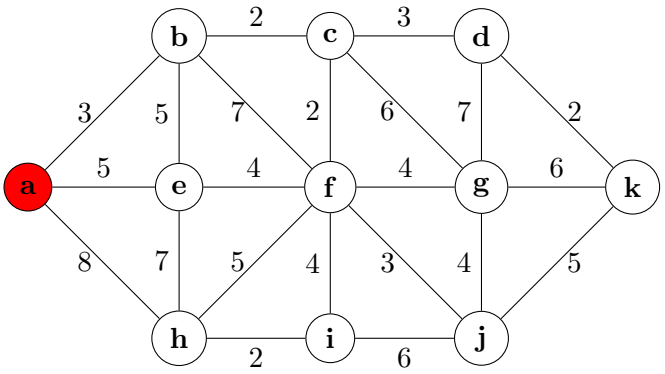
Adjacent Vertices in G	Adjacent Vertices in G'
a and b are adjacent	f(a)=a' and f(b)=b' are adjacent
a and e are adjacent	f(a)=a' and f(e)=e' are adjacent
b and c are adjacent	f(b)=b' and f(c)=c' are adjacent
c and d are adjacent	f(c)=c' and f(d)=d' are adjacent
d and e are adjacent	f(d)=d' and f(e)=e' are adjacent

By observing Table 1, we can easily see that for all x and y where x and y are vertices in G, if x and y are adjacent then $f(x) = x'$ and $f(y) = y'$ are adjacent in G'. Moreover, for all $f(x) = x'$ and $f(y) = y'$ where x' and y' are vertices in G', if $f(x) = x'$ and $f(y) = y'$ adjacent, then x and y are adjacent in G. Hence, we have proved that there exist a one to one and onto f function where $f : V \to V'$ such that x and y vertices are adjacent in G if and only if $f(x)$ and $f(y)$ are adjacent in G'. Thus, by the definition, we have proved that G and G' are isomorphic.

Answer 5

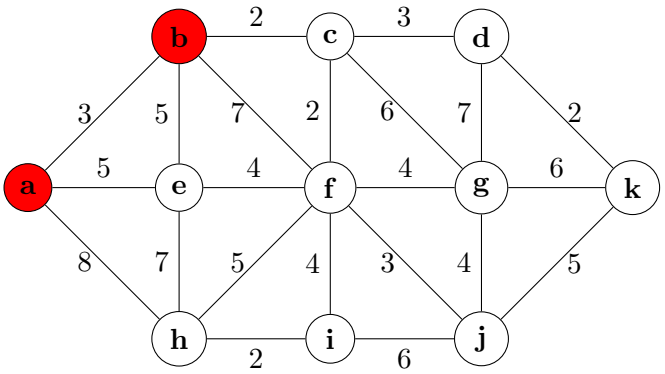
a)

We will start from the vertex a and initialize it with distance 0. All other vertices will be initialized with infinite distances. In each step, we will choose the unvisited vertex with the lowest distance as our visited vertex (v) and update the distance value of each u vertex adjacent to v as $dist(v) + weight(v, u)$ if $dist(v) + weight(v, u) < dist(u)$.In each step, visited nodes represented with red color, current distance values and current shortest paths from a to each node can be seen from the tables. When we visit the j and color it to red that means we found the shortest distance and the path placed in the table is the shortest path from a to j.



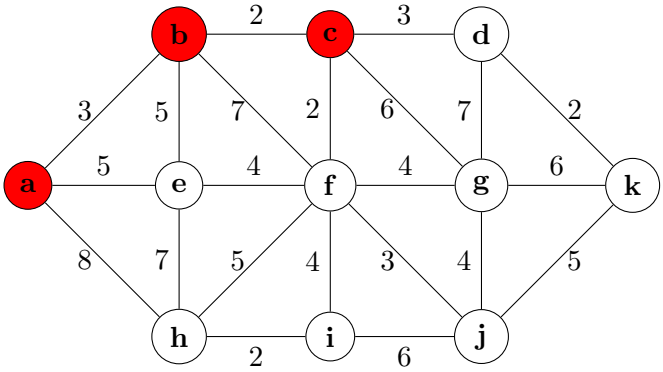
Vertex	Distance	Path
a	0	
b	3	a,b
c	inf	
d	inf	
e	5	a,e
f	inf	
g	inf	
h	8	a,h
i	inf	
j	inf	
k	inf	

Since b is the unvisited vertex with the lowest distance, choose it as next visited vertex.



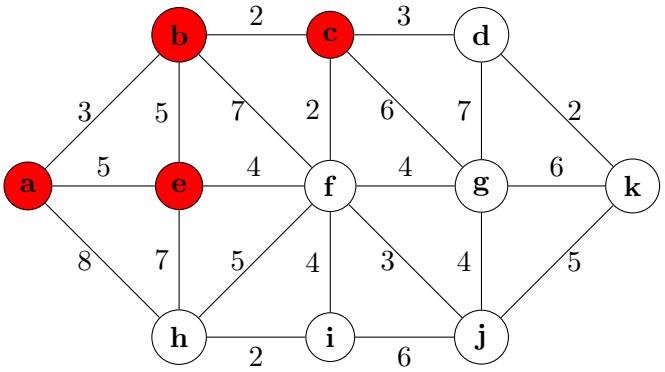
Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	inf	
e	5	a,e
f	10	a,b,f
g	inf	
h	8	a,h
i	inf	
j	inf	
k	inf	

Both c and e has same and the lowest distances. So I choose c as the next visited vertex.



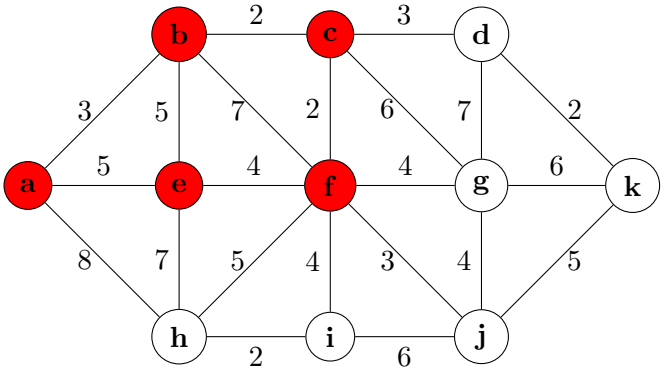
Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	8	a,b,c,d
e	5	a,e
f	7	a,b,c,f
g	11	a,b,c,g
h	8	a,h
i	inf	
j	inf	
k	inf	

e is the unvisited vertex with the lowest distance. So choose e as the next visited vertex.



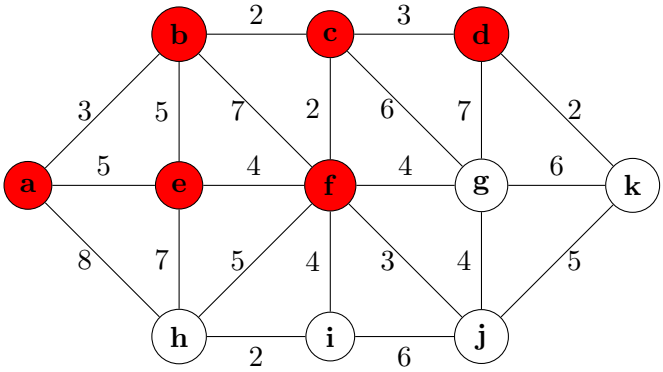
Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	8	a,b,c,d
e	5	a,e
f	7	a,b,c,f
g	11	a,b,c,g
h	8	a,h
i	inf	
j	inf	
k	inf	

f is the unvisited vertex with the lowest distance. So choose f as the next visited vertex.



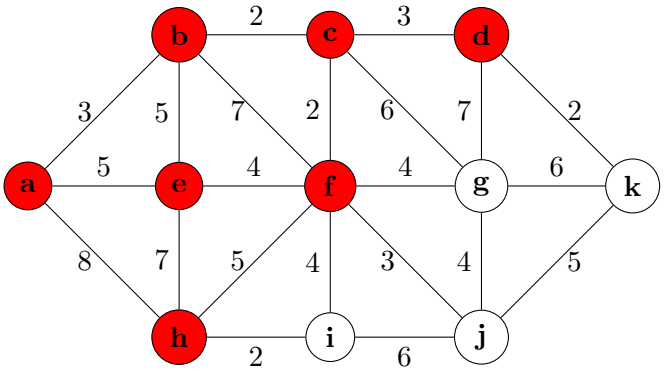
Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	8	a,b,c,d
e	5	a,e
f	7	a,b,c,f
g	11	a,b,c,g
h	8	a,h
i	11	a,b,c,f,i
j	10	a,b,c,f,j
k	inf	

Both d and h has same and the lowest distances. So I choose d as the next visited vertex.



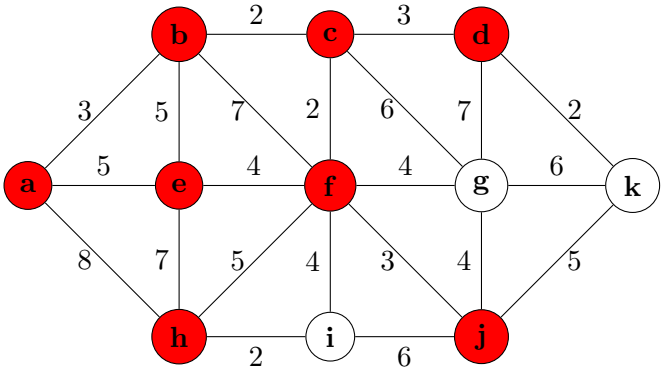
Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	8	a,b,c,d
e	5	a,e
f	7	a,b,c,f
g	11	a,b,c,g
h	8	a,h
i	11	a,b,c,f,i
j	10	a,b,c,f,j
k	10	a,b,c,d,k

h is the unvisited vertex with the lowest distance. So choose h as the next visited vertex.



Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	8	a,b,c,d
e	5	a,e
f	7	a,b,c,f
g	11	a,b,c,g
h	8	a,h
i	10	a,h,i
j	10	a,b,c,f,j
k	10	a,b,c,d,k

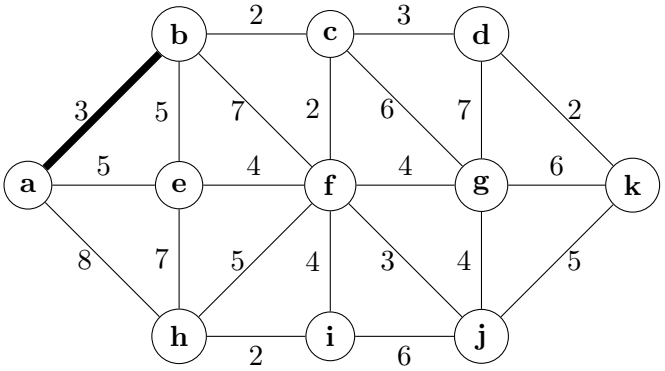
i, j and k has same and the lowest distance value. So I choose j as the next visited vertex.



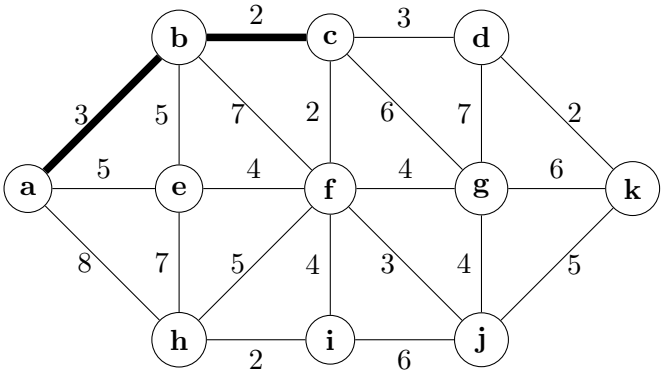
Vertex	Distance	Path
a	0	
b	3	a,b
c	5	a,b,c
d	8	a,b,c,d
e	5	a,e
f	7	a,b,c,f
g	11	a,b,c,g
h	8	a,h
i	10	a,h,i
j	10	a,b,c,f,j
k	10	a,b,c,d,k

Since we have visited j, we have found the shortest path from a to j by applying Dijkstra's Shortest Path Algorithm. According to the table shortest distance between a and j is 10 and the shortest path from a to j is a,b,c,f,j.

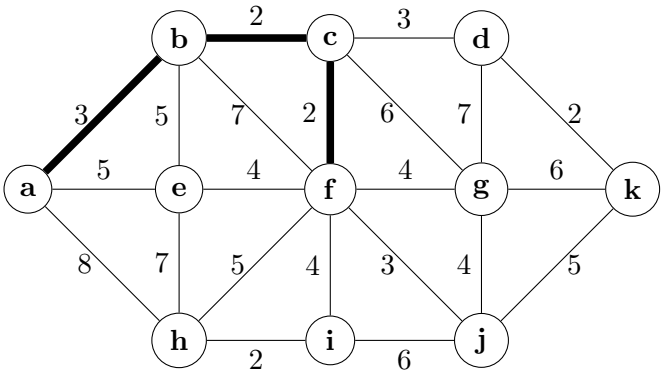
b)



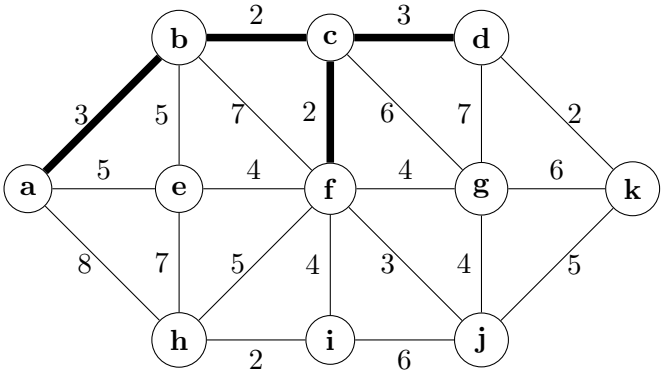
Since we chose a as a root vertex, to apply Prim's rule we should find the edge of minimum weight incident to a. Since the weight of a, b is smaller than the other edges incident to a and a, b doesn't cause a simple circuit on the spanning tree, we should add a, b to our spanning tree and b is now visited.



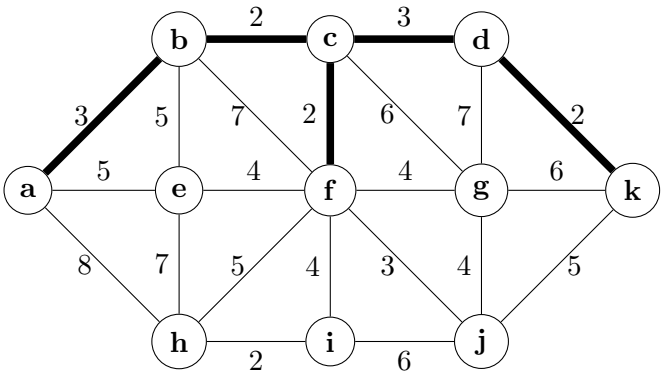
a, h, a, e, b, e, b, f and b, c are incident edges to the current vertices which are a and b of our spanning tree. Since the weight of b, c is lower than the other incident edges and doesn't cause a simple circuit on the spanning tree, according to Prim's Algorithm we should add b, c to our spanning tree and c is now visited.



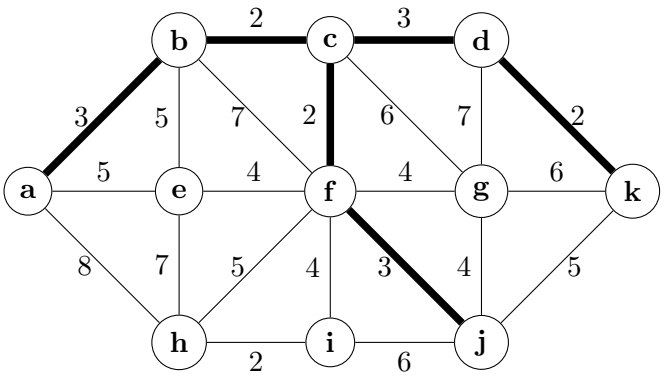
If we control each incident edges to the current vertices our spanning tree has, we can easily see that the one with smallest weight is c, f whose weight is 2. Since c, f has the smallest weight among the incident edges and doesn't create any simple circuits on the spanning tree according to the Prim's Algorithm we should add c, f to our spanning tree.



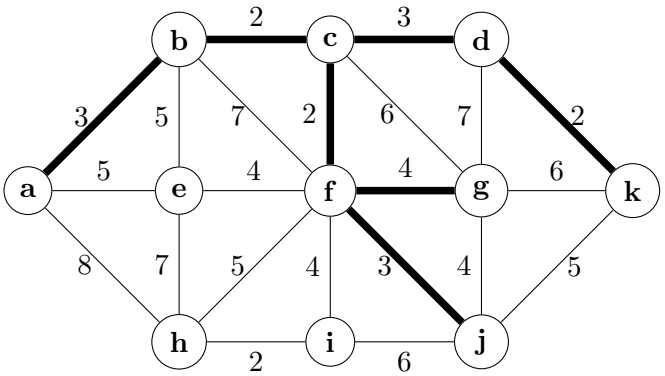
If we control each incident edges to the current vertices our spanning tree has, we can easily see that both c, d and f, j has the lowest weight which is 3. Since both c, d and f, j doesn't create any simple circuits on the spanning tree according to the Prim's Algorithm, we can add either of them, so that at this point I choose to add c, d to our spanning tree.



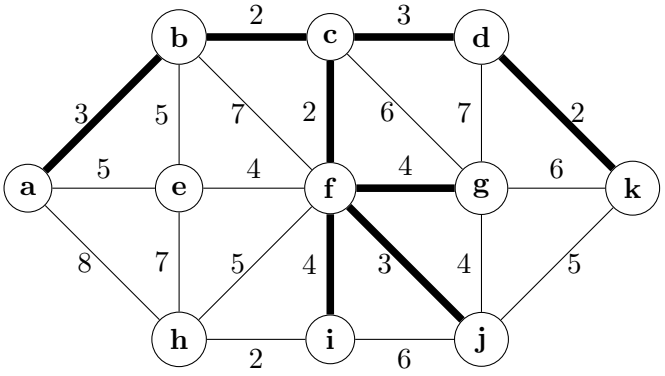
If we control each incident edges to the current vertices our spanning tree has, we can easily see that d, k has the lowest weight which is 2. Since d, k doesn't create any simple circuits on the spanning tree according to the Prim's Algorithm we should add d, k to the spanning tree.



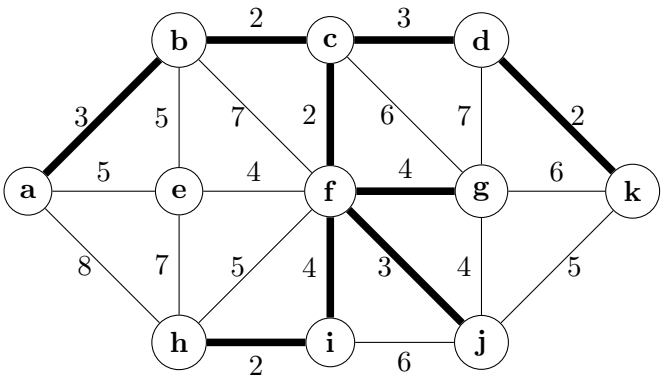
If we control each incident edges to the current vertices our spanning tree has, we can easily see that f, j has the lowest weight which is 3. Since f, j doesn't create any simple circuits on the spanning tree according to the Prim's Algorithm we should add f, j to the spanning tree.



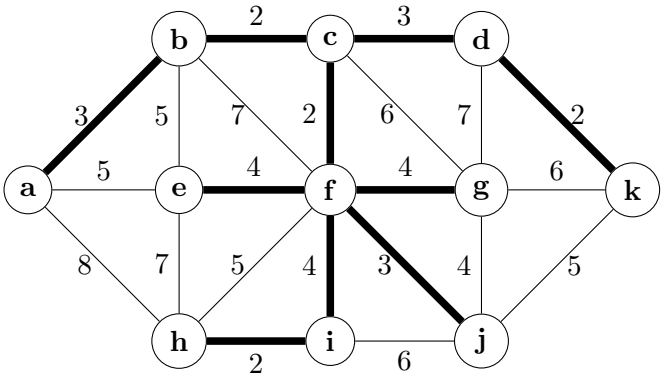
If we control each incident edges to the current vertices our spanning tree has, we can easily see that f, g, f, i, f, e have the lowest weight which is 4. Since each of those edges doesn't create any simple circuits on the spanning tree according to the Prim's Algorithm we can add any of them to the spanning tree. At this point, I choose f, g to add our spanning tree.



If we control each incident edges to the current vertices our spanning tree has, we can easily see that g, j, f, i, f, e have the lowest weight which is 4. However, we can't add g, j to the spanning tree since it creates simple circuit on the spanning tree. Both f, i and f, e doesn't cause simple circuit on the spanning tree. At this point, I choose f, i to add the spanning tree.

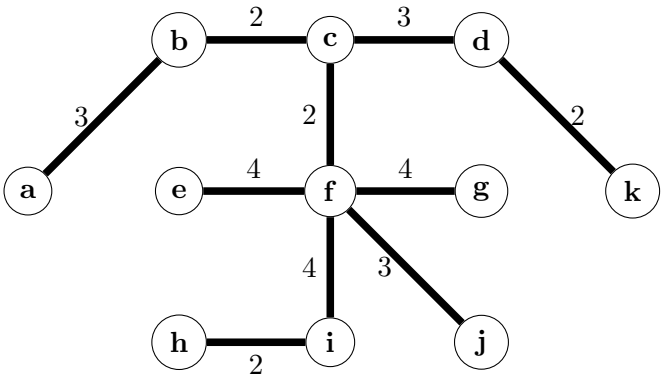


If we control each incident edges to the current vertices our spanning tree has, we can easily see that i, h has the lowest weight which is 2. Since i, h doesn't create any simple circuit on the spanning tree, we should add it to the spanning tree.



At this point, we have visited all the vertices on the graph except e. If we control each incident edges to the current vertices our spanning tree has, we can easily see that g, j and f, e has the lowest weight which is 4. However, since g, j create simple circuit on the spanning tree, according to Prim's Algorithm we can't add it to the spanning tree. So we should add f, e to the spanning tree.

Since we add *vertex number* $-1 = 10$ edges to our spanning tree we should stop, we obtained our spanning tree using Prim's Algorithm;



Answer 6

a)

Number of Vertices: 7
Number of Edges: 6
Height of the T: 3

b)

Preorder: a,b,c,d,e,f,g

c)

Postorder: b,d,f,g,e,c,a

d)

Inorder: b,a,d,c,f,e,g

e)

Since every internal vertex of T which are a,c and e has exactly two children, by the definition T is a full binary tree.

f)

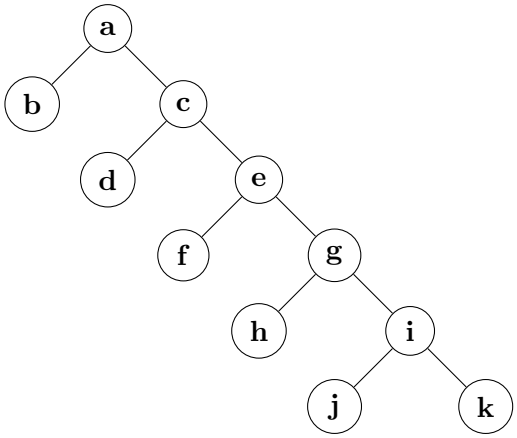
Since b has no children, level 2 is not completely full. Hence, by the definition, T is not a complete binary tree.

g)

According to the definition of binary search tree, for an arbitrary binary tree which can be entitled as A, if A is a binary search tree under comparison with respect to $\leq$ relation defined on $Z \times Z$ then for every chosen vertex of A, key of the chosen vertex must not be smaller than the keys of all vertices in its left subtree and must not be bigger than the keys of all vertices in its right subtree. As can be observed, the key of c vertex of T which is 24 is bigger than the key of f which placed in the right subtree of c. Hence, T doesn't satisfy the definition of binary search tree under comparison with respect to $\leq$ relation. Thus, T is not a binary search tree.

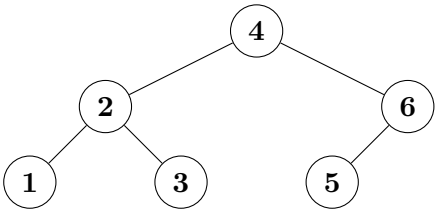
h)

Minimum number of nodes for a full binary tree with height 5 can be obtained from the below case;



As can be seen, given tree is a full binary tree with 5 height and 11 nodes. Thus, minimum number of nodes for a full binary tree with height 5 is 11.

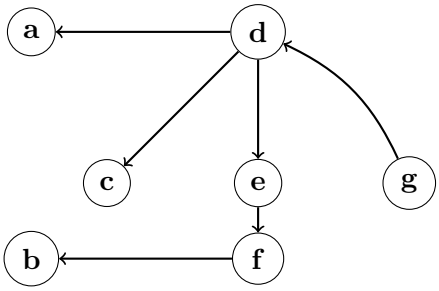
i)



j)

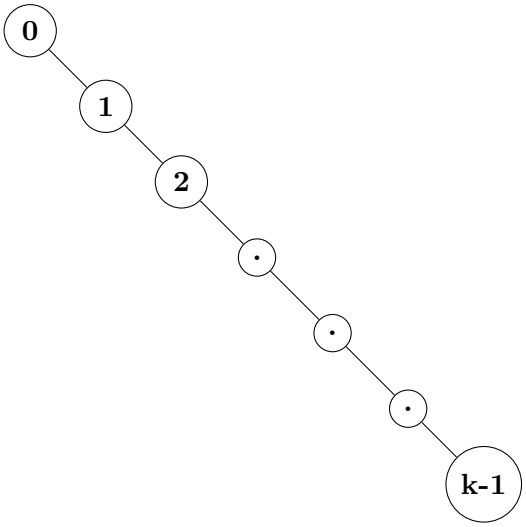
For searching 1: 4,2,1
For searching 6: 4,6

k)



l)

To obtain the maximum height, binary search tree should be in the linear form.For instance;



In the described case, height of the tree would be equal to the number of edges which is k-1. Thus, maximum height to create a binary search tree containing k vertices is k-1.