

Student Information

Full Name : Murat Bolu

Id Number : 2521300

Answer 1

The graph G can be represented by a pair (V, E) where V is the set of vertices, and E is the set of edges. $V = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}\}$ and $E = \{\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{a}, \mathbf{c}\}, \{\mathbf{a}, \mathbf{e}\}, \{\mathbf{b}, \mathbf{c}\}, \{\mathbf{b}, \mathbf{e}\}, \{\mathbf{c}, \mathbf{d}\}, \{\mathbf{d}, \mathbf{e}\}\}$. Therefore, $|V| = 5$, and $|E| = 7$.

- a) $\sum_{v \in V} \deg(v) = 2 \cdot |E| = 14$ by handshaking lemma, since all edges increase the degree of vertices they connect by one. The sum of degrees of all vertices is twice the number of edges since all edges are counted twice.

b) Adjacency matrix is $M = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$ where the item M_{ij} represents the existence of

an edge between the vertices i and j . The vertices are enumerated in the order they are written above. There are 14 non-zero entries in the adjacency matrix representation of G .

c) Incidence matrix is $M = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$ where the item M_{ij} represents the incidence

of the vertex i with the edge j . The rows represent vertices and columns represent edges. The edges are enumerated in the order they are written above. There are 21 zero entries in the incidence matrix representation of G .

- d) There is not, since \mathbf{c} and \mathbf{e} are not connected, only one can be chosen, and only two choices are $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ and $\{\mathbf{a}, \mathbf{b}, \mathbf{d}, \mathbf{e}\}$, both of which are obviously not connected since \mathbf{d} is not connected to \mathbf{b} .
- e) No, it is not, since $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ are all connected, it is not possible to place them in two disjoint and independent sets such that there is no edge connecting between two elements in the same set. One can exhaust all possibilities by trying $\{\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}\}\}$, $\{\{\mathbf{a}, \mathbf{c}\}, \{\mathbf{b}\}\}$, $\{\{\mathbf{b}, \mathbf{c}\}, \{\mathbf{a}\}\}$. Since a subgraph of G is not bipartite, adding new vertices to that subgraph would not be able to fix that and therefore G is not bipartite.
- f) For every undirected edge, we can choose two directions, and therefore we have $2^{|E|} = 2^7 = 128$ different directed graphs that have G as their underlying graph.

- g) The vertices $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{e}\}$ have odd degrees, therefore we cannot visit all edges, i.e. create a Euler path, since when we enter a vertex, we have to leave it. The only exception to this rule is for the first and the last vertices, which there are two of. A simple path can be represented by the sequence of vertices such as $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{a}, \mathbf{c})$ in order. The path contains 6 edges in total. Therefore, the longest simple path is of length 6.
- h) There is only one, and it is the graph itself, since all vertices are connected, i.e. there exists a path between every pair of vertices. There are $\binom{5}{2} = 10$ pairs, and the path between each pair is as follows: $\{(\mathbf{a}, \mathbf{b}), (\mathbf{a}, \mathbf{c}), (\mathbf{a}, \mathbf{c}, \mathbf{d}), (\mathbf{a}, \mathbf{e}), (\mathbf{b}, \mathbf{c}), (\mathbf{b}, \mathbf{c}, \mathbf{d}), (\mathbf{b}, \mathbf{e}), (\mathbf{c}, \mathbf{d}), (\mathbf{c}, \mathbf{d}, \mathbf{e}), (\mathbf{d}, \mathbf{e})\}$.
- i) Since the graph does not have an Euler path as explained in item j), it does not have an Euler circuit since Euler circuits are a subset of Euler paths.
- j) The graph G does not have an Euler path, since there are 4 vertices with odd degrees, $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{e}\}$. There must be either no vertices with odd degrees or exactly two vertices with an odd degree to create an Euler path.
- k) Yes, it has many, one example being $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{a})$.
- l) Yes, it has many, one example being $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e})$.

Answer 2

The graphs G and H are isomorphic if and only if there exists a one-to-one and onto function f such that the domain of f is the set of vertices of G , the codomain of f is the set of vertices of H , and for all pairs of vertices u and v , u and v are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H .

The graph G can be represented by a pair (V_G, E_G) where V_G is the set of vertices, and E_G is the set of edges.

$$V_G = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}\}$$

$$E_G = \{\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{b}, \mathbf{c}\}, \{\mathbf{c}, \mathbf{d}\}, \{\mathbf{d}, \mathbf{e}\}, \{\mathbf{e}, \mathbf{a}\}\}$$

The graph H can be represented by a pair (V_H, E_H) where V_H is the set of vertices, and E_H is the set of edges.

$$V_H = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}', \mathbf{e}'\}$$

$$E_H = \{\{\mathbf{a}', \mathbf{b}'\}, \{\mathbf{b}', \mathbf{c}'\}, \{\mathbf{c}', \mathbf{d}'\}, \{\mathbf{d}', \mathbf{e}'\}, \{\mathbf{e}', \mathbf{a}'\}\}$$

Therefore, the function f exists and is as follows:

$$f : V_G \mapsto V_H, \quad f = \{(\mathbf{a}, \mathbf{a}'), (\mathbf{b}, \mathbf{b}'), (\mathbf{c}, \mathbf{c}'), (\mathbf{d}, \mathbf{d}'), (\mathbf{e}, \mathbf{e}')\}$$

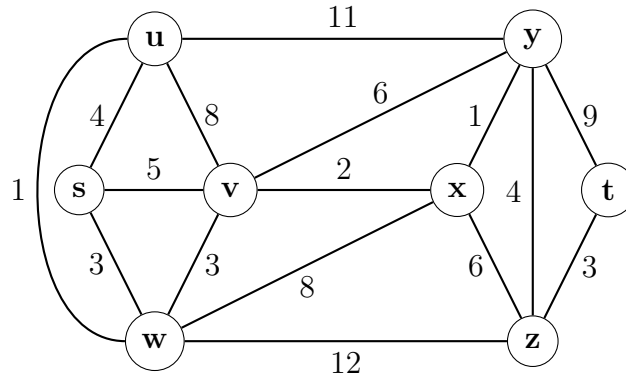
Answer 3

The graph G can be represented by a triple (V, E, W) where V is the set of vertices, E is the set of edges, and W is the weight function from the set of edges E to real numbers \mathbb{R} .

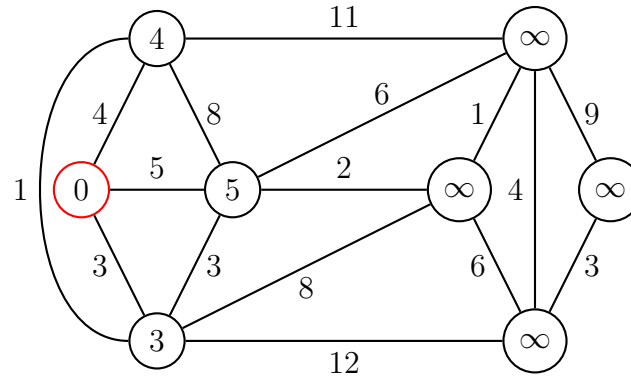
$$V = \{\mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$$

$$E = \{\{\mathbf{s}, \mathbf{u}\}, \{\mathbf{s}, \mathbf{v}\}, \{\mathbf{s}, \mathbf{w}\}, \{\mathbf{t}, \mathbf{y}\}, \{\mathbf{t}, \mathbf{z}\}, \{\mathbf{u}, \mathbf{v}\}, \{\mathbf{u}, \mathbf{w}\}, \{\mathbf{u}, \mathbf{y}\}, \\ \{\mathbf{v}, \mathbf{w}\}, \{\mathbf{v}, \mathbf{x}\}, \{\mathbf{v}, \mathbf{y}\}, \{\mathbf{w}, \mathbf{x}\}, \{\mathbf{w}, \mathbf{z}\}, \{\mathbf{x}, \mathbf{y}\}, \{\mathbf{x}, \mathbf{z}\}, \{\mathbf{y}, \mathbf{z}\}\}$$

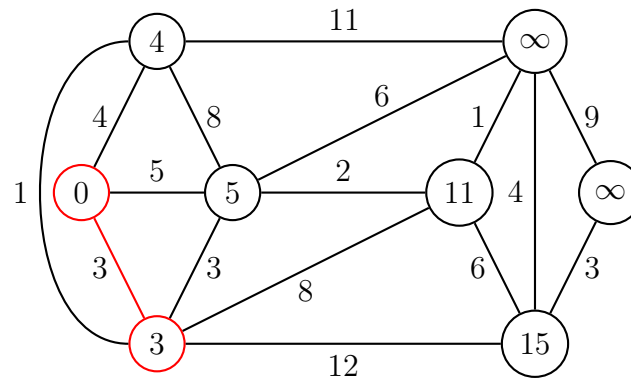
$$W : E \mapsto \mathbb{R}, \quad W = \{(\{\mathbf{s}, \mathbf{u}\}, 4), (\{\mathbf{s}, \mathbf{v}\}, 5), (\{\mathbf{s}, \mathbf{w}\}, 3), (\{\mathbf{t}, \mathbf{y}\}, 9), \\ (\{\mathbf{t}, \mathbf{z}\}, 3), (\{\mathbf{u}, \mathbf{v}\}, 8), (\{\mathbf{u}, \mathbf{w}\}, 1), (\{\mathbf{u}, \mathbf{y}\}, 11), \\ (\{\mathbf{v}, \mathbf{w}\}, 3), (\{\mathbf{v}, \mathbf{x}\}, 2), (\{\mathbf{v}, \mathbf{y}\}, 6), (\{\mathbf{w}, \mathbf{x}\}, 8), \\ (\{\mathbf{w}, \mathbf{z}\}, 12), (\{\mathbf{x}, \mathbf{y}\}, 1), (\{\mathbf{x}, \mathbf{z}\}, 6), (\{\mathbf{y}, \mathbf{z}\}, 4)\}$$



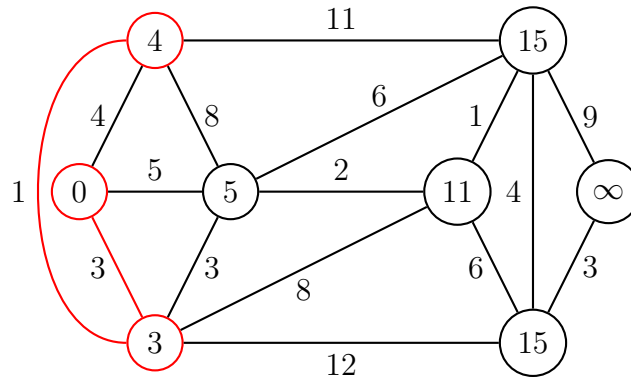
Let the distance to the starting vertex be zero. For all vertices other than the starting vertex, if they have an edge with the starting vertex, make the distance to them equal to the weight of that edge, otherwise mark the distance as infinity. Let the parent vertex of all vertices that have an edge with the starting vertex be the starting vertex. The parent vertex of the starting vertex is undefined, and the parent vertex of all other vertices are yet to be determined.



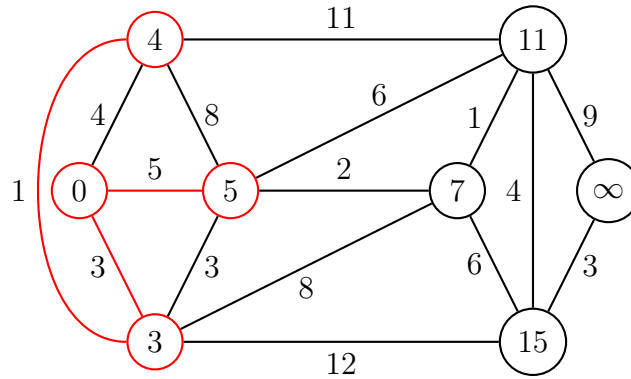
Since the vertex with the smallest distance is **w**, visit **w** and update the distances of **x** and **z** to be the sum of the distance of **w** and the weight of the edge between **w** and that vertex. Mark the parent vertex of **w** as **s**. The distance of **v** is not updated since the sum of the distance of **w** and the weight of the edge between **w** and **v**, which is 6, is greater than the current distance of **v**, which is 5.



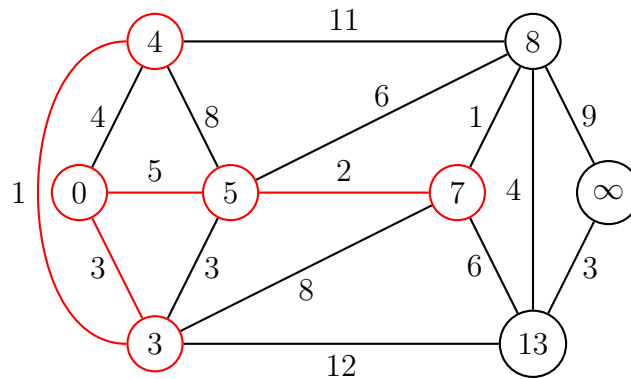
Since the unvisited vertex with the smallest distance is **u**, visit **u** and repeat the steps above.



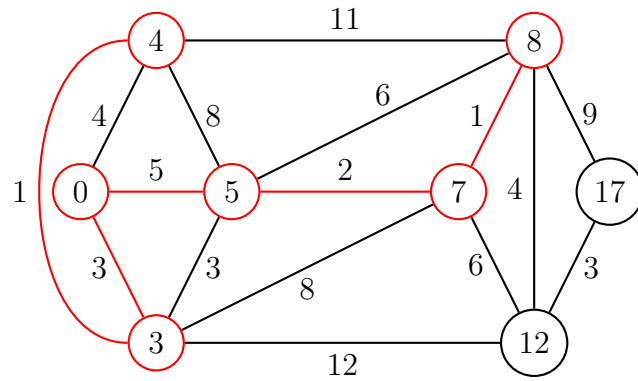
Visit the vertex **v** and repeat the steps above. Since its parent vertex was **s**, one must “color” the edge between **s** and **v**, not any other edge. This step is crucial to generate the correct shortest path from the starting vertex to the ending vertex.



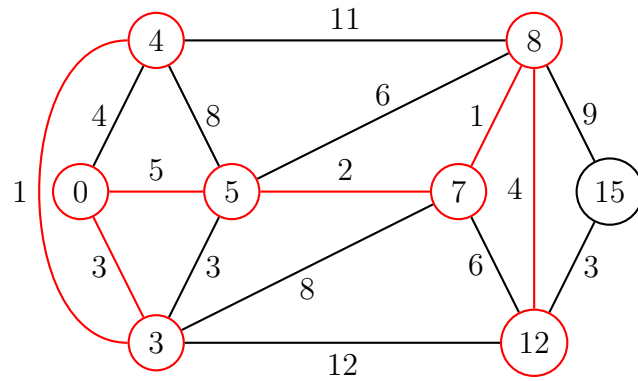
Visit the vertex **x** and repeat the steps above.



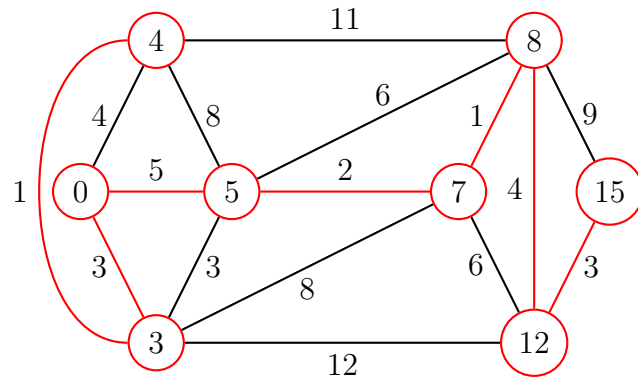
Visit the vertex **y** and repeat the steps above.



Visit the vertex **z** and repeat the steps above.



Visit the vertex **t** and halt the algorithm, since we visited the desired ending vertex.



Therefore, the shortest path between the vertex **s** and the vertex **t** is **(s, v, x, y, z, t)** and its length is 15.

Answer 4

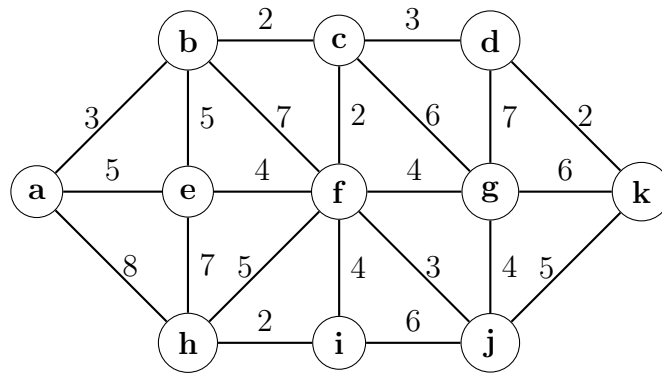
The graph G can be represented by a triple (V, E, W) where V is the set of vertices, E is the set of edges, and W is the weight function from the set of edges E to real numbers \mathbb{R} .

$$V = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$$

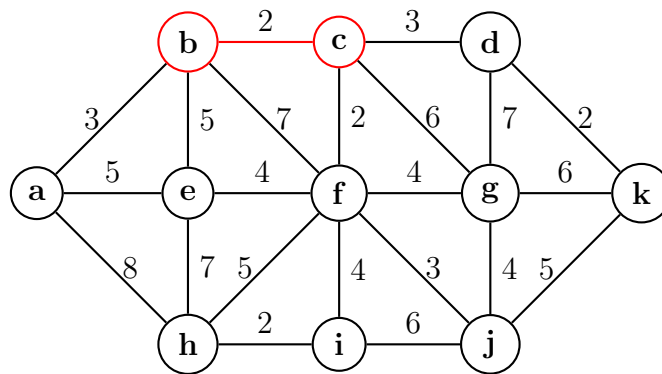
$$\begin{aligned} E = \{ & \{\mathbf{a}, \mathbf{b}\}, \{\mathbf{a}, \mathbf{e}\}, \{\mathbf{a}, \mathbf{h}\}, \\ & \{\mathbf{b}, \mathbf{c}\}, \{\mathbf{b}, \mathbf{e}\}, \{\mathbf{b}, \mathbf{f}\}, \\ & \{\mathbf{c}, \mathbf{d}\}, \{\mathbf{c}, \mathbf{f}\}, \{\mathbf{c}, \mathbf{g}\}, \\ & \{\mathbf{d}, \mathbf{g}\}, \{\mathbf{d}, \mathbf{k}\}, \\ & \{\mathbf{e}, \mathbf{f}\}, \{\mathbf{e}, \mathbf{h}\}, \\ & \{\mathbf{f}, \mathbf{g}\}, \{\mathbf{f}, \mathbf{h}\}, \{\mathbf{f}, \mathbf{i}\}, \{\mathbf{f}, \mathbf{j}\}, \\ & \{\mathbf{g}, \mathbf{k}\}, \{\mathbf{g}, \mathbf{j}\}, \\ & \{\mathbf{k}, \mathbf{j}\}, \\ & \{\mathbf{h}, \mathbf{i}\}, \\ & \{\mathbf{i}, \mathbf{j}\}, \} \end{aligned}$$

$$\begin{aligned} W : E \mapsto \mathbb{R}, \quad W = \{ & (\{\mathbf{a}, \mathbf{b}\}, 3), (\{\mathbf{a}, \mathbf{e}\}, 5), (\{\mathbf{a}, \mathbf{h}\}, 8), \\ & (\{\mathbf{b}, \mathbf{c}\}, 2), (\{\mathbf{b}, \mathbf{e}\}, 5), (\{\mathbf{b}, \mathbf{f}\}, 7), \\ & (\{\mathbf{c}, \mathbf{d}\}, 3), (\{\mathbf{c}, \mathbf{f}\}, 2), (\{\mathbf{c}, \mathbf{g}\}, 6), \\ & (\{\mathbf{d}, \mathbf{g}\}, 7), (\{\mathbf{d}, \mathbf{k}\}, 2), \\ & (\{\mathbf{e}, \mathbf{f}\}, 4), (\{\mathbf{e}, \mathbf{h}\}, 7), \\ & (\{\mathbf{f}, \mathbf{g}\}, 4), (\{\mathbf{f}, \mathbf{h}\}, 5), (\{\mathbf{f}, \mathbf{i}\}, 4), (\{\mathbf{f}, \mathbf{j}\}, 3), \\ & (\{\mathbf{g}, \mathbf{k}\}, 6), (\{\mathbf{g}, \mathbf{j}\}, 4), \\ & (\{\mathbf{k}, \mathbf{j}\}, 5), \\ & (\{\mathbf{h}, \mathbf{i}\}, 2), \\ & (\{\mathbf{i}, \mathbf{j}\}, 6), \} \end{aligned}$$

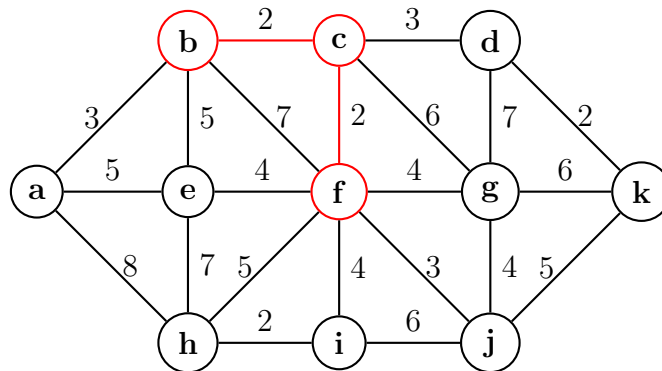
I am going to use Kruskal's algorithm to generate the minimum spanning tree.

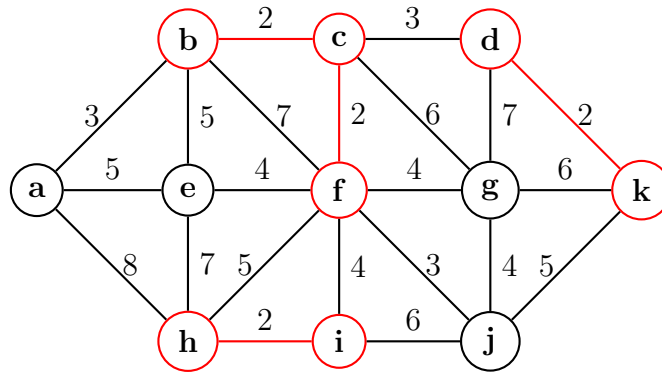
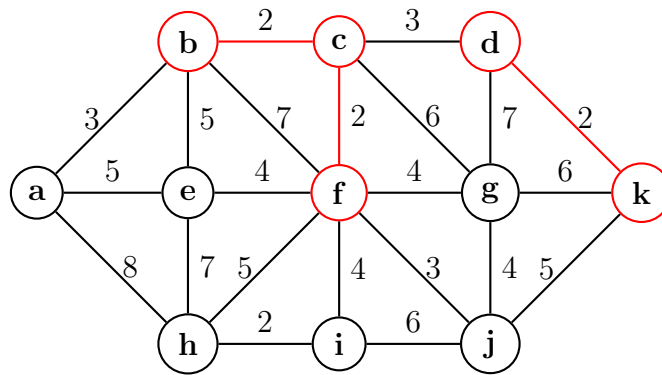


Create the minimum spanning tree by choosing the unchosen edges with the smallest weights that do not create a cycle within the previously created sub-trees. The smallest edges have the weight 2. We can choose any one of them, and in the end the total sum of the weights of the edges of the minimum spanning tree will be the same. However, the composition of the tree may be different.

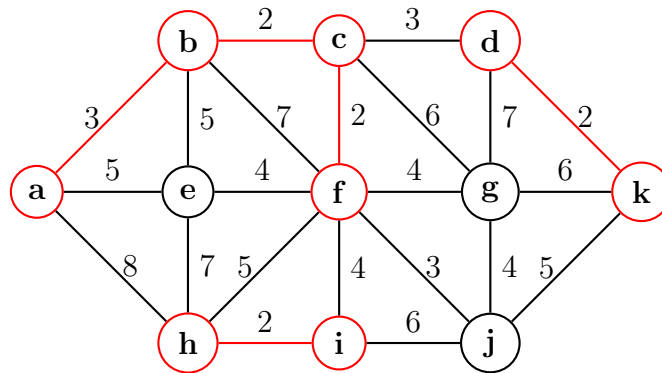


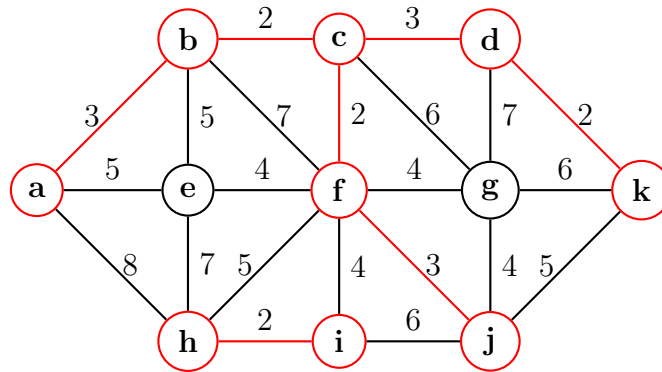
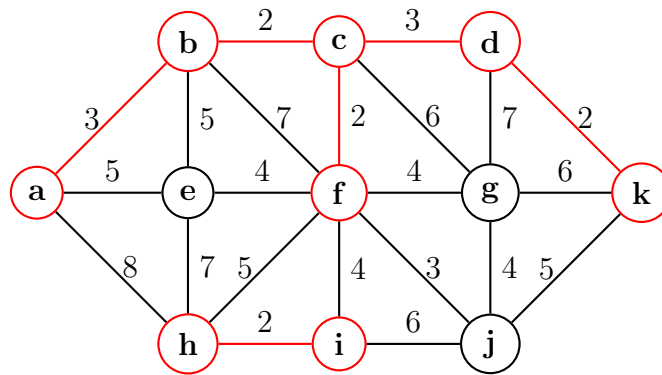
Continue adding the edges with the weight 2, making sure that edges do not create a cycle within a tree.



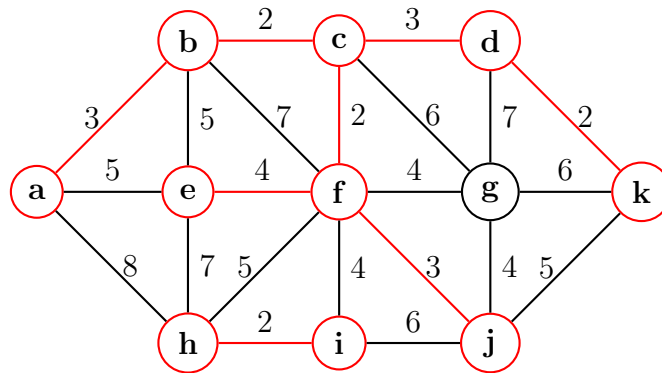


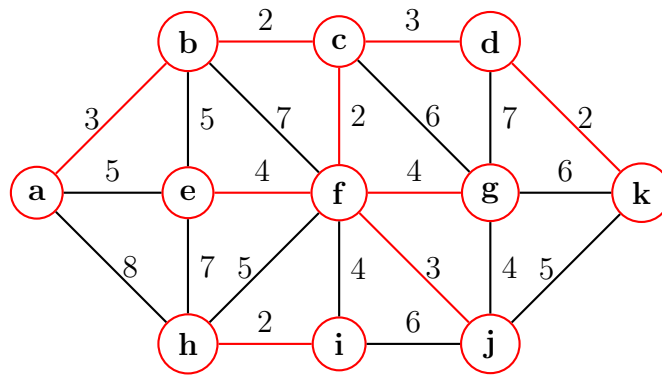
Since the all edges with weight 2 are exhausted, continue adding edges with the minimum weight, in this case, 3.





Continue with adding edges with weight 4, making sure that no cycles are created and the minimum spanning tree has not been generated yet.

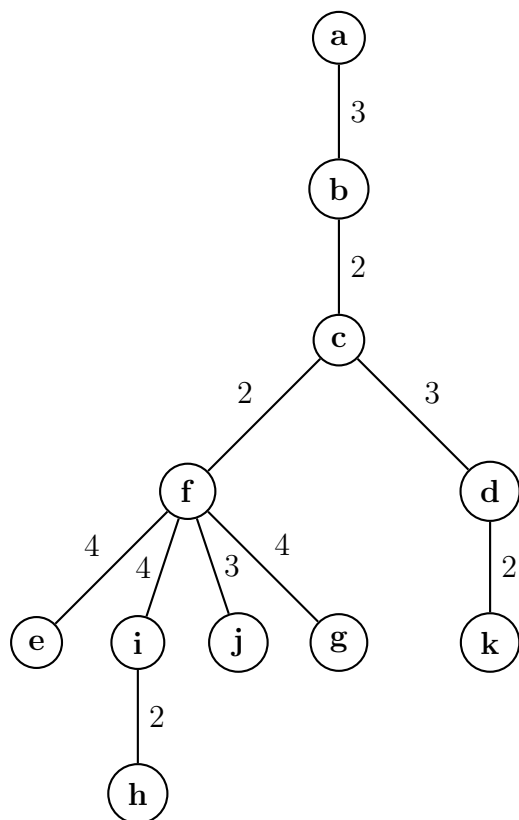




Since there are 11 vertices, and the number of edges added are 10, we must have generated our minimum spanning tree and we can halt the algorithm.

- a) The order edges are added is as follows: $\{b, c\}$, $\{c, f\}$, $\{d, k\}$, $\{h, i\}$, $\{a, b\}$, $\{c, d\}$, $\{f, j\}$, $\{e, f\}$, $\{f, g\}$, $\{f, i\}$.

b) The minimum spanning tree is as follows:



- c) It is not, because there were multiple edges with weight 4. We made arbitrary decisions, which affected the composition of the minimum spanning tree. We chose the edge $\{f, g\}$ to connect the vertex g to the tree, however, we could have chosen the edge $\{g, j\}$ which also has a weight of 4. That would have changed the composition of the minimum spanning tree. In that case, the minimum spanning would be as follows:

