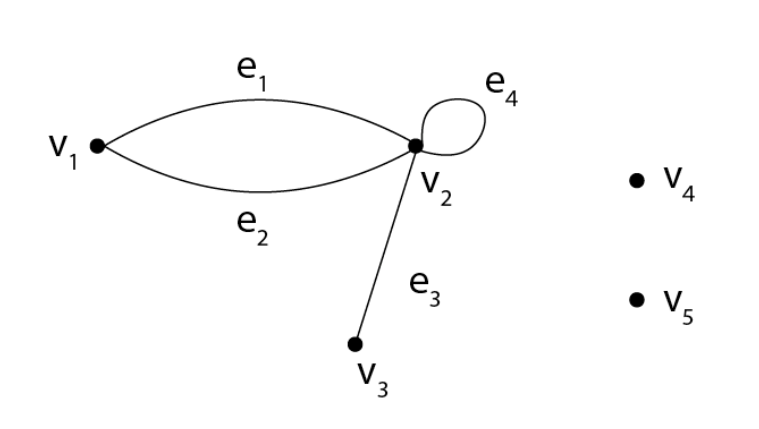
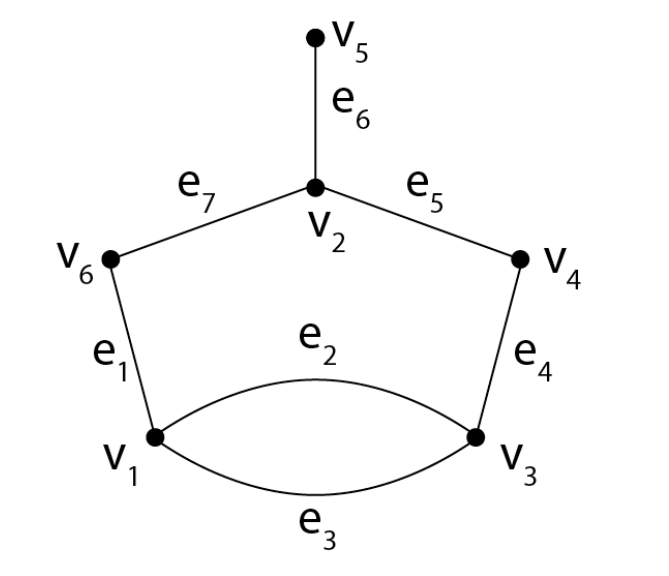
**Exercise Set 10.1**

1. V(G) = {v1, v2, v3, v4}, E(G) = {e1, e2, e3}  
  
Edge-endpoint function:  
  
Edge Endpoints  
e1 {v1, v2}  
e2 {v1, v3}  
e3 {v3}

3.



5.

8.

(i) e1, e2, and e3 are incident on v1.  
(ii) v1, v2, and v3 are adjacent to v3.  
(iii) e2, e8, e9, and e3 are adjacent to e1.  
(iv) Loops are e6 and e7.  
(v) e8 and e9 are parallel; e4 and e5 are parallel.  
(vi) v6 is an isolated vertex.  
(vii) degree of v3 = 5  
(viii) total degree = 20

17: A graph with five vertices with degrees 1, 2, 3, 3 and 5 does not exist because the sum of the degrees of the vertices is odd (14), this violates the handshake theorem1.

18: A graph with four vertices with degrees 1, 2, 3 and 3 also does not exist because the sum of the degrees of the vertices is an odd number (9), this cannot happen in a simple graph2.

**Exercise Set 10.2**

1.

a. Voel Vie 10 V5e9V2e201: This sequence does not follow the standard format for describing paths in a graph. Need more clear information for analysis.

b. V4e7V2e9V5e10V1C3V2e9V5: This sequence appears to describe a path using vertices (V) and edges (e), but includes “C3” which does not conform to standard notation, making the analysis unclear .

c. U2: This sequence does not correspond to any vertex or edge in the provided graph.

e. U2V3V4V5V2V4V3V2: Similarly, this range contains “U” that does not match any vertex labels in the image.

d. U50203040405: This sequence also contains elements (“U” and numbers without the prefix ‘e’ or ‘v’) that do not match any labels in the image.

f. e5e8e10e3: This represents a path passing through edges e5, e8, e10 and e3. If these edges connect properly according to their endpoints, the sequence will be considered a trail if no vertices are repeated and all edges are distinct.

4.

a. Paths from ( v1 ) to ( v4 ): A path is a sequence of edges where each vertex (except for the first and last) is distinct. In the given graph, there is only one path from ( v1 ) to ( v4 ), which is ( v1 ), ( e1 ), ( v2 ), ( e5 ), ( v3 ), ( e6 ), ( v4 ).

b. Trails from ( v1 ) to ( v4 ): A trail is a walk where all edges are distinct. Since the graph has a loop at ( v2 ) (edges ( e2 ) and ( e3 )), there can be multiple trails from ( v1 ) to ( v4 ) by traversing the loop different numbers of times. However, without additional rules about traversing loops, the exact number of trails cannot be determined.

c. Walks from ( v1 ) to ( v4 ): A walk can include repeated vertices and edges. Therefore, there are infinitely many walks from ( v1 ) to ( v4 ) because the loop at ( v2 ) can be traversed any number of times.

8.

Graph a.: Has 2 connected components; a pentagon with vertices from 'a' to 'e', and a triangle with vertices from 'f' to 'h'.

Graph b.: There is only 1 connected component because all vertices are connected to the central vertex 'v'.

Graph c.: Has 1 connected component; All the vertices from ‘a’ to ‘j’ are connected together to form a shape like a person.

Graph d.: Has 1 connected component; The three vertices ( v\_1 ), ( v\_2 ), and ( v\_3 ) are all connected to each other vertically.

12: If all vertices have an even degree, then it has an Euler circuit.

13: If it has more than two vertices with an odd degree, it does not have an Euler circuit.

14: If it has exactly two vertices with an odd degree, it has an Euler path but not an Euler circuit.

15: If it’s disconnected (not all vertices are connected by edges), it does not have an Euler circuit.

16: If it has any loops (edges that connect a vertex to itself), it may still have an Euler circuit if the rest of the graph satisfies the conditions.

17: If it has any multiple edges (more than one edge connecting the same two vertices), it does not affect the existence of an Euler circuit.

19: Determine the degree of each vertex. If there are exactly two vertices with an odd degree, one being ‘u’ and the other ‘w’, then an Euler path exists from ‘u’ to ‘w’.

20: Same as above, check the degrees of vertices. If ‘u’ and ‘w’ are the only vertices with an odd degree, then there is an Euler path from ‘u’ to ‘w’.

21: Again, check the degrees of vertices. If ‘u’ and ‘w’ have odd degrees and no other vertex has an odd degree, an Euler path exists from ‘u’ to ‘w’.

23: To find a Hamiltonian circuit, start at any vertex and visit every other vertex exactly once before returning to the starting vertex. If all vertices are connected in such a way that you can accomplish this, then a Hamiltonian circuit exists.

24: Similarly, start at any vertex and attempt to visit every other vertex exactly once. If you can return to the starting vertex without revisiting any vertex, then a Hamiltonian circuit exists.

**Exercise Set 10.3**

2.

Graph (a): This graph has three vertices ( v\_1, v\_2, ) and ( v\_3 ). The adjacency matrix is a 3x3 matrix where each cell (i, j) represents the presence of an edge from vertex ( v\_i ) to vertex ( v\_j ). If there is an edge, the cell contains a 1; otherwise, it contains a 0.

Graph (b): This graph has four vertices ( v\_1, v\_2, v\_3, ) and ( v\_4 ). The adjacency matrix is a 4x4 matrix with the same rules as above.

3.

Graph (a): This graph has three vertices ( v\_1, v\_2, ) and ( v\_3 ). The adjacency matrix is a 3x3 matrix where each cell (i, j) represents the presence of an edge from vertex ( v\_i ) to vertex ( v\_j ). If there is an edge, the cell contains a 1; otherwise, it contains a 0.

Graph (b): This graph has four vertices ( v\_1, v\_2, v\_3, ) and ( v\_4 ). The adjacency matrix is a 4x4 matrix with the same rules as above.

4.

Graph (a): This graph has five vertices (v1 to v5) and four edges (e1 to e4), arranged in a non-linear layout.

Graph (b): This graph has six vertices (v1 to v6) and six edges (e1 to e6), forming two separate loops.

Graph (c): Refers to ( K\_4 ), the complete graph on four vertices, which means every pair of distinct vertices is connected by a unique edge.

Graph (d): Refers to ( K\_{2,3} ), the complete bipartite graph on two sets of vertices, one with 2 and the other with 3 vertices, with every vertex from one set connected to every vertex from the other set.

5.

Matrix ‘a.’: This is a 3x3 matrix with elements [1 0 1], [0 1 2], [1 2 0]. To find a graph that corresponds to this matrix, you would create a graph with three vertices. An edge would be present between the first and third vertices (since the first and third elements of the first row are 1), between the second and third vertices (since the second element of the second row and the third element of the third row are 2), and so on.

Matrix ‘b.’: This is another 3x3 matrix with elements [0 2 0], [2 1 0], [0 0 1]. Similarly, you would create a graph with three vertices. There would be two edges between the first and second vertices (since the second element of the first row and the first element of the second row are 2), one edge from the second vertex to itself (since the second element of the second row is 1), and one edge from the third vertex to itself (since the third element of the third row is 1).

19.

Matrix Multiplication: The problem involves calculating higher powers of a matrix ( A ), specifically ( A^2 ) and ( A^3 ). This is done by multiplying the matrix by itself.

Graph Theory: The graph ( G ) corresponds to the adjacency matrix ( A ). The number of walks of a certain length between vertices in ( G ) can be found by examining the entries of ( A^2 ) and ( A^3 ).

Walks in Graphs: A walk of length two from ( v\_1 ) to ( v\_3 ) corresponds to the entry in the first row and third column of ( A^2 ). Similarly, a walk of length three from ( v\_1 ) to ( v\_3 ) corresponds to the same entry in ( A^3 ).

20.

Matrix Multiplication: The problem involves calculating higher powers of a matrix ( A ), specifically ( A^2 ) and ( A^3 ). This is done by multiplying the matrix by itself.

Graph Theory: The graph ( G ) corresponds to the adjacency matrix ( A ). The number of walks of a certain length between vertices in ( G ) can be found by examining the entries of ( A^2 ) and ( A^3 ).

Walks in Graphs: A walk of length two from ( v\_1 ) to ( v\_3 ) corresponds to the entry in the first row and third column of ( A^2 ). Similarly, a walk of length three from ( v\_1 ) to ( v\_3 ) corresponds to the same entry in ( A^3 ).

**Exercise Set 10.4**

1, 2.

Graph Isomorphism: Two graphs ( G ) and ( G’ ) are isomorphic, denoted as ( G \cong G’ ), if there is a one-to-one correspondence between their vertices and edges that preserves the adjacency and incidence relations.

Finding Isomorphism: To determine if ( G \cong G’ ), you need to find a function ( v: V(G) \rightarrow V(G’) ) for vertices and ( e: E(G) \rightarrow E(G’) ) for edges that define the isomorphism. This means each vertex and edge in ( G ) must match exactly one vertex and edge in ( G’ ), respectively.

Invariant for Graph Isomorphism: If two graphs do not share an isomorphism, they will have different graph invariants, such as the number of vertices, the number of edges, degrees of vertices, or other properties that are preserved under isomorphism.

6, 7.

Graph Isomorphism: Two graphs ( G ) and ( G’ ) are isomorphic, denoted as ( G \cong G’ ), if there is a one-to-one correspondence between their vertices and edges that preserves the adjacency and incidence relations.

Finding Isomorphism: To determine if ( G \cong G’ ), you need to find a function ( v: V(G) \rightarrow V(G’) ) for vertices and ( e: E(G) \rightarrow E(G’) ) for edges that define the isomorphism. This means each vertex and edge in ( G ) must match exactly one vertex and edge in ( G’ ), respectively.

Invariant for Graph Isomorphism: If two graphs do not share an isomorphism, they will have different graph invariants, such as the number of vertices, the number of edges, degrees of vertices, or other properties that are preserved under isomorphism.