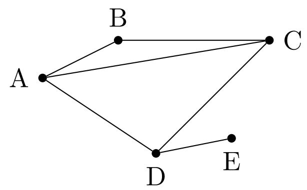
GRAPH THEORY

A DEFINITIONS

A.1 DETERMINING THE DEGREE OF VERTICES

Ex 1:



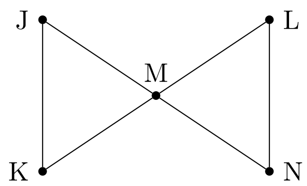
Count the degree of the vertices:

- $\deg(A) = \boxed{3}$
- $\deg(B) = \boxed{2}$
- $\deg(C) = \boxed{3}$
- $\deg(D) = \boxed{3}$
- $\deg(E) = \boxed{1}$

Answer: The degree of a vertex is the number of edges connected to it.

- Vertex A is connected to B, C, and D.
- Vertex B is connected to A and C.
- Vertex C is connected to A, B, and D.
- Vertex D is connected to A, C, and E.
- Vertex E is connected to D.

Ex 2:



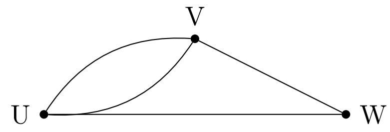
Count the degree of the vertices:

- $\deg(J) = \boxed{2}$
- $\deg(K) = \boxed{2}$
- $\deg(L) = \boxed{2}$
- $\deg(M) = \boxed{4}$
- $\deg(N) = \boxed{2}$

Answer: The degree of a vertex is the number of edges connected to it.

- Vertex J is connected to K and M.
- Vertex K is connected to J and M.
- Vertex L is connected to N and M.
- Vertex M is connected to J, K, L, and N.
- Vertex N is connected to L and M.

Ex 3:



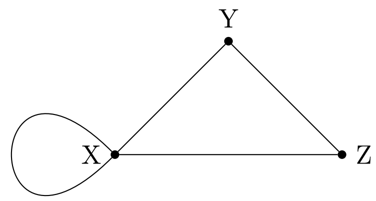
Count the degree of the vertices:

- $\deg(U) = \boxed{3}$
- $\deg(V) = \boxed{3}$
- $\deg(W) = \boxed{2}$

Answer: The degree is the count of incident edges.

- Vertex U is connected to W (1 edge) and has a double connection to V (2 edges). Total $= 3$.
- Vertex V has a double connection to U (2 edges) and is connected to W (1 edge). Total $= 3$.
- Vertex W is connected to U and V. Total $= 2$.

Ex 4:



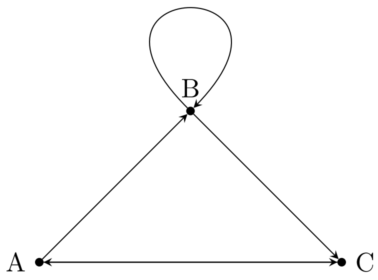
Count the degree of the vertices (remember that a loop counts for 2):

- $\deg(X) = \boxed{4}$
- $\deg(Y) = \boxed{2}$
- $\deg(Z) = \boxed{2}$

Answer: The degree of a vertex is the number of edge ends connected to it.

- Vertex X is connected to Y and Z. It also has a loop. The loop contributes 2 to the degree. Total: $1 + 1 + 2 = 4$.
- Vertex Y is connected to X and Z.
- Vertex Z is connected to X and Y.

Ex 5:



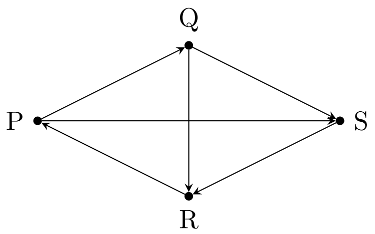
Determine the in-degree ($\deg_{in}$) and out-degree ($\deg_{out}$) of the vertices:

- Vertex A: $\deg_{in}(A) = \boxed{1}$, $\deg_{out}(A) = \boxed{2}$
- Vertex B: $\deg_{in}(B) = \boxed{2}$, $\deg_{out}(B) = \boxed{2}$
- Vertex C: $\deg_{in}(C) = \boxed{2}$, $\deg_{out}(C) = \boxed{1}$

Answer: The in-degree is the number of edges coming into a vertex. The out-degree is the number of edges going out of a vertex.

- Vertex A: 1 edge in (from C), 2 edges out (to B and C).
- Vertex B: 2 edges in (from A and the loop), 2 edges out (to C and the loop).
- Vertex C: 2 edges in (from A and B), 1 edge out (to A).

Ex 6:



Count the in-degree and out-degree:

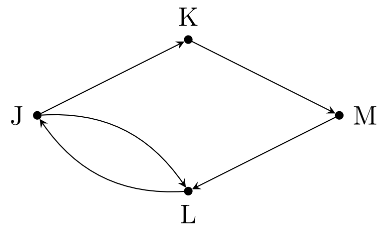
- $\deg_{in}(P) = \boxed{1}$, $\deg_{out}(P) = \boxed{2}$
- $\deg_{in}(Q) = \boxed{1}$, $\deg_{out}(Q) = \boxed{2}$
- $\deg_{in}(R) = \boxed{2}$, $\deg_{out}(R) = \boxed{1}$
- $\deg_{in}(S) = \boxed{2}$, $\deg_{out}(S) = \boxed{1}$

Answer:

- P: Incoming from R (1). Outgoing to Q, S (2).
- Q: Incoming from P (1). Outgoing to R, S (2).
- R: Incoming from S, Q (2). Outgoing to P (1).
- S: Incoming from Q, P (2). Outgoing to R (1).

A.2 IDENTIFYING PATHS AND CIRCUITS

Ex 7: Consider the following directed graph (digraph):

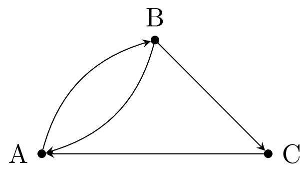


1. Is it possible to go directly from K to J?
2. Find a path from K to J.

Answer:

1. No, there is no direct arrow from K to J.
2. A valid path following the arrows is $K \to M \to L \to J$.

Ex 8: Consider the following directed graph (digraph):

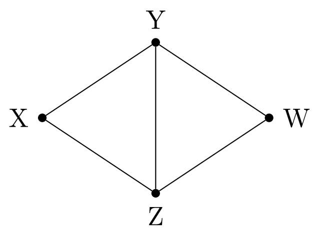


1. Is it possible to go directly from A to C?
2. Find a circuit starting and ending at A.

Answer:

1. No, there is no direct arrow from A to C. You must go through B.
2. One possible circuit is $A \to B \to A$. Another one is $A \to B \to C \to A$.

Ex 9: Consider the following undirected graph:

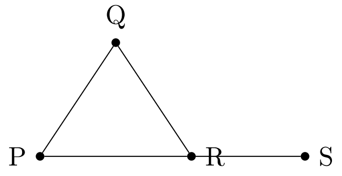


1. Is it possible to go directly from X to W?
2. Find a cycle starting and ending at Y that passes through Z.

Answer:

1. No, there is no single edge connecting X directly to W. You must pass through Y or Z.
2. A possible cycle is $Y \to Z \to X \to Y$. Another one is $Y \to Z \to W \to Y$.

Ex 10: Consider the following undirected graph:



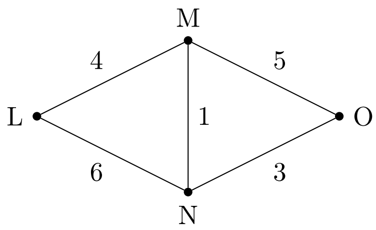
1. Is it possible to go from P to S without passing through R?
2. Find a path from Q to S.

Answer:

1. No, vertex R is the only connection to vertex S. Any path to S must go through R.
2. A possible path is $Q \to R \to S$. Another (longer) one is $Q \to P \to R \to S$.

A.3 ANALYZING PATHS IN WEIGHTED GRAPHS

Ex 11: Consider the following weighted graph representing travel costs between cities:



1. What is the weight of edge MN?

$$\boxed{1}$$

2. Calculate the total weight of the path $L \to M \to O \to N \to L$.

$$\boxed{18}$$

3. Find the lowest total weight of the path from L to O.

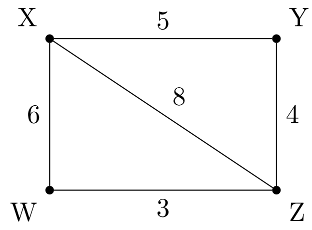
$$\boxed{8}$$

Answer:

1. The weight of edge MN is the number associated with the line connecting vertex M and vertex N. Here, it is 1.
2. We sum the weights of the consecutive edges in the path:
$$4(L \to M) + 5(M \to O) + 3(O \to N) + 6(N \to L) = 18$$
3. We compare the total weights of the possible paths from L to O:
   - $L \to M \to O$: $4 + 5 = 9$
   - $L \to N \to O$: $6 + 3 = 9$
   - $L \to M \to N \to O$: $4 + 1 + 3 = 8$

   The lowest total weight is 8.

Ex 12: Consider the following weighted graph representing driving times (in hours) between towns:



1. What is the weight of edge XZ?

$$\boxed{8}$$

2. Calculate the total weight of the path $W \to X \to Y \to Z$.

$$\boxed{15}$$

3. Find the lowest total weight of the path from W to Y.

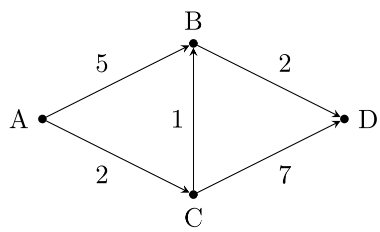
$$\boxed{7}$$

Answer:

1. The weight of edge XZ is the number associated with the diagonal line connecting vertex X and vertex Z. Here, it is 8.
2. We sum the weights of the consecutive edges in the path:
$$6(W \to X) + 5(X \to Y) + 4(Y \to Z) = 15$$
3. We compare the total weights of the possible paths from W to Y:
   - $W \to X \to Y$: $6 + 5 = 11$
   - $W \to Z \to Y$: $3 + 4 = 7$
   - $W \to X \to Z \to Y$: $6 + 8 + 4 = 18$ (Not efficient)
   - $W \to Z \to X \to Y$: $3 + 8 + 5 = 16$ (Not efficient)

   The lowest total weight is 7.

Ex 13: Consider the following weighted directed graph representing flight costs between airports. Note that the connections are one-way (indicated by arrows).



1. What is the weight of the directed edge from C to B?

$$\boxed{1}$$

2. Calculate the total weight of the path $A \to B \to D$.

$$\boxed{7}$$

3. Find the lowest total weight of a path from A to D.

$\boxed{5}$

Answer:

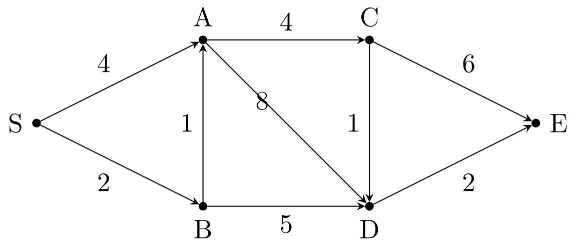
1. The weight corresponds to the arrow pointing from C to B. The value is 1.
2. We sum the weights of the path following the arrows:

$$5(A \to B) + 2(B \to D) = 7$$

3. We must find all valid paths from A to D respecting the direction of the arrows:
   - Path 1: $A \to B \to D$. Weight $= 5 + 2 = 7$.
   - Path 2: $A \to C \to D$. Weight $= 2 + 7 = 9$.
   - Path 3: $A \to C \to B \to D$. Weight $= 2 + 1 + 2 = 5$.

   The lowest total weight is 5.

Ex 14: Consider the following weighted directed graph representing latency (in ms) in a complex computer network from a source server S to a destination server E.



1. What is the weight of the directed edge from B to A?

$$\boxed{1}$$

2. Calculate the total weight of the path $S \to B \to A \to C \to D \to E$.

$$\boxed{10}$$

3. Find the lowest total weight of a path from S to E.

$$\boxed{9}$$

Answer:

1. The weight corresponds to the vertical arrow pointing upwards from B to A. The value is 1.
2. We sum the weights of the edges in the path:

$$2(S \to B) + 1(B \to A) + 4(A \to C) + 1(C \to D) + 2(D \to E) = 10$$

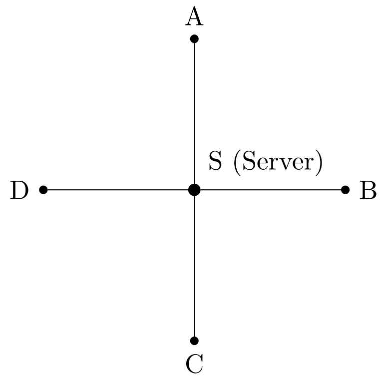
3. To find the lowest weight, we analyze the main paths:
   - **Top path:** $S \to A \to C \to E$: $4 + 4 + 6 = 14$.
   - **Bottom path:** $S \to B \to D \to E$: $2 + 5 + 2 = 9$.
   - **Crossing Down via A:** $S \to A \to D \to E$: $4 + 8 + 2 = 14$.

   The lowest total weight is 9 (Path $S \to B \to D \to E$).

A.4 MODELLING SITUATIONS WITH GRAPHS

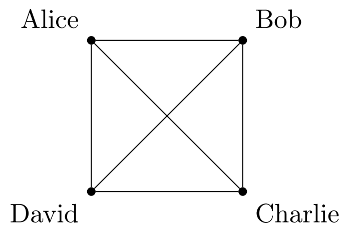
Ex 15: A computer network consists of 4 computers (labeled A, B, C, and D) connected to a central server (labeled S). Every computer is connected directly to the server, but not to each other.
Draw a graph to illustrate this network.

Answer: The graph has 5 vertices: $V = \{S, A, B, C, D\}$.
The edges represent the connections: $E = \{SA, SB, SC, SD\}$.
This forms a "star graph".



Ex 16: Four friends (Alice, Bob, Charlie, and David) meet for dinner. Everyone shakes hands with everyone else exactly once. Draw a graph to represent these handshakes, where vertices represent the people and edges represent the handshakes.

Answer: The graph has 4 vertices: $V = \{A, B, C, D\}$.
Since everyone shakes hands with everyone, every vertex is connected to every other vertex.
This forms a "complete graph" (K_4).

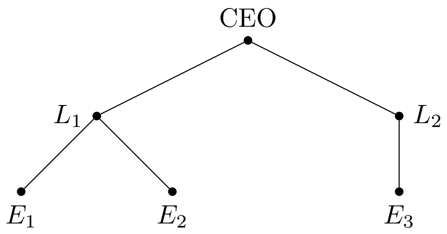


Ex 17: A train line serves four stations: North Station, Central Station, South Station, and Airport. The track goes from North to Central, then from Central to South, and finally from South to Airport. The train travels the same path in both directions.
Draw a graph to illustrate the connections between these stations.

Answer: The vertices represent the stations.
The edges represent the direct rail links.
This forms a "path graph".

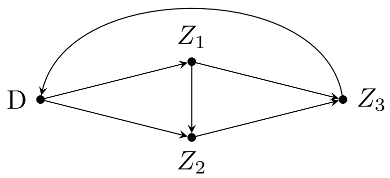
Ex 18: In a small business, a CEO manages two team leaders (L_1 and L_2). Team leader L_1 supervises two employees (E_1 and E_2), while team leader L_2 supervises one employee (E_3).
Draw a graph representing this hierarchy.

Answer: The vertices are the people in the company.
The edges represent the supervision relationship.
This structure forms a "tree".



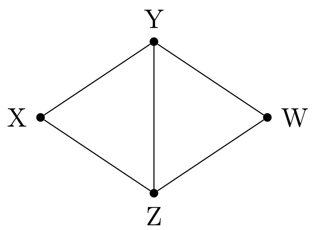
Ex 19: A food delivery company has a distribution center (D) and three delivery zones (Z_1, Z_2, Z_3). The delivery trucks leave the distribution center to go to Z_1 or Z_2. From Z_1, they can go to Z_2 or Z_3. From Z_2, they can only go to Z_3. Once in Z_3, they return directly to the distribution center D.
Draw a directed graph to illustrate these delivery routes.

Answer: The graph has 4 vertices: $V = \{D, Z_1, Z_2, Z_3\}$.
The directed edges represent the permitted routes.
This forms a directed graph (digraph).



A.5 CLASSIFYING SEQUENCES OF VERTICES

MCQ 20: Consider the following graph:



How is the sequence $X \to Y \to Z \to W$ classified?

☒ A path

☒ A trail

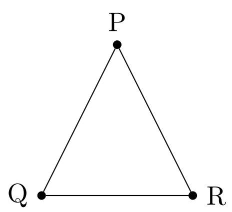
☐ A cycle

☐ A circuit

Answer: The sequence is $X \to Y \to Z \to W$.

- **Path**: True. No vertex is repeated (X, Y, Z, W are all distinct). A path is automatically a trail.
- **Trail**: True. Since no vertex is repeated, no edge is repeated either.
- **Cycle**: False. The start vertex (X) is not the end vertex (W).
- **Circuit**: False. The start vertex is not the end vertex.

MCQ 21: Consider the following graph:



How is the sequence $P \to Q \to R \to P$ classified?

☐ A path

☒ A trail

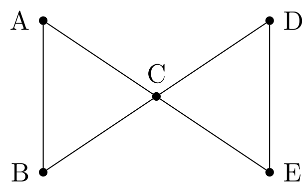
☒ A cycle

☒ A circuit

Answer: The sequence is $P \to Q \to R \to P$.

- **Path**: False. The start vertex P is repeated at the end. By definition, a path has no repeated vertices.
- **Trail**: True. No edge is repeated.
- **Cycle**: True. It is a closed path (start=end) with no repeated internal vertices or edges.
- **Circuit**: True. It is a closed trail. A cycle is a specific type of circuit.

MCQ 22: Consider the following graph ("Bowtie"):



How is the sequence $A \to B \to C \to D \to E \to C \to A$ classified?

☐ A path

☒ A trail

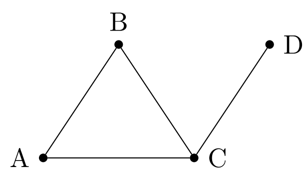
☐ A cycle

☒ A circuit

Answer: The sequence is $A \to B \to C \to D \to E \to C \to A$.

- **Path**: False. Vertex C is visited twice.
- **Trail**: True. No edge is repeated.
- **Cycle**: False. Vertex C is repeated internally.
- **Circuit**: True. It is a closed trail (start=end, no repeated edges).

MCQ 23: Consider the following graph:



How is the sequence $D \to C \to B \to A \to C$ classified?

☐ A path

☒ A trail

☐ A cycle

☐ A circuit

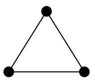
Answer: The sequence is $D \to C \to B \to A \to C$.

- **Path**: False. The vertex C is repeated.
- **Trail**: True. The edges traversed are $(D, C), (C, B), (B, A), (A, C)$. They are all distinct.
- **Cycle**: False. The sequence is not closed (start $D \neq$ end C).
- **Circuit**: False. The sequence is not closed.

B PROPERTIES OF GRAPHS

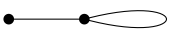
B.1 IDENTIFYING GRAPH PROPERTIES

MCQ 24: Which of the following graphs is a **simple graph**?

☒ 

☐ 

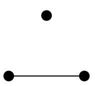
☐ 

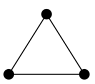
☐ 

Answer: A simple graph has no loops and no multiple edges between vertices.

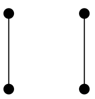
- The first graph is a triangle. It has no loops and single edges between vertices. It is simple.
- The second graph has two edges connecting the same pair of vertices (multiple edges).
- The third graph has a loop.
- The fourth graph has a loop on the right vertex.

MCQ 25: Which of the following graphs is **connected**?

☐ 

☒ 

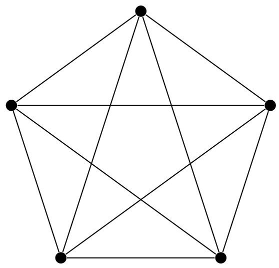
☐

☐ 

Answer: A graph is connected if there is a path between any pair of vertices.

- The first graph has an isolated vertex at the top. Not connected.
- The second graph is a triangle; you can travel between any two vertices. It is connected.
- The third graph has an isolated vertex on the right. Not connected.
- The fourth graph consists of two separate vertical lines. You cannot go from left to right. Not connected.

Ex 26: Consider the complete graph K_5 shown below.



1. How many vertices does it have?

$$\boxed{5}$$

2. How many edges does it have?

$$\boxed{10}$$

Answer:

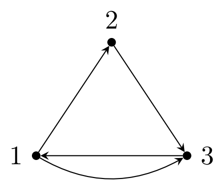
1. By definition, K_5 has $n = 5$ vertices.
2. In a complete graph, every vertex is connected to every other vertex. The number of edges is $\frac{n(n-1)}{2}$.

$$\text{Edges} = \frac{5(5-1)}{2} = \frac{20}{2} = 10$$

C ADJACENCY MATRICES

C.1 WRITING ADJACENCY MATRICES

Ex 27: Write the adjacency matrix $\mathbf{M}$ for the following directed graph:



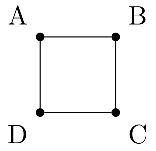
Answer: The vertices are 1, 2, 3.

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Explanation:

- Row 1: edges $1 \to 2$ and $1 \to 3$ (bend right). So 1 in col 2 and 3.
- Row 2: edge $2 \to 3$. So 1 in col 3.
- Row 3: edge $3 \to 1$. So 1 in col 1.

Ex 28: Write the adjacency matrix $\mathbf{M}$ for the following undirected graph:



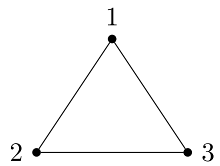
Answer: The vertices are A, B, C, D.

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Explanation:

- Row A: connected to B and D. So 1 in col 2 and 4.
- Row B: connected to A and C. So 1 in col 1 and 3.
- Row C: connected to B and D. So 1 in col 2 and 4.
- Row D: connected to A and C. So 1 in col 1 and 3.

Ex 29: Write the adjacency matrix **M** for the following undirected graph:



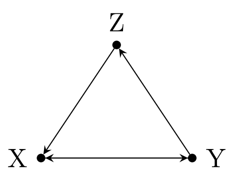
Answer: The vertices are 1, 2, 3.

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Explanation:

- Every vertex is connected to every other vertex (K_3).
- The matrix is symmetric with 0s on the diagonal.

Ex 30: Write the adjacency matrix **M** for the following directed graph:



Answer: The vertices are X, Y, Z.

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Explanation:

- Row X: edge $X \to Y$. So 1 in col 2.
- Row Y: edges $Y \to X$ and $Y \to Z$. So 1 in col 1 and 3.
- Row Z: edge $Z \to X$. So 1 in col 1.

C.2 DETERMINING THE NUMBER OF WALKS

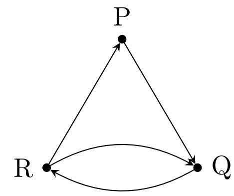
Ex 31: Consider the directed graph H given by the following adjacency matrix (vertices are ordered alphabetically: P, Q, R):

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

1. Draw the directed graph H.
2. Find the number of walks of length 2 from vertex P to vertex R.
3. Find the number of walks of length 3 from vertex R to vertex Q.

Answer:

1. The graph has vertices P, Q, R. The directed edges (arrows) are $P \to Q$, $Q \to R$, $R \to P$, and $R \to Q$.



2. We calculate $\mathbf{M}^2$:

$$\mathbf{M}^2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

The element $(1, 3)$ corresponds to walks from P to R. There is **1** such walk ($P \to Q \to R$).

3. We calculate $\mathbf{M}^3$:

$$\mathbf{M}^3 = \mathbf{M}^2 \times \mathbf{M} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

The element at row 3, column 2 is 1. There is **1** walk of length 3 from vertex R to vertex Q ($R \to Q \to R \to Q$).

Ex 32: Consider the undirected graph K given by the following adjacency matrix (vertices are ordered alphabetically: X, Y, Z):

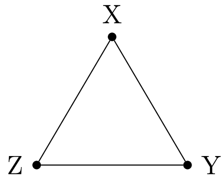
$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

1. Draw the undirected graph K.
2. Find the number of walks of length 2 from vertex X to vertex Y.

3. Find the number of walks of length 3 from vertex Y to vertex Z.

Answer:

1. The graph has vertices X, Y, Z. Since $\mathbf{M}$ is symmetric, the edges are undirected: (X,Y), (X,Z), and (Y,Z). It is a triangle.



2. We calculate $\mathbf{M}^2$:
$$\begin{aligned}\mathbf{M}^2 &= \begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix} \times \begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}\\ &= \begin{pmatrix}2&1&1\\1&2&1\\1&1&2\end{pmatrix}\end{aligned}$$
The element $(1,2)$ corresponds to walks from X to Y. There is **1** such walk ($X \to Z \to Y$).

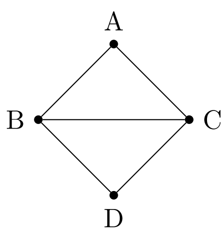
3. We calculate $\mathbf{M}^3$:
$$\begin{aligned}\mathbf{M}^3 &= \mathbf{M}^2 \times \mathbf{M}\\ &= \begin{pmatrix}2&1&1\\1&2&1\\1&1&2\end{pmatrix} \times \begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}\\ &= \begin{pmatrix}2&3&3\\3&2&3\\3&3&2\end{pmatrix}\end{aligned}$$
The element at row 2, column 3 (corresponding to Y to Z) is 3. There are **3** walks of length 3 ($Y \to X \to Y \to Z$, $Y \to Z \to Y \to Z$, $Y \to Z \to X \to Z$).

Ex 33: Consider the graph G given by the following adjacency matrix (vertices are ordered alphabetically: A, B, C, D):
$$\mathbf{M} = \begin{pmatrix}0&1&1&0\\1&0&1&1\\1&1&0&1\\0&1&1&0\end{pmatrix}$$

1. Draw the non-directed graph G.
2. Find the number of walks of length 2 from vertex A to vertex D.
3. Find the number of walks of length 3 from vertex B to itself.

Answer:

1. The graph has vertices A, B, C, D. The edges are (A,B), (A,C), (B,C), (B,D), (C,D).



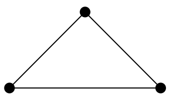
2. We calculate $\mathbf{M}^2$:
$$\begin{aligned}\mathbf{M}^2 &= \begin{pmatrix}0&1&1&0\\1&0&1&1\\1&1&0&1\\0&1&1&0\end{pmatrix} \times \begin{pmatrix}0&1&1&0\\1&0&1&1\\1&1&0&1\\0&1&1&0\end{pmatrix}\\ &= \begin{pmatrix}2&1&1&2\\1&3&2&1\\1&2&3&1\\2&1&1&2\end{pmatrix}\end{aligned}$$
The element $(1,4)$ corresponds to walks from A to D. There are **2** such walks ($A \to B \to D$ and $A \to C \to D$).

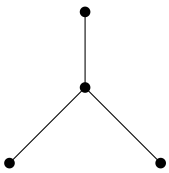
3. We calculate $\mathbf{M}^3$:
$$\begin{aligned}\mathbf{M}^3 &= \mathbf{M}^2 \times \mathbf{M}\\ &= \begin{pmatrix}2&1&1&2\\1&3&2&1\\1&2&3&1\\2&1&1&2\end{pmatrix} \times \begin{pmatrix}0&1&1&0\\1&0&1&1\\1&1&0&1\\0&1&1&0\end{pmatrix}\\ &= \begin{pmatrix}2&5&5&2\\5&4&5&5\\5&5&4&5\\2&5&5&2\end{pmatrix}\end{aligned}$$
The element at row 2, column 2 is 4. There are **4** walks of length 3 from vertex B to itself ($B \to A \to C \to B$, $B \to C \to A \to B$, $B \to C \to D \to B$, $B \to D \to C \to B$).

D TREES AND MINIMUM SPANNING TREES

D.1 IDENTIFYING TREES

MCQ 34: Which of the following graphs are **trees**?

☐ 

☒ 

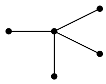
☐

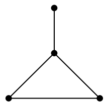
☒

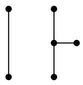
Answer: A tree must be connected and have no cycles.

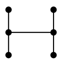
- The first graph is a triangle (cycle), so it is not a tree.
- The second graph is connected and has no cycles. It is a tree.
- The third graph is not connected (two separate components). It is not a tree (it's a forest).
- The fourth graph is a simple path. It is connected and has no cycles. It is a tree.

MCQ 35: Which of the following graphs are **trees**?

☒ 

☐ 

☐ 

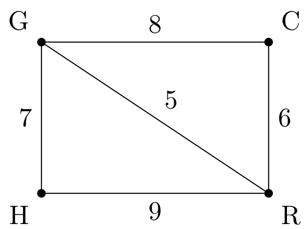
☒ 

Answer: A tree must be connected and have no cycles.

- The first graph connects all vertices without forming any closed loops. It is a tree.
- The second graph contains a triangle (a cycle of length 3) at the bottom. It is not a tree.
- The third graph consists of two separate components (a line on the left and a T-shape on the right). It is not connected, so it is not a tree (it is a forest).
- The fourth graph (H-shape) connects all vertices and has no closed loops. It is a tree.

D.2 OPTIMIZING NETWORKS

Ex 36: A city council plans to build bike lanes connecting four parks: Green Park (G), Central Park (C), Riverside Park (R), and Hilltop Park (H). The costs (in thousands of euros) to build lanes between these parks are given in the weighted graph below. The council wants to connect all parks with the minimum total cost.

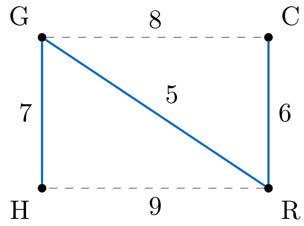


1. List the edges in ascending order of weight.
2. Determine the edges that form the Minimum Spanning Tree (MST).
3. Calculate the minimum cost to connect all the parks.

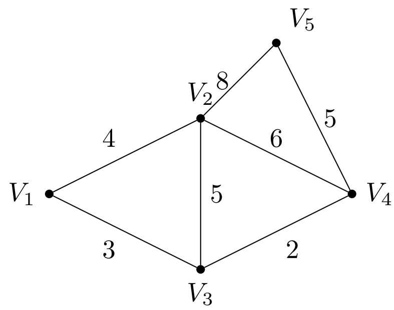
Answer:

1. **Edges ordered by weight:**
   - GR (5)
   - CR (6)
   - GH (7)
   - GC (8)
   - HR (9)
2. **Kruskal's Algorithm:**
   - Select GR (5).
   - Select CR (6). No cycle.
   - Select GH (7). No cycle.
   - All vertices (G, C, R, H) are connected.
   - Reject GC (8) and HR (9) as they form cycles.
3. **Total Cost:** $5 + 6 + 7 = 18$ (thousand euros).

The Minimum Spanning Tree is shown below:



Ex 37: A utility company needs to connect five rural villages (V_1, V_2, V_3, V_4, V_5) to the power grid. The distances (in km) between the villages are given in the weighted graph below.
The company wants to minimize the total length of power lines installed.

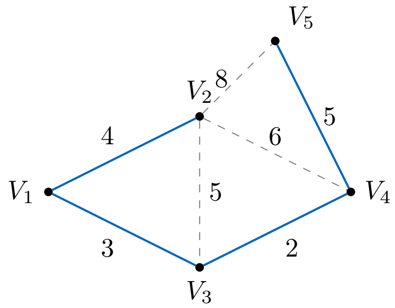


1. List the edges in ascending order of weight.
2. Determine the edges that form the Minimum Spanning Tree (MST).
3. Calculate the minimum total length of the power lines.

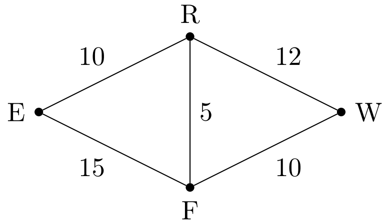
Answer:

1. **Edges ordered by weight:**
   - V_3V_4 (2)
   - V_1V_3 (3)
   - V_1V_2 (4)
   - V_2V_3 (5)
   - V_4V_5 (5)
   - V_2V_4 (6)
   - V_2V_5 (8)
2. **Kruskal's Algorithm:**
   - Select V_3V_4 (2).
   - Select V_1V_3 (3). No cycle.
   - Select V_1V_2 (4). No cycle.
   - Reject V_2V_3 (5) (forms cycle $V_1 - V_2 - V_3 - V_1$).
   - Select V_4V_5 (5). No cycle.
   - All vertices (V_1 to V_5) are connected. Stop.
3. **Total Length:** $2 + 3 + 4 + 5 = 14$ km.

The Minimum Spanning Tree is shown below:



Ex 38: An amusement park wants to install a fiber optic network to connect four main zones: the Entrance (E), the Roller Coasters (R), the Food Court (F), and the Water Park (W). The costs of laying the cables between these zones (in thousands of dollars) are shown in the weighted graph below.
The park wants to connect all zones with the minimum possible cost.

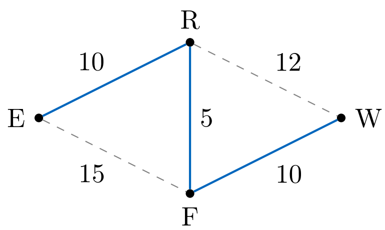


1. List the edges in ascending order of weight.
2. Determine the edges that form the Minimum Spanning Tree (MST).
3. Calculate the minimum cost to connect all the zones.

Answer:

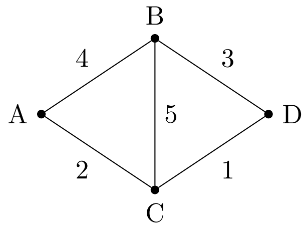
1. **Edges ordered by weight:**
   - RF (5)
   - ER (10)
   - FW (10)
   - RW (12)
   - EF (15)
2. **Kruskal's Algorithm:**
   - Select RF (5).
   - Select ER (10). No cycle.
   - Select FW (10). No cycle.
   - All vertices (E, R, F, W) are now connected.
   - Reject RW (12) as it connects R and W which are already connected via F.
   - Reject EF (15) as it connects E and F which are already connected via R.
3. **Total Cost:** $5 + 10 + 10 = 25$ (thousand dollars).

The Minimum Spanning Tree is shown below:



D.3 DETERMINING THE MINIMUM SPANNING TREE

Ex 39: Consider the weighted graph below:



1. List the edges in ascending order of weight.
2. Using Kruskal's algorithm or Prim's algorithm, find the total weight of the Minimum Spanning Tree (MST).

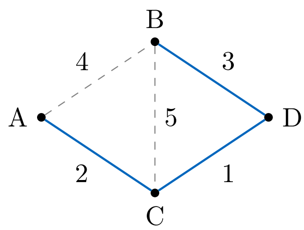
Answer:

1. Edges ordered by weight:
   - CD (1)
   - AC (2)
   - BD (3)
   - AB (4)
   - BC (5)
2. **Kruskal's Algorithm:**
   - Select CD (1).
   - Select AC (2). No cycle.
   - Select BD (3). No cycle.
   - (Reject AB and BC as they would form cycles).

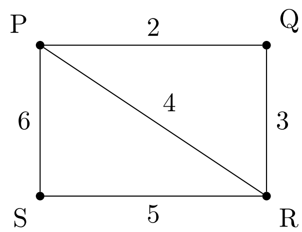
   Prim's Algorithm (Starting from A):
   - Start at A. Available edges: $AC(2), AB(4)$. Select AC. Tree: $\{A, C\}$.
   - Available from tree: $CD(1), CB(5), AB(4)$. Select CD. Tree: $\{A, C, D\}$.
   - Available from tree: $BD(3), CB(5), AB(4)$. Select BD. Tree: $\{A, C, D, B\}$.

   The MST consists of edges CD, AC, BD. Total weight $= 1 + 2 + 3 = 6$.

The Minimum Spanning Tree is shown below:



Ex 40: Consider the weighted graph below:



1. List the edges in ascending order of weight.
2. Using Kruskal's algorithm or Prim's algorithm, find the total weight of the Minimum Spanning Tree (MST).

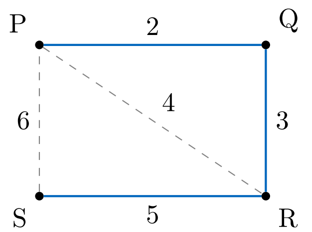
Answer:

1. Edges ordered by weight:
   - PQ (2)
   - QR (3)
   - PR (4)
   - RS (5)
   - SP (6)
2. **Kruskal's Algorithm:**
   - Select PQ (2).
   - Select QR (3). No cycle.
   - Consider PR (4). This forms a cycle $P - Q - R - P$. **Reject**.
   - Select RS (5). No cycle.

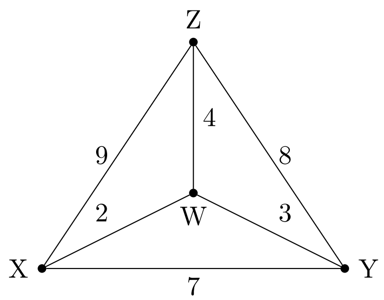
   Prim's Algorithm (Starting from P):
   - Start at P. Available edges: $PQ(2), PR(4), PS(6)$. Select PQ. Tree: $\{P, Q\}$.
   - Available from tree: $QR(3), PR(4), PS(6)$. Select QR. Tree: $\{P, Q, R\}$.
   - Available from tree: $RS(5), PS(6)$ ($PR(4)$ is not available). Select RS. Tree: $\{P, Q, R, S\}$.

   The MST consists of edges PQ, QR, RS. Total weight $= 2 + 3 + 5 = 10$.

The Minimum Spanning Tree is shown below:



Ex 41: Consider the weighted graph below with a central node:



1. List the edges in ascending order of weight.
2. Using Kruskal's algorithm or Prim's algorithm, find the total weight of the Minimum Spanning Tree (MST).

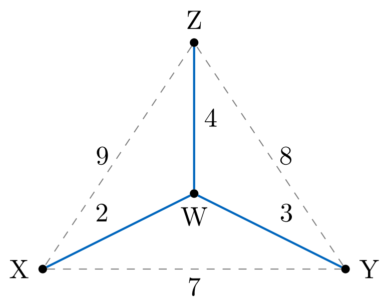
Answer:

1. Edges ordered by weight:
   - WX (2)
   - WY (3)
   - WZ (4)
   - XY (7)
   - YZ (8)
   - XZ (9)
2. **Kruskal's Algorithm:**
   - Select WX (2).
   - Select WY (3). No cycle.
   - Select WZ (4). No cycle.
   - All nodes (X, Y, Z, W) are connected.

   Prim's Algorithm (Starting from W):
   - Start at W. Available edges: $WX(2), WY(3), WZ(4)$. Select WX. Tree: $\{W, X\}$.
   - Available from tree: $WY(3), WZ(4), XY(7), XZ(9)$. Select WY. Tree: $\{W, X, Y\}$.
   - Available from tree: $WZ(4), YZ(8), XZ(9)$.. Select WZ. Tree: $\{W, X, Y, Z\}$.

   The MST consists of edges WX, WY, WZ. Total weight $= 2 + 3 + 4 = 9$.

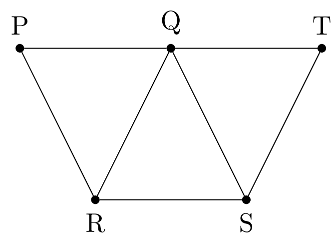
The Minimum Spanning Tree is shown below:



E EULERIAN GRAPHS

E.1 IDENTIFYING EULERIAN CIRCUITS AND TRAILS

Ex 42: Consider the following graph:



Does this graph contain an Eulerian circuit? If not, does it contain an Eulerian trail? Justify your answer.

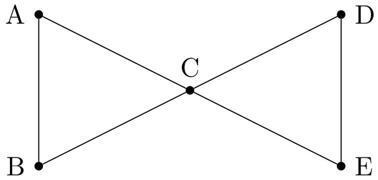
Answer: Let's find the degree of each vertex:

- $\deg(P) = 2$ (even)
- $\deg(Q) = 4$ (even)
- $\deg(R) = 3$ (odd)

- $\deg(S) = 3$ (odd)
- $\deg(T) = 2$ (even)

Since there are vertices with odd degrees (R and S), the graph does **not** have an Eulerian circuit.
However, there are exactly two vertices with an odd degree. Therefore, the graph has an **Eulerian trail**. The trail must start at one odd vertex (R or S) and end at the other (S or R).
For example: $R \to P \to Q \to R \to S \to T \to Q \to S$.

Ex 43: Consider the following graph:



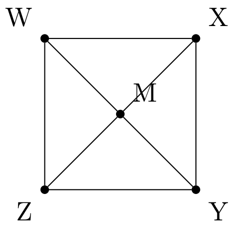
Does this graph contain an Eulerian circuit? If not, does it contain an Eulerian trail? Justify your answer.

Answer: Let's find the degree of each vertex:

- $\deg(A) = 2$ (even)
- $\deg(B) = 2$ (even)
- $\deg(C) = 4$ (even)
- $\deg(D) = 2$ (even)
- $\deg(E) = 2$ (even)

Since **all vertices have an even degree** and the graph is connected, it contains an **Eulerian circuit**.
For example, a Eulerian circuit is: $C \to A \to B \to C \to D \to E \to C$.

Ex 44: Consider the following graph:



Does this graph contain an Eulerian circuit? If not, does it contain an Eulerian trail? Justify your answer.

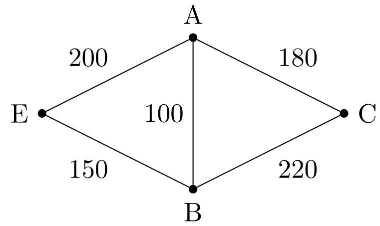
Answer: Let's find the degree of each vertex:

- $\deg(M) = 4$ (even)
- $\deg(W) = 3$ (odd)
- $\deg(X) = 3$ (odd)
- $\deg(Y) = 3$ (odd)
- $\deg(Z) = 3$ (odd)

This graph has **4 vertices with an odd degree** (W, X, Y, Z). Since a graph can have an Eulerian trail only if it has 0 or 2 odd vertices, this graph contains **neither an Eulerian circuit nor an Eulerian trail**.

E.2 SOLVING THE CHINESE POSTMAN PROBLEM

Ex 45: A park ranger needs to inspect every path in a park shown by the graph below. The lengths of the paths are given in meters. The ranger starts at the entrance E and must return to E.



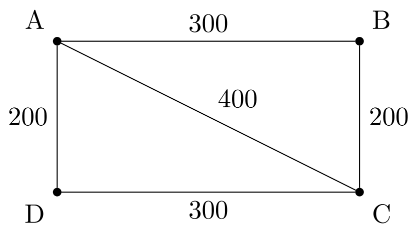
1. Explain why it is impossible to traverse every path exactly once and return to E.
2. Solve the Chinese Postman Problem to find the minimum distance the ranger must travel.

Answer:

1. We check the degrees of the vertices: $\deg(E) = 2$, $\deg(A) = 3$, $\deg(B) = 3$, $\deg(C) = 2$. Vertices A and B have odd degrees. An Eulerian circuit exists only if all vertices have even degrees. Therefore, it is impossible to traverse every edge exactly once and return to the start.
2. To solve the Chinese Postman Problem:
   - Total weight of the graph: $200+150+100+180+220 = 850$ m.
   - Odd vertices: A and B. We must repeat the path between them.
   - Shortest path between A and B: The direct edge AB has weight 100. (Alternative path $A \to E \to B$ is $200 + 150 = 350$, much longer).
   - Add this weight to the total: $850 + 100 = 950$ m.
   - **Example of a valid walk:** $E \to A \to C \to B \to A \to B \to E$. (Note that the edge AB is traversed twice).

   The minimum distance is 950 m.

Ex 46: A snowplow must clear snow from all the roads connecting four intersections A, B, C, and D. The distances in meters are shown in the graph below. The snowplow starts at the garage located at intersection A and must return to A at the end of the shift.

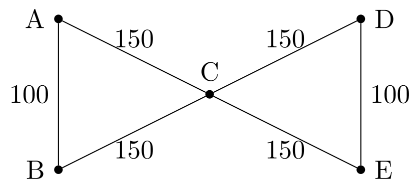


1. Explain why an Eulerian circuit does not exist in this graph.
2. Solve the Chinese Postman Problem to find the minimum distance the snowplow must travel to clear every road and return to A.

Answer:

1. We calculate the degrees of the vertices: $\deg(B) = 2$, $\deg(D) = 2$ (Even). $\deg(A) = 3$, $\deg(C) = 3$ (Odd). Since vertices A and C have odd degrees, the graph is not Eulerian (it has an Eulerian trail but not a circuit). Therefore, some edges must be repeated to return to the start.
2. To solve the Chinese Postman Problem:
   - Total weight of the graph edges: $300+200+300+200+400 = 1400$ m.
   - Odd vertices: A and C. We need to traverse the shortest path between them an extra time.
   - Possible paths between A and C:
     - Direct (AC): 400 m.
     - Via B ($A \to B \to C$): $300 + 200 = 500$ m.
     - Via D ($A \to D \to C$): $200 + 300 = 500$ m.
   - The shortest path is the direct edge AC (400 m).
   - Total minimum distance $= 1400 + 400 = 1800$ m.
   - **Example of a valid walk:** $A \to B \to C \to A \to C \to D \to A$. (Note that the diagonal AC is traversed twice).

Ex 47: A maintenance worker needs to paint the lines on the paths of a small park, represented by the graph below. The lengths are in meters. The worker starts at the central intersection C and must return to C after painting every path exactly once.



1. Determine the degree of each vertex.
2. Does this graph have an Eulerian circuit? If so, calculate the total distance the worker will travel.

Answer:

1. Degrees of vertices:
   - $\deg(A) = 2$ (connected to B and C)
   - $\deg(B) = 2$ (connected to A and C)
   - $\deg(C) = 4$ (connected to A, B, D, and E)
   - $\deg(D) = 2$ (connected to C and E)
   - $\deg(E) = 2$ (connected to C and D)
2. Since every vertex has an even degree and the graph is connected, an Eulerian circuit exists. This means it is possible to traverse every edge exactly once and return to the start. The total distance is simply the sum of the weights of all edges:

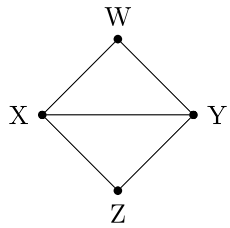
   $$100(AB)+150(AC)+150(BC)+150(CD)+150(CE)+100(DE) = 800 \text{ m}$$

   The worker will travel 800 meters.
   Example of a valid walk: $C \to A \to B \to C \to D \to E \to C$. Every edge is traversed exactly once.

F HAMILTONIAN GRAPHS

F.1 IDENTIFYING HAMILTONIAN PATHS AND CYCLES

Ex 48: Consider the following graph:

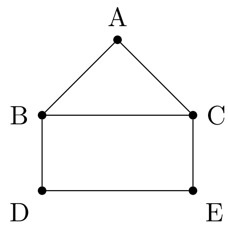


1. Find a Hamiltonian path from W to Z.
2. Does this graph contain a Hamiltonian cycle? If yes, list the vertices in order.

Answer:

1. A Hamiltonian path visits every vertex exactly once. Path: $W \to X \to Y \to Z$.
2. Yes. Cycle: $W \to X \to Z \to Y \to W$.

Ex 49: Consider the following graph:

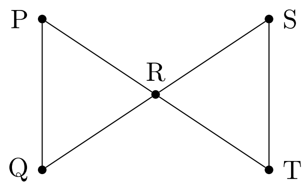


1. Find a Hamiltonian path from D to E.
2. Does this graph contain a Hamiltonian cycle? If yes, list the vertices in order.

Answer:

1. A Hamiltonian path visits every vertex exactly once. Path: $D \to B \to A \to C \to E$.
2. Yes. Cycle: $D \to B \to A \to C \to E \to D$.

Ex 50: Consider the "Bowtie" graph below:



1. Find a Hamiltonian path from P to T.
2. Does this graph contain a Hamiltonian cycle? Explain why or why not.

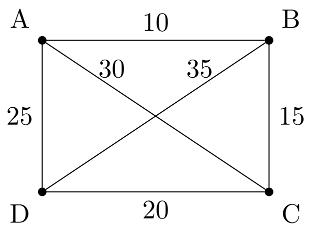
Answer:

1. Path: $P \to Q \to R \to S \to T$. This visits every vertex exactly once.

2. No. Vertex R is a "cut vertex" (or articulation point). To visit all vertices in a cycle, you would need to pass through R to get from the left side to the right side, and pass through R again to get back. This would require visiting R twice, which violates the definition of a Hamiltonian cycle.

F.2 FINDING OPTIMAL TOUR BOUNDS

Ex 51: A delivery driver needs to visit 4 locations (A, B, C, D) starting and ending at the depot A. The distances (in km) are given by the weighted complete graph below.



Use the **Nearest Neighbour Algorithm** starting at vertex A to find an upper bound for the optimal tour.

Answer:

1. Start at A.

2. Nearest unvisited neighbors from A: B(10), D(25), C(30). Choose **B** (10). Path: $A \to B$.

3. From B, nearest unvisited neighbors: C(15), D(35). Choose **C** (15). Path: $A \to B \to C$.

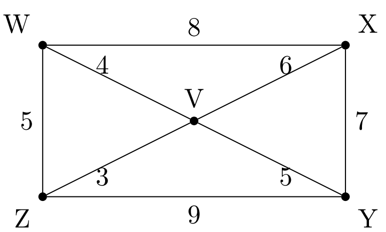
4. From C, nearest unvisited neighbor: D(20). Choose **D** (20). Path: $A \to B \to C \to D$.

5. From D, return to start (A). Distance DA is 25. Path: $A \to B \to C \to D \to A$.

6. Total weight: $10 + 15 + 20 + 25 = 70$ km.

The upper bound is 70 km.

Ex 52: A network administrator wants to find the shortest loop to check 5 servers arranged in a star topology with a ring connection. The graph below shows the latency (ms). The central server is V.



Use the **Deleted Vertex Algorithm** (deleting the central vertex V) to find a lower bound for the optimal tour.

Answer:

1. Delete vertex V and its connecting edges (VW, VX, VY, VZ).

2. Remaining vertices: W, X, Y, Z. Remaining edges (the ring): $WX(8), XY(7), YZ(9), ZW(5)$.

3. Find the MST of the remaining graph (W, X, Y, Z):

   (a) Smallest edge: ZW (5).

   (b) Next smallest: XY (7).

   (c) Next smallest: WX (8). (No cycle formed).

   (d) (YZ is rejected as it closes the cycle).

   (e) MST Weight $= 5 + 7 + 8 = 20$.

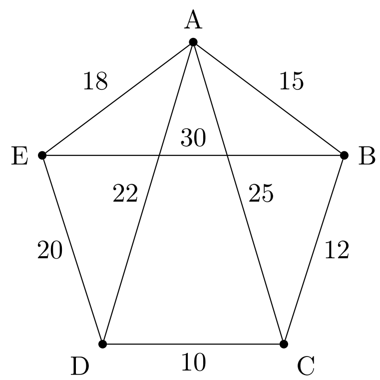
4. Add the two shortest edges connected to the deleted vertex V:

   (a) Edges from V: $VZ(3), VW(4), VY(5), VX(6)$.

   (b) Two shortest: $VZ(3)$ and $VW(4)$. Sum $= 7$.

5. Lower Bound $=$ MST Weight $+$ Two shortest edges $= 20 + 7 = 27$ ms.

Ex 53: A politician is campaigning and needs to visit 5 towns: A, B, C, D, E. The travel times (in minutes) are given in the weighted graph below. The politician starts at town A and must visit every town exactly once before returning to A.



Use the **Nearest Neighbour Algorithm** starting at vertex A to find an upper bound for the travel time.

Answer:

1. Start at vertex **A**.

2. List the edges connecting A to unvisited vertices: $AB(15)$, $AE(18)$, $AD(22)$, $AC(25)$.

3. Select the edge with the minimum weight: AB (15). Move to **B**.

4. From **B**, the unvisited neighbors are $C(12)$ and $E(30)$. Select the nearest: C (12). Move to **C**.

5. From **C**, the only unvisited neighbor directly connected is D (10). Select D. Move to **D**.

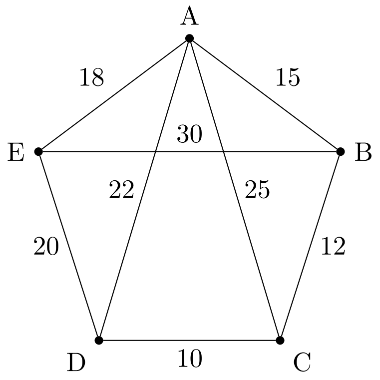
6. From **D**, the only unvisited neighbor is E (20). Select E. Move to **E**.

7. From **E**, return to the start vertex **A**. The edge is EA (18).

8. **Total Travel Time:** $15 + 12 + 10 + 20 + 18 = 75$ minutes.

9. The upper bound for the travel time is **75 minutes**.

Ex 54: A politician is campaigning and needs to visit 5 towns: A, B, C, D, E. The travel times (in minutes) are given in the weighted graph below. The politician wants to estimate the minimum time for a tour.



Use the **Deleted Vertex Algorithm** (deleting vertex A) to find a lower bound for the optimal tour.

Answer:

1. Delete vertex A and its connecting edges (AB, AC, AD, AE).
2. Remaining vertices: B, C, D, E. Remaining edges: $BC(12), CD(10), DE(20), EB(30)$.
3. Find the MST of the remaining graph (B, C, D, E):
   (a) Smallest edge: CD (10).
   (b) Next smallest: BC (12).
   (c) Next smallest: DE (20). (No cycle formed).
   (d) (EB is 30, not needed as all vertices are connected).
   (e) MST Weight $= 10 + 12 + 20 = 42$.
4. Add the two shortest edges connected to the deleted vertex A:
   (a) Edges from A: $AB(15), AE(18), AD(22), AC(25)$.
   (b) Two shortest: $AB(15)$ and $AE(18)$. Sum $= 33$.
5. Lower Bound $=$ MST Weight $+$ Two shortest edges $= 42 + 33 = 75$ minutes.