Mathematics I

Week 11: Practice for graded assignment

Vicky Kumar Sharma
Course Instructor

Plan for this session

- How to join?
  - Join on webex - click on link sent to you
  - Join on pear deck - joinpd.com (enter code: see top right on webex)
  - Keep a notebook and pen ready for solving problems

- For every question - 5 to 15 minutes allotted
  - Question will be shown in a slide for solving
  - If you are done solving, enter your answer on joinpd.com
  - Presenter will provide a solution
  - Questions and discussion

Sample Question - your screen on joinpd.com

How to participate?
joinpd.com
code: see above

Description of the problem.

Question to be answered.

Desktop

Answer box

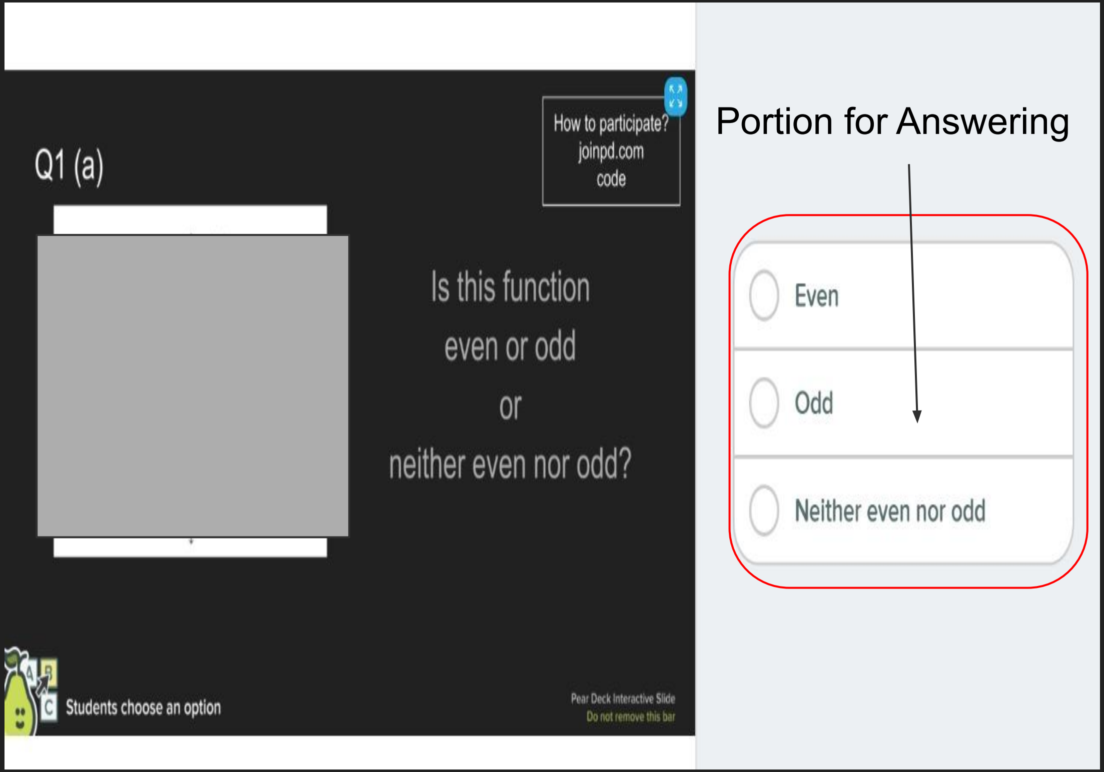
enter a number or a choice or some text

Mobile

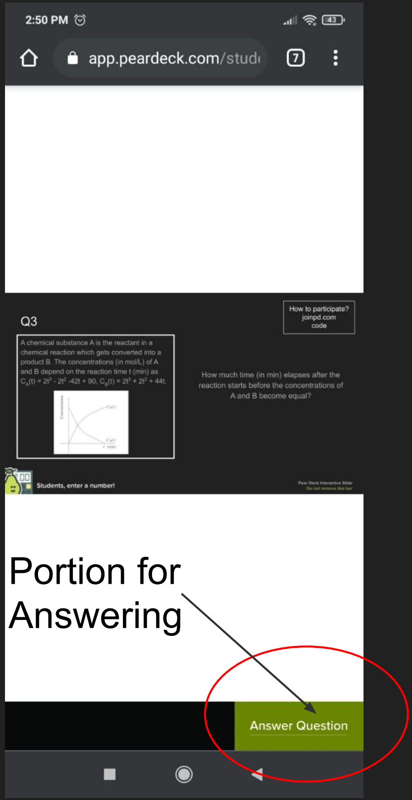
Answer question

Example Screenshots

Laptop/Desktop



Mobile



Test Problem

How to participate?
joinpd.com
code: see above

This is a problem to test if everything is working.

What is your favourite sport?

Dijkstra’s algorithm

How to participate?
joinpd.com
code: see above

Dijkstra’s algorithm is used to find the shortest path from a fixed source vertex to every other vertex in a graph G.

Dijkstra’s algorithm

How to participate?
joinpd.com
code: see above

Dijkstra’s algorithm is used to find the shortest path from a fixed source vertex to every other vertex in a graph G.

Facts:

- It is applicable for weighted graphs with positive weight edges.
- Not applicable for graphs having negative weight edges.

Dijkstra’s algorithm

How to participate?
joinpd.com
code: see above

Dijkstra’s algorithm is used to find the shortest path from a fixed source vertex to every other vertex in a graph G.

Facts:

- It is applicable for weighted graphs with positive weight edges.
- Not applicable for graphs having negative weight edges.

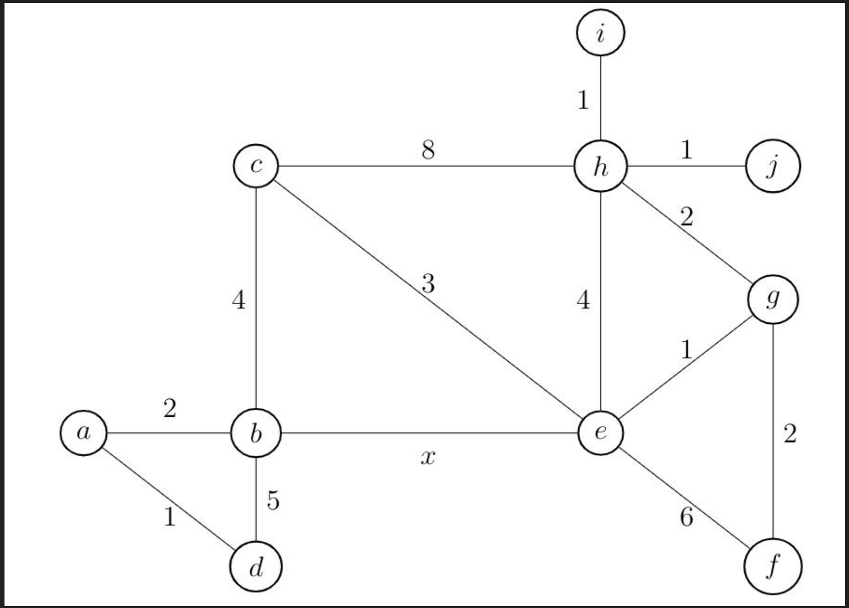
Algorithm:

- Start with the source vertex ‘i’.
- Initialize distance value of vertex ‘i’ to 0 and ∞ to every other vertex in G.
- Update the distance values of neighbouring vertices of vertex ‘i’.
- Each time a new vertex is visited, update the minimum distance values of its neighbours.

Q1

How to participate?
joinpd.com
code: see above

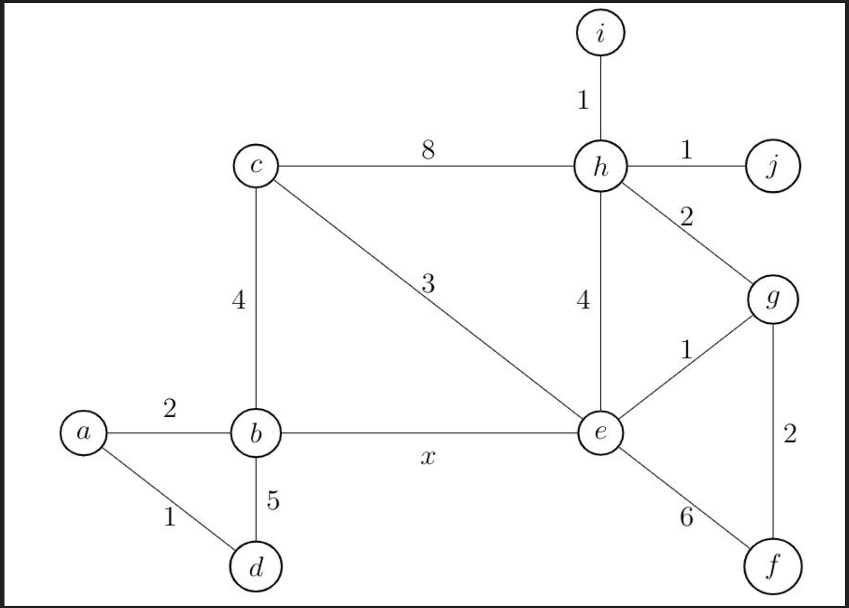
An undirected weighted graph G is shown below.



Q1

How to participate?
joinpd.com
code: see above

An undirected weighted graph G is shown below.



Find the set of all positive integer values of ‘x’ such that if we use Dijkstra’s algorithm, the length of the shortest path from vertex ‘a’ to vertex ‘j’ is less than 13.

(a) {1,2,3,4,5}

(b) {1,2,3,4,5,6}

(c) {1,2,3,4,5,6,7}

(d) {1,2,3,4,5,6,7,8}

Solution:

How to participate?
joinpd.com
code: see above

Step 1:

Remove the edge (b,e) from the graph and then find the shortest path from vertex ‘a’ to every other vertices using Dijkstra’s algorithm.

How to participate?
joinpd.com
code: see above

Solution:

Step 1:

Remove the edge (b,e) from the graph and then find the shortest path from vertex ‘a’ to every other vertices using Dijkstra’s algorithm.

Step 2: Start from vertex ‘a’, as it is the source vertex. Initialize distance value 0 to vertex ‘a’ and ∞ to all other vertices in the graph.

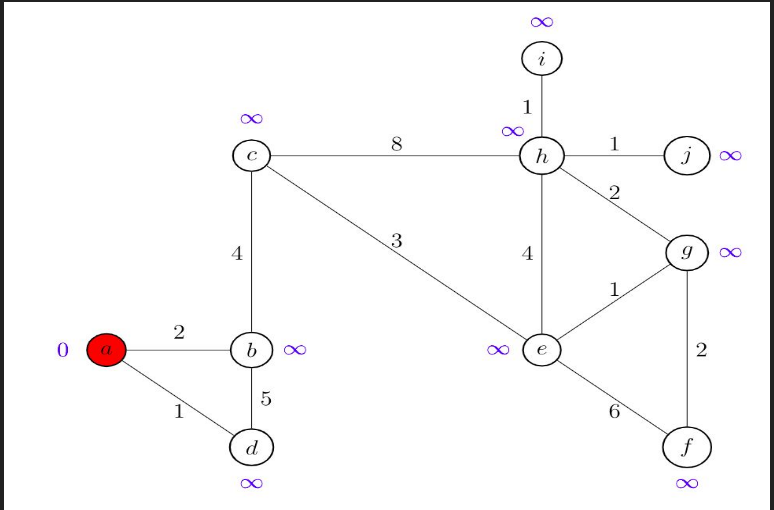
How to participate?
joinpd.com
code: see above

Solution:

Step 1:

Remove the edge (b,e) from the graph and then find the shortest path from vertex ‘a’ to every other vertices using Dijkstra’s algorithm.

Step 2: Start from vertex ‘a’, as it is the source vertex. Initialize distance value 0 to vertex ‘a’ and ∞ to all other vertices in the graph.



How to participate?
joinpd.com
code: see above

Solution:

Step 3:

Using Dijkstra’s algorithm, update the distance values of each of the vertex in the graph that are adjacent to ‘a’, which will be the shortest distance from the vertex ‘a’, in terms of weights.

OBJ

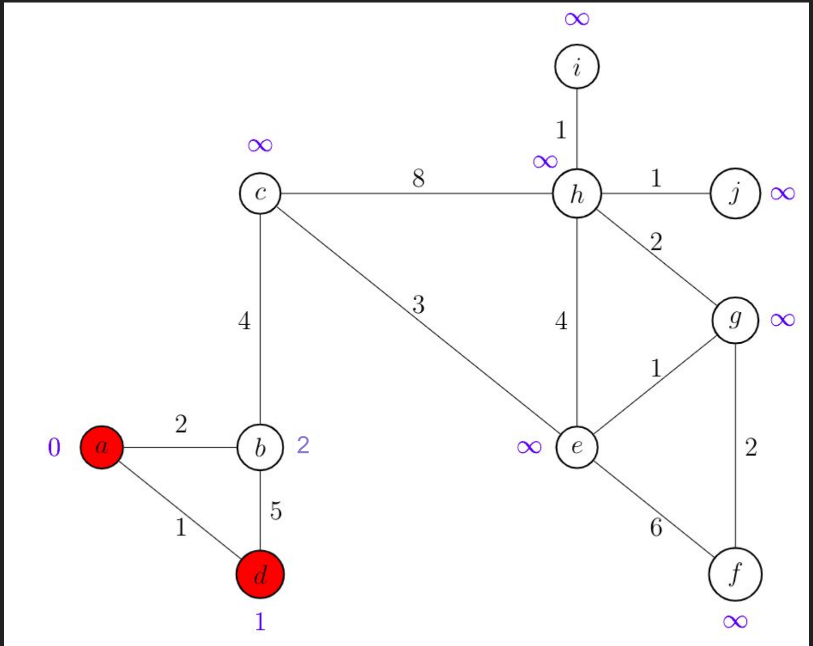
OBJ

How to participate?
joinpd.com
code: see above

Solution:

Step 3:

Using Dijkstra's algorithm, update the distance values of each of the vertex in the graph that are adjacent to 'a', which will be the shortest distance from the vertex 'a', in terms of weights.

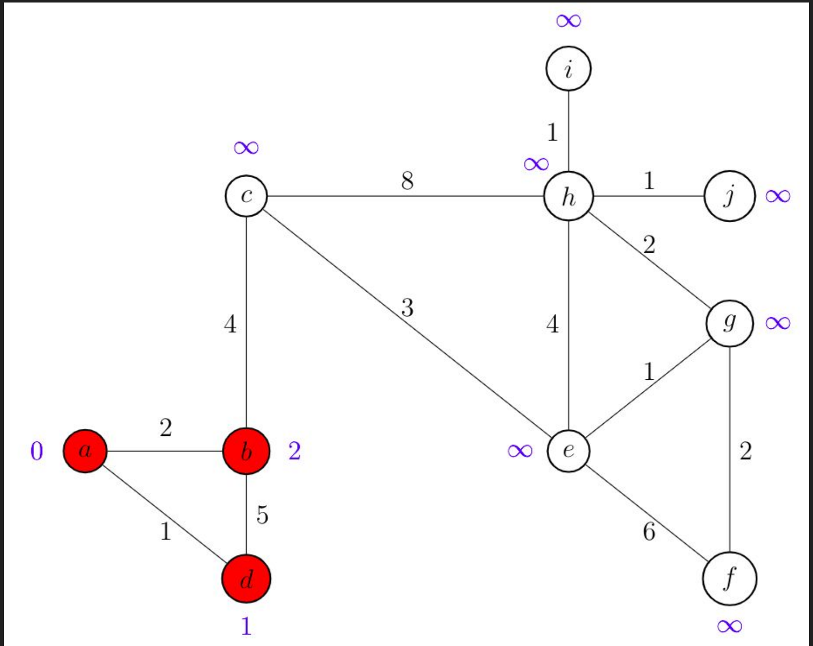


How to participate?
joinpd.com
code: see above

Solution:

Step 3:

Update the distance values of each of the vertex in the graph which are neighbours of ‘d’.

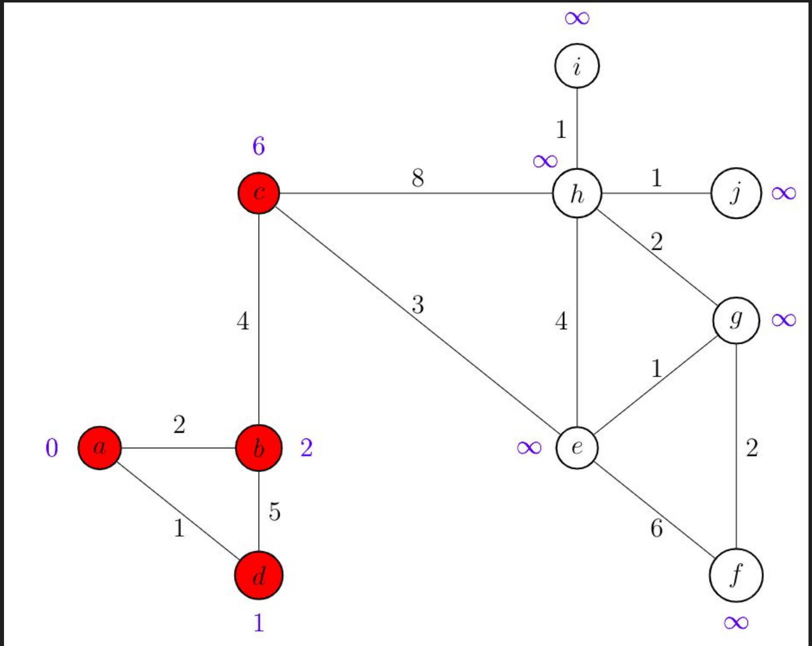


How to participate?
joinpd.com
code: see above

Solution:

Step 3:

Update the distance values of each of the vertex in the graph which are neighbours of ‘b’.

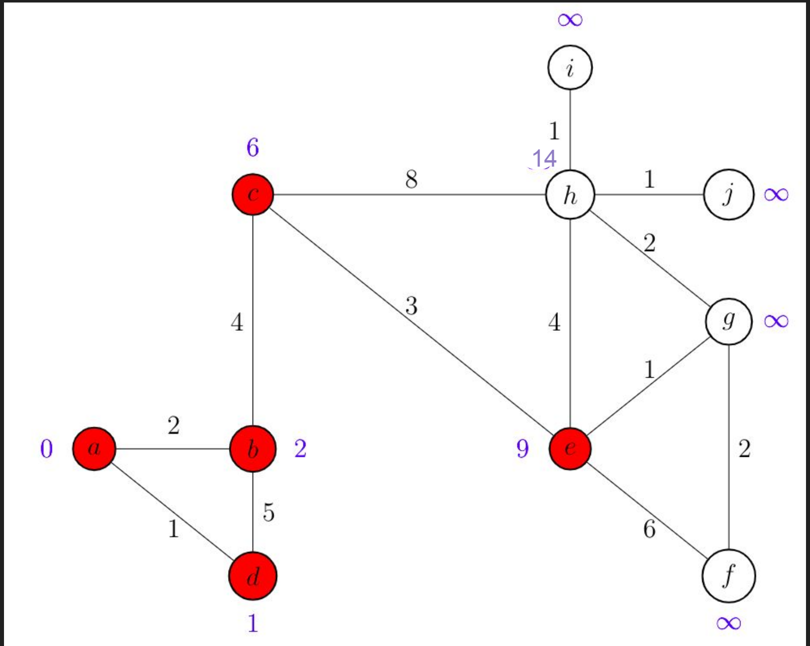


How to participate?
joinpd.com
code: see above

Solution:

Step 3:

Update the distance values of each of the vertex in the graph which are neighbours of ‘c’.



OBJ

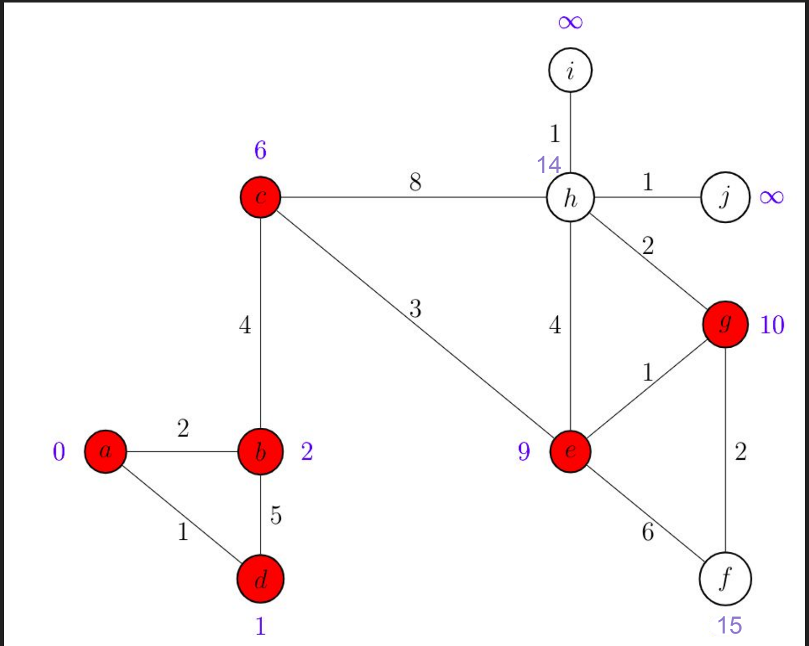
OBJ

How to participate?
joinpd.com
code: see above

Solution:

Step 3:

Update the distance values of each of the vertex in the graph which are neighbours of ‘e’.

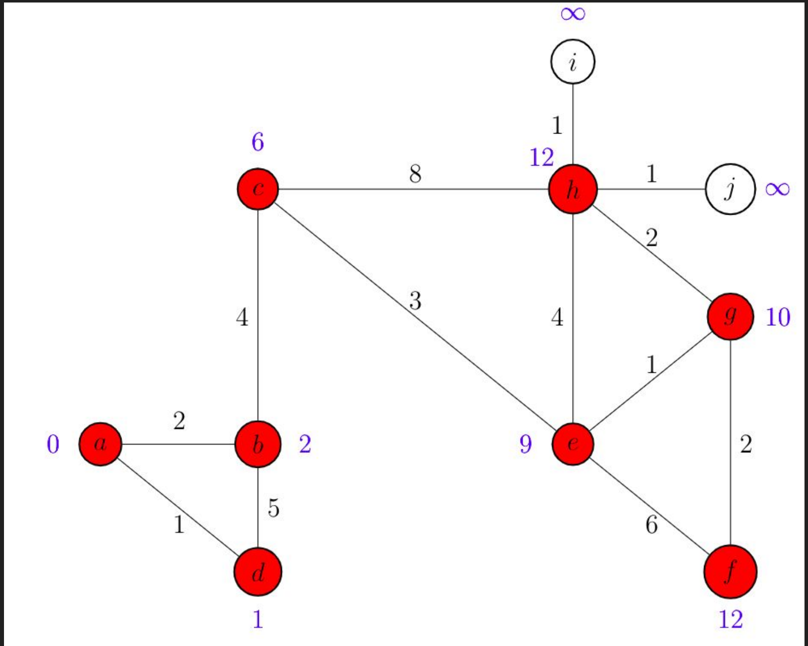


Solution:

How to participate?
joinpd.com
code: see above

Step 3:

Update the distance values of each of the vertex in the graph which are neighbours of ‘g’

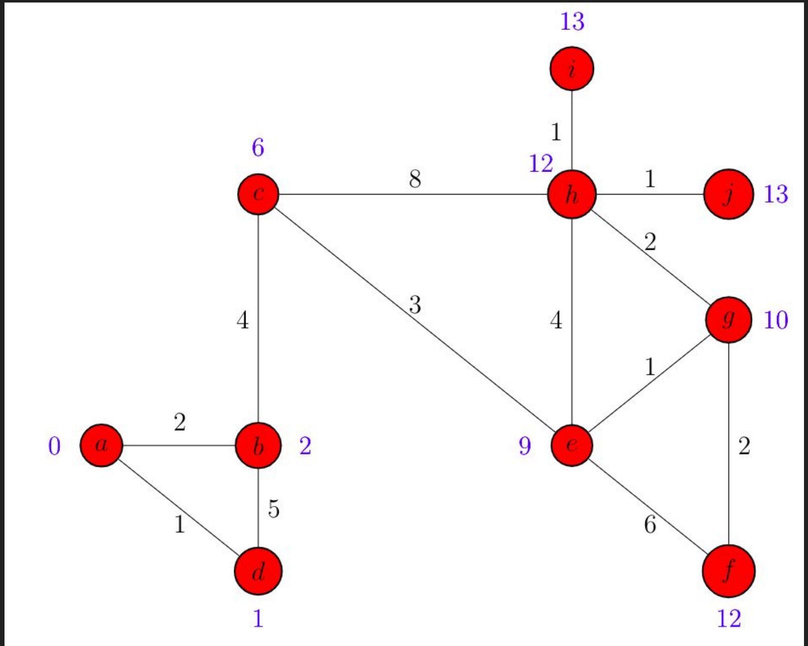


Solution:

Step 3:

Update the distance values of each of the vertex in the graph that are neighbours of 'h'.

How to participate?
joinpd.com
code: see above



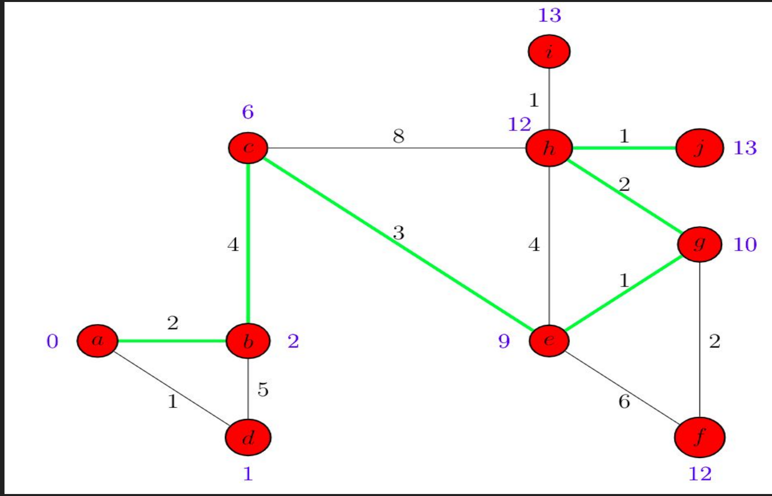
OBJ
OBJ

How to participate?
joinpd.com
code: see above

Solution:

Step 4:

The length of the shortest path from vertex ‘a’ to vertex ‘j’ we obtained is 13.



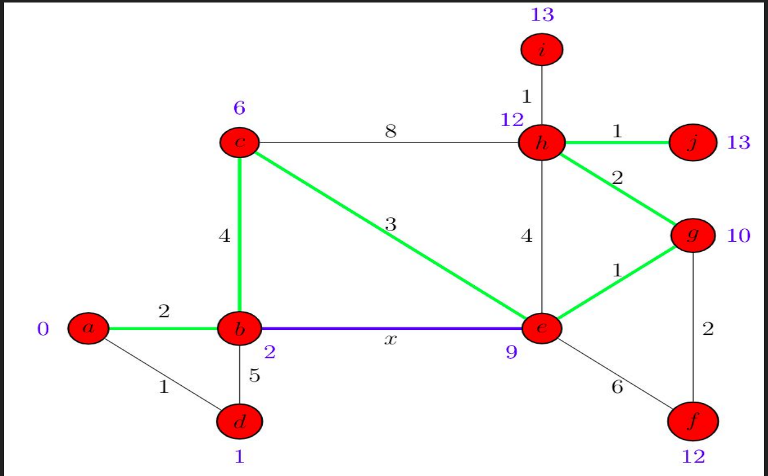
But it is given that the length of the shortest path from vertex ‘a’ to vertex ‘j’ is less than 13.

Therefore, the path a → b → c → e → g → h → j is not the shortest..!!!

Solution:

How to participate?
joinpd.com
code: see above

Step 4: Consider the edge (b,e).

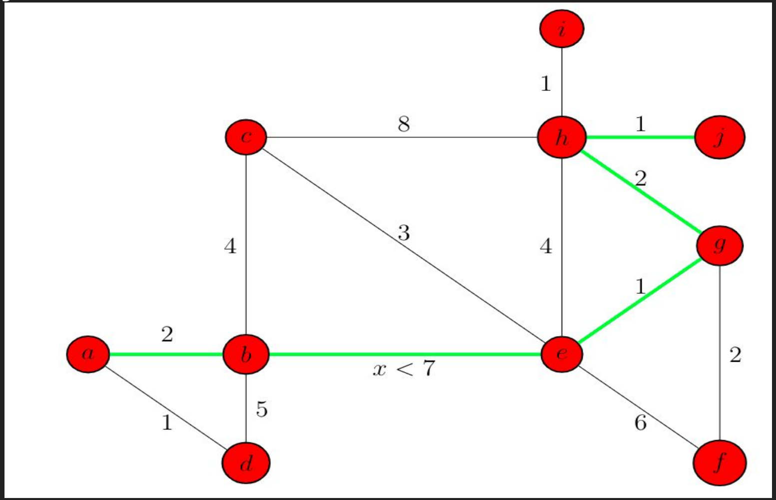


- Length of path from ‘b’ to ‘e’ is 7, now edge (b,e) is added so weight of this edge (b,e) should be less than 7.
- W((b,e)) < W((b,c)) + W((c,e)) $\Rightarrow$ x < 7

[OBJ]

How to participate?
joinpd.com
code: see above

Hence, the value of ‘x’ should be less than 7 in order to have the length shortest path from vertex ‘a’ to vertex ‘j’ less than 13.

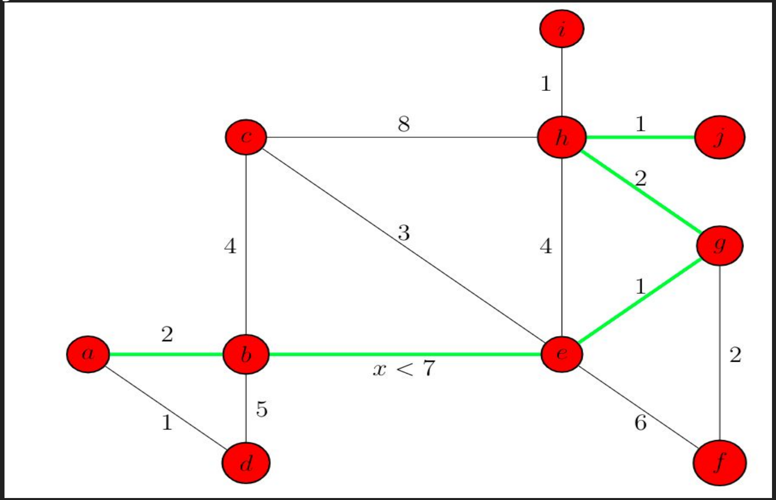


The shortest path will be a → b → e → g → h → j

OBJ

How to participate?
joinpd.com
code: see above

Hence, the value of ‘x’ should be less than 7 in order to have the length shortest path from vertex ‘a’ to vertex ‘j’ less than 13.



The shortest path will be a → b → e → g → h → j

Therefore, the set of all possible positive integer values of ‘x’ is {1,2,3,4,5,6}

Answer: (b)

Q2

How to participate?
joinpd.com
code: see above

A company has branches in each of five cities C_0, C_1, ...,C_4. The fare (in thousands of rupees) for a direct flight from C_i to C_j is given by the (i, j)th entry in the following matrix (∞ indicates that there is no direct flight):

$$\begin{bmatrix} 0 & 2 & \infty & \infty & 4 \\ 2 & 0 & 3 & 9 & \infty \\ \infty & 3 & 0 & 1 & 5 \\ \infty & 9 & 1 & 0 & 10 \\ 4 & \infty & 5 & 10 & 0 \end{bmatrix}$$

Q2.1

How to participate?
joinpd.com
code: see above

A company has branches in each of five cities C_0, C_1, ...,C_4. The fare (in thousands of rupees) for a direct flight from C_i to C_j is given by the (i, j)th entry in the following matrix (∞ indicates that there is no direct flight):

$$\begin{bmatrix} 0 & 2 & \infty & \infty & 4 \\ 2 & 0 & 3 & 9 & \infty \\ \infty & 3 & 0 & 1 & 5 \\ \infty & 9 & 1 & 0 & 10 \\ 4 & \infty & 5 & 10 & 0 \end{bmatrix}$$

An employee of that company wanted to travel from the city C_0 to the city C_3. If she travelled by the cheapest route possible, then the total fare (in rupees) she paid for flight journey was

<u>Sample answer</u>:
If you got 10, then enter 10000

Students, enter a number!

Solution:

How to participate?
joinpd.com
code: see above

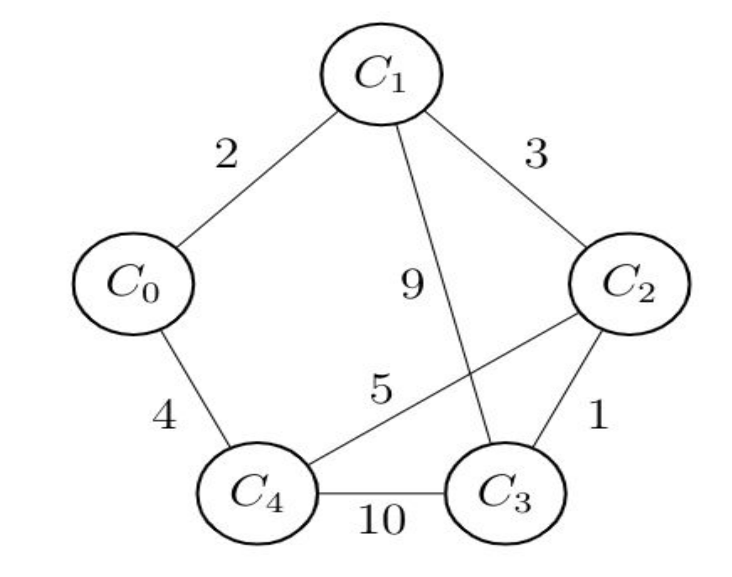
Step 1: Draw a graph G that represents the given adjacency matrix.

How to participate?
joinpd.com
code: see above

Solution:

Step 1: Draw a graph G that represents the given adjacency matrix.

$$\begin{array}{c|c} . & \begin{matrix} c_0 & c_1 & c_2 & c_3 & c_4 \end{matrix} \\ \begin{matrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix} & \begin{bmatrix} 0 & 2 & \infty & \infty & 4 \\ 2 & 0 & 3 & 9 & \infty \\ \infty & 3 & 0 & 1 & 5 \\ \infty & 9 & 1 & 0 & 10 \\ 4 & \infty & 5 & 10 & 0 \end{bmatrix} \end{array}$$



Solution:

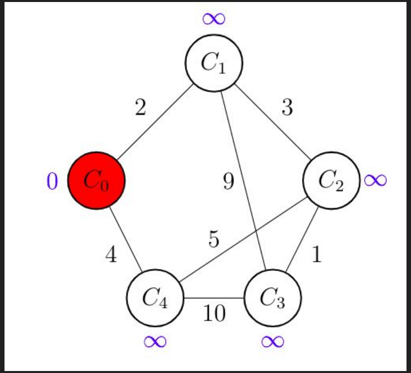
How to participate?
joinpd.com
code: see above

Step 2: Use Dijkstra's algorithm, starting with vertex C_0

Solution:

How to participate?
joinpd.com
code: see above

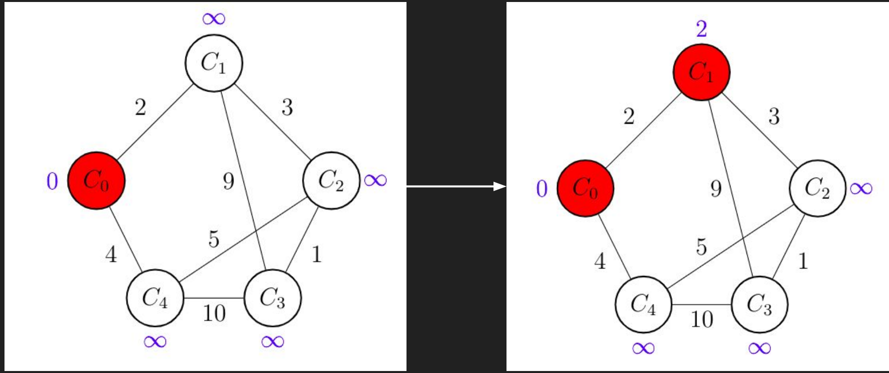
Step 2: Use Dijkstra's algorithm, starting with vertex C_0



Solution:

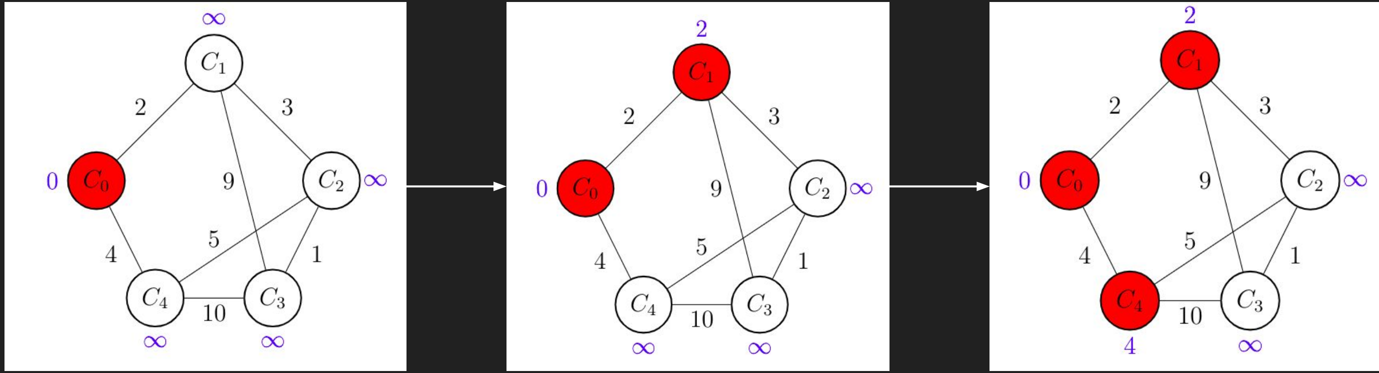
How to participate?
joinpd.com
code: see above

Step 2: Use Dijkstra's algorithm, starting with vertex C_0



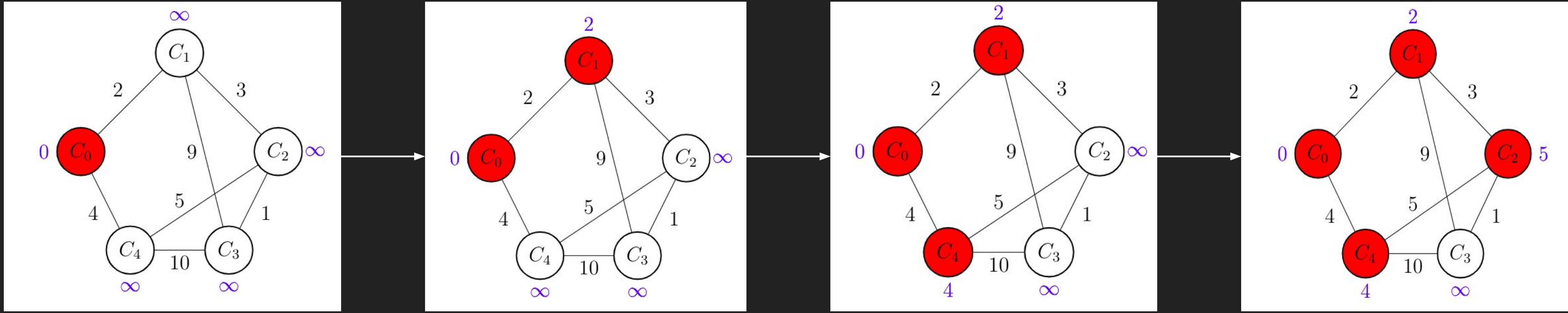
Solution:

Step 2: Use Dijkstra's algorithm, starting with vertex C_0

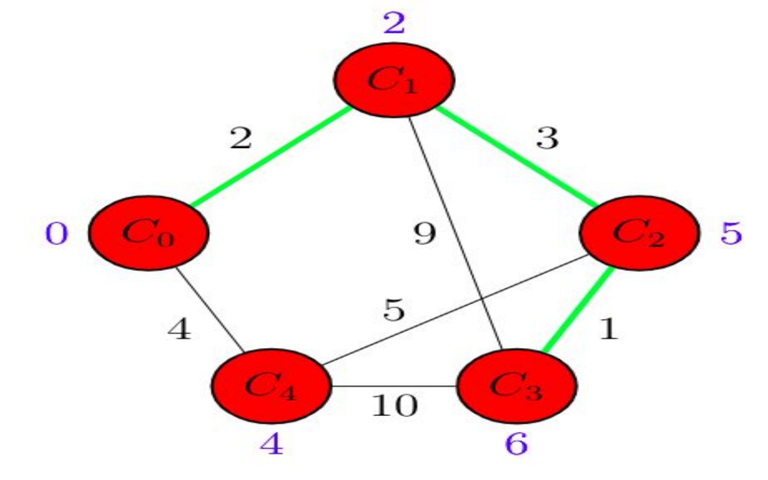


Solution:

Step 2: Use Dijkstra's algorithm, starting with vertex C_0



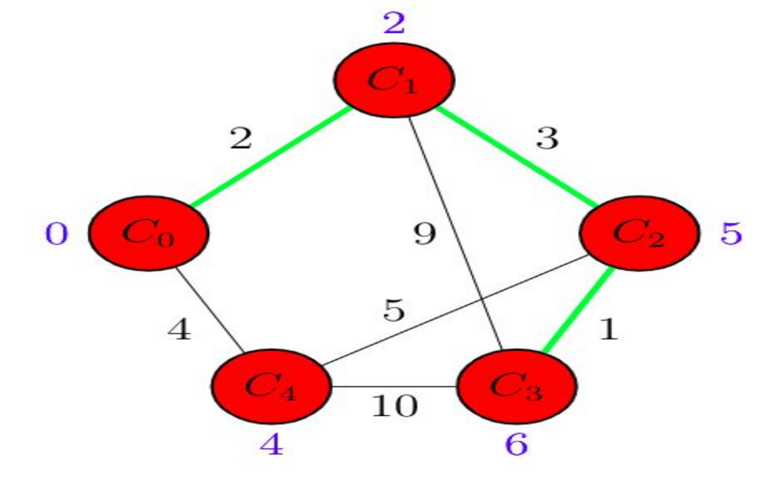
Solution:



Step 3:

- The shortest path from C_0 to C_3 is $C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$ (green path).
- The length of the shortest path is 2 + 3 + 1 = 6.

Solution:



How to participate?
joinpd.com
code: see above

Step 3:

- The shortest path from C_0 to C_3 is $C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$ (green path).
- The length of the shortest path is 2 + 3 + 1 = 6.

Hence, the total fare she paid for flight journey by travelling in cheapest route is 6,000 rupees.

Answer: 6000

Q2.2

How to participate?
joinpd.com
code: see above

A company has branches in each of five cities C_0, C_1, ...,C_4. The fare (in thousands of rupees) for a direct flight from C_i to C_j is given by the (i, j)th entry in the following matrix (∞ indicates that there is no direct flight):

$$\begin{bmatrix} 0 & 2 & \infty & \infty & 4 \\ 2 & 0 & 3 & 9 & \infty \\ \infty & 3 & 0 & 1 & 5 \\ \infty & 9 & 1 & 0 & 10 \\ 4 & \infty & 5 & 10 & 0 \end{bmatrix}$$

Q2.2

How to participate?
joinpd.com
code: see above

A company has branches in each of five cities C_0, C_1, ...,C_4. The fare (in thousands of rupees) for a direct flight from C_i to C_j is given by the (i, j)th entry in the following matrix (∞ indicates that there is no direct flight):

$$\begin{bmatrix} 0 & 2 & \infty & \infty & 4 \\ 2 & 0 & 3 & 9 & \infty \\ \infty & 3 & 0 & 1 & 5 \\ \infty & 9 & 1 & 0 & 10 \\ 4 & \infty & 5 & 10 & 0 \end{bmatrix}$$

If an inspection team member wanted to inspect all the branches of the company starting from C_0 and ending at C_3, visiting each branch exactly once, then which route should he choose in order to pay minimum fare for flight journey?

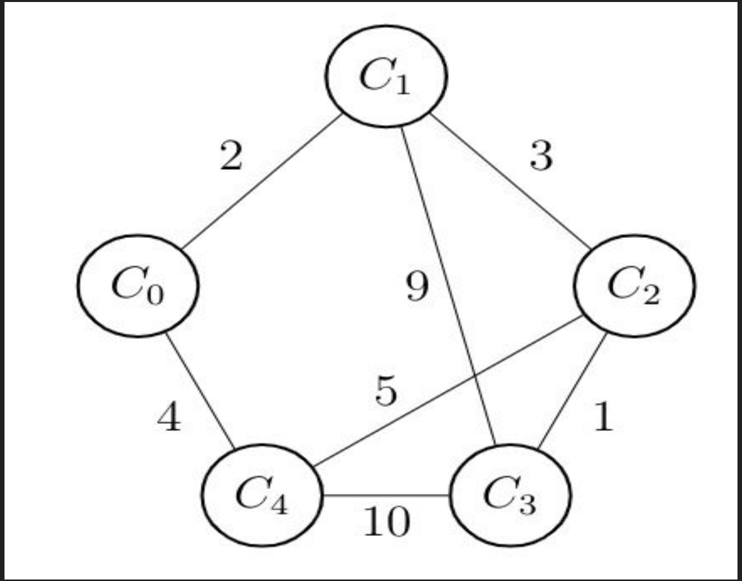
<u>Sample answer</u>: If your route is $C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4$, then enter C0, C1, C2, C3, C4

Students, write your response!

Solution:

How to participate?
joinpd.com
code: see above

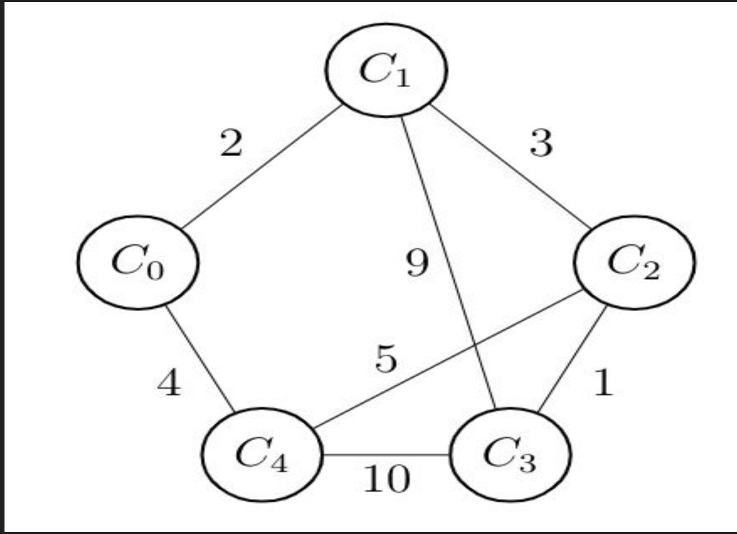
Step 1: Draw the graph G which represents the given adjacency matrix.



Solution:

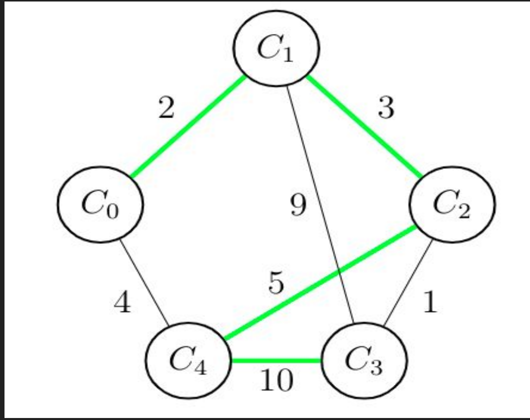
How to participate?
joinpd.com
code: see above

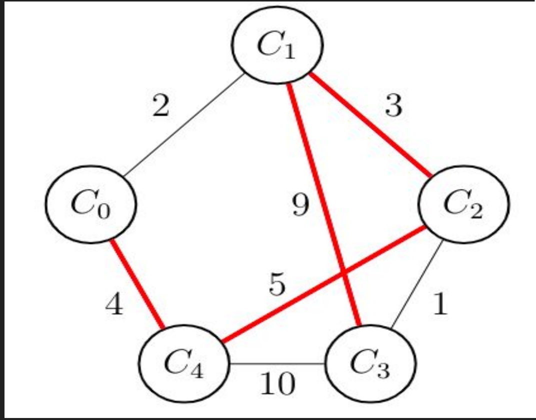
Step 2: The inspection team member starts at C_0 and ends at C_3 , in between he has to travel C_1,C_2, and C_4.



- Try to explore all possible paths starting from C_0 to C_3.
- Identify the paths that contains all the vertices (there can be more than one).

Solution:

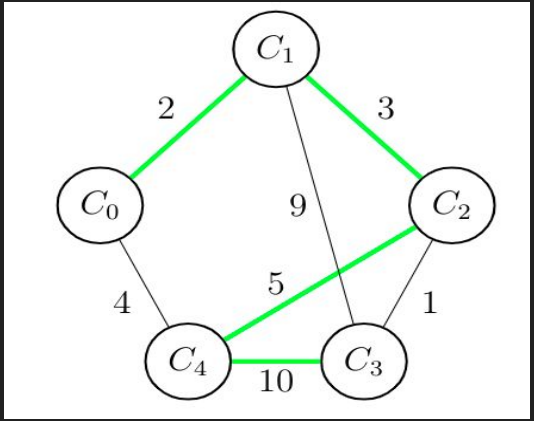


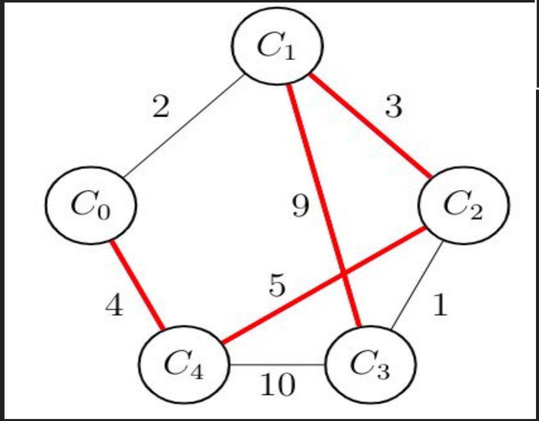


Step 3: Calculate the length of the two paths and choose the minimum length path.

- The length of the path $C_0 \to C_1 \to C_2 \to C_4 \to C_3$ (green path) = 2 + 3 + 5 + 10 = 20.
- The length of the path $C_0 \to C_4 \to C_2 \to C_1 \to C_3$ (red path) = 4 + 5 + 3 + 9 = 21.

Solution:





Step 3: Calculate the length of the two paths and choose the minimum length path.

- The length of the path $C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3$ (green path) = 2 + 3 + 5 + 10 = 20.
- The length of the path $C_0 \rightarrow C_4 \rightarrow C_2 \rightarrow C_1 \rightarrow C_3$ (red path) = 4 + 5 + 3 + 9 = 21.

Hence, the inspection team member should choose the green path.

Answer: C0, C1, C2, C4, C3

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Q. What is the cost of the spanning tree?

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Q. What is the cost of the spanning tree?

- It is the sum of the weights of all the edges in the spanning tree.

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Q. What is the cost of the spanning tree?

- It is the sum of the weights of all the edges in the spanning tree.

Q. What is MCST?

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Q. What is the cost of the spanning tree?

- It is the sum of the weights of all the edges in the spanning tree.

Q. What is MCST?

- MCST is the spanning tree whose cost is minimum among all the spanning trees.

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Q. What is the cost of the spanning tree?

- It is the sum of the weights of all the edges in the spanning tree.

Q. What is MCST?

- MCST is the spanning tree whose cost is minimum among all the spanning trees.

Q. How to find MCST?

Minimum Cost Spanning Tree (MCST)

How to participate?
joinpd.com
code: see above

Q. What is spanning tree?

- Spanning tree of an undirected graph G is a subgraph that is a tree which includes all the vertices of G.

Q. How is spanning tree different from tree?

- Spanning tree is the subgraph of graph G and that sub graph should be a tree.

Q. What is the cost of the spanning tree?

- It is the sum of the weights of all the edges in the spanning tree.

Q. What is MCST?

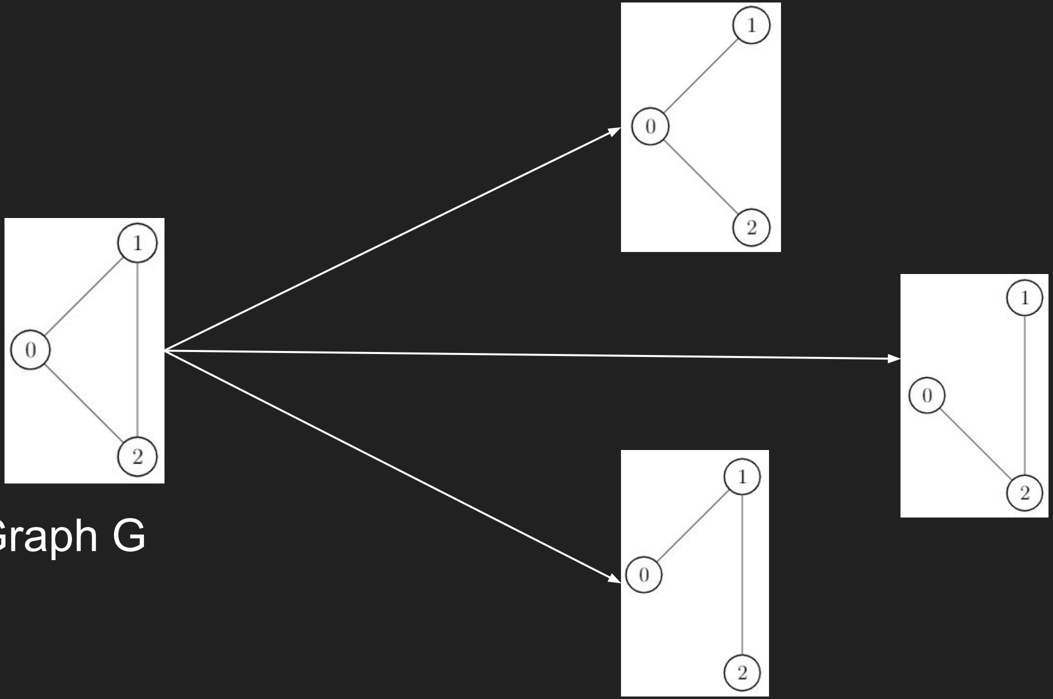
- MCST is the spanning tree whose cost is minimum among all the spanning trees.

Q. How to find MCST?

- Prim’s algorithm or Kruskal’s algorithm can be used to find MCST.

Example:

How to participate?
joinpd.com
code: see above



Graph G

Possible spanning trees

Prim’s Algorithm

How to participate?
joinpd.com
code: see above

Prim’s algorithm is used for finding MCST of a given graph G having ‘n’ vertices.

Prim’s Algorithm

How to participate?
joinpd.com
code: see above

Prim’s algorithm is used for finding MCST of a given graph G having ‘n’ vertices.

Step 1:

- Incrementally build an MCST of G.

Prim's Algorithm

How to participate?
joinpd.com
code: see above

Prim's algorithm is used for finding MCST of a given graph G having 'n' vertices.

Step 1:

- Incrementally build an MCST of G.

Step 2:

- Initialize TV = Φ and TE = Φ , where TV is the set of tree vertices added to MCST and TE is the set of tree edges added to MCST.

Prim’s Algorithm

How to participate?
joinpd.com
code: see above

Prim’s algorithm is used for finding MCST of a given graph G having ‘n’ vertices.

Step 1:

- Incrementally build an MCST of G.

Step 2:

- Initialize TV = Φ and TE = Φ , where TV is the set of tree vertices added to MCST and TE is the set of tree edges added to MCST.

Step 3:

- Choose the minimum weight edge e = (i, j)
- Set TV = {i, j} and TE = {e}

How to participate?
joinpd.com
code: see above

Prim's Algorithm

Prim's algorithm is used for finding MCST of a given graph G having 'n' vertices.

Step 1:

- Incrementally build an MCST of G.

Step 2:

- Initialize TV = Φ and TE = Φ , where TV is the set of tree vertices added to MCST and TE is the set of tree edges added to MCST.

Step 3:

- Choose the minimum weight edge e = (i, j)
- Set TV = {i, j} and TE = {e}

Step 4:

- Repeat (n-2) times
  - Choose minimum weight edge f = (u,v) such that u $\in$ TV and v $\notin$ TV.
  - Add 'v' to TV, 'f' to TE

Q3.1

How to participate?
joinpd.com
code: see above

Consider a weighted graph G with 7 vertices {P,Q,R,S,T,U,V}, which is represented by the following adjacency matrix.

$$\begin{bmatrix} 0 & 6 & 0 & 0 & 9 & 0 & 7 \\ 6 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 8 & 0 & 0 & 2 & 0 & 0 \\ 9 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 \\ 7 & 0 & 3 & 0 & 0 & 5 & 0 \end{bmatrix}.$$

Q3.1

Consider a weighted graph G with 7 vertices {P,Q,R,S,T,U,V}, which is represented by the following adjacency matrix.

$$\begin{bmatrix} 0 & 6 & 0 & 0 & 9 & 0 & 7 \\ 6 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 8 & 0 & 0 & 2 & 0 & 0 \\ 9 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 \\ 7 & 0 & 3 & 0 & 0 & 5 & 0 \end{bmatrix}.$$

If we perform Prim's algorithm starting with vertex R, then the order in which the vertices will be added to the set TV is

Enter the letters in order separated by comma.

Solution:

How to participate?
joinpd.com
code: see above

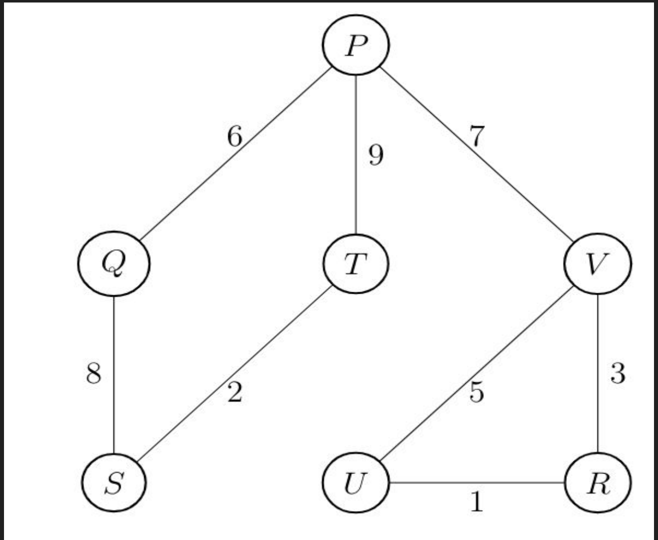
Step 1: Draw the graph G that represents the given adjacency matrix.

Solution:

How to participate?
joinpd.com
code: see above

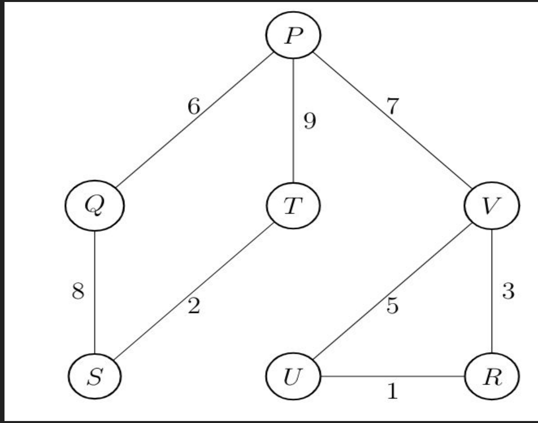
Step 1: Draw the graph G that represents the given adjacency matrix.

$$\begin{bmatrix} 0 & 6 & 0 & 0 & 9 & 0 & 7 \\ 6 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 8 & 0 & 0 & 2 & 0 & 0 \\ 9 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 \\ 7 & 0 & 3 & 0 & 0 & 5 & 0 \end{bmatrix}.$$



How to participate?
joinpd.com
code: see above

Solution:



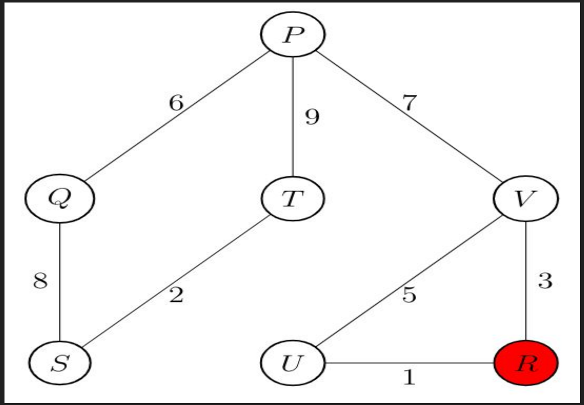
Step 2:

- Initialize TV = Φ , TE = Φ
- Start with vertex ‘R’.
- Update

TV = {R} and TE = Φ

How to participate?
joinpd.com
code: see above

Solution:



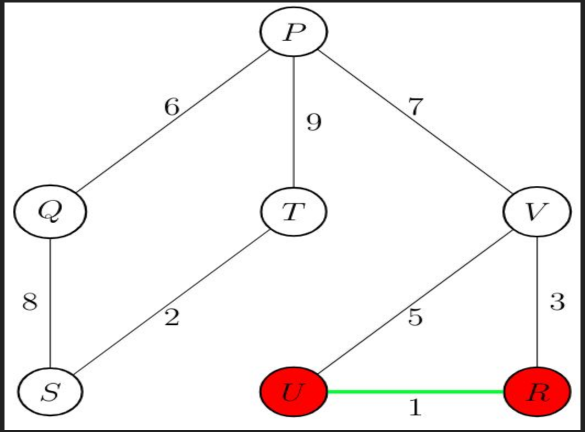
Step 3:

- Choose the minimum weight edge that is adjacent to vertex ‘R’.
- Edge (R,U) is the minimum weight edge.
- Add U to TV , (R,U) to TE.
- Update

TV = {R,U} and TE = { (R,U) }

How to participate?
joinpd.com
code: see above

Solution:

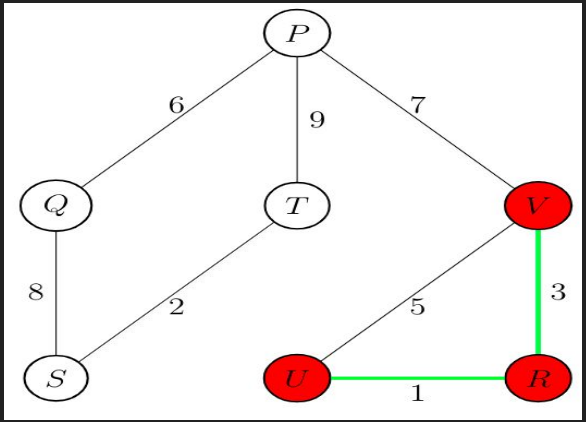


Step 3:

- Choose the minimum weight edge that is adjacent to vertex ‘R’ or ‘U’.
- Edge (R,V) is the minimum weight edge.
- Add V to TV , (R,V) to TE.
- Update

TV = {R,U,V} and TE = { (R,U) , (R,V)}

Solution:



How to participate?
joinpd.com
code: see above

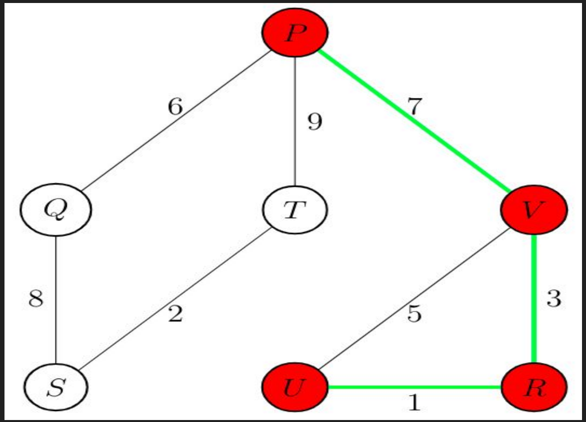
Step 3:

- Choose the minimum weight edge that is adjacent to vertex ‘R’ or ‘U’ or ‘V’.
- Edge (U,V) is not added to TE because it creates a cycle.
- Now, edge (V,P) will be the minimum edge.
- Add P to TV , (V,P) to TE.
- Update

TV = {R,U,V, P} and TE = { (R,U) , (R,V) , (V,P)}

Solution:

How to participate?
joinpd.com
code: see above

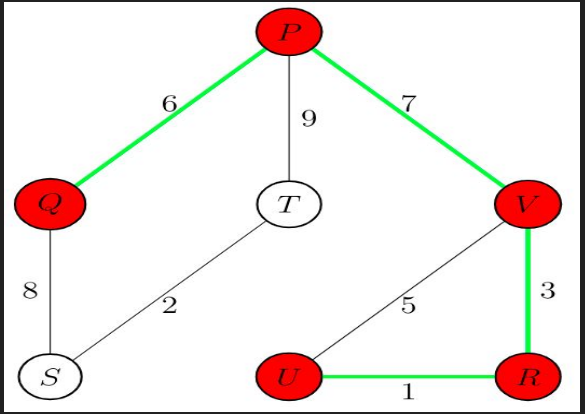


Step 3:

- Choose the minimum weight edge that is adjacent to vertex ‘P’
- Edge (P,Q) is the minimum weight edge.
- Add Q to TV , (P,Q) to TE.
- Update

TV = {R,U,V,P,Q} and TE = { (R,U) , (R,V) , (V,P) , (P,Q)}

Solution:



How to participate?
joinpd.com
code: see above

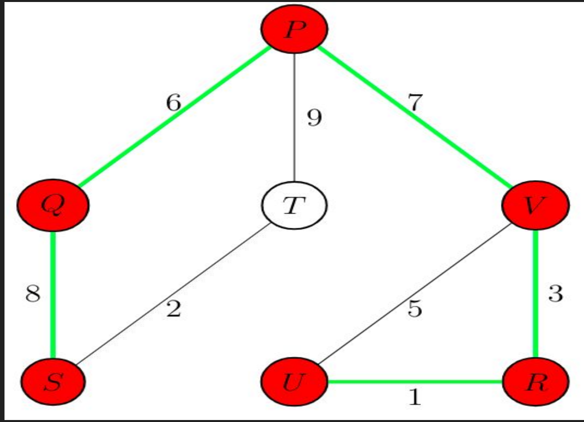
Step 3:

- Choose the minimum weight edge that is adjacent to vertex ‘P’ or ‘Q’.
- Edge (Q,S) is the minimum weight edge.
- Add S to TV , (Q,S) to TE.
- Update

TV = {R,U,V,P,Q,S} and TE = { (R,U) , (R,V) , (V,P) , (P,Q) , (Q,S)}

Solution:

How to participate?
joinpd.com
code: see above

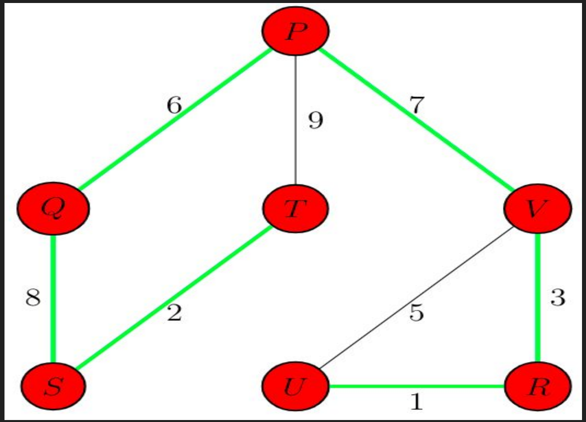


Step 3:

- Choose the minimum weight edge that is adjacent to vertex ‘P’ or ‘S’.
- Edge (S,T) is the minimum weight edge.
- Add T to TV , (S,T) to TE.
- Update

TV = {R,U,V,P,Q,S,T} and TE = { (R,U) , (R,V) , (V,P) , (P,Q) , (Q,S) , (S,T) }

Solution:



How to participate?
joinpd.com
code: see above

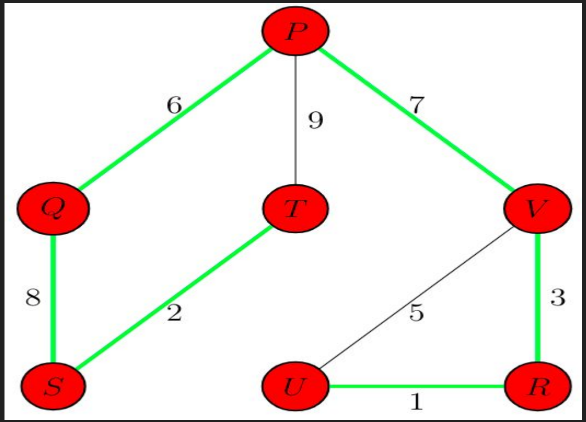
Step 4:

- Finally

TV = {R,U,V,P,Q,S,T} and TE = { (R,U) , (R,V) , (V,P) , (P,Q) , (Q,S) , (S,T) }

Solution:

How to participate?
joinpd.com
code: see above



Step 4:

- Finally

TV = {R,U,V,P,Q,S,T} and TE = { (R,U) , (R,V) , (V,P) , (P,Q) , (Q,S) , (S,T) }

Hence, the order in which the vertices is added to the set TV is R,U,V,P,Q,S,T.

Answer: R,U,V,P,Q,S,T

Kruskal’s Algorithm

How to participate?
joinpd.com
code: see above

Kruskal’s algorithm is used for finding MCST of a given graph G having ‘n’ vertices and ‘m’ edges.

Kruskal's Algorithm

How to participate?
joinpd.com
code: see above

Kruskal's algorithm is used for finding MCST of a given graph G having 'n' vertices and 'm' edges.

Step 1:

- Incrementally build an MCST of G.
- Let E = $\{e_0, e_1, e_2 ... e_{m-1}\}$ be the edges sorted in ascending order by weight.

Kruskal’s Algorithm

How to participate?
joinpd.com
code: see above

Kruskal’s algorithm is used for finding MCST of a given graph G having ‘n’ vertices and ‘m’ edges.

Step 1:

- Incrementally build an MCST of G.
- Let $E = \{e_0, e_1, e_2 ... e_{m-1}\}$ be the edges sorted in ascending order by weight.

Step 2:

- Initialize $TE = \Phi$, where TE is the set of tree edges added to MCST.

How to participate?
joinpd.com
code: see above

Kruskal's Algorithm

Kruskal's algorithm is used for finding MCST of a given graph G having 'n' vertices and 'm' edges.

Step 1:

- Incrementally build an MCST of G.
- Let E = $\{e_0, e_1, e_2 ... e_{m-1}\}$ be the edges sorted in ascending order by weight.

Step 2:

- Initialize TE = Φ , where TE is the set of tree edges added to MCST.

Step 3:

- Scan the edge set E from e_0 to e_{m-1}
  - If adding e_i to TE creates a cycle, then skip it
  - Otherwise add e_i to TE.

Q3.2

How to participate?
joinpd.com
code: see above

Consider a weighted graph G with 7 vertices {P,Q,R,S,T,U,V}, which is represented by the following adjacency matrix.

$$\begin{bmatrix} 0 & 6 & 0 & 0 & 9 & 0 & 7 \\ 6 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 8 & 0 & 0 & 2 & 0 & 0 \\ 9 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 \\ 7 & 0 & 3 & 0 & 0 & 5 & 0 \end{bmatrix}.$$

Q3.2

How to participate?
joinpd.com
code: see above

Consider a weighted graph G with 7 vertices {P,Q,R,S,T,U,V}, which is represented by the following adjacency matrix.

$$\begin{bmatrix} 0 & 6 & 0 & 0 & 9 & 0 & 7 \\ 6 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 8 & 0 & 0 & 2 & 0 & 0 \\ 9 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 \\ 7 & 0 & 3 & 0 & 0 & 5 & 0 \end{bmatrix}.$$

If we perform Kruskal's algorithm, then the order in which the edges are added to the set TE is

Consider edge between vertex i and vertex j as (i,j)

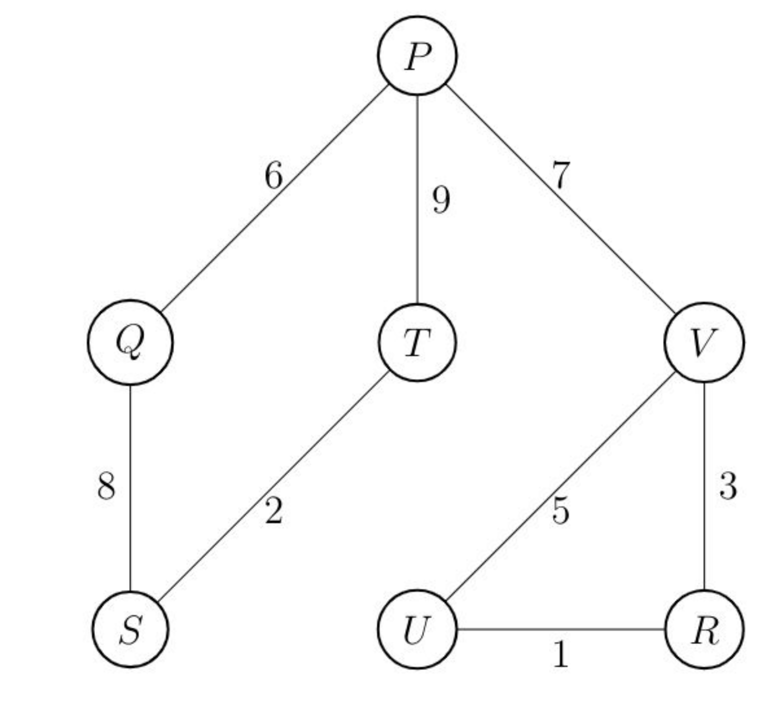
<u>Sample answer:</u>

(a,b) , (c,d) , (a,c) , ...,(d,f)

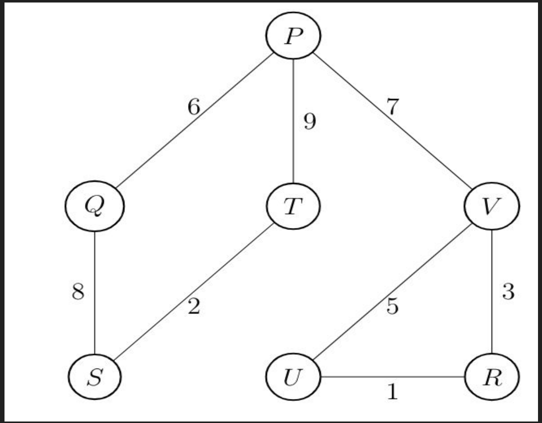
Solution:

How to participate?
joinpd.com
code: see above

Step 1: Draw the graph G that represents the given adjacency matrix.



Solution:

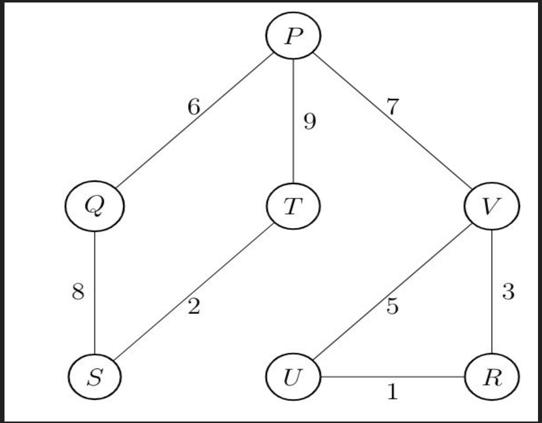


How to participate?
joinpd.com
code: see above

Step 2:

- Arrange the edges in the ascending order in terms of weights.
- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)

Solution:



How to participate?
joinpd.com
code: see above

Step 2:

- Arrange the edges in the ascending order in terms of weights.
- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)

Step 3:

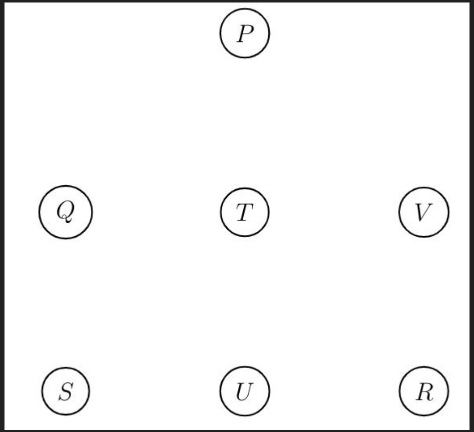
- Initialize TE = Φ
- Scan all edges starting from the edge (R,U).
- Start adding edges to TE and build MCST.

Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Add edge (R,U) to TE , which is the minimum cost edge.
- TE = {(R,U)}

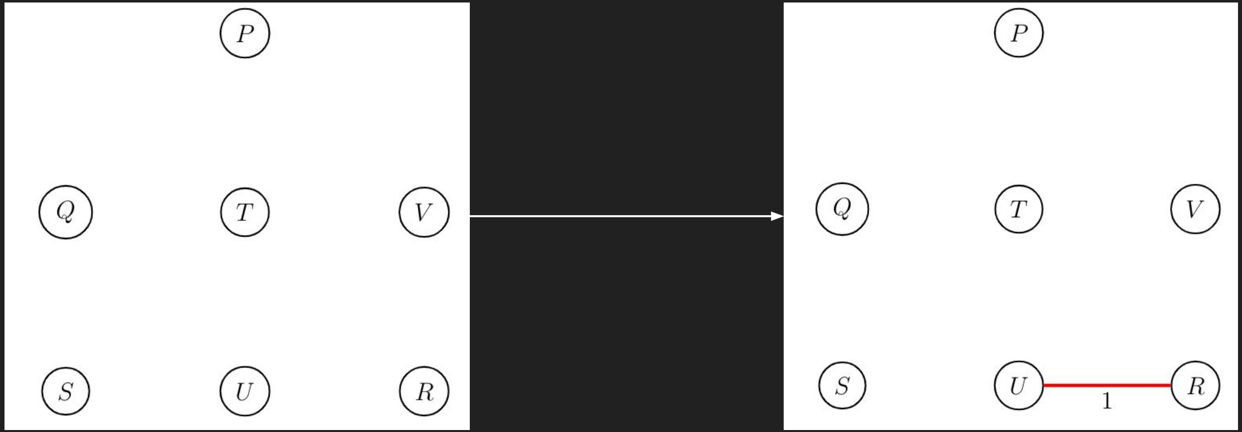


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- $\boxed{\text{(R,U)}}$, (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Add edge (R,U) to TE , which is the minimum cost edge.
- TE = {(R,U)}

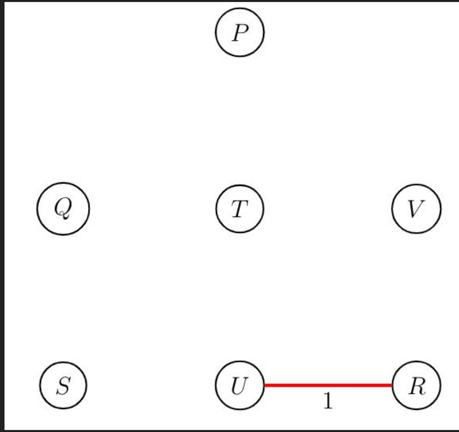


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , $\boxed{\text{(S,T)}}$, (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Add edge (S,T) to TE,
- TE = {(R,U) , (S,T)}

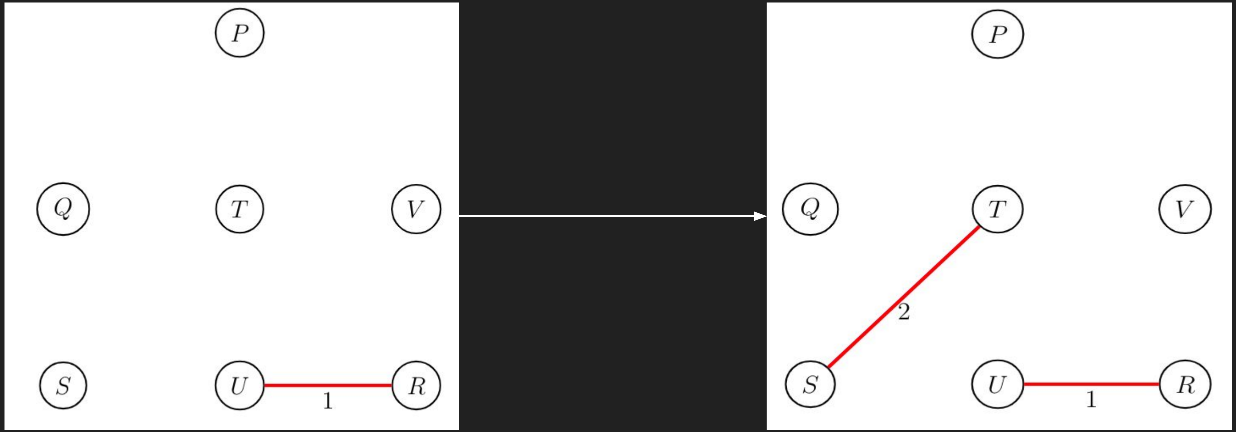


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , $\boxed{\text{(S,T)}}$, (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Add edge (S,T) to TE,
- TE = {(R,U) , (S,T)}

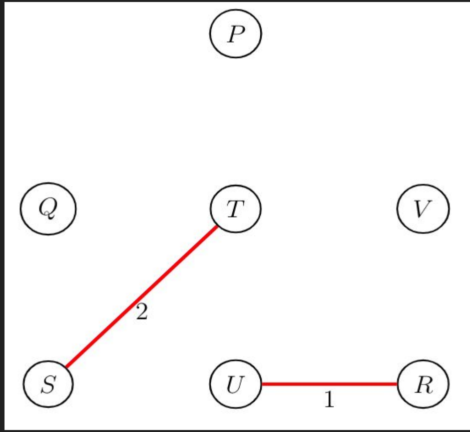


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , (S,T) , $\boxed{\text{(R,V)}}$, (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (R,V) to TE, does not create a cycle. Add (R,V) to TE.
- TE = {(R,U) , (S,T), (R,V) }

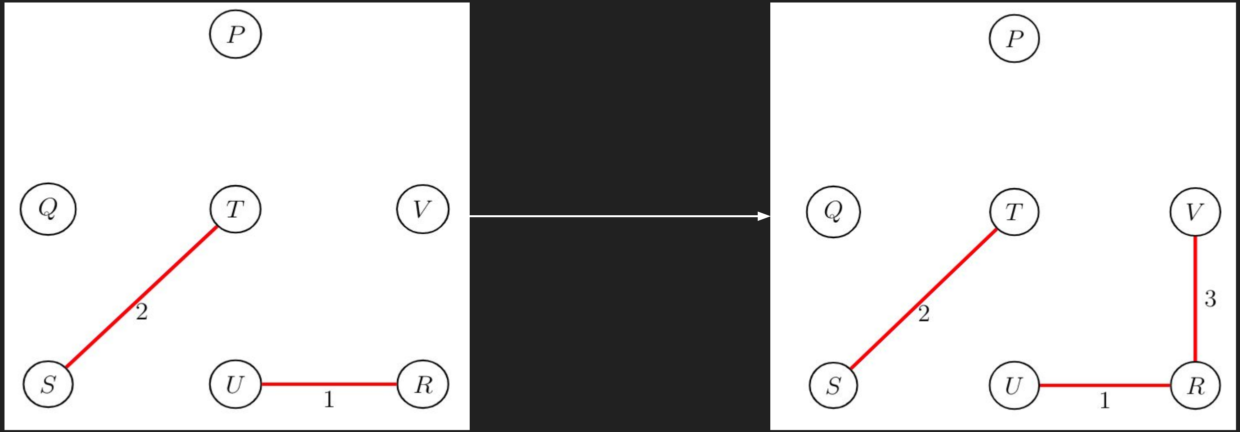


How to participate?
joinpd.com
code: see above

Solution:

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (R,V) to TE, does not create a cycle. Add (R,V) to TE.
- TE = {(R,U) , (S,T), (R,V) }

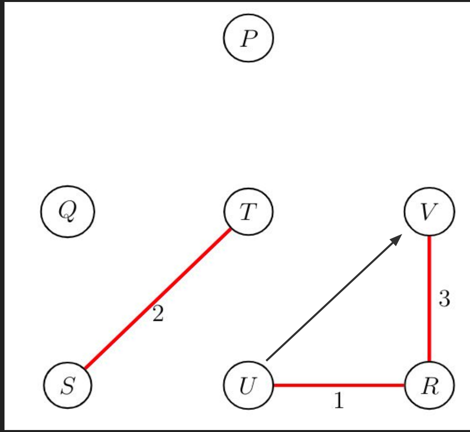


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , (S,T) , (R,V) , $\boxed{\text{(U,V)}}$, (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (U,V) to TE, creates a cycle (R→ U → V → R).
- Skip (U,V)
- TE = {(R,U) , (S,T), (R,V) }

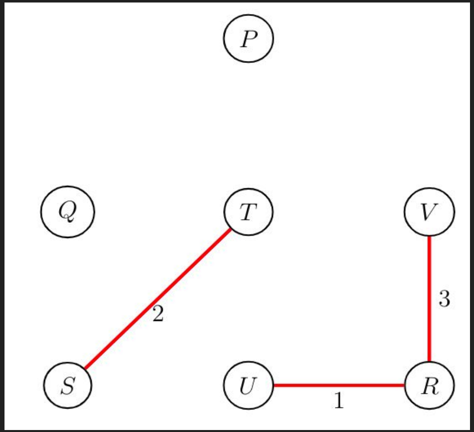


How to participate?
joinpd.com
code: see above

Solution:

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (P,Q) to TE, does not create a cycle. Add (P,Q) to TE
- TE = {(R,U) , (S,T), (R,V) , (P,Q)}

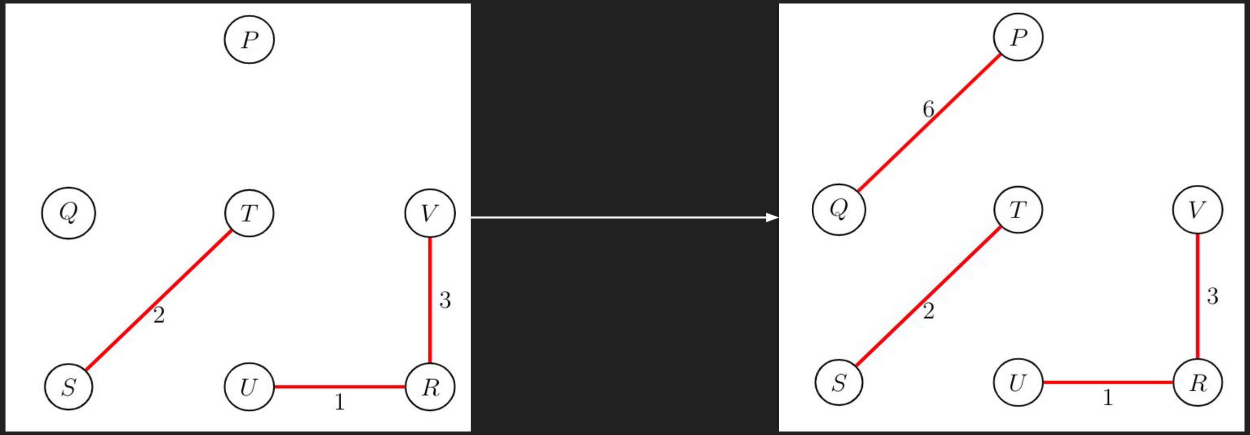


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , $\boxed{\text{(P,Q)}}$, (P,V) , (Q,S) , (P,T)
- Adding (P,Q) to TE, does not create a cycle. Add (P,Q) to TE
- TE = {(R,U) , (S,T), (R,V) , (P,Q)}

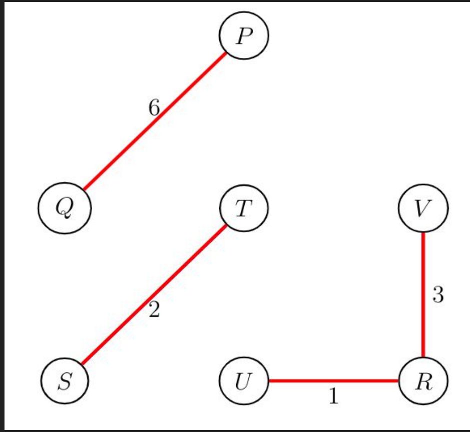


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , $\boxed{\text{(P,V)}}$, (Q,S) , (P,T)
- Adding (P,V) to TE, does not create a cycle. Add (P,V) to TE
- TE = {(R,U) , (S,T), (R,V) , (P,Q) , (P,V)}

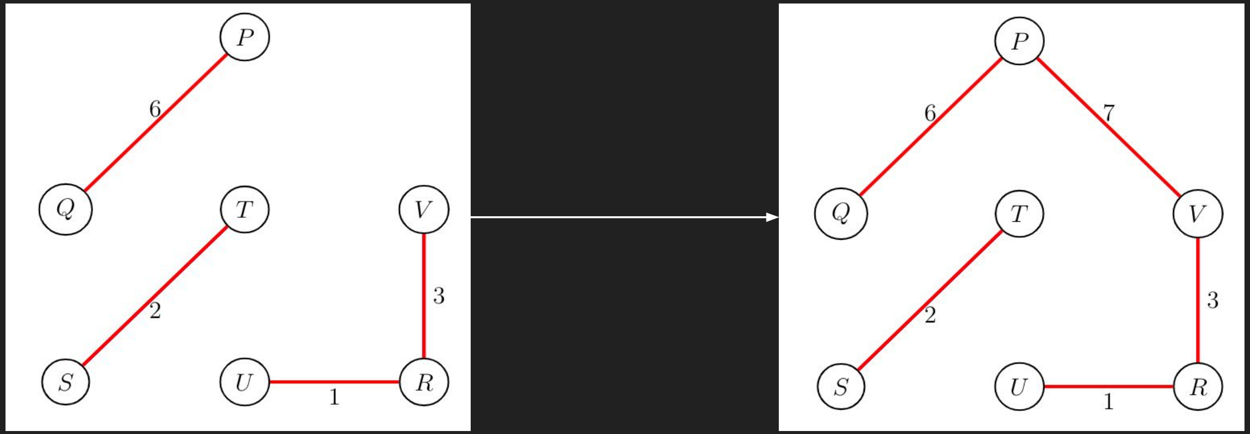


Solution:

How to participate?
joinpd.com
code: see above

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , $\boxed{\text{(P,V)}}$, (Q,S) , (P,T)
- Adding (P,V) to TE, does not create a cycle. Add (P,V) to TE
- TE = {(R,U) , (S,T), (R,V) , (P,Q) , (P,V)}

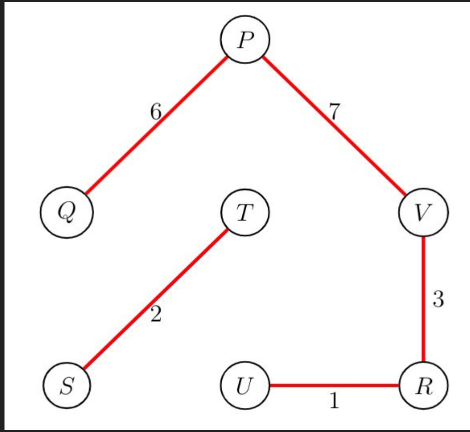


How to participate?
joinpd.com
code: see above

Solution:

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (Q,S) to TE, does not create a cycle. Add (Q,S) to TE
- TE = {(R,U) , (S,T), (R,V) , (P,Q) , (P,V) , (Q,S)}

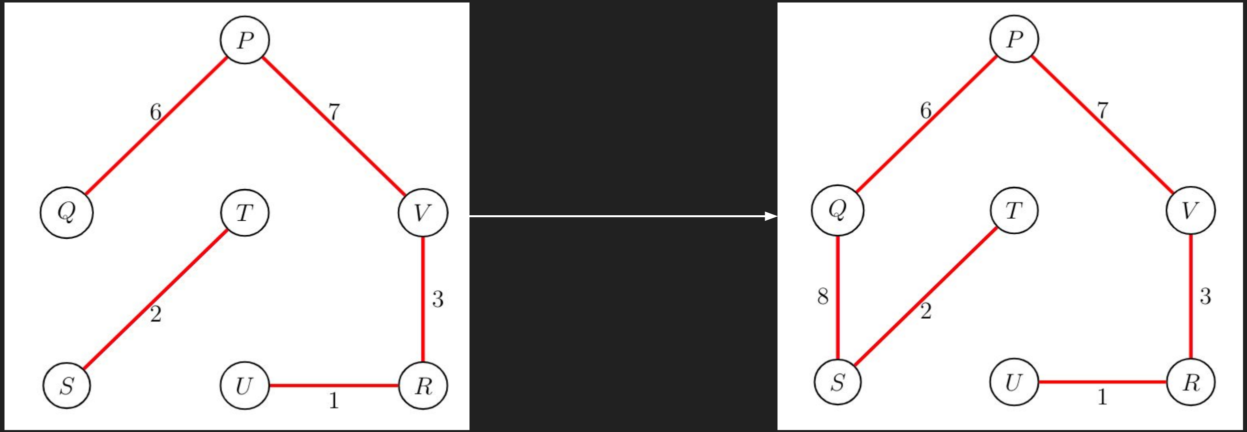


How to participate?
joinpd.com
code: see above

Solution:

Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (Q,S) to TE, does not create a cycle. Add (Q,S) to TE
- TE = {(R,U) , (S,T), (R,V) , (P,Q) , (P,V) , (Q,S)}



Solution:

How to participate?
joinpd.com
code: see above

Step 4:

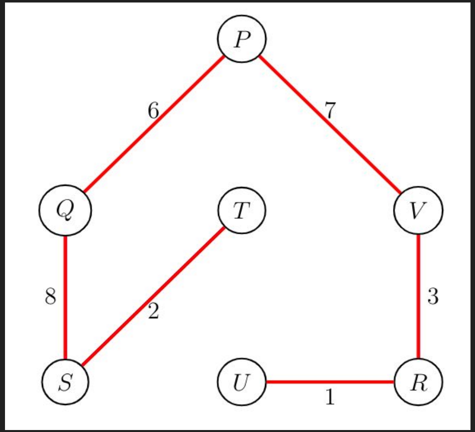
- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (P,T) to TE, creates a cycle (P → Q → S → T → P). Skip (P,T)
- TE = {(R,U) , (S,T), (R,V) , (P,Q) , (P,V)}

Solution:

How to participate?
joinpd.com
code: see above

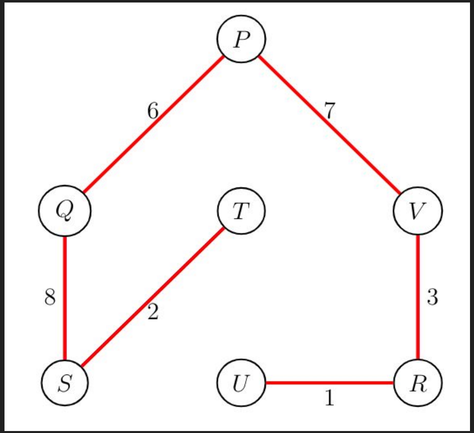
Step 4:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- Adding (P,T) to TE, creates a cycle (P → Q → S → T → P). Skip (P,T)
- TE = {(R,U) , (S,T), (R,V) , (P,Q) , (P,V)}



How to participate?
joinpd.com
code: see above

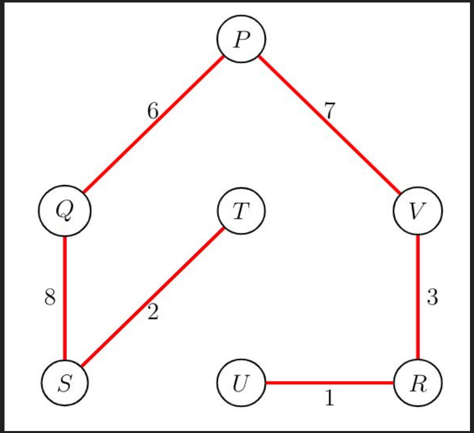
Solution:



Step 5:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- (U,V) and (P,T) are not added to MCST.

Solution:



Step 5:

- (R,U) , (S,T) , (R,V) , (U,V) , (P,Q) , (P,V) , (Q,S) , (P,T)
- (U,V) and (P,T) are not added to MCST.
- The order in which the edges are added to the set TE when we perform Kruskal's algorithm is (R,U) , (S,T) , (R,V) , (P,Q) , (P,V) , (Q,S)

Answer: (R,U) , (S,T) , (R,V) , (P,Q) , (P,V) , (Q,S)

Some things to keep in mind while solving problems:

1. Generally which algorithm we should use to find shortest path from a given source vertex?

Some things to keep in mind while solving problems:

1. Generally which algorithm we should use to find shortest path from a given source vertex?

    Answer: Dijkstra’s algorithm, it is not MCST problem.

Some things to keep in mind while solving problems:

1. Generally which algorithm we should use to find shortest path from a given source vertex?

   Answer: Dijkstra’s algorithm, it is not MCST problem.
2. Dijkstra’s algorithm will have unique shortest path with some of the edges having

   Equal weights-----Answer: NO

   Unequal weights---

Some things to keep in mind while solving problems:

1. Generally which algorithm we should use to find shortest path from a given source vertex?

    Answer: Dijkstra’s algorithm, it is not MCST problem.

2. Dijkstra’s algorithm will have unique shortest path with some of the edges having

    Equal weights-----Answer: NO

    Unequal weights---Answer: Need to be always true

3. Which algorithm should be used for finding the minimum cost in which all the edges are connected

Some things to keep in mind while solving problems:

1. Generally which algorithm we should use to find shortest path from a given source vertex?
   Answer: Dijkstra’s algorithm, it is not MCST problem.
2. Dijkstra’s algorithm will have unique shortest path with some of the edges having
   Equal weights-----Answer: NO
   Unequal weights---Answer: Need to be always true
3. Which algorithm should be used for finding the minimum cost in which all the edges are connected
   Answer: MCST problem so either Prim’s or Kruskal's algorithm

Name the algorithm to be used:

1. To find the shortest path from a fixed source vertex to every other vertex in a graph G (No negative edge weights) -------Dijkstra’s algorithm
2. To find the shortest path from a fixed source vertex to every other vertex in a graph G (With or without negative edge weights, no negative cycle)----Bellman-Ford algorithm
3. All pair shortest path (With or without negative edge weights, no negative cycle)----Floyd-Warshall algorithm
4. MCST----Prim’s or Kruskal's algorithm

Bellman-Ford Algorithm

How to participate?
joinpd.com
code: see above

Bellman-Ford Algorithm

How to participate?
joinpd.com
code: see above

- This algorithm is used for finding the shortest path from a source vertex to every other vertex in the graph.

How to participate?
joinpd.com
code: see above

Bellman-Ford Algorithm

- This algorithm is used for finding the shortest path from a source vertex to every other vertex in the graph.
- Facts:
  - This algorithm is applicable for both negative and positive edge weights.
  - Not applicable for graphs having a negative weight cycle.

Bellman-Ford Algorithm

How to participate?
joinpd.com
code: see above

- This algorithm is used for finding the shortest path from a source vertex to every other vertex in the graph.
- Facts:
  - This algorithm is applicable for both negative and positive edge weights.
  - Not applicable for graphs having a negative weight cycle.
- Algorithm:
  - D(j) is the minimum distance known so far from source vertex 'j'
  - Initialize:
    - $D(0) = 0$ if $j = 0$; otherwise $D(j) = \infty$

How to participate?
joinpd.com
code: see above

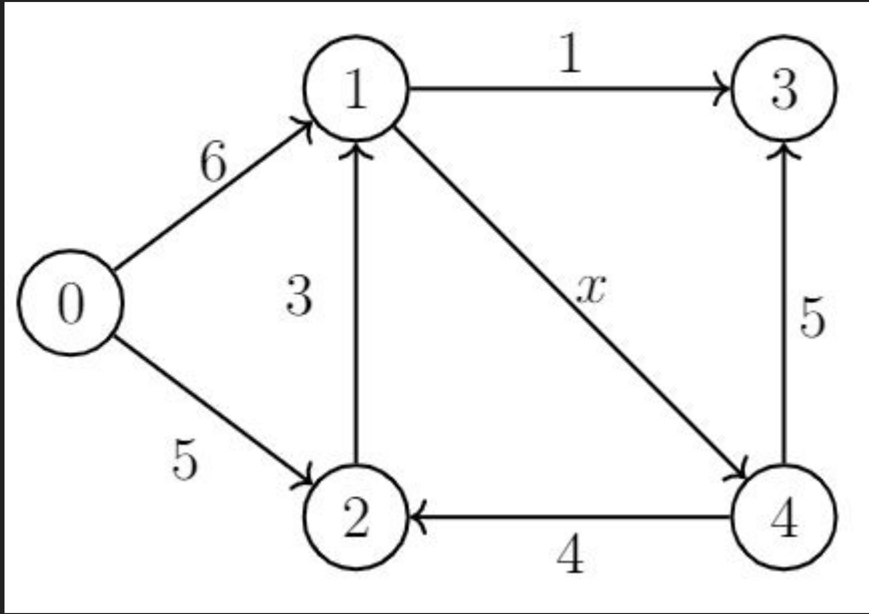
Bellman-Ford Algorithm

- This algorithm is used for finding the shortest path from a source vertex to every other vertex in the graph.
- Facts:
  - This algorithm is applicable for both negative and positive edge weights.
  - Not applicable for graphs having a negative weight cycle.
- Algorithm:
  - D(j) is the minimum distance known so far from source vertex ‘j’
  - Initialize:
    - $D(0) = 0$ if $j = 0$; otherwise $D(j) = \infty$
  - Repeat (n - 1) times
    - For every vertex $j = \{ 0, 1, 2, \dots (n-1) \}$ for each edge $(j , k) \epsilon E$
    - $D(k) = \min \{D(k), D(j) + W(j, k)\}$

Q4

How to participate?
joinpd.com
code: see above

For what values of ‘x’ can we use the Bellman-Ford algorithm to find the shortest path from a source vertex 0 to every other vertex in the graph given below?



Which of the following can be a possible values of ‘x’ ?

a) (-6, ∞)

b) (-14, ∞)

c) (-7, ∞)

d) (-21, ∞)

How to participate?
joinpd.com
code: see above

Solution:

Step 1:

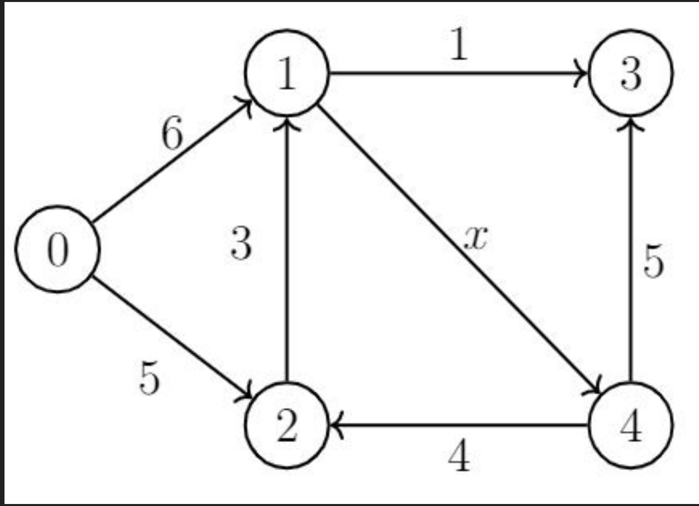
- Recall Bellman Ford algorithm is not applicable for negative weight cycle.
    - Notice that after (n-1) iteration the shortest path values doesn't converge.

Step 2:

- Notice vertices 1, 4 and 2 form a cycle.
    - If 'x' < (-7), then vertices 1, 4 and 2 form negative weight cycle which is not acceptable (not defined).

Step 3:

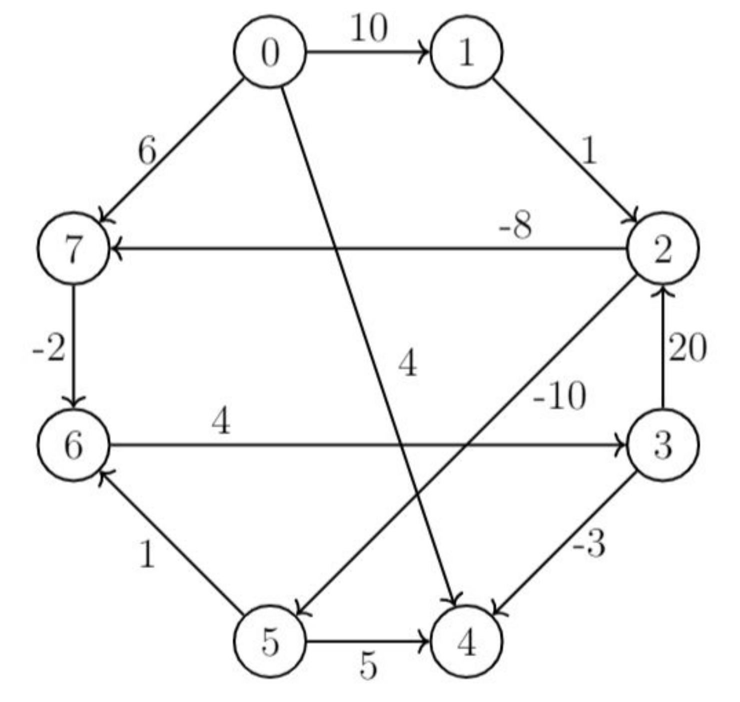
- Therefore, 'x' can take values between (-7, ∞)



Example 1

How to participate?
joinpd.com
code: see above

Consider the graph G shown below, perform the Bellman-Ford algorithm.



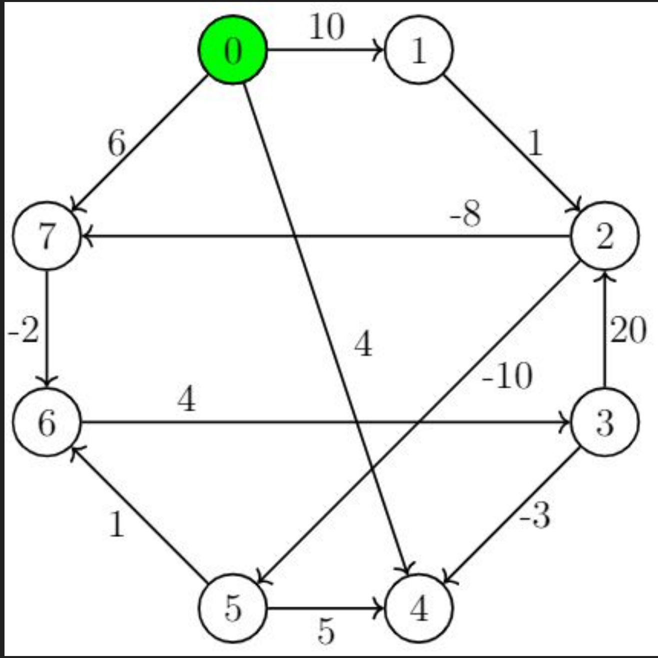
Suppose the source vertex is 0. We need to perform the Bellman-Ford algorithm upto 3 iterations.

Solution:

How to participate?
joinpd.com
code: see above

Step 1: Initialization

v								D(v)
0	0							
1	∞							
2	∞							
3	∞							
4	∞							
5	∞							
6	∞							
7	∞							

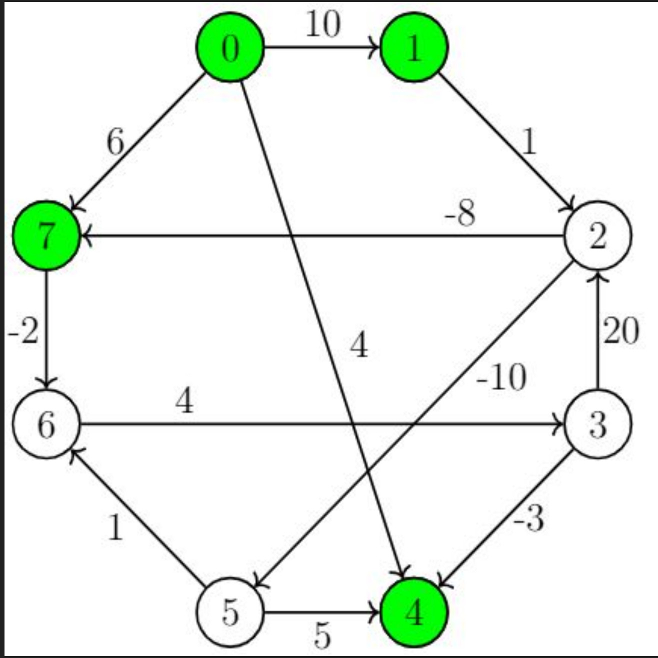


Solution:

How to participate?
joinpd.com
code: see above

Step 2: First iteration

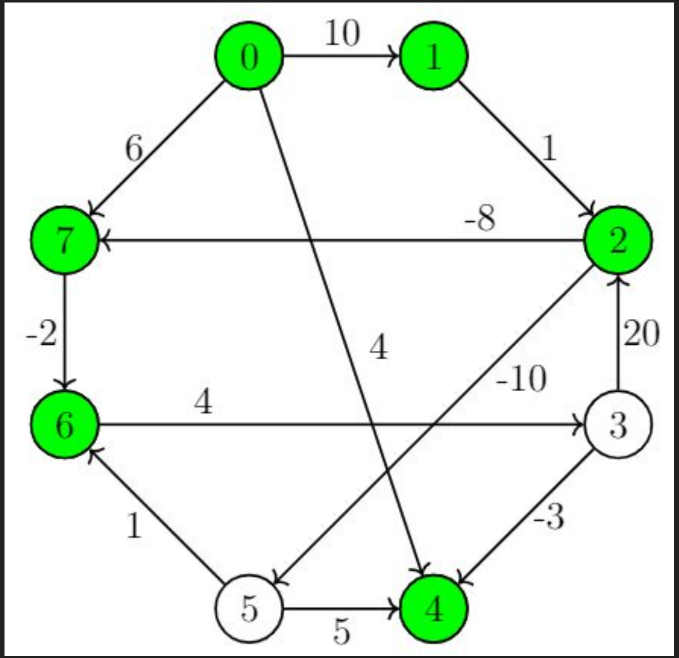
v								D(v)
0	0	0						
1	∞	10						
2	∞	∞						
3	∞	∞						
4	∞	4						
5	∞	∞						
6	∞	∞						
7	∞	6						



Solution:

Step 3: Second iteration

v								D(v)
0	0	0	0					
1	∞	10	10					
2	∞	∞	11					
3	∞	∞	∞					
4	∞	4	4					
5	∞	∞	∞					
6	∞	∞	4					
7	∞	6	6					

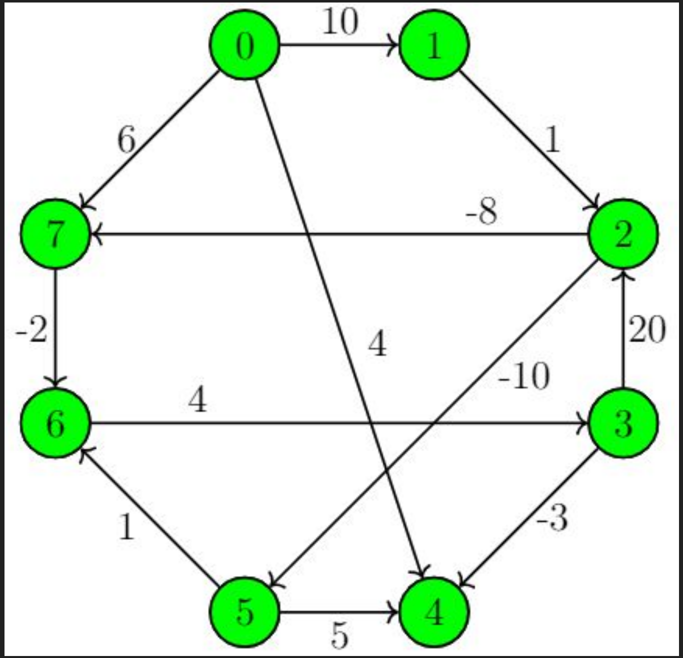


How to participate?
joinpd.com
code: see above

Solution:

Step 4: Third iteration

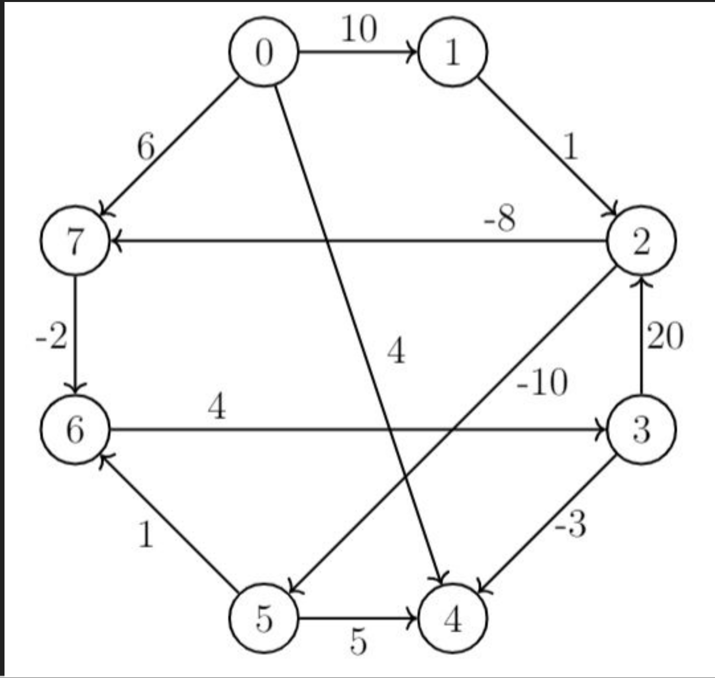
v								D(v)
0	0	0	0	0				
1	∞	10	10	10				
2	∞	∞	11	11				
3	∞	∞	∞	8				
4	∞	4	4	4				
5	∞	∞	∞	1				
6	∞	∞	4	4				
7	∞	6	6	3				



Q5.

How to participate?
joinpd.com
code: see above

While using Bellman-Ford Algorithm for the graph shown below, let D(v) be the shortest distance of vertex ‘v’ from the source vertex after 7 iterations.



What are the value of

1. D(3)
2. D(4)
3. D(5)

Sample example for answer writing
D(1) = 10

Students, write your response!

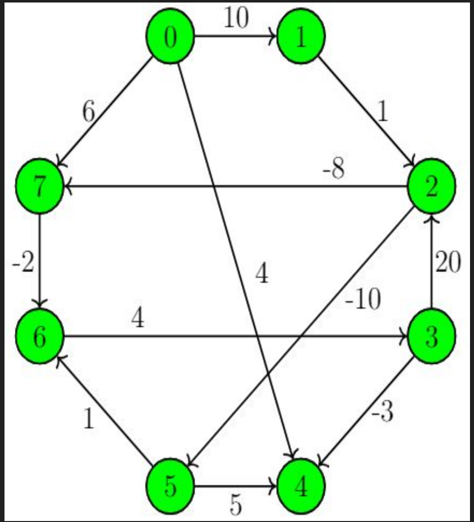
Solution:

How to participate?
joinpd.com
code: see above

Step 5: Fourth iteration -> Fifth iteration

v								D(v)
0	0	0	0	0	0			
1	∞	10	10	10	10			
2	∞	∞	11	11	11			
3	∞	∞	∞	8	8			
4	∞	4	4	4	4			
5	∞	∞	∞	1	1			
6	∞	∞	4	4	1			
7	∞	6	6	3	3			

v								D(v)
0	0	0	0	0	0	0		
1	∞	10	10	10	10	10		
2	∞	∞	11	11	11	11		
3	∞	∞	∞	8	8	5		
4	∞	4	4	4	4	4		
5	∞	∞	∞	1	1	1		
6	∞	∞	4	4	1	1		
7	∞	6	6	3	3	3		



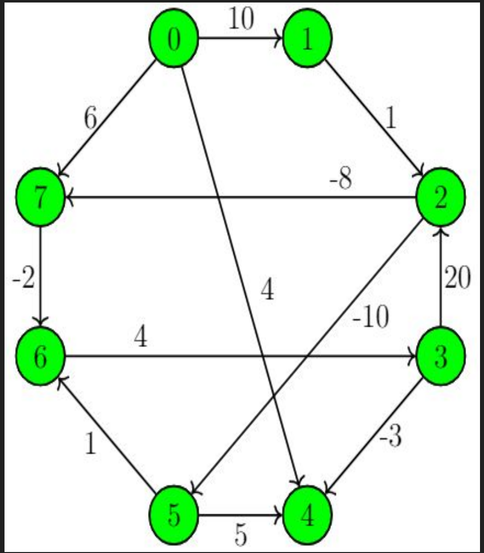
How to participate?
joinpd.com
code: see above

Solution:

Step 1: Sixth iteration -> Seventh iteration

v								D(v)
0	0	0	0	0	0	0	0	
1	∞	10	10	10	10	10	10	
2	∞	∞	11	11	11	11	11	
3	∞	∞	∞	8	8	5	5	
4	∞	4	4	4	4	4	2	
5	∞	∞	∞	1	1	1	1	
6	∞	∞	4	4	1	1	1	
7	∞	6	6	3	3	3	3	

v								D(v)
0	0	0	0	0	0	0	0	0
1	∞	10	10	10	10	10	10	10
2	∞	∞	11	11	11	11	11	11
3	∞	∞	∞	8	8	5	5	5
4	∞	4	4	4	4	4	2	2
5	∞	∞	∞	1	1	1	1	1
6	∞	∞	4	4	1	1	1	1
7	∞	6	6	3	3	3	3	3



We get D(3) = 5, D(4) = 2, D(5) = 1

Floyd-Warshall Algorithm

How to participate?
joinpd.com
code: see above

Floyd-Warshall Algorithm

How to participate?
joinpd.com
code: see above

- Floyd-Warshall algorithm is used to compute all pairs shortest paths.

Floyd-Warshall Algorithm

How to participate?
joinpd.com
code: see above

- Floyd-Warshall algorithm is used to compute all pairs shortest paths.

FACTS:

- It is applicable for directed weighted graphs of both negative and positive edge weights.
- Not applicable for graphs having negative weight cycles.

Floyd-Warshall Algorithm

How to participate?
joinpd.com
code: see above

- Floyd-Warshall algorithm is used to compute all pairs shortest paths.

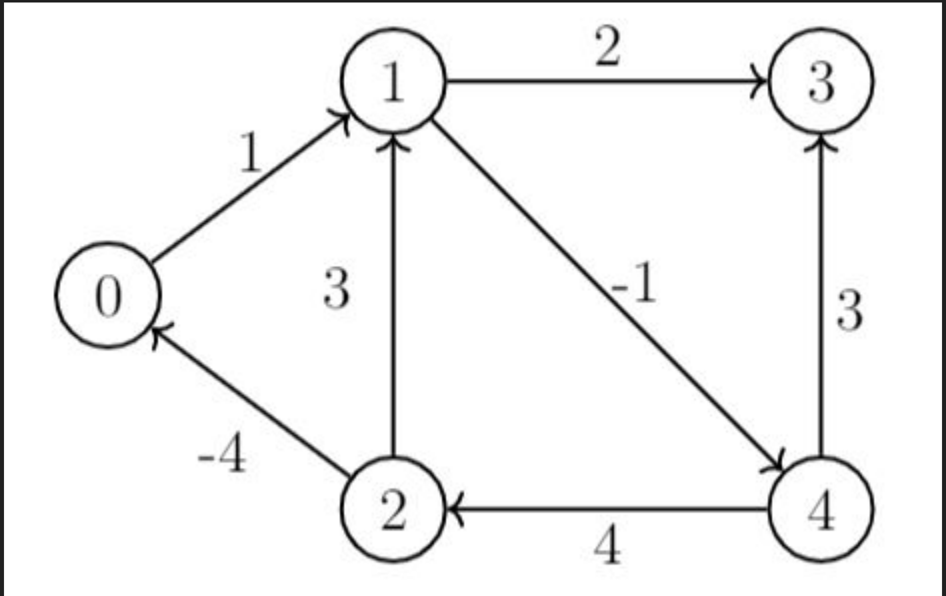
FACTS:

- It is applicable for directed weighted graphs of both negative and positive edge weights.
- Not applicable for graphs having negative weight cycles.

- SP^k [i, j] is the length of the shortest path from vertex 'i' to vertex 'j' using vertices in { 0, 1, 2, 3 ...k-1}
  - For example: If we compute SP^3 it means we need to find the shortest distance from any vertex 'i' to vertex 'j' via vertices {0, 1, 2}
  - If no path between vertices then simply the distance is ∞

Example

Use the below graph



Let us see what are the values of SP^0

Hint: Use Floyd-Warshall Algorithm

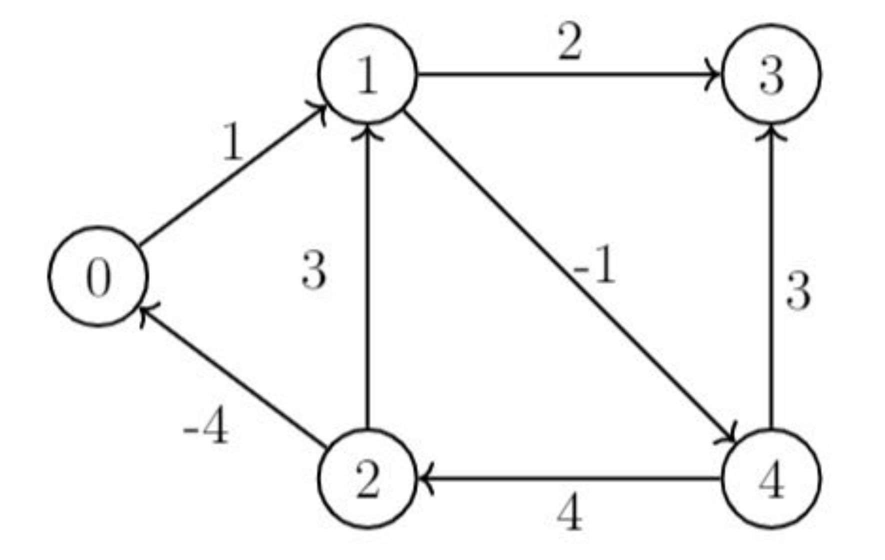
How to participate?
joinpd.com
code: see above

Solution:

SP^0 :

- $SP^0(i, j) = W(i, j)$, is the weight of an edge from vertex (i) to vertex (j).
- If no edge then consider path to be (∞).

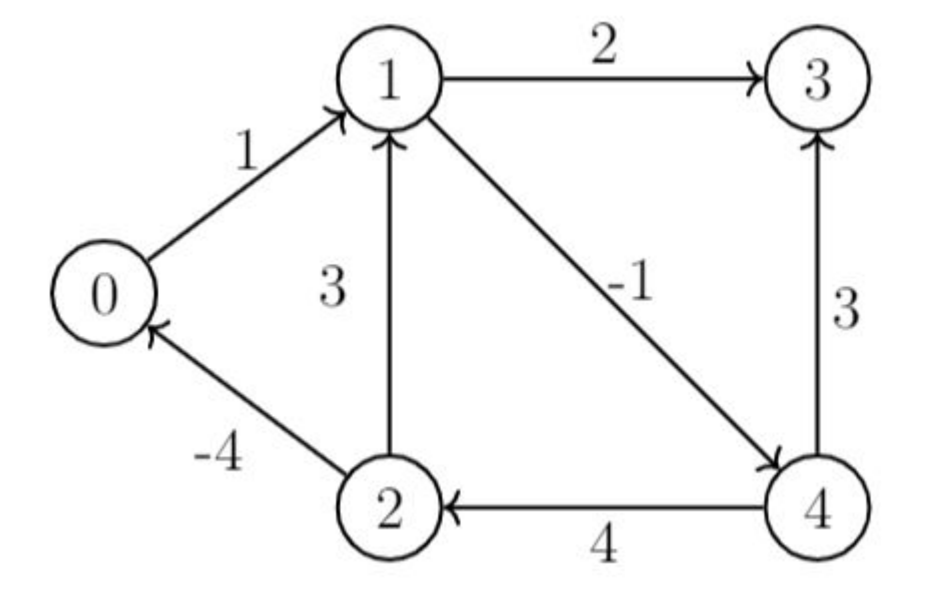
SP^0	0	1	2	3	4
0	∞	1	∞	∞	∞
1	∞	∞	∞	2	-1
2	-4	3	∞	∞	∞
3	∞	∞	∞	∞	∞
4	∞	∞	4	3	∞



Example

How to participate?
joinpd.com
code: see above

Use the below graph and answer the question shown on right hand side.



Let us see what are the values of SP^3

Hint: Use Floyd-Warshall Algorithm

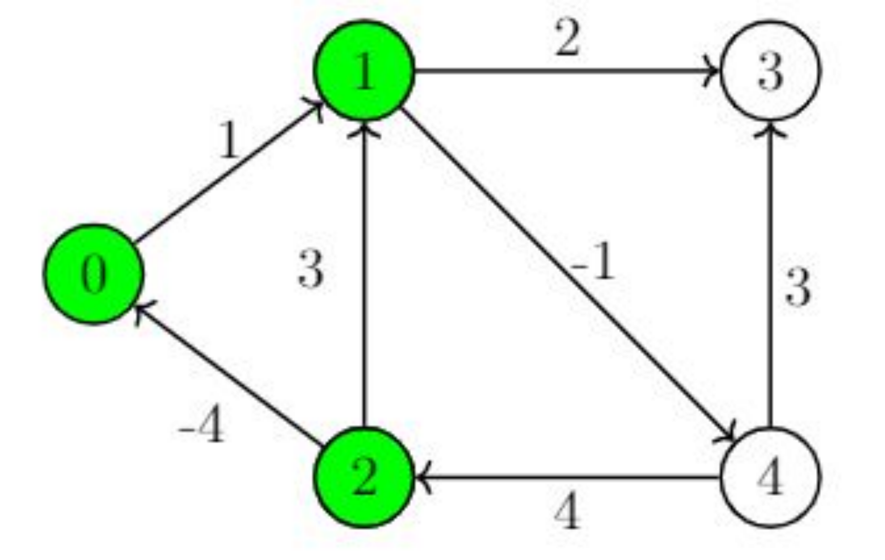
How to participate?
joinpd.com
code: see above

Solution:

SP^3 :

- It means we need to find the shortest distance from any vertex (i) to vertex (j) via vertices {0, 1, 2}

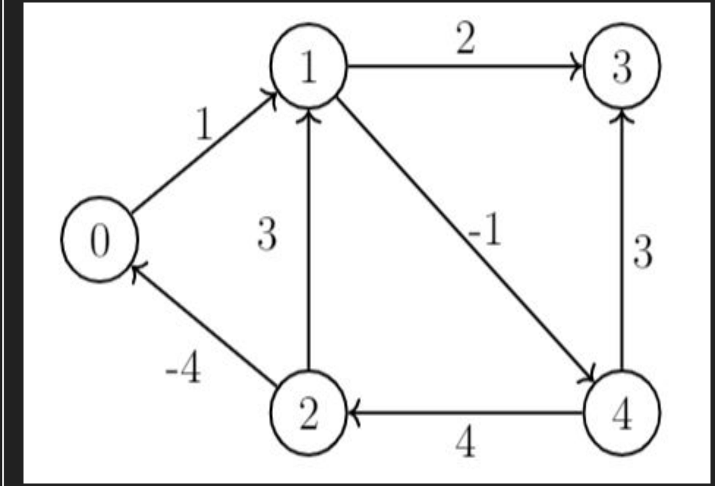
SP^3	0	1	2	3	4
0	∞	1	∞	3	0
1	∞	∞	∞	2	-1
2	-4	-3	∞	-1	-4
3	∞	∞	∞	∞	∞
4	0	1	4	3	0



Q6.1

How to participate?
joinpd.com
code: see above

Use the below graph and answer the question shown on right hand side.



Hint: Use Floyd-Warshall Algorithm

What is SP^4 ?

(a)

SP^4	0	1	2	3	4
0	∞	1	∞	3	0
1	∞	∞	∞	2	-1
2	-4	-3	∞	-1	-4
3	∞	∞	∞	∞	∞
4	0	1	4	3	0

(b)

SP^4	0	1	2	3	4
0	∞	1	∞	∞	∞
1	∞	∞	∞	2	-1
2	-4	-3	∞	-1	-4
3	∞	∞	∞	∞	∞
4	∞	1	4	3	∞

(c)

SP^4	0	1	2	3	4
0	∞	1	∞	∞	∞
1	∞	∞	∞	2	-1
2	-4	-3	∞	-1	-4
3	∞	∞	∞	∞	∞
4	0	1	4	3	∞

(d)

SP^4	0	1	2	3	4
0	∞	1	∞	∞	∞
1	∞	∞	∞	2	-1
2	-4	-3	∞	∞	-4
3	∞	∞	∞	∞	∞
4	∞	1	4	3	0

Students choose an option

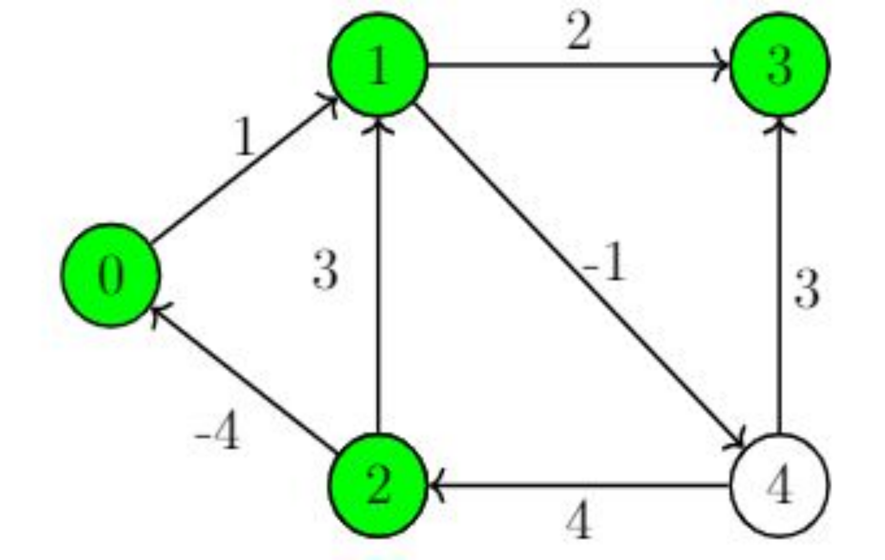
How to participate?
joinpd.com
code: see above

Solution:

SP^4 :

- It means we need to find the shortest distance from any vertex ‘i’ to vertex ‘j’ via vertices {0, 1, 2, 3}.
- Option (a) is right

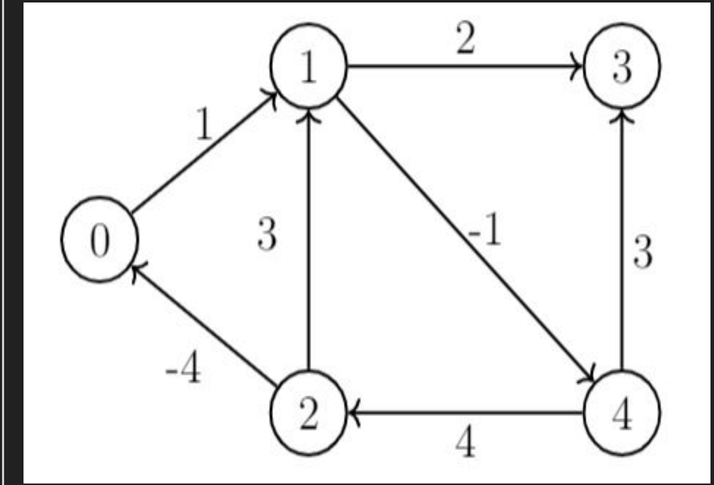
SP^4	0	1	2	3	4
0	∞	1	∞	3	0
1	∞	∞	∞	2	-1
2	-4	-3	∞	-1	-4
3	∞	∞	∞	∞	∞
4	0	1	4	3	0



Q6.2

How to participate?
joinpd.com
code: see above

Use the below graph and answer the question shown on right hand side.



Hint: Use Floyd-Warshall Algorithm

What is SP^5 ?

(a)

SP^5	0	1	2	3	4
0	0	1	4	3	-1
1	-1	0	3	2	-1
2	-4	-3	0	-1	-4
3	∞	∞	-4	∞	∞
4	0	1	4	3	0

(b)

SP^5	0	1	2	3	4
0	0	1	4	3	0
1	-1	0	3	2	-1
2	-4	-3	0	-1	-4
3	∞	∞	-4	∞	∞
4	0	1	4	3	0

(c)

SP^5	0	1	2	3	4
0	0	1	4	3	0
1	-1	0	3	2	-1
2	-4	-3	0	-1	-4
3	∞	∞	∞	∞	∞
4	0	1	4	3	0

(d)

SP^5	0	1	2	3	4
0	0	1	4	3	0
1	∞	∞	∞	2	-1
2	-4	3	∞	∞	∞
3	∞	∞	∞	∞	∞
4	∞	∞	3	4	∞

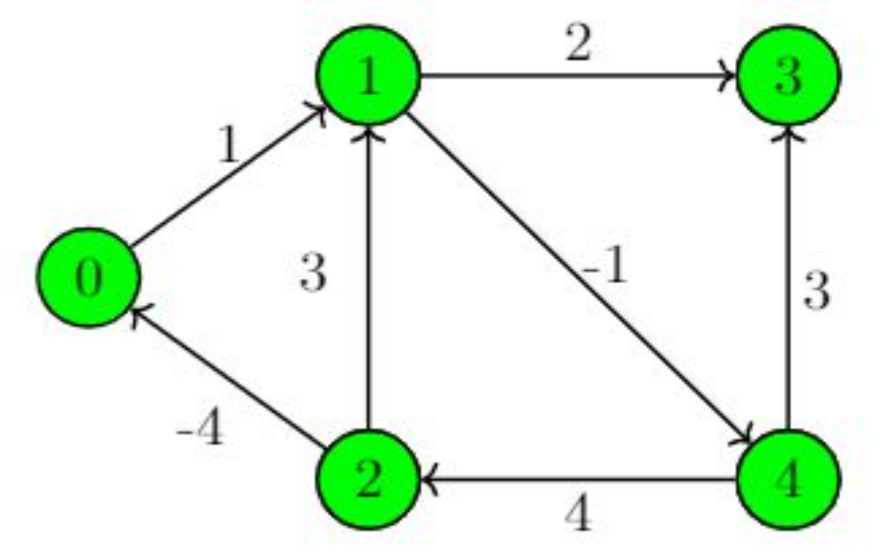
Solution:

How to participate?
joinpd.com
code: see above

SP^5 :

- It means we need to find the shortest distance from any vertex (i) to vertex (j) via vertices {0, 1, 2, 3, 4}.
- Option (c) is right

SP^5	0	1	2	3	4
0	0	1	4	3	0
1	-1	0	3	2	-1
2	-4	-3	0	-1	-4
3	∞	∞	∞	∞	∞
4	0	1	4	3	0



Thank You!!!

Vicky Kumar Sharma
Course Instructor