ESSENTIALS OF PROBLEM SOLVING(ACSD05)

Don’t write or place any image in this area

SYLLABUS

MODULE - I GRAPH THEORY

Graph terminology, digraphs, weighted graphs, complete graphs, graph complements, bipartite graphs, graph combinations, isomorphisms, matrix representations of graphs, incidence and adjacency matrices, degree sequence.

MODULE - II GRAPH ROUTES

Eulerian circuit: Konigsberg bridge problem, touring a graph; Eulerian graphs, Hamiltonian cycles, the traveling salesman problem; Shortest paths: Dijkstra’s algorithm, walks using matrices.

MODULE - III GRAPH COLORING AND GRAPH ALGORITHMS

Four color theorem, vertex coloring, edge coloring, coloring variations, first-fit coloring algorithm.

Graph traversal: depth-first search, bread-first search and its applications; Minimum spanning trees: Kruskal’s and Prim’s algorithm, union-find structure.

SYLLABUS

MODULE - IV: ALGEBRAIC AND TRANSCENDENTAL EQUATIONS

Algebraic equations, method of false position, bisection method, iteration method, Newton-Raphson method, Secant method, Ramanujan's Method, Muller's method (Approximation up to 2 decimals only).

MODULE - V: NUMERICAL INTEGRATION AND ORDINARY DIFFERENTIATIAL EQUATIONS

Trapezoidal rule, Simpson's 1/3 rule, Simpson's 3/8 rule, Solution by Taylor's series, Euler's method of solving an ordinary differential equation numerically, Runge-Kutta's second order method of solving ordinary differential equations (Approximation up to 2 decimals only).

MODULE - I GRAPH THEORY

Graph terminology, digraphs, weighted graphs, complete graphs, graph complements, bipartite graphs, graph combinations, isomorphisms, matrix representations of graphs, incidence and adjacency matrices, degree sequence.

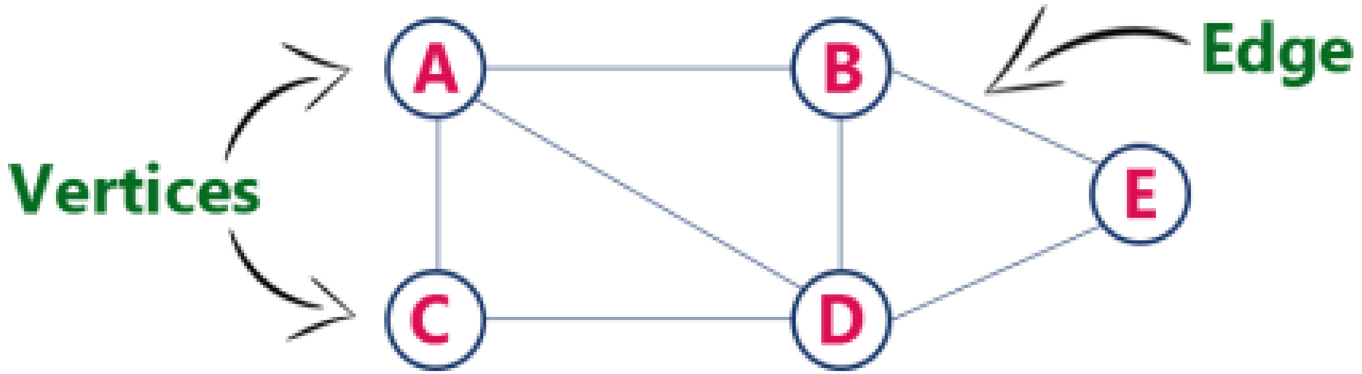
Graph terminology

Definition: **A graph** is collection of points called **vertices** & collection of lines called **edges** each of which joins either a pair of points or single points to itself.

- Mathematically graph G is an ordered pair of (V, E)
- V is the vertex-set whose elements are called the vertices, or nodes of the graph. This set is often denoted by V(G) or just V.
- E is the edge-set whose elements are called the edges, or connections between vertices of the graph. This set is often denoted by E(G) or just E.Each edge e_{ij} is associated with an ordered pair of vertices (V_i,V_j).

Graph terminology

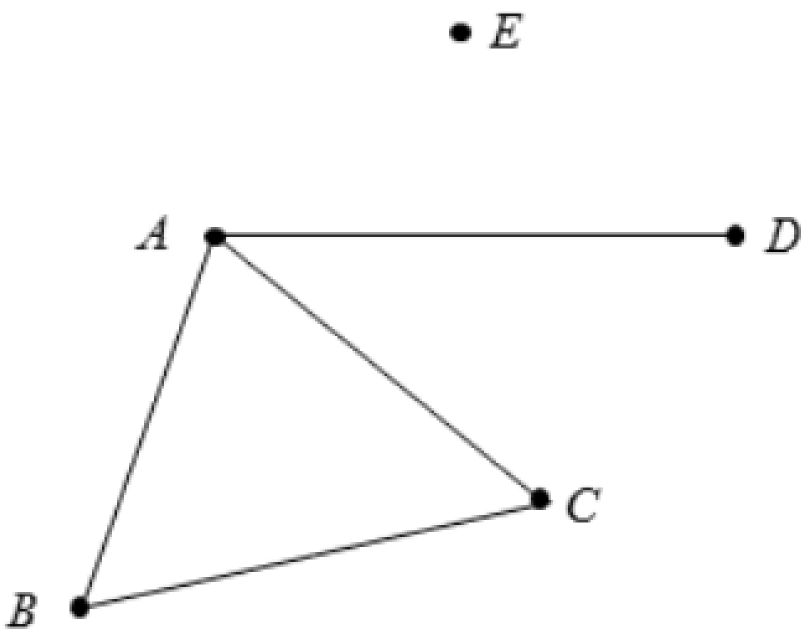
Example:



graph G can be defined as G = (V , E)
Where V = {A,B,C,D,E} and
E = {(A,B),(A,C)(A,D),(B,D),(C,D),(B,E),(E,D)}.
This is a graph with 5 vertices and 6 edges.

Graph terminology

Example:



In the above graph,

V = {A, B, C, D, E}

E = {AB, BC, CA, AD}

Graph terminology

Graph theory is the study of the relationship between edges and vertices.

Applications of Graph Theory:

1. Computer Science

- In computer science graph theory is used for the study of algorithms.
- Graphs are used to define the flow of computation.
- Graphs are used to represent networks of communication.
- Graphs are used to represent data organization.
- Graph theory is used to find shortest path in road or a network.
- In Google Maps, various locations are represented as vertices or nodes and the roads are represented as edges and graph theory is used to find the shortest path between two nodes.

Graph terminology

2. Electrical Engineering

In Electrical Engineering, graph theory is used in designing of circuit connections. These circuit connections are named as topologies. Some topologies are series, bridge, star and parallel topologies.

3. Physics and Chemistry

- In physics and chemistry, graph theory is used to **study molecules**.
- The **3D structure of complicated simulated atomic structures** can be studied quantitatively by gathering statistics on graph-theoretic properties related to the topology of the atoms.
- Graph is also helpful in **constructing the molecular structure** as well as lattice of the molecule. It also helps us to show the bond relation in between atoms and molecules, also help in comparing structure of one molecule to other.

4. Computer Network

- In computer network, the relationships among interconnected computers within the network, follow the principles of graph theory.
- Graph theory is also used in network security.
- We can use the vertex coloring algorithm to find a proper coloring of the map with four colors.
- Vertex coloring algorithm may be used for assigning at most four different frequencies for any GSM (Grouped Special Mobile) mobile phone networks.

Graph terminology

5. Social Sciences

- Graph theory is also used in sociology. For example, to explore rumor spreading, or to measure actors' prestige notably through the use of social network analysis software.
- Friendship graphs describe whether people know each other or not.
- In influence graphs model, certain people can influence the behavior of others.
- In collaboration graphs model to check whether two people work together in a particular way, such as acting in a movie together.

Graph terminology

6. Biology

- Nodes in biological networks represent bimolecular such as genes, proteins or metabolites, and edges connecting these nodes indicate functional, physical or chemical interactions between the corresponding bimolecular.
- Graph theory is used in transcriptional regulation networks.
- It is also used in Metabolic networks.
- In PPI (Protein - Protein interaction) networks graph theory is also useful.
- Characterizing drug - drug target relationships.

Graph terminology

7. Mathematics

In mathematics, operational research is the important field. Graph theory provides many useful applications in operational research.

Graph terminology

Types of edges

1.Undirected Edge - An undirected edge is a bidirectional edge. If there is an undirected edge between vertices u and v then edge (u , v) is equal to edge (u , v).

2.Directed Edge - A directed edge is a unidirectional edge. If there is a directed edge between vertices u and v then edge (u ,v) is not equal to edge (v ,u).

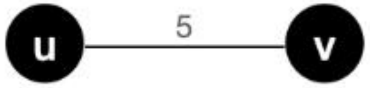
3.Weighted Edge - A weighted edge is an edge with cost on it.

4. Self-Loop Edge - An edge of a graph that starts and ends at the same vertex.

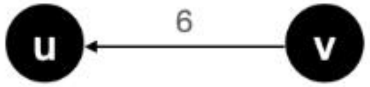
Graph terminology

Undirected edge - edge have no direction.
The edge (u, v) is identical to edge (v, u)

Directed edge - edge have direction
The edge (u, v) is the edge from node u to node v



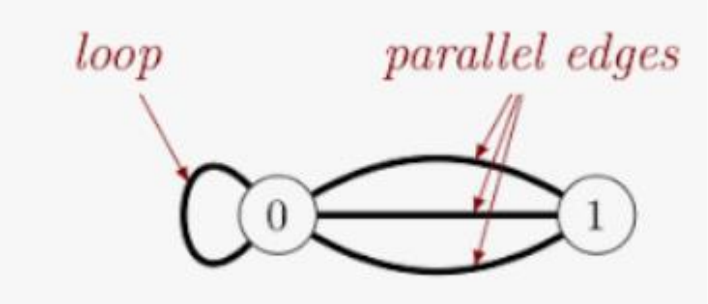
Weighted edge
Edge contains a certain weight to represent cost, distance, quantity etc

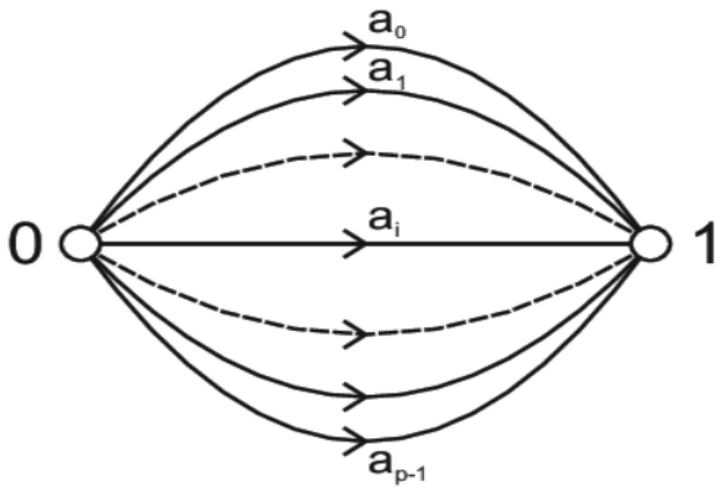


Directed weighted edge
Edge (v, u) has direction as well as weight

5. Parallel Edge

In an undirected graph, two or more edges that are incident to the same two vertices, or in a directed graph, two or more edges with both the same tail vertex and the same head vertex





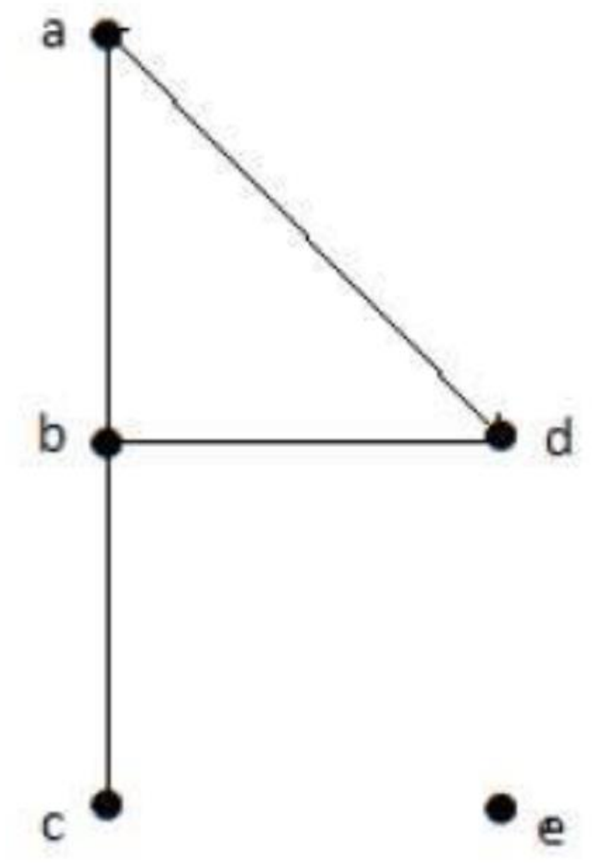
6. Outgoing Edge: A directed edge is said to be outgoing edge on its orign vertex.

7.Incoming Edge: A directed edge is said to be incoming edge on its destination vertex.

Graph terminology

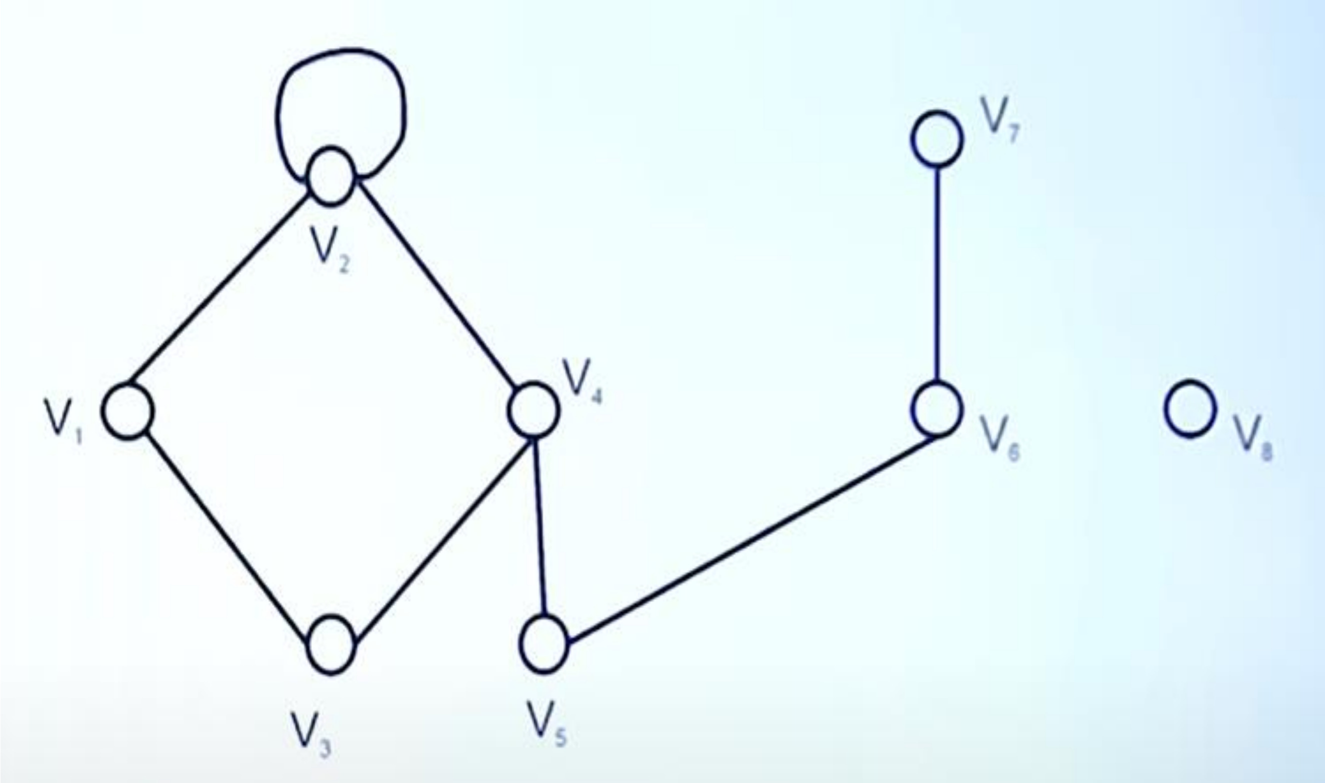
Degree of Vertex of an Undirected Graph

In an undirected graph, total number of edges connected to a vertex is said to be degree of that vertex with self loop counted twice. Eg:



deg(a) = 2, as there are 2 edges meeting at vertex 'a'.
deg(b) = 3, as there are 3 edges meeting at vertex 'b'.
deg(c) = 1, as there is 1 edge formed at vertex 'c'
So 'c' is a pendent vertex.
deg(d) = 2, as there are 2 edges meeting at vertex 'd'.
deg(e) = 0, as there are 0 edges formed at vertex 'e'.
So 'e' is an isolated vertex.

Graph terminology



Here $d(V_1) = 2$, $d(V_2) = 4$, $d(V_3) = 2$, $d(V_4) = 3$

$d(V_5) = 2$, $d(V_6) = 2$, $d(V_7) = 1$, $d(V_8) = 0$

Degree of Vertex in a Directed Graph

In a directed graph, each vertex has an indegree and an outdegree. The degree of a vertex is equal to the sum of the in-degree and out-degree of a vertex, i.e., $deg(V)=deg^{-}(V)+deg^{+}(V)$

Indegree of a Graph

Indegree of vertex V is the number of edges which are coming into the vertex V.

Notation −: $deg^{-}(V)$.

Outdegree of a Graph

Outdegree of vertex V is the number of edges which are going out from the vertex V.

Notation −: $deg^{+}(V)$.

Graph terminology

Theorem :

Let G = (V, E) be a graph with directed edges.

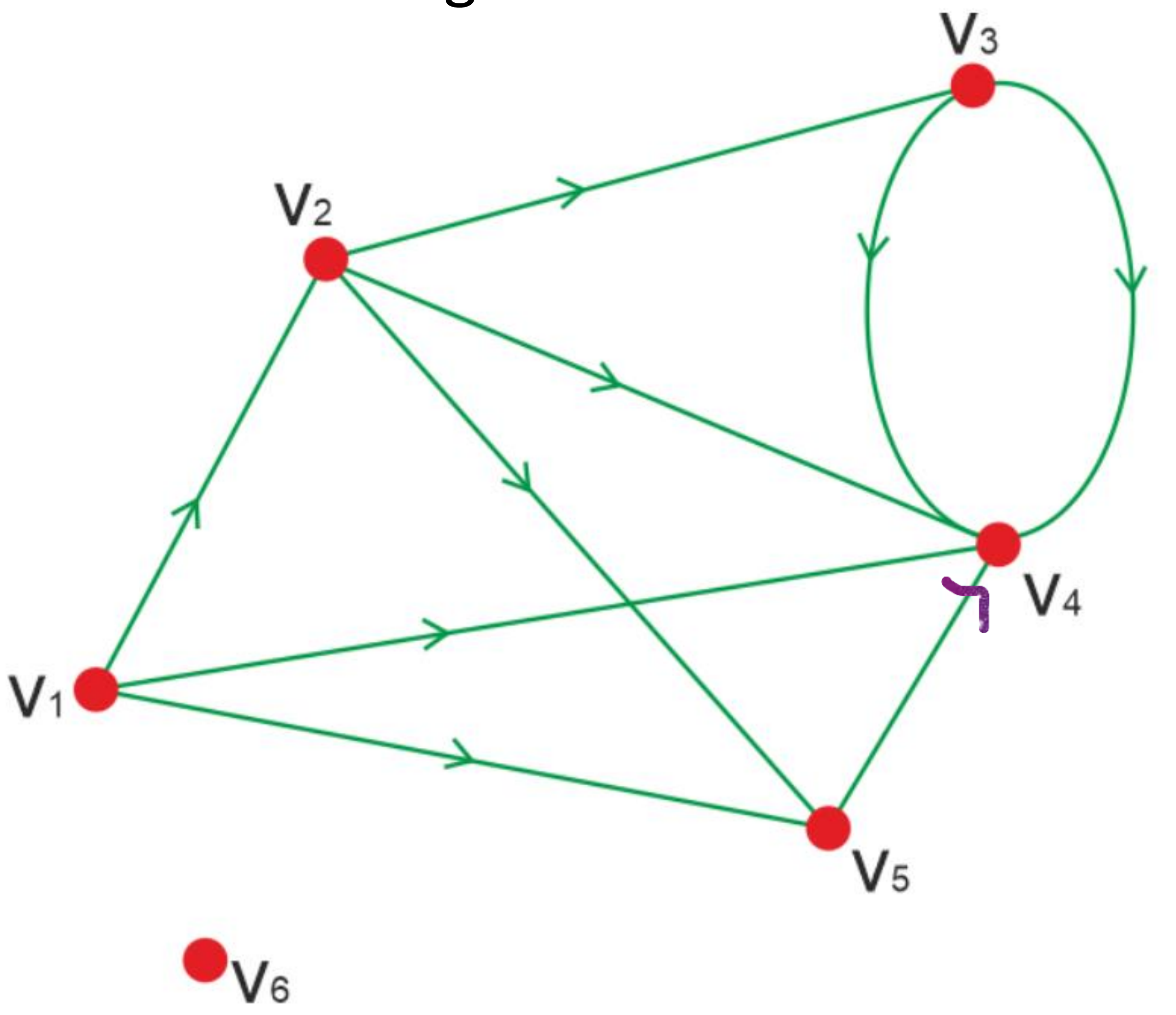
Then:

$$|E| = \sum_{v \in V} deg^-(v) = \sum_{v \in V} deg^+(v)$$

Proof: The first sum counts the number of outgoing edges over all vertices an d the second sum counts the number of incoming edges over all vertices. It f ollows that both sums equal the number of edges in the graph.

Graph terminology

Example 1: Determine the degree of each vertex.



Graph terminology

As we can see that the above graph contains the total 6 vertex, i.e., v1, v2, v3, v4, v5 and v6.

In-degree:

In-degree of a vertex v1 = deg(v1) = 0

In-degree of a vertex v2 = deg(v2) = 1

In-degree of a vertex v3 = deg(v3) = 1

In-degree of a vertex v4 = deg(v4) = 5

In-degree of a vertex v5 = deg(v5) = 2

In-degree of a vertex v6 = deg(v6) = 0

Graph terminology

Out-degree:

Out-degree of a vertex v1 = deg(v1) = 3

Out-degree of a vertex v2 = deg(v2) = 3

Out-degree of a vertex v3 = deg(v3) = 2

Out-degree of a vertex v4 = deg(v4) = 0

Out-degree of a vertex v5 = deg(v5) = 1

Out-degree of a vertex v6 = deg(v6) = 0

Degree of a vertex

With the help of the definition described above, we know that the degree of a vertex $Deg(v) = deg^{-}(v) + deg^{+}(v)$. Now we will calculate it with the help of this formula like this:

Degree of a vertex v1 = deg(v1) = 0+3 = 2

Degree of a vertex v2 = deg(v2) = 1+3 = 4

Degree of a vertex v3 = deg(v3) = 1+2 = 3

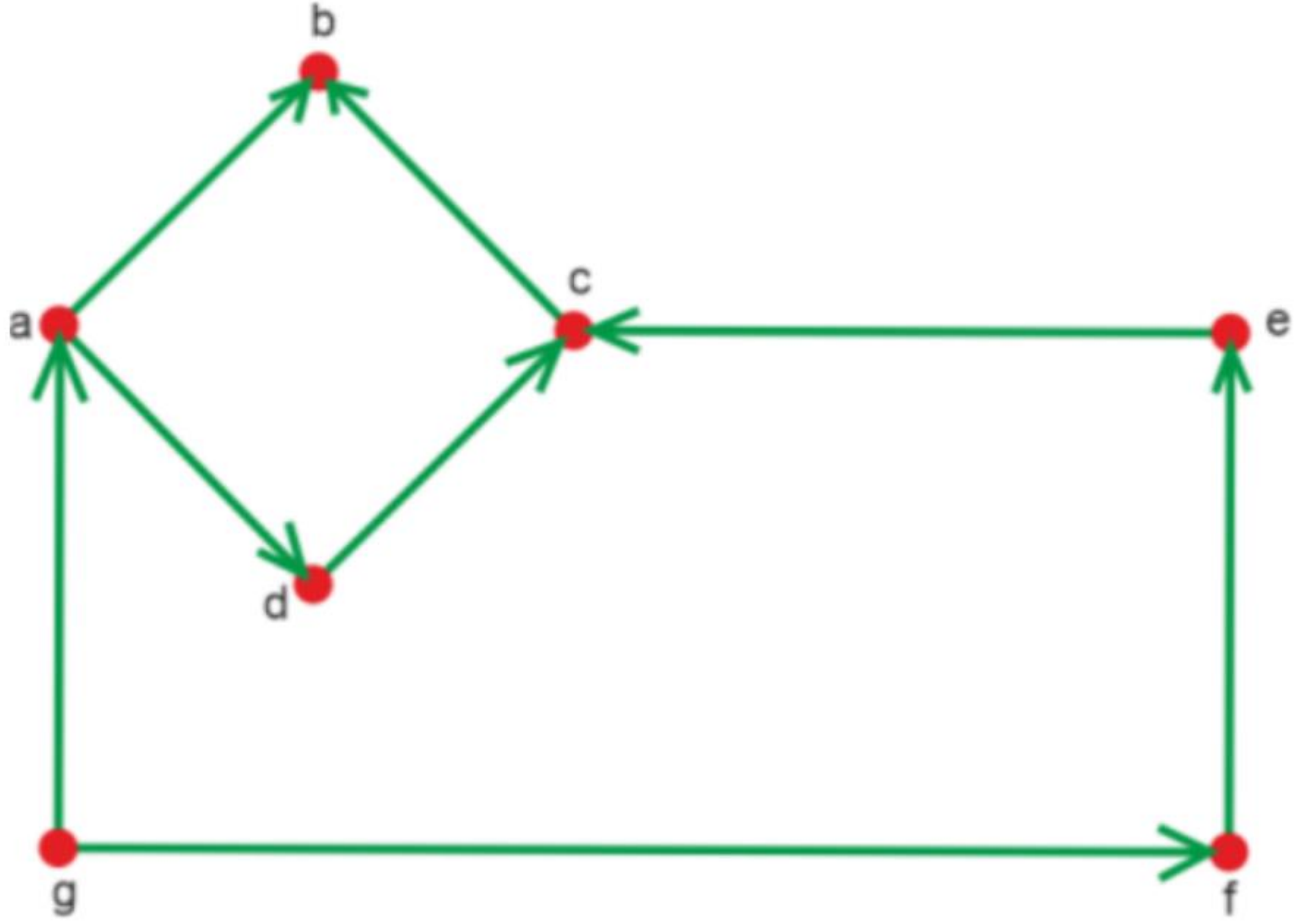
Degree of a vertex v4 = deg(v4) = 5+0 = 5

Degree of a vertex v5 = deg(v5) = 2+1 = 3

Degree of a vertex v6 = deg(v6) = 0+0 = 0

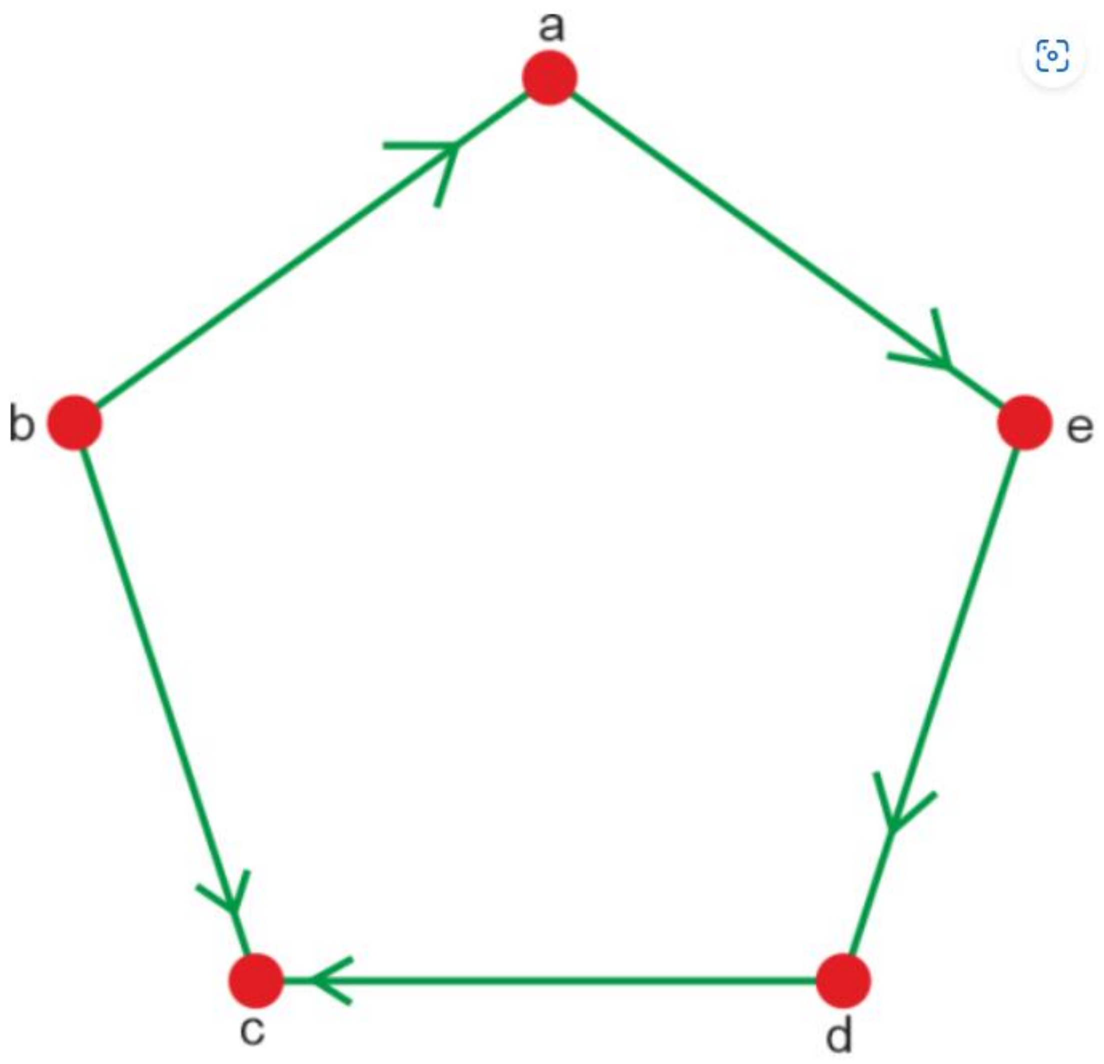
Graph terminology

Example 2:

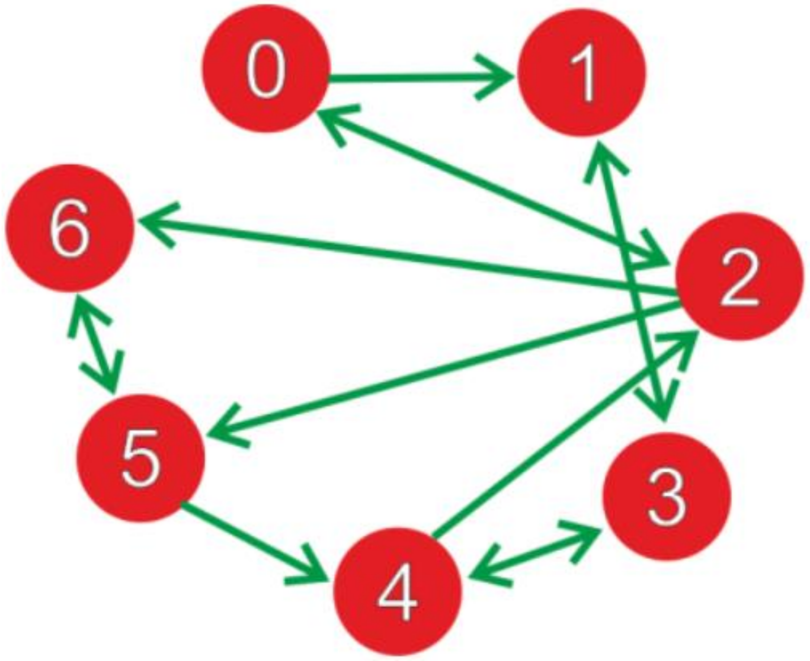


Graph terminology

Example 3

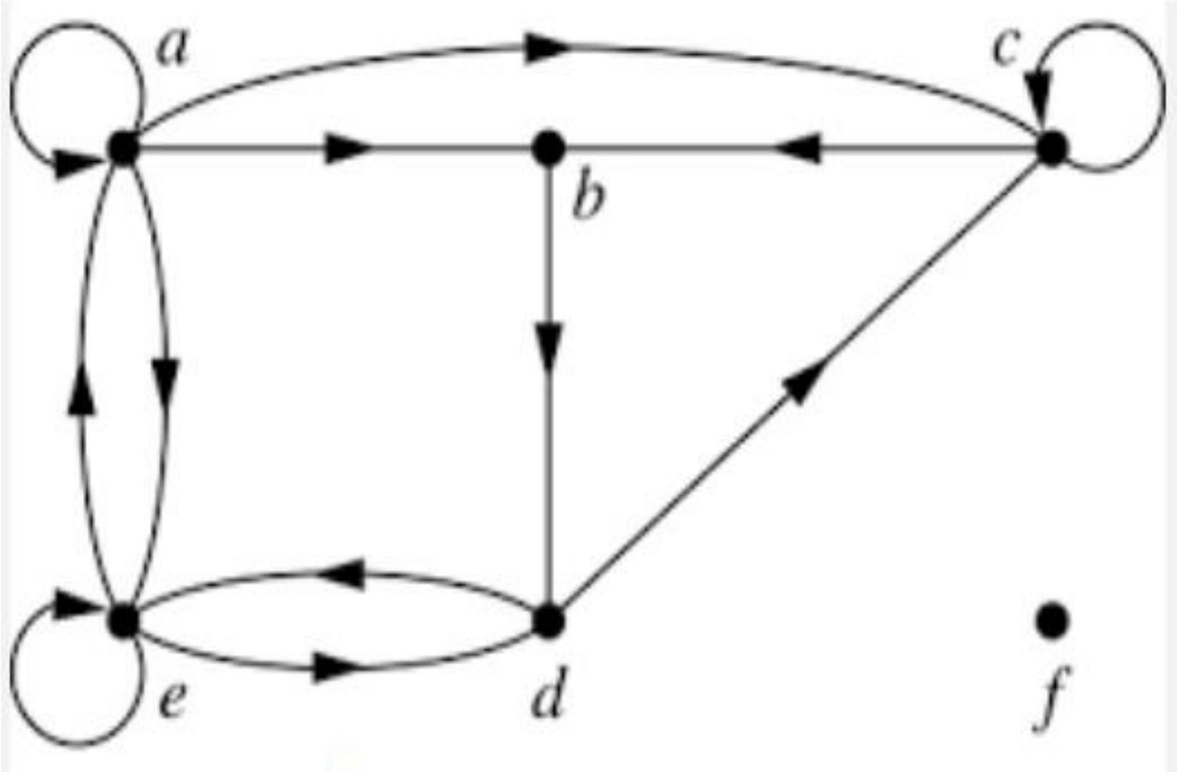


Example 4



Graph terminology

Example 5



$\deg^-(a) = 2, \deg^-(b) = 2, \deg^-(c) = 3, \deg^-(d) = 2,$
$\deg^-(e) = 3, \deg^-(f) = 0.$

$\deg^+(a) = 4, \deg^+(b) = 1, \deg^+(c) = 2, \deg^+(d) = 2,$
$\deg^+(e) = 3, \deg^+(f) = 0.$

Graph terminology

Types of vertices

vertex: a point or node in a graph.

Isolated Vertex: A vertex with degree zero is called an isolated vertex.eg:

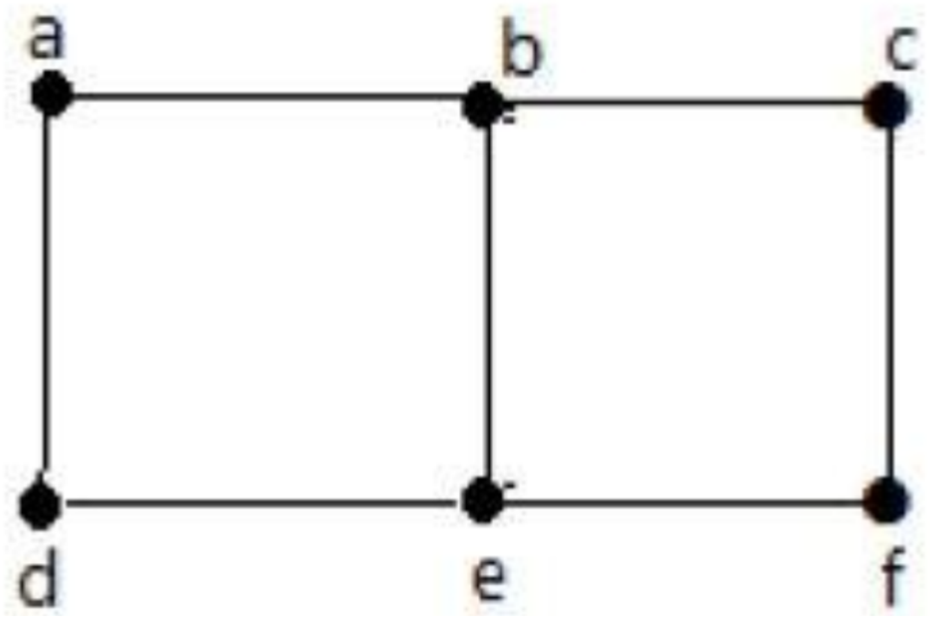
Pendent Vertex: A vertex with degree one is called a pendent vertex.

Graph terminology

adjacent vertices: Two distinct vertices are said be adjacent if there is a common edge connecting them. Or When there is an edge from one node to another then these nodes are called adjacent nodes.

adjacent edges: Two edges are said to be adjacent, if there is a common vertex between the two edges. Here, the adjacency of edges is maintained by the single vertex that is connecting two edges.

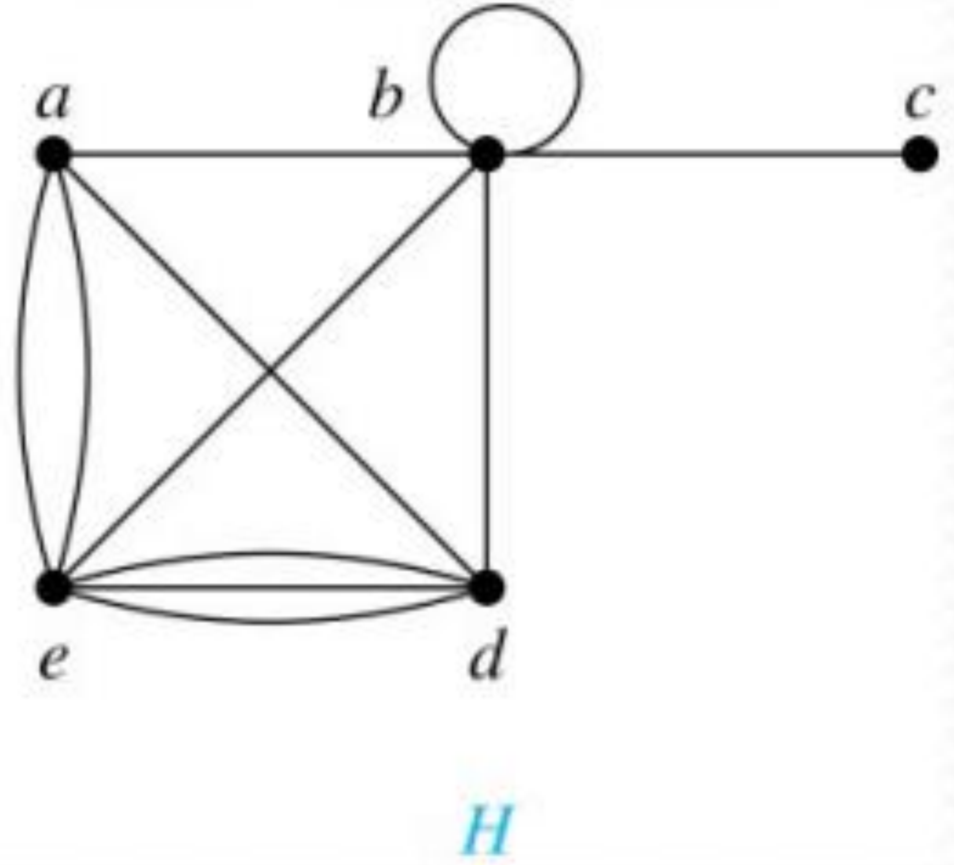
Graph terminology



- 'a' and 'b' are the adjacent vertices, as there is a common edge 'ab' between them.
- 'a' and 'd' are the adjacent vertices, as there is a common edge 'ad' between them.
- ab' and 'be' are the adjacent edges, as there is a common vertex 'b' between them.
- be' and 'de' are the adjacent edges, as there is a common vertex 'e' between them.

Graph terminology

What are the degrees and neighborhoods of the vertices in the graph H?



Graph terminology

Solution: H:

deg(a) = 4, deg(b) = deg(e) = 6, deg(c) = 1, deg(d) = 5.

N(a) = {b, d, e}, N(b) = {a, b, c, d, e}, N(c) = {b},

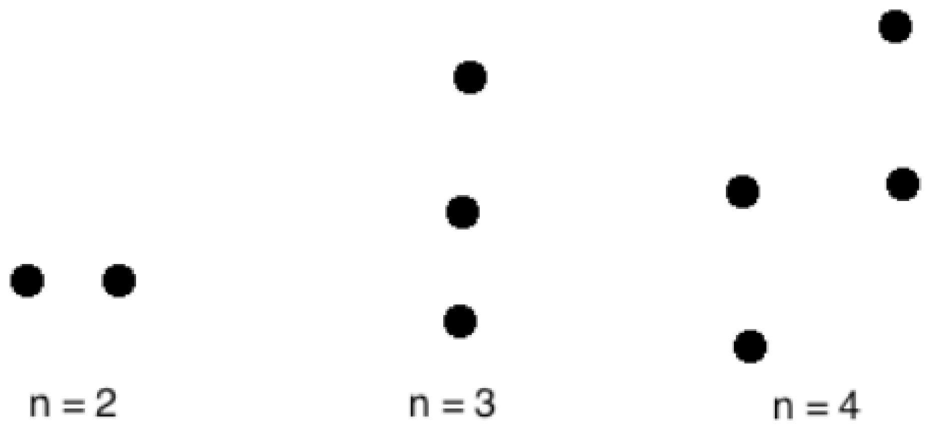
N(d) = {a, b, e}, N(e) = {a, b ,d}.

Types of Graphs

1. Null Graph

A null graph is a graph in which there are no edges between its vertices. A null graph is also called empty graph.

Example



Graph terminology

2. Trivial Graph

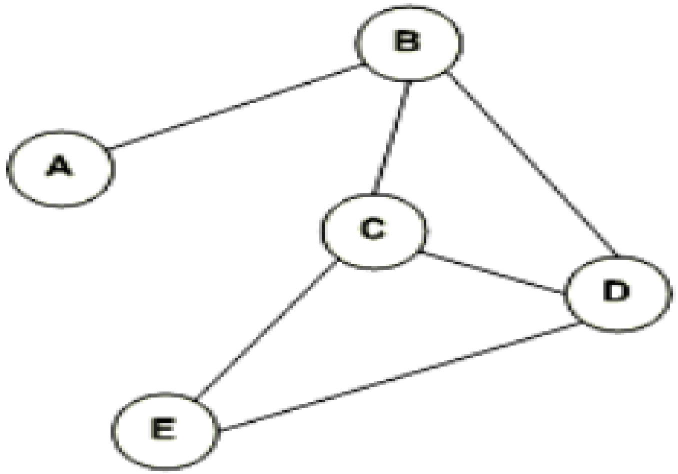
A trivial graph is the graph which has only one vertex.

Example

3. Undirected Graph

An undirected graph is a graph whose edges are not directed.

Example



4.Directed Graph

Definition : A directed graph, or digraph, is a graph G = (V, A) that consists of a vertex set V (G) and an arc set A(G). An arc is an ordered pair of vertices.

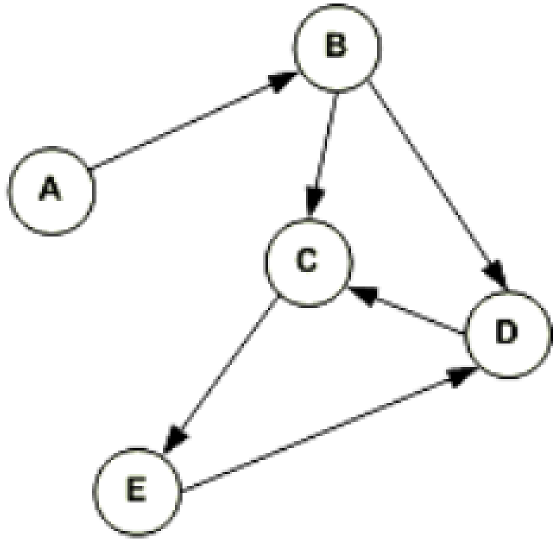
A directed graph is a graph in which the edges are directed by arrows.

- Given an arc xy, the head is the starting vertex x and

the tail is the ending vertex y.

- A is the head of arc AB and the tail is B

Example

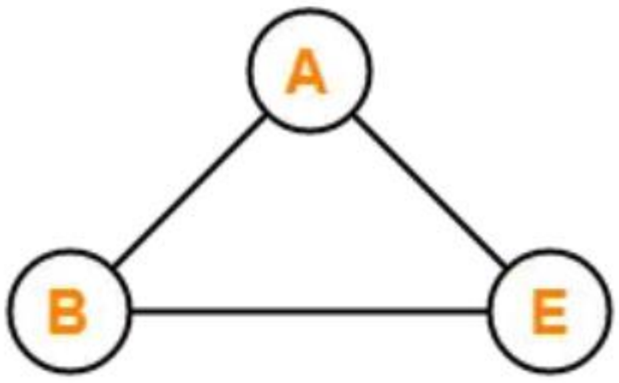


5. Simple Graph

A simple graph is the undirected graph with no parallel edges and no loops.
A simple graph which has n vertices, the degree of every vertex is at most n -1.

Here,

- This graph consists of three vertices and three edges.
- There are neither self loops nor parallel edges.
- Therefore, it is a simple graph.



Example of Simple Graph

6. Complete Graph

A simple graph G is complete if every pair of distinct vertices is adjacent. The complete graph on n vertices is denoted Kn. (Or) A graph in which any V node is adjacent to all other nodes present in the graph is known as a complete graph. A complete graph with n vertices is represented by Kn.

For complete graphs,

Each vertex is connected with all the remaining vertices through exactly one edge.

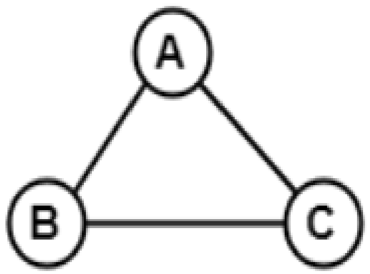
Properties of Kn

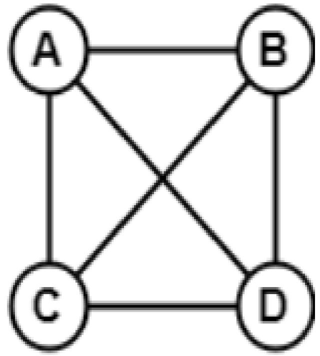
(1) Each vertex in Kn has degree n − 1.

(2) Kn has n(n − 1)/ 2 edges.

(3) Kn contains the most edges out of all simple graphs on n vertices.

Example

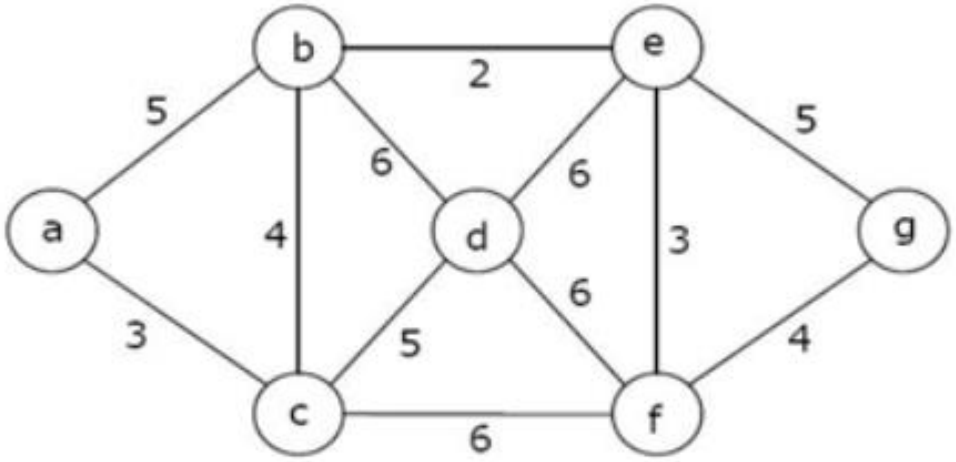




7. Weighted Graph

A weighted graph is a graph whose edges have been labeled with some weights or numbers. The length of a path in a weighted graph is the sum of the weights of all the edges in the path.

Example



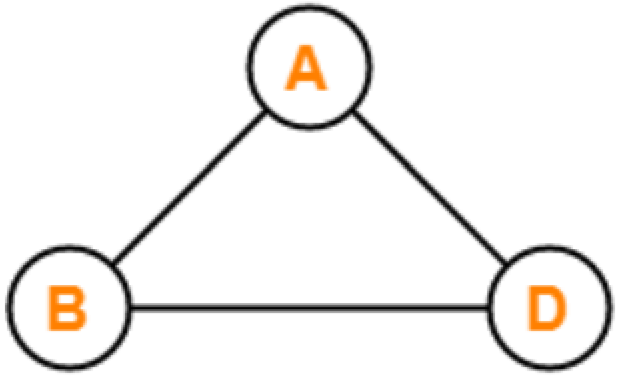
In the above graph, if path is a -> b -> c -> d -> e -> g then the length of the path is 5 + 4 + 5 + 6 + 5 = 25.

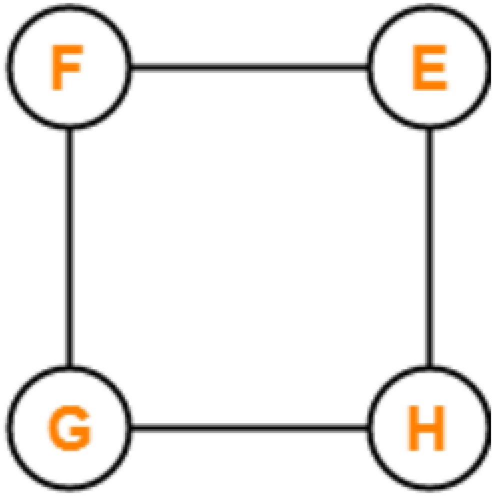
8. Regular Graph

A graph in which degree of all the vertices is same is called as a regular graph. If all the vertices in a graph are of degree ‘k’, then it is called as a “k-regular graph“

Examples

- In these graphs, All the vertices have degree-2. Therefore, they are 2-Regular graphs.



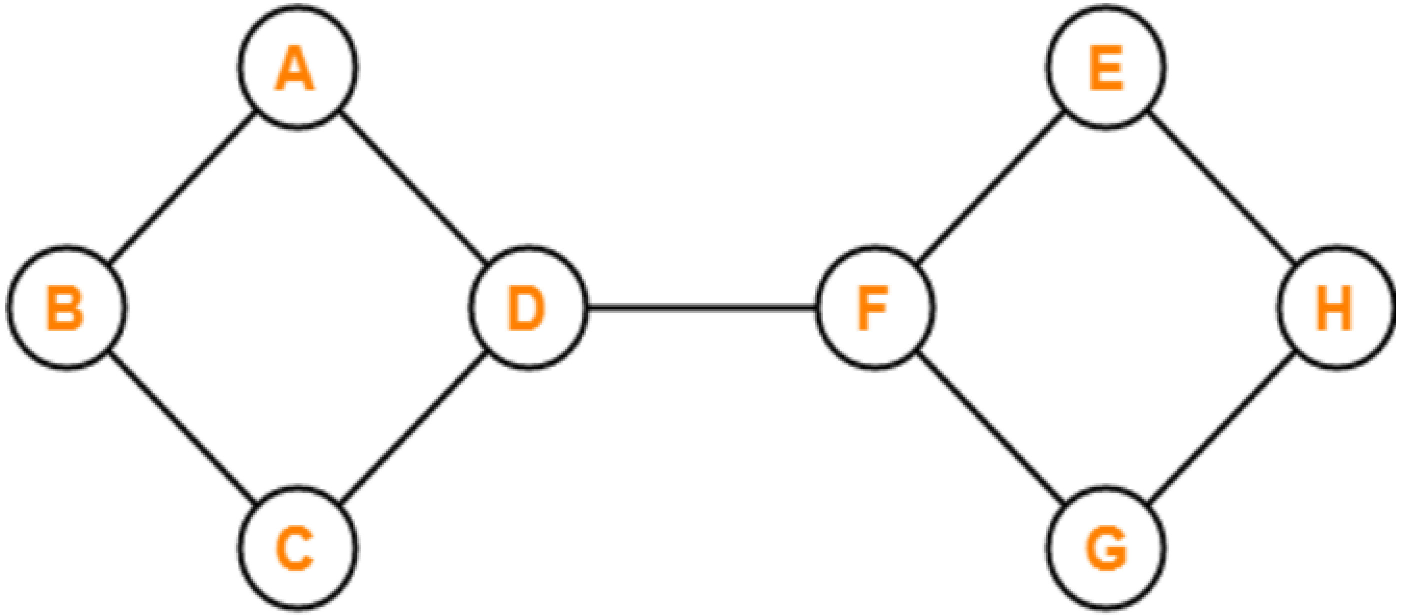


9. Connected Graph

A graph in which we can visit from any one vertex to any other vertex is called as a connected graph. In connected graph, at least one path exists between every pair of vertices.

Here,

- In this graph, we can visit from any one vertex to any other vertex.
- There exists at least one path between every pair of vertices.
- Therefore, it is a connected graph.



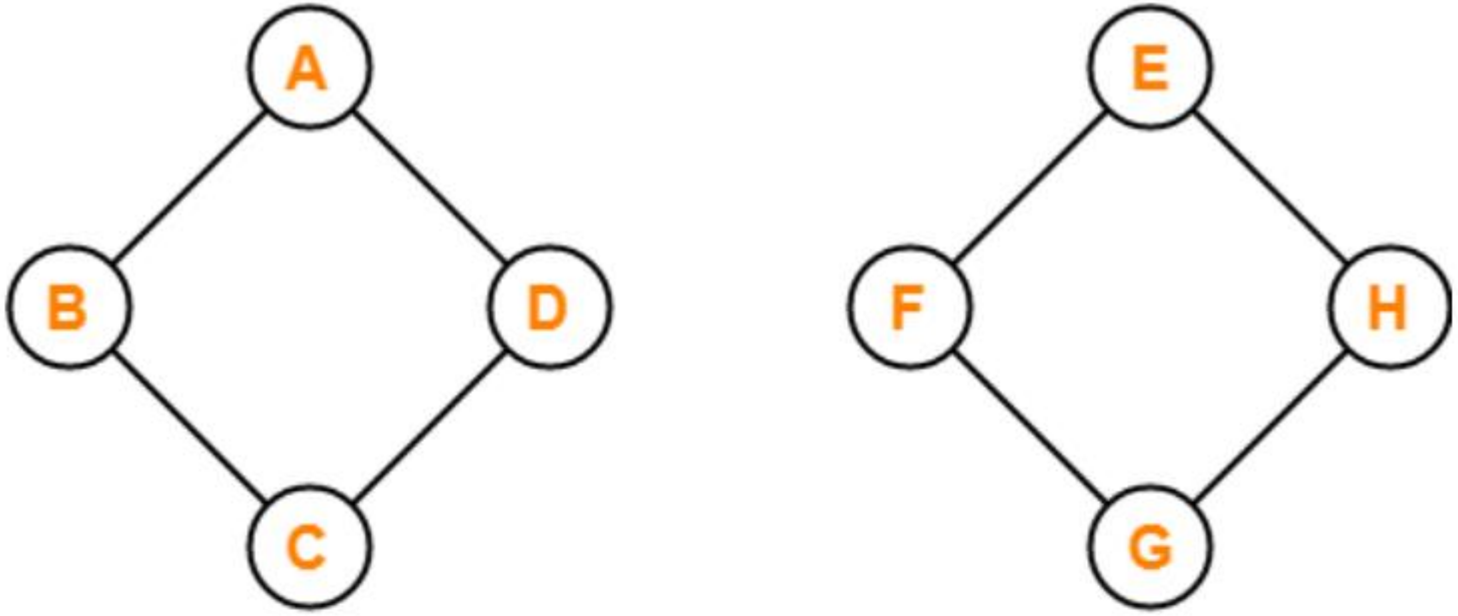
Example of Connected Graph

10. Disconnected Graph

A graph in which there does not exist any path between at least one pair of vertices is called as a disconnected graph.

Here,

- This graph consists of two independent components which are disconnected.
- It is not possible to visit from the vertices of one component to the vertices of other component.
- Therefore, it is a disconnected graph.

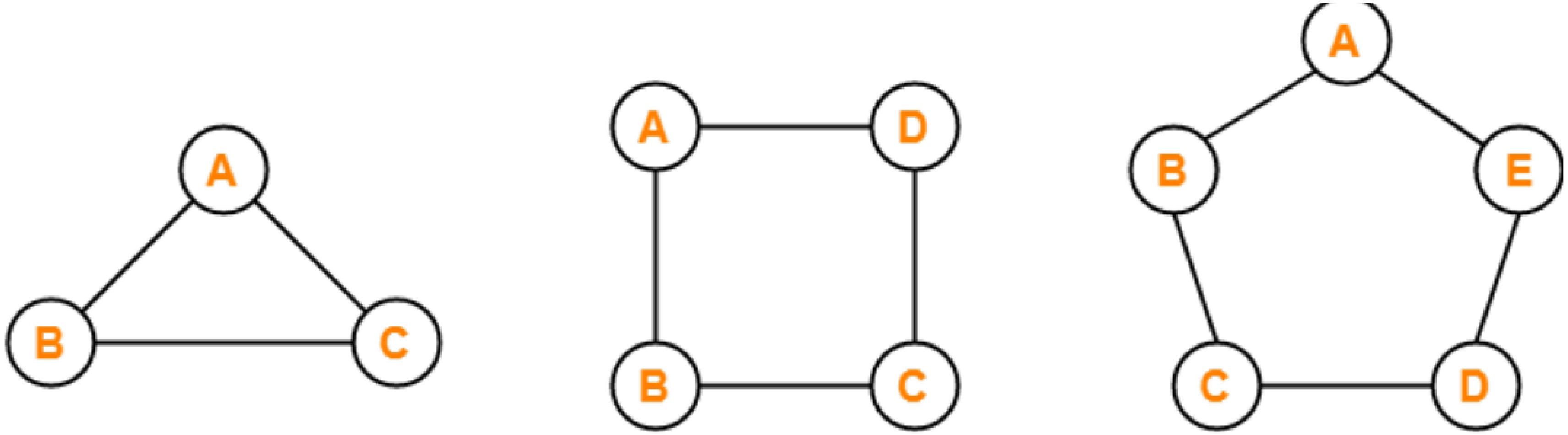


Example of Disconnected Graph

11. Cycle Graph

A simple graph of ‘n’ vertices (n>=3) and n edges forming a cycle of length ‘n’ is called as a cycle graph. In a cycle graph, all the vertices are of degree 2.

- In these graphs, Each vertex is having degree 2.
- Therefore, they are cycle graphs.



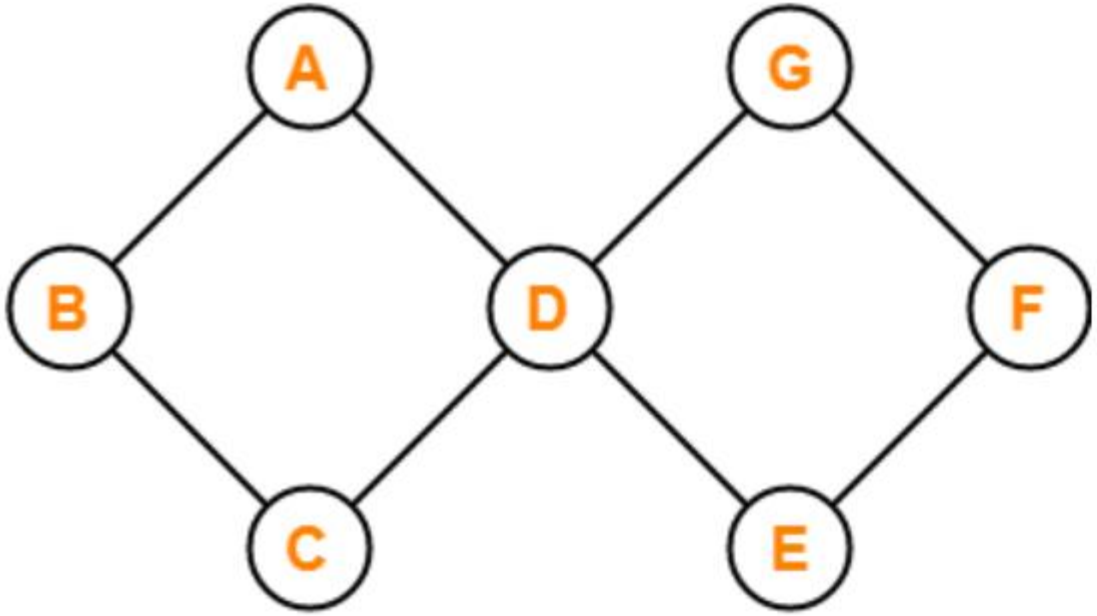
Examples of Cycle Graph

12. Cyclic Graph

A graph containing at least one cycle in it is called as a cyclic graph.

Here,

- This graph contains two cycles in it.
- Therefore, it is a cyclic graph.



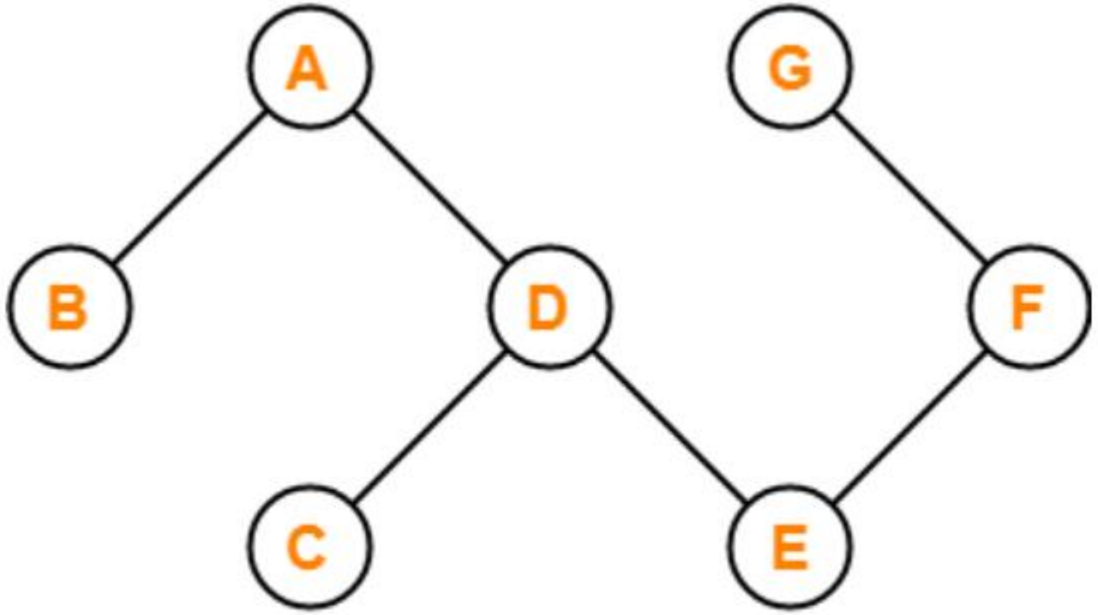
Example of Cyclic Graph

13. Acyclic Graph

A graph not containing any cycle in it is called as an acyclic graph

Here,

- This graph do not contain any cycle in it.
- Therefore, it is an acyclic graph.



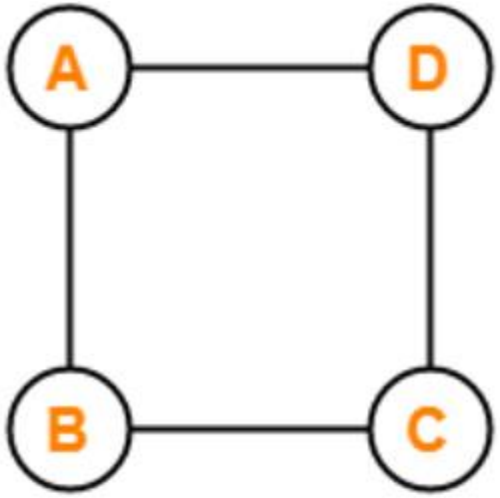
Example of Acyclic Graph

14. Finite Graph

A graph consisting of finite number of vertices and edges is called as a finite graph.

Here,

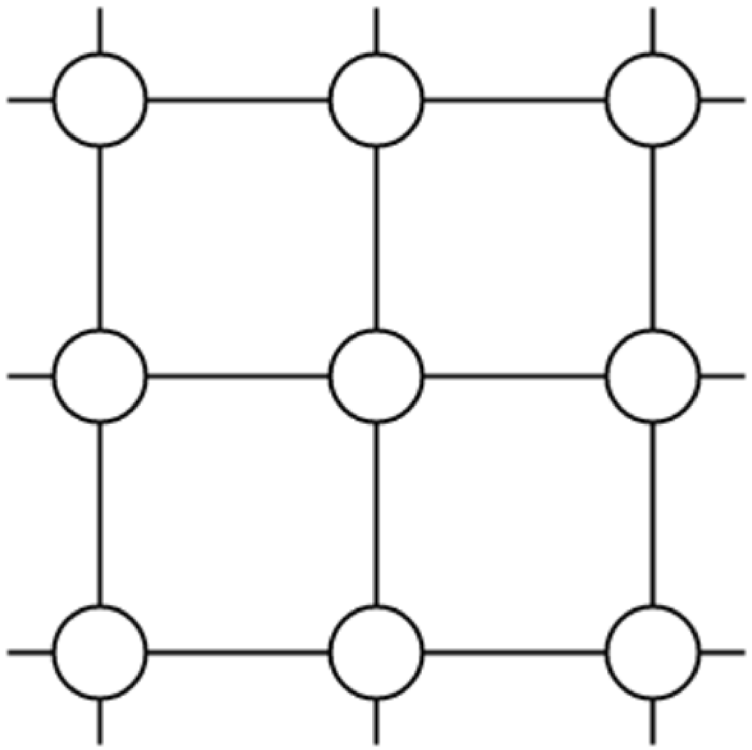
- This graph consists of finite number of vertices and edges.
- Therefore, it is a finite graph.



Example of Finite Graph

15. Infinite Graph

A graph consisting of infinite number of vertices and edges is called as an infinite graph.



Example of Infinite Graph

Here,

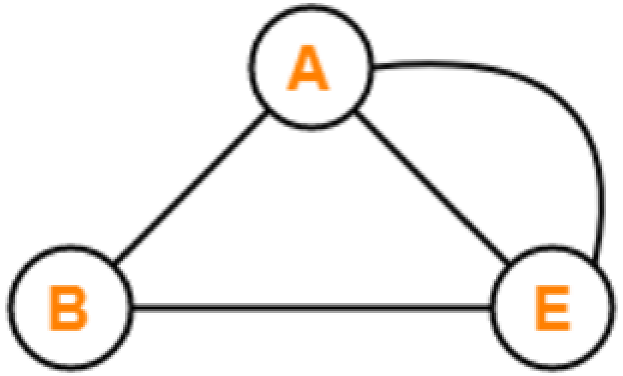
- This graph consists of infinite number of vertices and edges.
- Therefore, it is an infinite graph.

16. Multi Graph

A graph having no self loops but having parallel edge(s) in it is called as a multi graph.

Here,

- This graph consists of three vertices and four edges out of which one edge is a parallel edge.
- There are no self loops but a parallel edge is present.
- Therefore, it is a multi graph



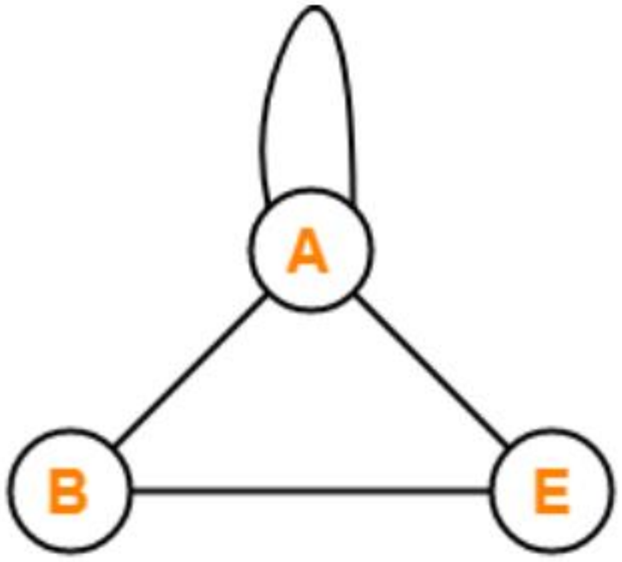
Example of Multi Graph

17. Pseudo Graph

A graph having no parallel edges but having self loop(s) in it is called as a pseudo graph.

Here,

- This graph consists of three vertices and four edges out of which one edge is a self loop.
- There are no parallel edges but a self loop is present.
- Therefore, it is a pseudo graph.



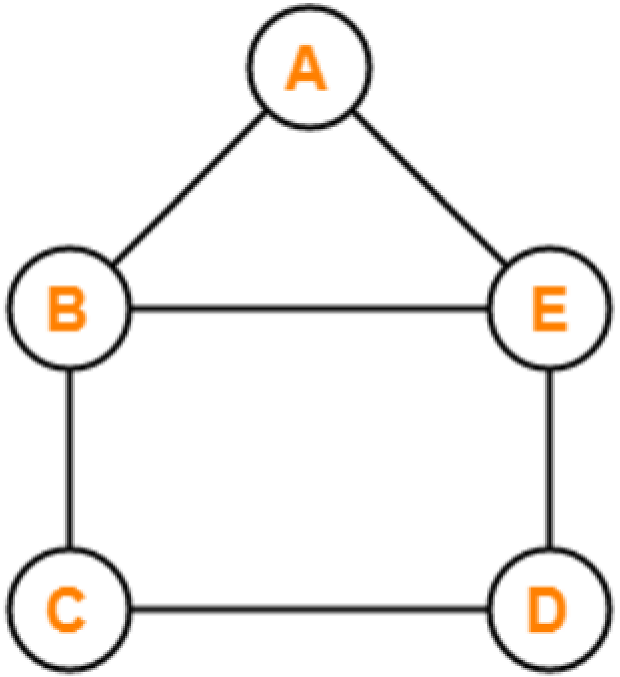
Example of Pseudo Graph

18. Planar Graph

A planar graph is a graph that we can draw in a plane such that no two edges of it cross each other.

Here,

- This graph can be drawn in a plane without crossing any edges.
- Therefore, it is a planar graph.



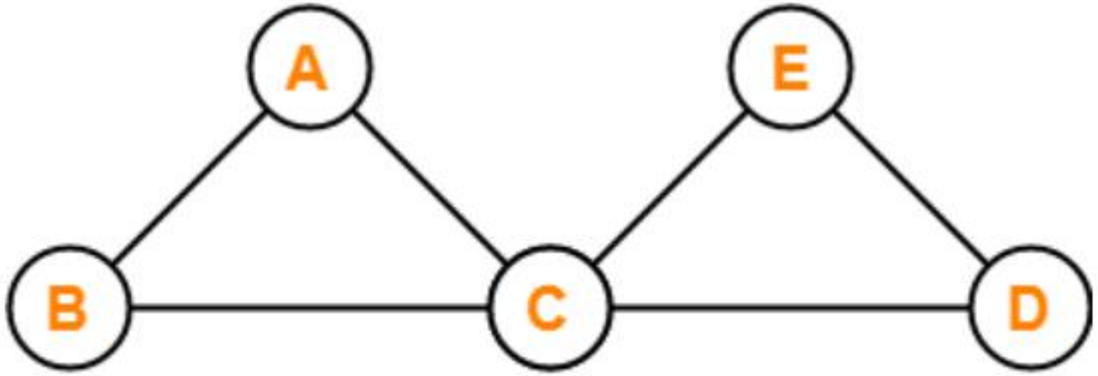
Example of Planar Graph

19. Euler Graph

Euler Graph is a connected graph in which all the vertices are even degree.

Here,

- This graph is a connected graph. The degree of all the vertices is even.
- Therefore, it is an Euler graph.



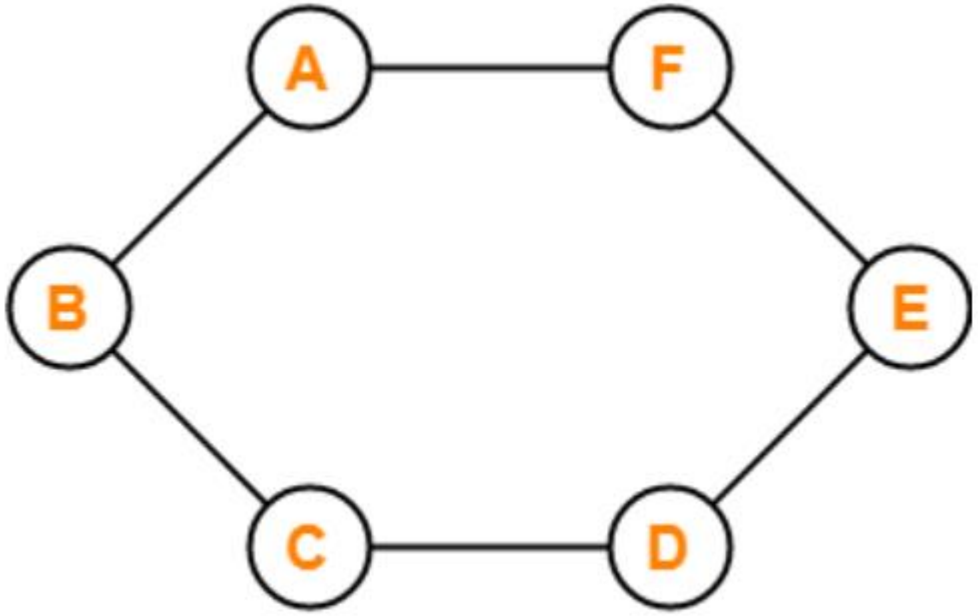
Example of Euler Graph

20. Hamiltonian Graph

If there exists a closed walk in the connected graph that visits every vertex of the graph exactly once (except starting vertex) without repeating the edges, then such a graph is called as a Hamiltonian graph.

Here,

- This graph contains a closed walk ABCDEFG that visits all the vertices (except starting vertex) exactly once.
- All the vertices are visited without repeating the edges.
- Therefore, it is a Hamiltonian Graph.



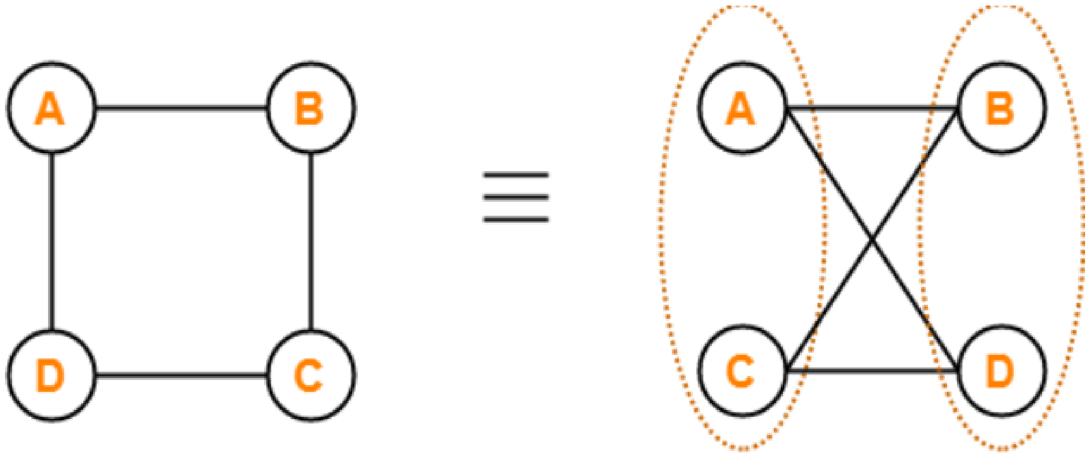
Example of Hamiltonian Graph

21. Bipartite Graph

Definition : A graph G is bipartite if the vertices can be partitioned into two sets X and Y so that every edge has one endpoint in X and the other in Y .

Here,

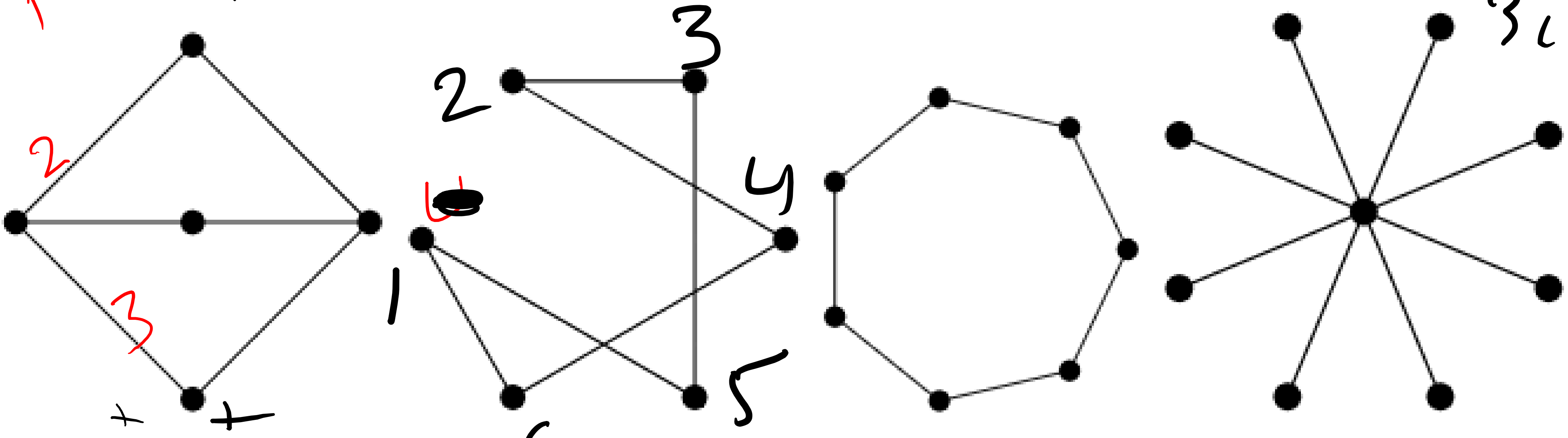
- The vertices of the graph can be decomposed into two sets.
- The two sets are X = {A, C} and Y = {B, D}.
- The vertices of set X join only with the vertices of set Y and vice-versa.
- The vertices within the same set do not join.
- Therefore, it is a bipartite graph.



Example of Bipartite Graph

Graph terminology

Which of the graphs below are bipartite? Justify your answers.



$1 \to 5, 6$

$2 \to 3, 4$

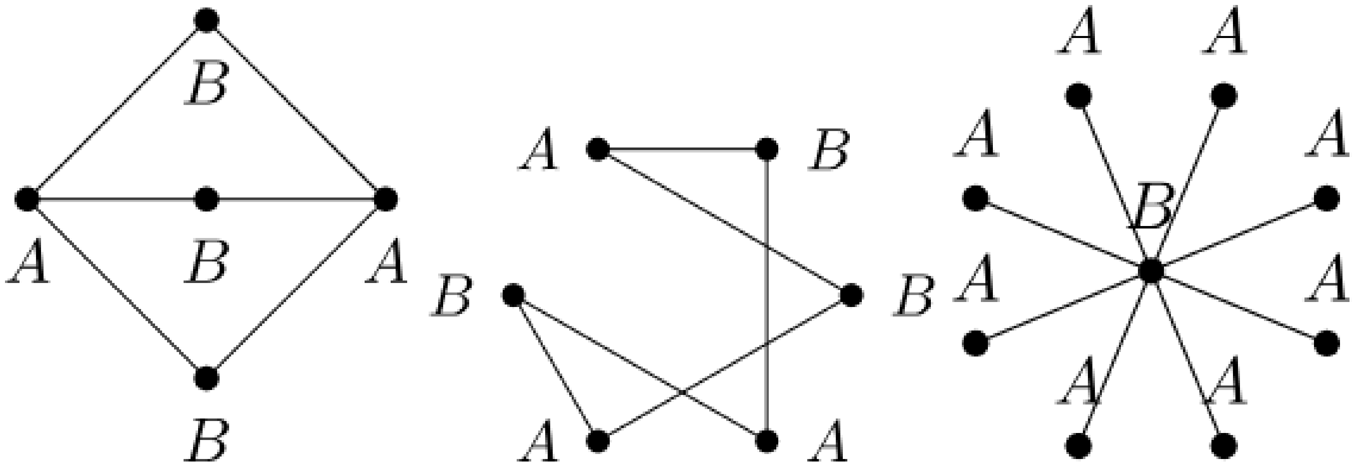
$3 \to 2, 5$

$4 \to 2, 6$

$6 \to 1, 4$

Graph terminology

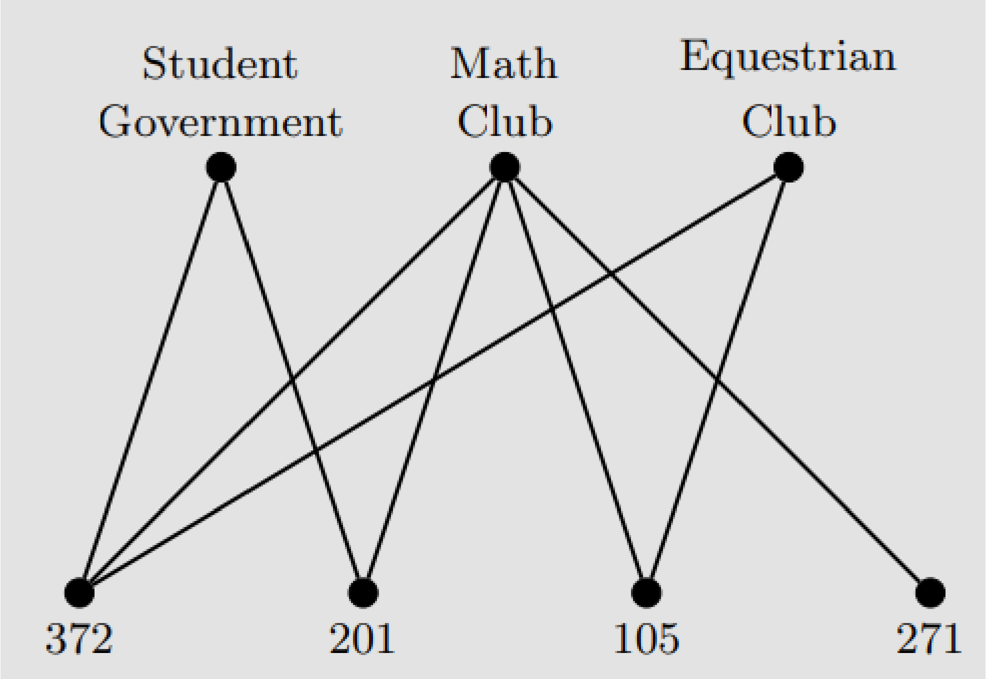
Three of the graphs are bipartite. The one which is not is second from the right. To see that the three graphs are bipartite, we can just give the bipartition into two sets A and B,, as labeled below:



Graph terminology

Example : Three student organizations (Student Government, Math Club, and the Equestrian Club) are holding meetings on Thursday afternoon. The only available rooms are 105, 201, 271, and 372. Based on membership and room size, the Student Government can only use 201 or 372, Equestrian Club can use 105 or 372, and Math Club can use any of the four rooms. Draw a graph that depicts these restrictions.

Solution: Each organization and room is represented by a vertex, and an edge denotes when an organization is able to use a room.



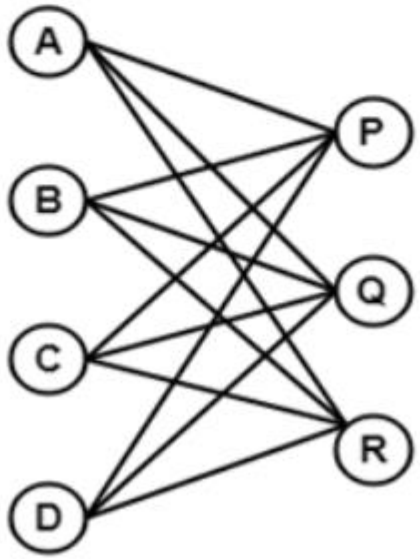
Note that edges do not occur between two organizations or between two rooms, as these would be nonsensical in the context of the problem. The graph above is a bipartite graph.

22.Complete Bipartite Graph

- A **complete bipartite graph** is a bipartite graph in which each vertex in the first set is joined to each vertex in the second set by exactly one edge.
- A complete bipartite graph is a bipartite graph which is complete.

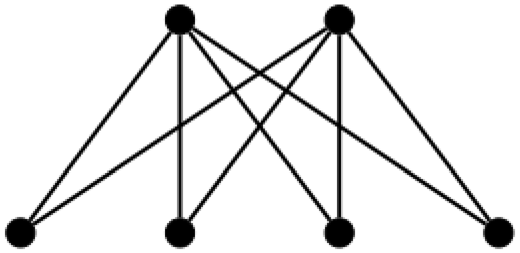
Complete Bipartite graph = Bipartite graph + Complete graph

The above graph is known as K4,3.

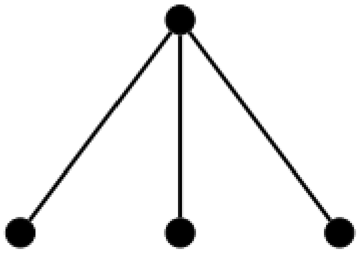


Graph terminology

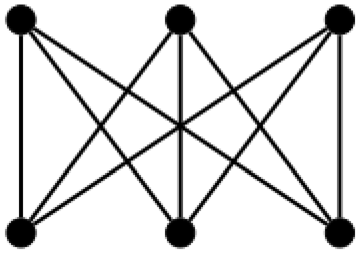
The following are a few complete bipartite graphs. It is customary to write m and n in increasing order (so K2,3 versus K3,2), but it is not required.



$K_{2,4}$

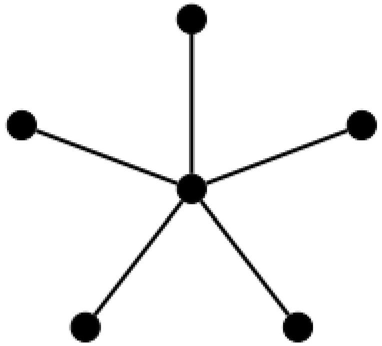


$K_{1,3}$

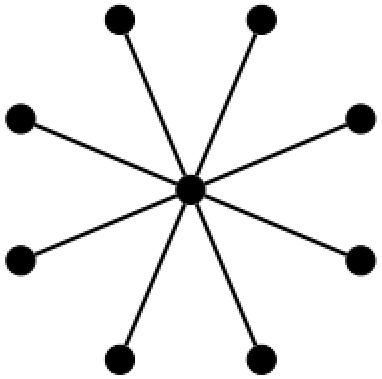


$K_{3,3}$

When m = 1, we call K1,n a star since we could draw these with a singular vertex in the center and the remaining vertices surrounding it, as seen below with K1,5 and K1,8.



$K_{1,5}$

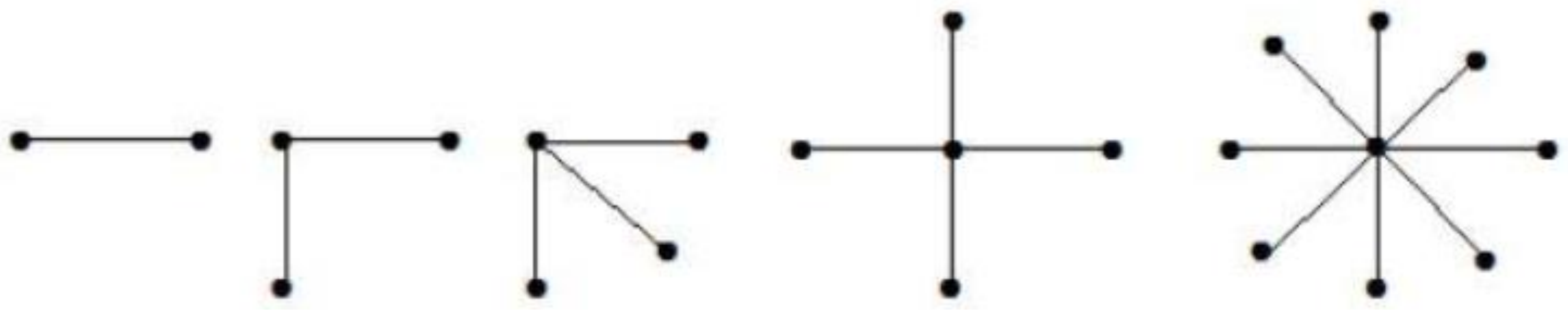


$K_{1,8}$

23. Star Graph

A star graph is a complete bipartite graph in which n-1 vertices have degree 1 and a single vertex have degree (n -1). This exactly looks like a star where (n - 1) vertices are connected to a single central vertex.

A star graph with n vertices is denoted by S_n.



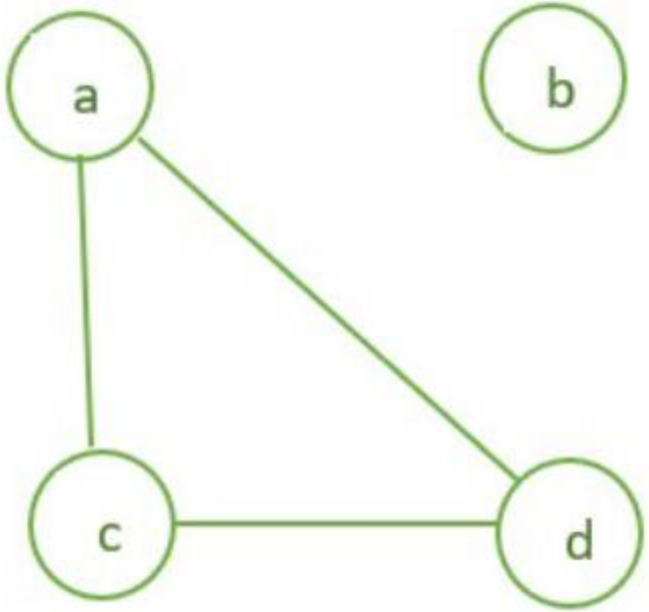
In the above example, out of n vertices, all the (n-1) vertices are connected to a single vertex. Hence, it is a star graph.

Graph complements

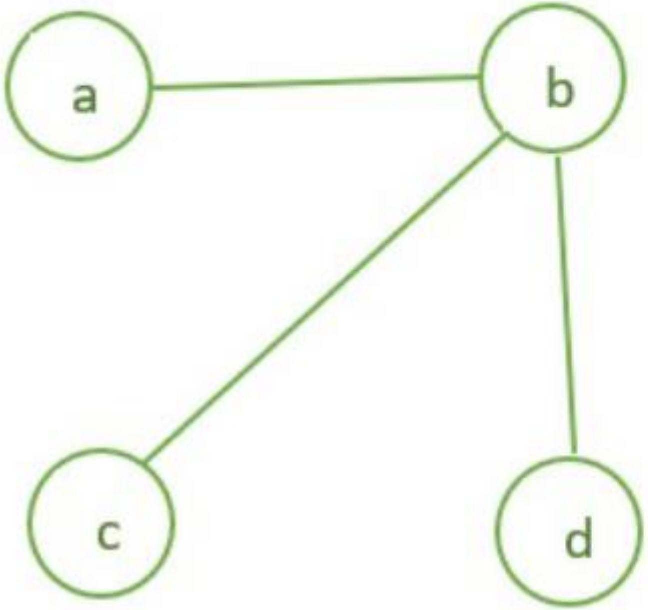
- The complement of a graph G is a graph G’ on the same set of vertices as of G such that there will be an edge between two vertices (v, e) in G’, if and only if there is no edge in between (v, e) in G.
- Complement of graph **G(v, e)** is denoted by **G'(v, e’).**
- The number of vertices remains unchanged in the complement of the graph.

Graph complements

Example:



Graph



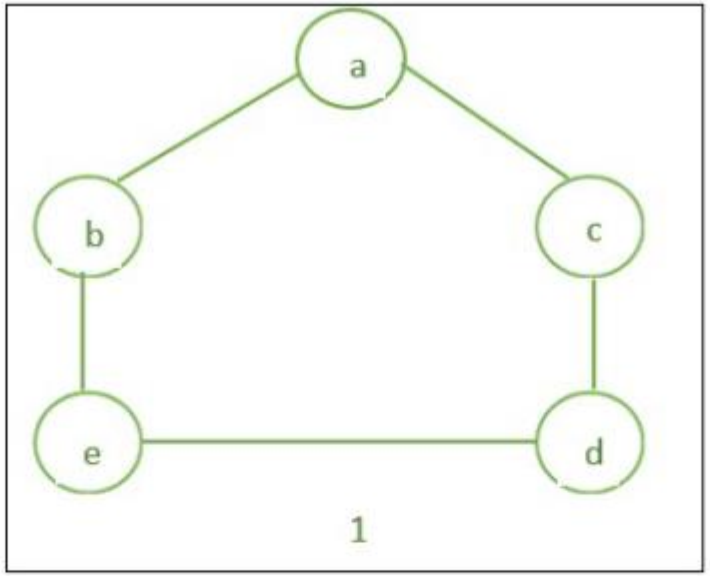
Complemented Graph

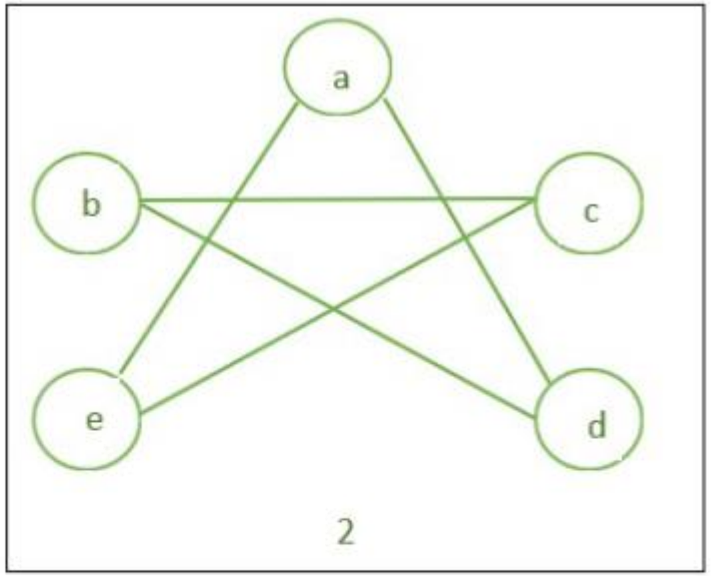
In the above example in graph G there is a edge between (a, d),(a, c),(a, d). Complement of Graph G is G' having edges between (a, b),(b, c),(b, d).

Graph complements

Properties of Complemented Graph

1. If E be the set of edges of graph G’ then E(G’)={ (u, v) | (u, v) ∉ E(G) }

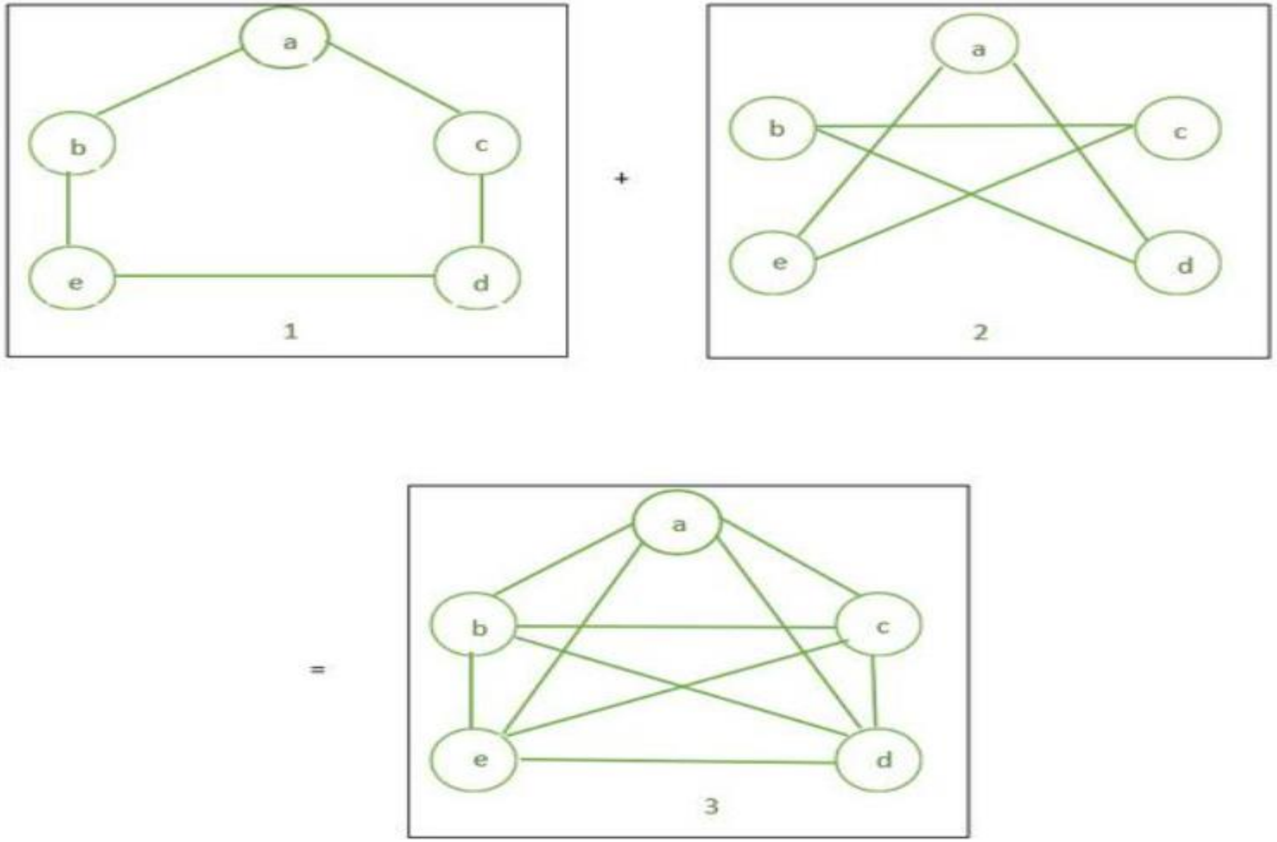




Graph and its Complement

Graph complements

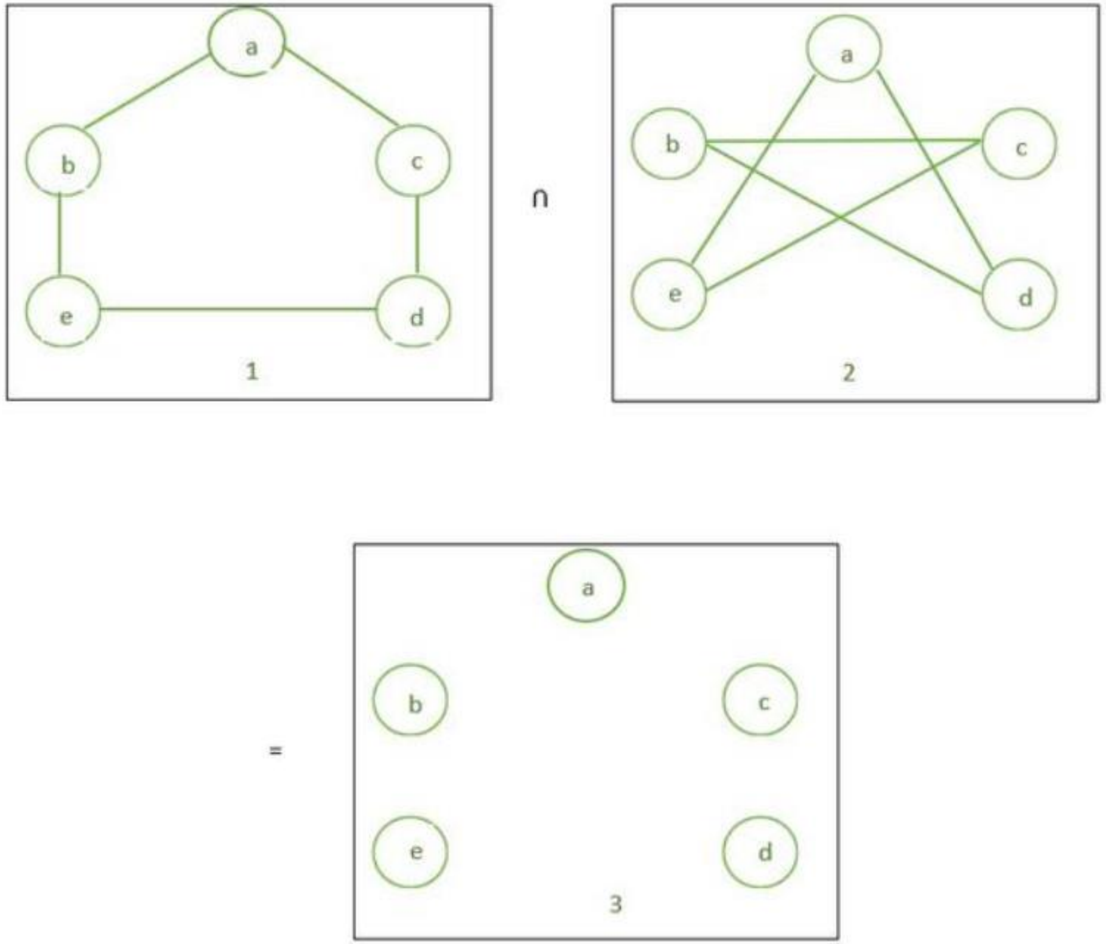
2. Union of graph G and its complement G’ will give a complete graph(Kn).



$$E(G')+E(G) = E(Kn) = n(n-1)\div 2.$$

Graph complements

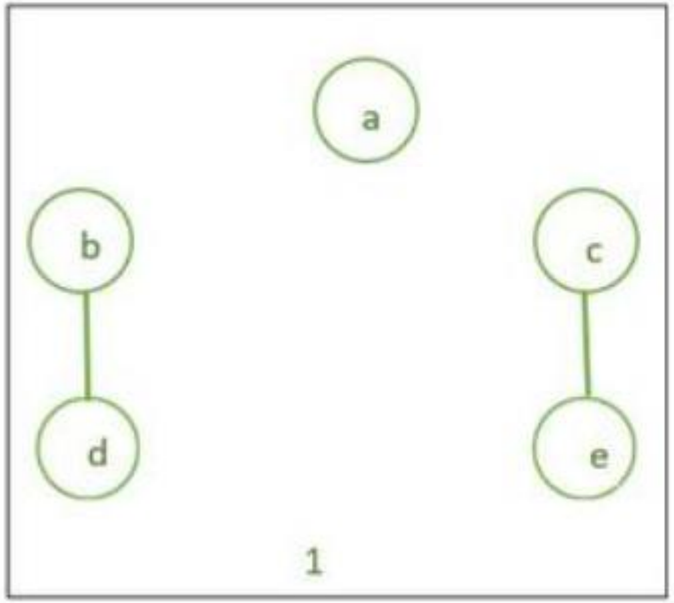
3. The intersection of two complement graphs has no edges, also known as null graph

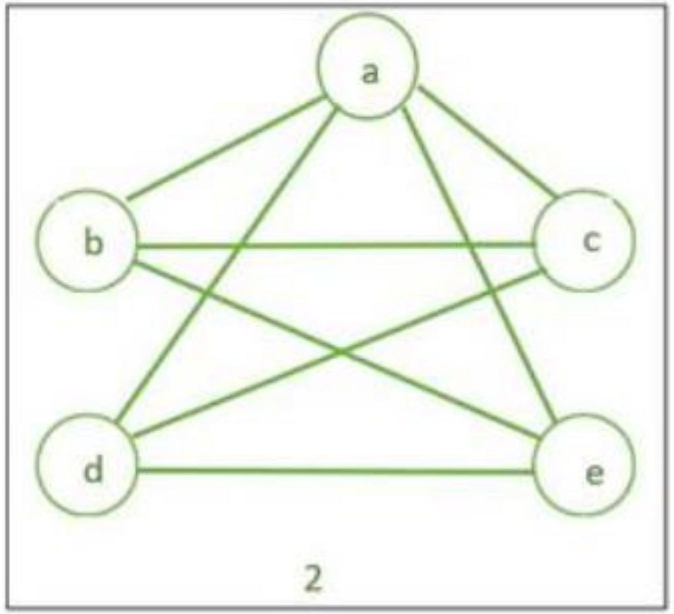


The intersection of the graph and complemented graph

Graph complements

4. If G is a disconnected graph then its complement G’ would be a connected graph.

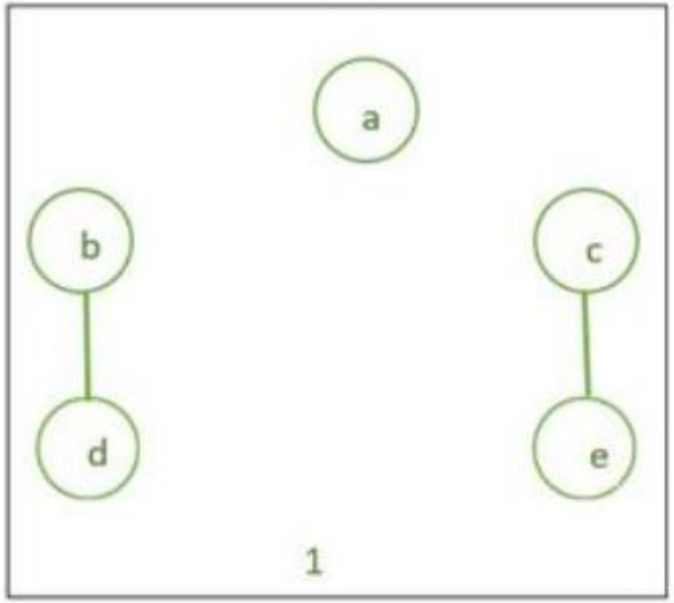


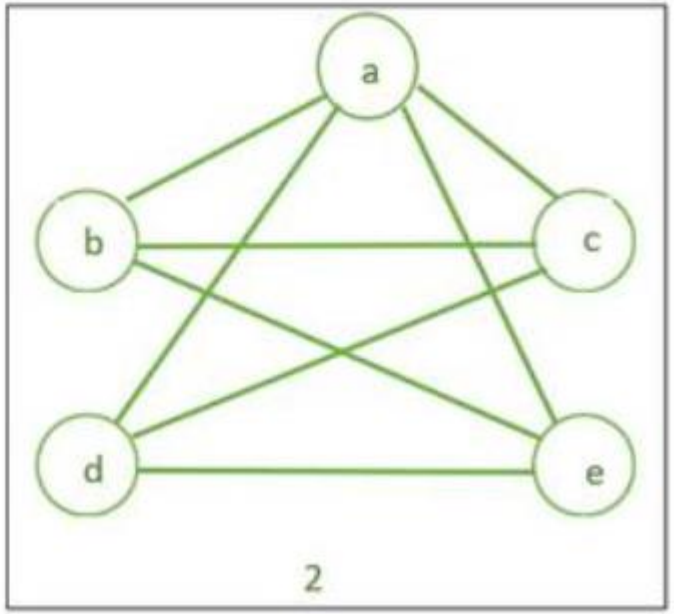


The complement of a Disconnected graph is connected

Graph complements

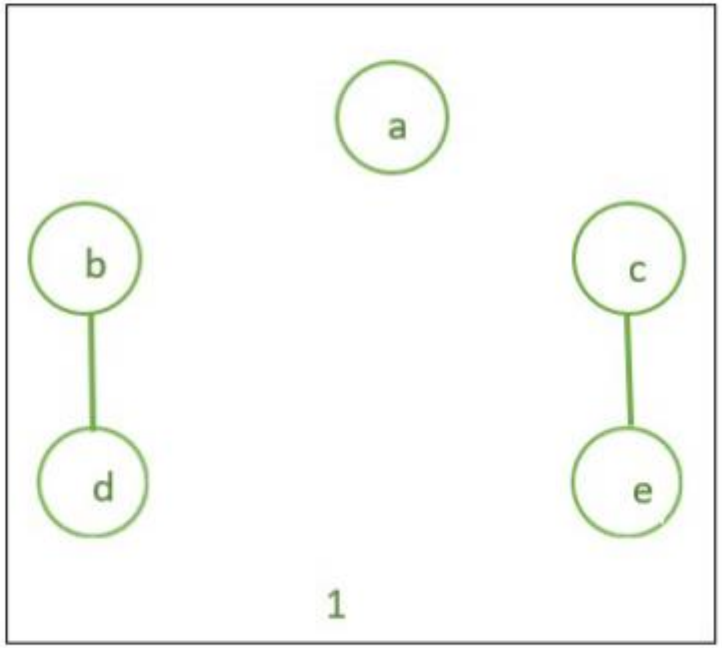
5. Number of vertices of a Graph and its Complement are Same.

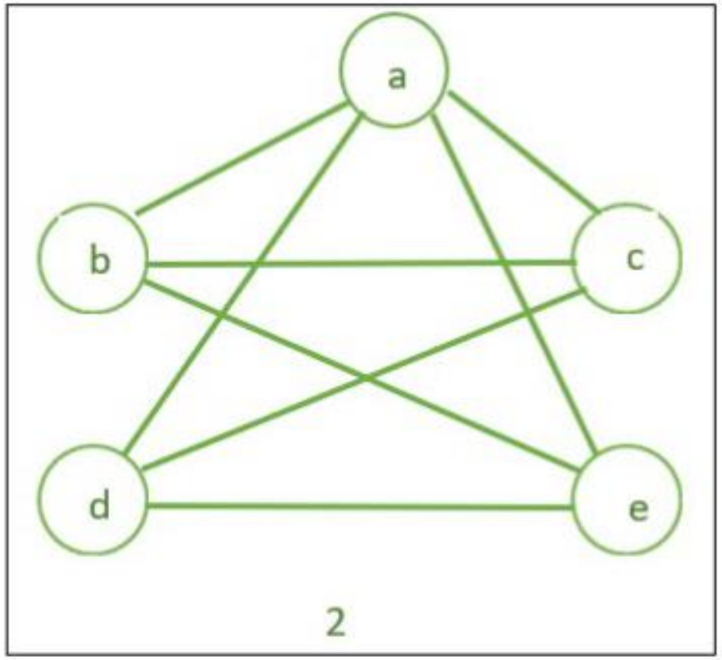




Graph complements

6. Size of a Graph and its complement cannot be the same. The size of a graph is the number of edges in it.





The size of Graph 1 is 2 and the Size of its Complement Graph 2 is 8

Graph complements

Question 1. Consider a simple graph G, where E denotes the edges and V denotes the vertices |E(G)|= 30, |E(G')|= 36. Find |V(G)|=?

Solution:

We know,

E(G')+E(G)=E(Kn)=n(n-1)÷2.

⇒ 36+30=n(n-1)÷2

⇒ 66=n(n-1)÷2

⇒ 66×2=n2-n

⇒ n2-n-132=0

⇒ n2-12n+11n-132=0

⇒ n(n-12)+11(n-12)=0

⇒ (n-12)(n+11)=0

Therefore, n=12 and n=-11.

Since vertices cannot be negative.   n=12.

Graph complements

Question 2. Consider a simple graph G, where E denotes the edges and V denotes the vertices $|E(G)| = 12$, $|V(G)| = 8$. Find the number of edges in complement graph $|E(G')| = ?$.

Solution:

We know,

$E(G')+E(G)=E(K_n)=n(n-1)\div 2.$

$\Rightarrow E(G') + 12 = 8(8-1)\div 2$ [here n denotes number of vertices, i.e. given 8]

$\Rightarrow E(G')+12 = 8(7)\div 2$

$\Rightarrow E(G')+12= 4\times 7$

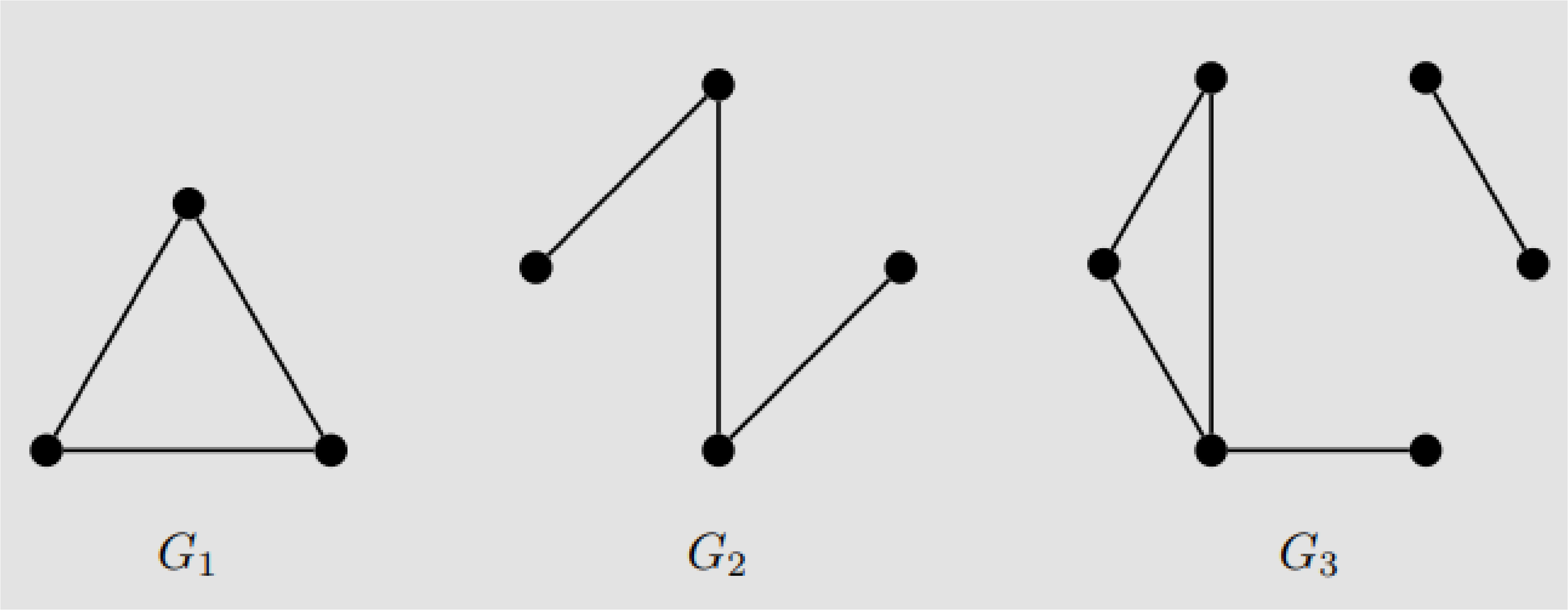
$\Rightarrow E(G')+12=28$

$\Rightarrow E(G')=28-12$

$\Rightarrow E(G')=16$

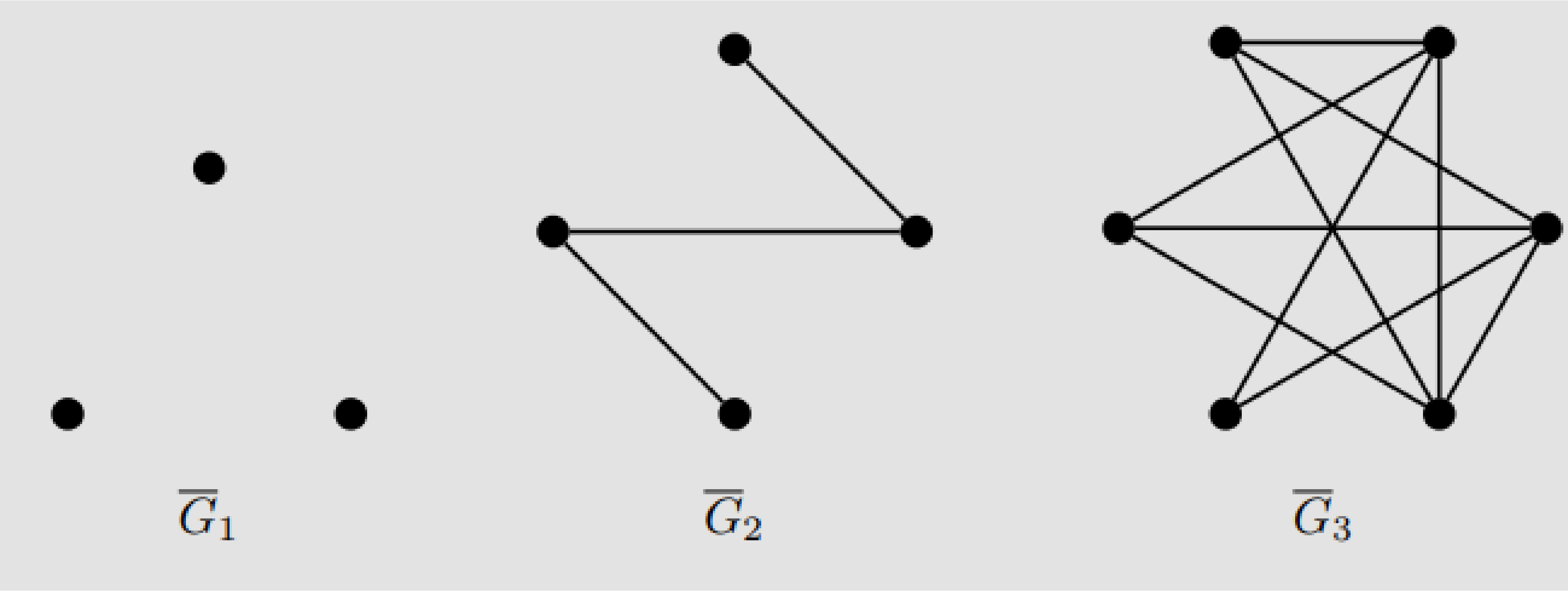
Graph complements

3. Find the complements of each graph shown below.



Graph complements

Solution: For each graph we simply add an edge where there wasn't one before and remove the current edges.



Subgraph

Definition :A subgraph H of a graph G is a graph where H contains some of the edges and vertices of G; that is, $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

induced subgraph

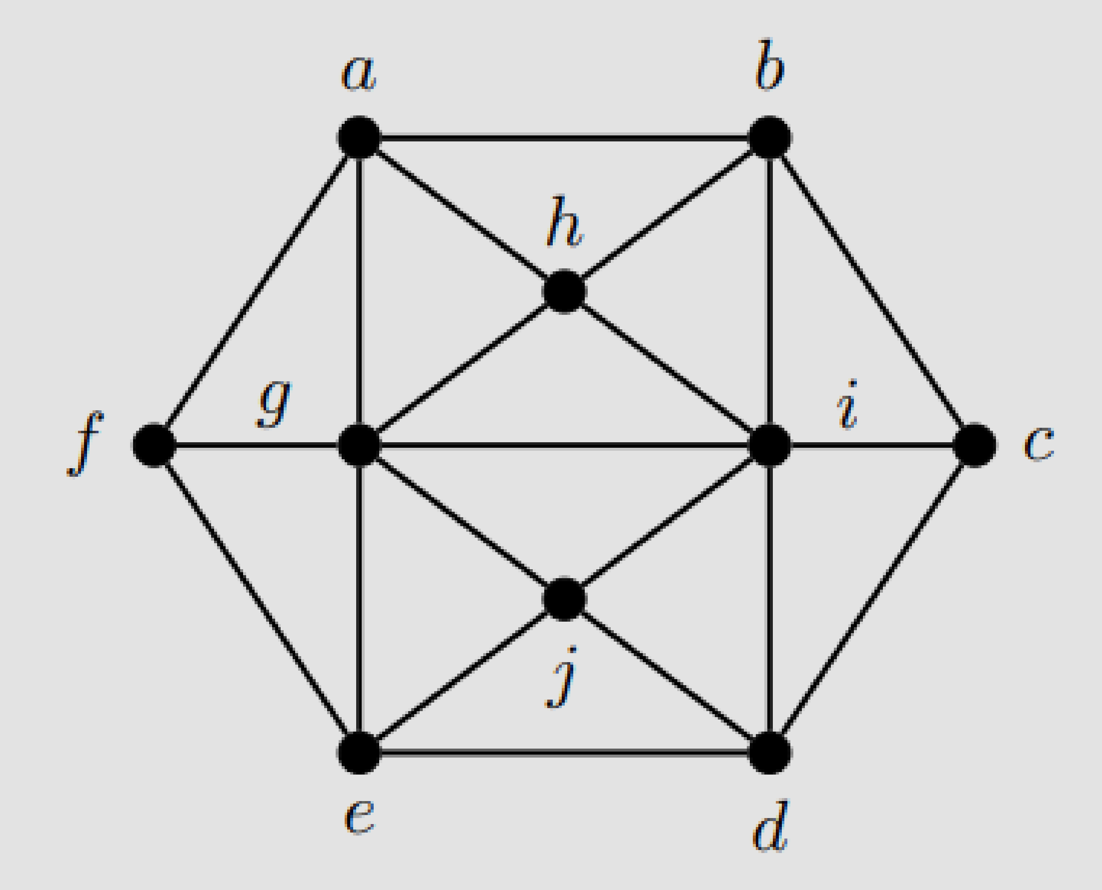
Definition : Given a graph $G = (V, E)$, an induced subgraph is a subgraph $G[V']$ where $V' \subseteq V$ and every available edge from G between the vertices in V' is included. Here all the edges are present between the chosen vertices.

spanning subgraph

spanning subgraph contains all the vertices of G but not necessarily all the edges of G; that is, $V(H) = V(G)$ and $E(H) \subseteq E(G)$. Here all the vertices are present between the chosen edges.

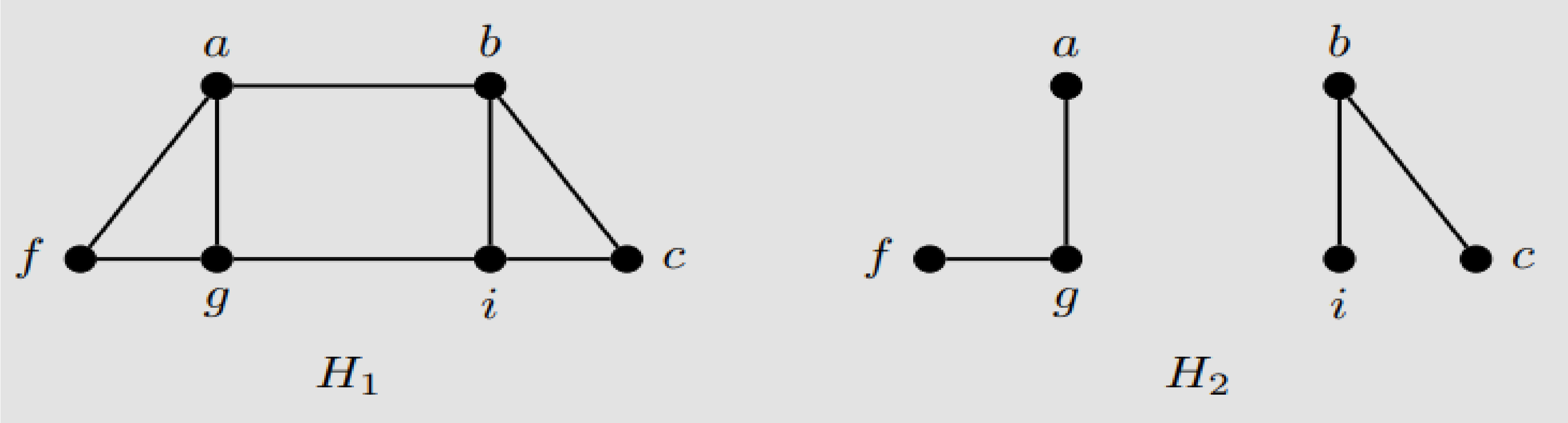
Sub Graph

Example : Consider the graph G below. Find two subgraphs of G, both of which have vertex set V' = {a, b, c, f, g, i}.



Sub Graph

Solution: Two possible solutions are shown below. Note that the graph H1 on the left contains every edge from G amongst the vertices in V' , whereas the graph H2 on the right does not since some of the available edges are missing (namely, ab, af, ci, and gi).



The graph shown on the left above is a special type, called an induced subgraph, since all the edges are present between the chosen vertices. Another special type of subgraph, called a spanning subgraph, includes all the vertices of the original graph

1. SUM OF GRAPHS

- **Definition:** Given two graphs G and H and if the vertex-sets are disjoint that is $V(G) \cap V(H) = \emptyset$, then we call the disjoint union the sum, denoted G + H.

or

- If we have 2 graphs, G1 & G2 such that their vertices intersection is null ($V(G1) \cap V(G2) = \emptyset$) , then the sum : G1 + G2 is defined as the graph whose vertex set V(G1+G2) is V(G1) + V(G2) and the edge set consists of these edges, which are inG1 & in G2 & the edges contained by joining each vertex of G1 to each vertex of G2.

Graph combinations

Example : Draw the addition of 2 graphs shown G1 & G2.

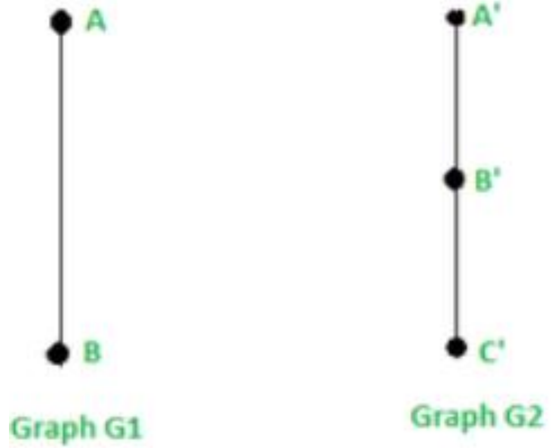
Here : $V(G1) \cap V(G2) = \emptyset$

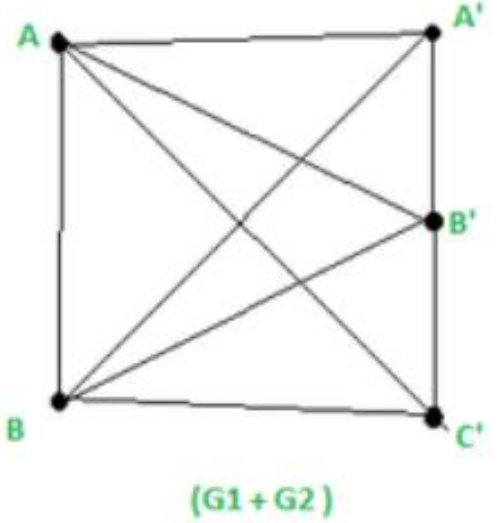
- The already contained edges in G1 are, E(G1) : {{A, B}} and the vertices are : V(G1) = {A, B}
- The already contained edges in G2 are, E(G2) : {{A', B'} ,{B',C'} }and the vertices are : V(G2) = {A', B', C'}
- So the graph , G1 + G2 will have

(i) vertices as : V(G1+G2) = V(G1) + V(G2) = { A, B. A', B', C'}

(ii) and E(G1 + G2) = E(G1) + E(G2) + edges contained by joining each vertex of G1 to each vertex of G2 =

{ {A, B}, {A', B'} ,{B',C'} , {A,A'} , {A, B'}, {A, C'}, {B,A'} , {B, B'}, {B, C'}}

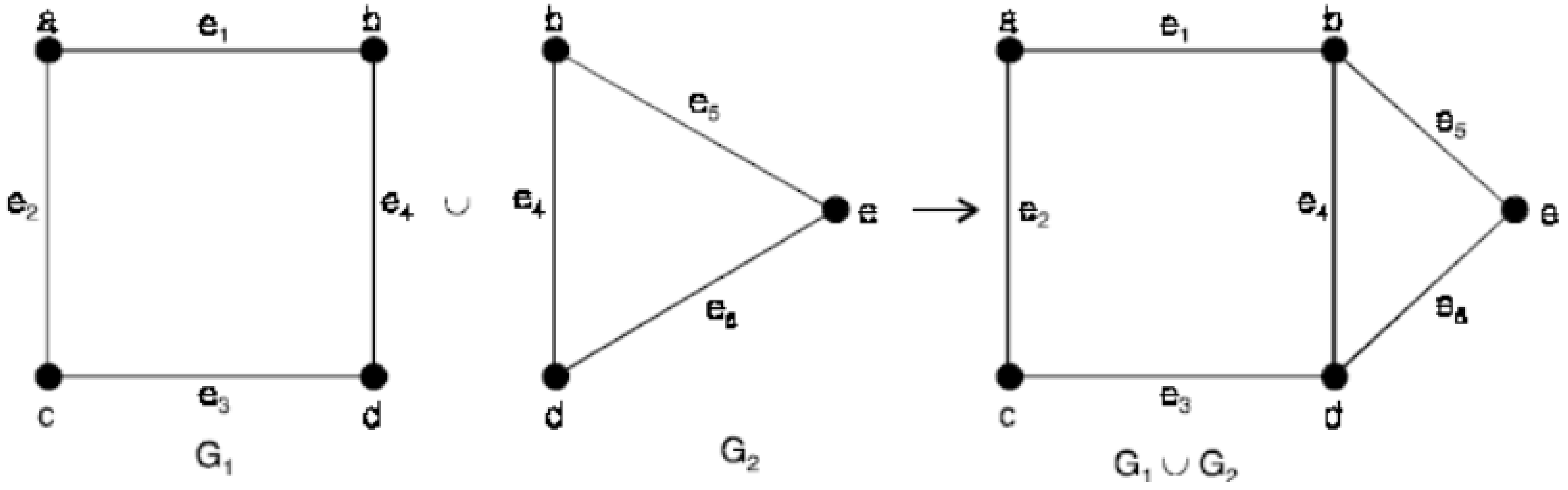




2. UNION OF GRAPHS

- **Definition:** Given two graphs G and H, the union G ∪ H is the graph with vertex-set V (G ∪ H) =V (G) ∪ V (H) and edge-set E(G∪H)=E(G) ∪ E(H).

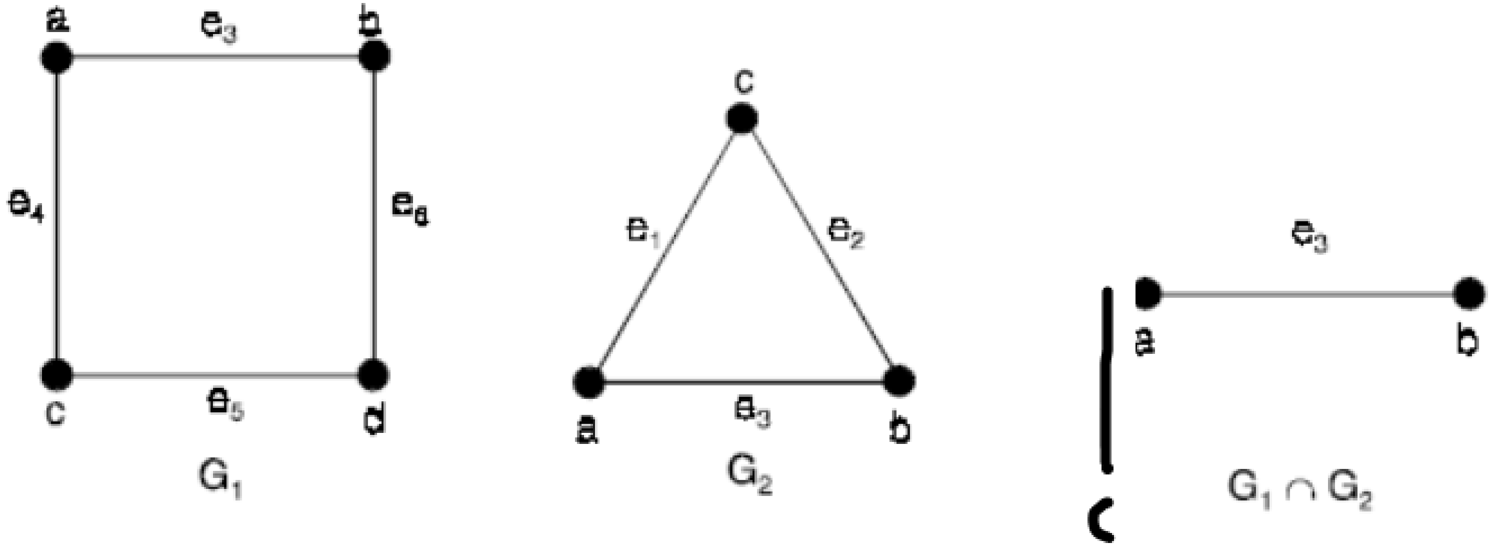
Example:



3. INTERSECTION OF GRAPHS

- **Definition:** Given two graphs G and H with at least one vertex in common, then their intersection G ∩ H is the graph with vertex-set V (G ∩ H) =V (G) ∩ V (H) and edge-set E(G ∩ H)=E(G) ∩ E(H).

Example:



Graph Isomorphisms

Isomorphic Graphs

A graph can exist in different forms having the same number of vertices, edges, and also the same edge connectivity. Such graphs are called isomorphic graphs.

Two graphs G1 and G2 are said to be isomorphic if −

1. Both graphs have same number of vertices and edges.
2. The degree of vertices in both graphs are same.
3. One to one mapping of vertices
4. Both graphs preserves adjacency and non-adjacency i.e , for every pair of vertices in G,

   u and v are adjacent in G ⇔ f(u) and f(v) are adjacent in H.

   u and v are non adjacent in G ⇔ f(u) and f(v) are non adjacent in H.

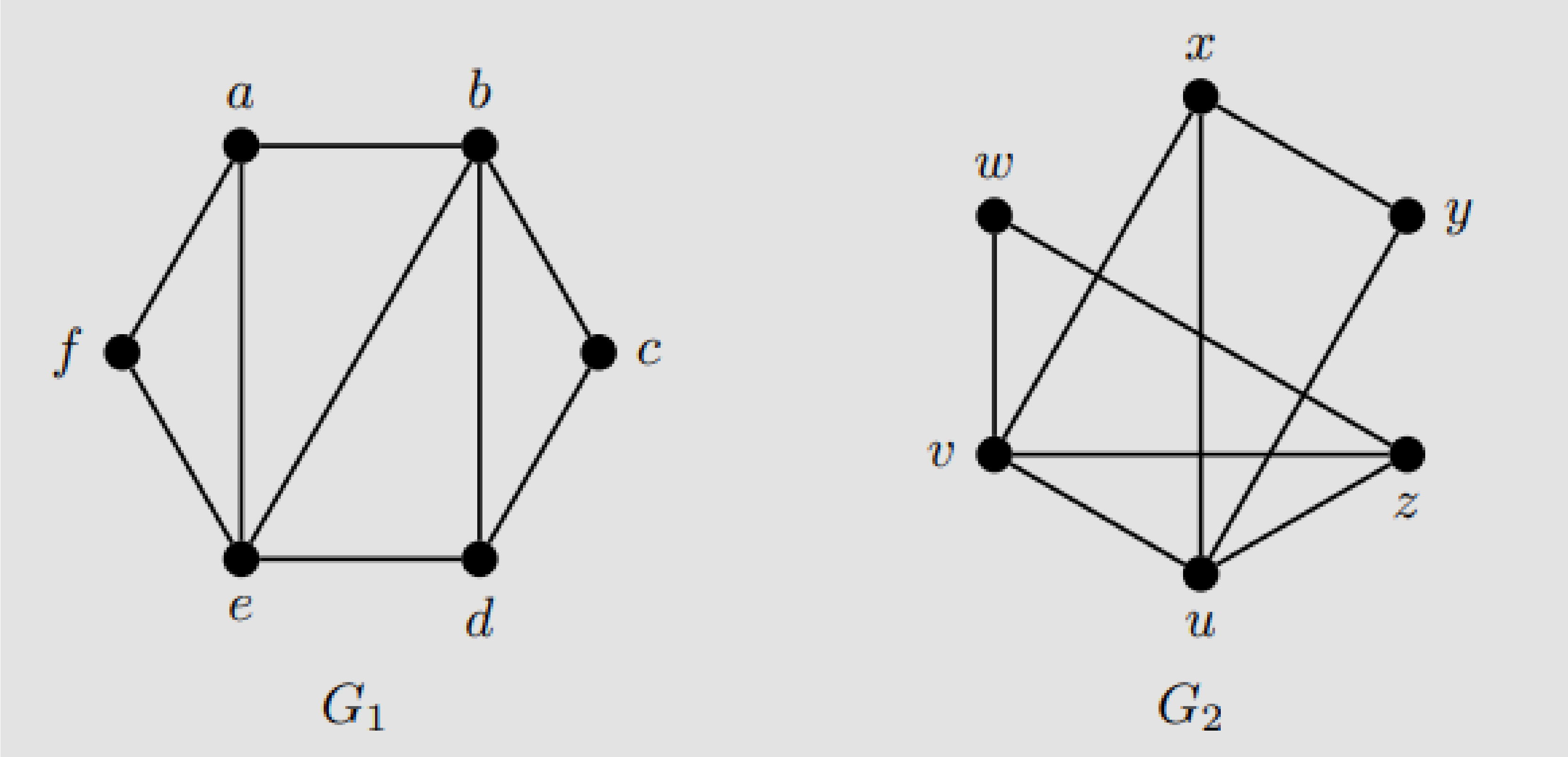
5. Adjacency matrix of both the graphs is similar

Isomorphisms for simple graphs

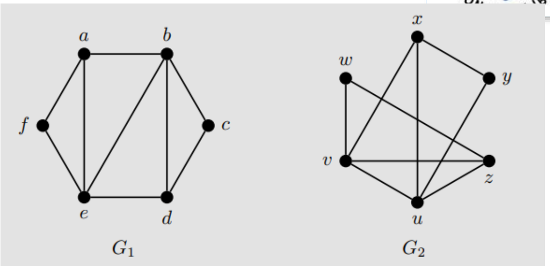
definition: Two simple graphs G and H are isomorphic, denoted G ~= H, if there exists a structure-preserving vertex bijection f : $V_G \rightarrow V_H$. Such a function f between the vertex-sets of G and H is called an isomorphism from G to H.

Graph Isomorphisms

Example : Determine if the following pair of graphs are isomorphic. If so, give the vertex pairings; if not, explain what property is different among the graphs.



Graph Isomorphisms



1. **Both graphs have same number of vertices and edges.**

	G_1	G_2
No of Vertices	6	6
No of edges	9	9

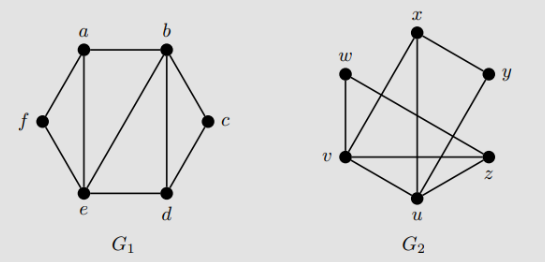
2. **The degree of vertices in both graphs are same.**

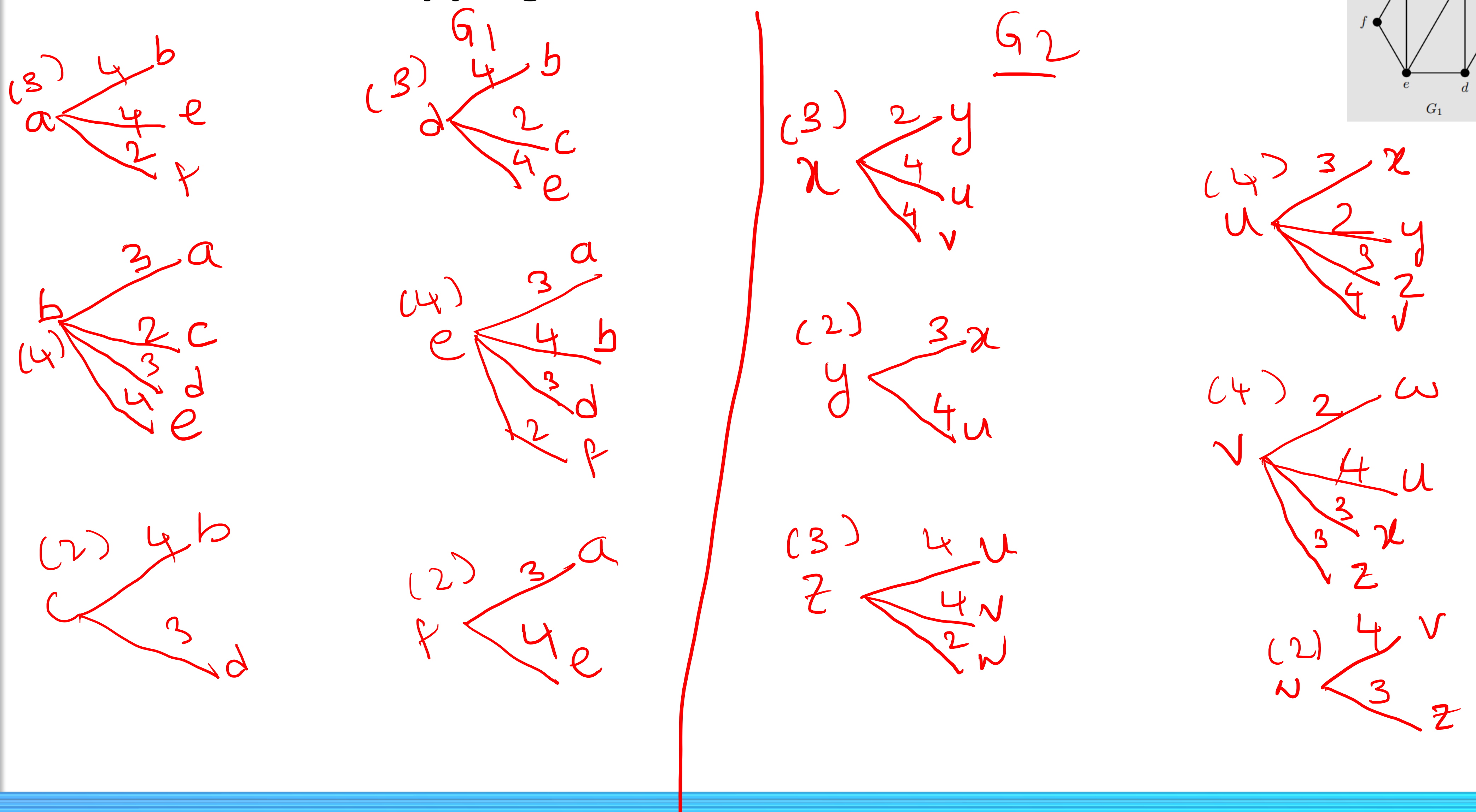
Degree of Vertices

G_1	G_2
a → 3	x → 3
b → 4	y → 2
c → 2	z → 3
d → 3	u → 4
e → 4	v → 4
f → 2	w → 2

Degree Sequence 4, 4, 3, 3, 2, 2 4, 4, 3, 3, 2, 2

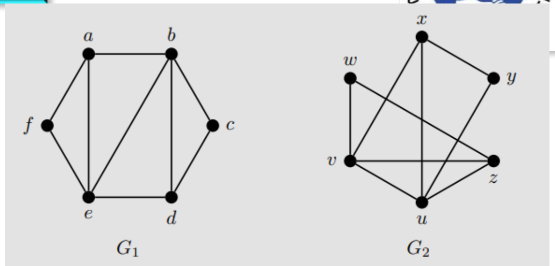
3. One to one mapping of vertices





a ⟷ x
b ⟷ v
c ⟷ w
d ⟷ z
e ⟷ u
f ⟷ y

Graph Isomorphisms



4. Both graphs preserves adjacency and non-adjacency i.e , for every pair of vertices in G,

u and v are adjacent in G ⇔ f(u) and f(v) are adjacent in H.

u and v are non adjacent in G ⇔ f(u) and f(v) are non adjacent in H.

edge in G_1		edge in G_2
ab	⟷	xv
bc	⟷	vw
cd	⟷	wz
de	⟷	zu
ef	⟷	uy
fa	⟷	yx
ae	⟷	xu
be	⟷	vu
bd	⟷	vz

a ⟷ x

b ⟷ v

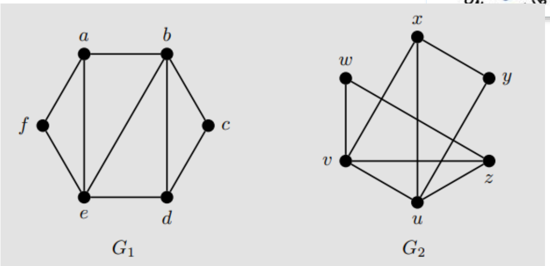
c ⟷ w

d ⟷ z

e ⟷ u

f ⟷ y

5. Adjacency matrix of both the graphs is similar



$G_1 \rightarrow$

$$
\begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
\hline
a & 0 & 1 & 0 & 0 & 1 & 1 \\
b & 1 & 0 & 1 & 1 & 1 & 0 \\
c & 0 & 1 & 0 & 1 & 0 & 0 \\
d & 0 & 1 & 1 & 0 & 1 & 0 \\
e & 1 & 1 & 0 & 1 & 0 & 1 \\
f & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

$G_2 \rightarrow$

$$
\begin{array}{c|cccccc}
 & x & v & w & z & u & y \\
\hline
x & 0 & 1 & 0 & 0 & 1 & 1 \\
v & 1 & 0 & 1 & 1 & 1 & 0 \\
w & 0 & 1 & 0 & 1 & 0 & 0 \\
z & 0 & 1 & 1 & 0 & 1 & 0 \\
u & 1 & 1 & 0 & 1 & 0 & 1 \\
y & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

$a \longleftrightarrow x$

$b \longleftrightarrow v$

$c \longleftrightarrow w$

$d \longleftrightarrow z$

$e \longleftrightarrow u$

$f \longleftrightarrow y$

Since all the conditions are satisfied, the two graphs are isomorphic

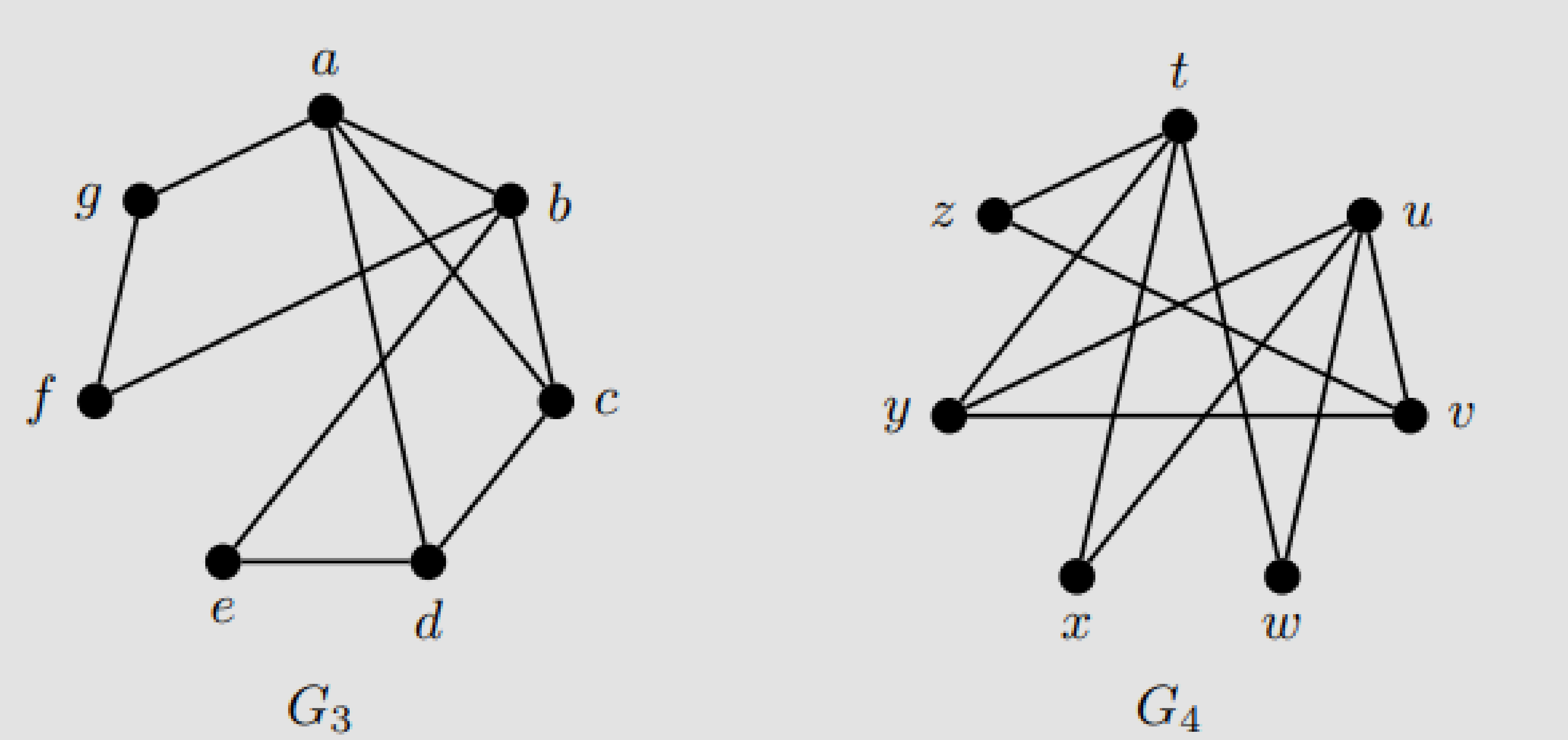
Graph Isomorphisms

Solution: First note that both graphs have six vertices and nine edges, with two vertices each of degrees 4, 3, and 2. Since all edge relationships are maintained, we know G1 and G2 are isomorphic.

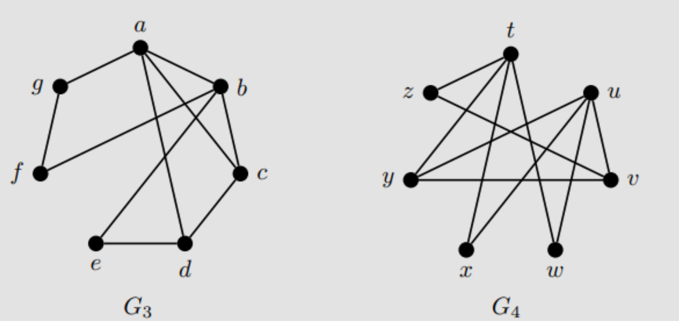
$V(G_1) \longleftrightarrow V(G_2)$	Edges	
$a \longleftrightarrow x$	$ab \longleftrightarrow xv$	✓
$b \longleftrightarrow v$	$ae \longleftrightarrow xu$	✓
$c \longleftrightarrow w$	$af \longleftrightarrow xy$	✓
$d \longleftrightarrow z$	$bc \longleftrightarrow vw$	✓
$e \longleftrightarrow u$	$bd \longleftrightarrow vz$	✓
$f \longleftrightarrow y$	$be \longleftrightarrow vu$	✓
	$cd \longleftrightarrow wz$	✓
	$de \longleftrightarrow zu$	✓
	$ef \longleftrightarrow uy$	✓

Graph Isomorphisms

Example : Determine if the following pair of graphs are isomorphic. If so, give the vertex pairings; if not, explain what property is different among the graphs.



Graph Isomorphisms



1. Both graphs have same number of vertices and edges.

	G3	G4
no of vertices	7	7
no of edges	10	10

2. The degree of vertices in both graphs are same.

G3

a → 4
b → 4
c → 3
d → 3
e → 2
f → 2
g → 2

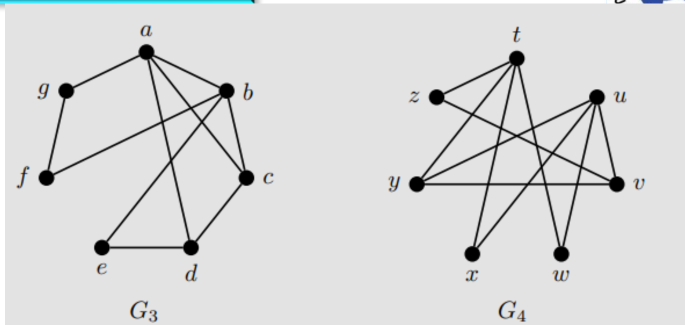
degree Sequence = 4, 4, 3, 3, 2, 2, 2

G4

t → 4
u → 4
v → 3
w → 2
x → 2
y → 3
z → 2

degree Sequence = 4, 4, 3, 3, 2, 2, 2

3. One to one mapping of vertices



G3

(4) a — 4 b, 3 c, 3 d, 2 g

(4) b — 4 a, 3 c, 2 e, 2 f

(4) c — 4 a, 4 b, 3 d

(3) d — 4 a, 3 c, 2 e

(2) e — 4 b, 3 d

(2) f — 4 b, 2 g

(2) g — 4 a, 2 f

G4

(4) t — 2 z, 3 y, 2 x, 2 w

(4) u — 3 y, 4 x, 2 w, 3 v

(3) v — 4 u, 2 z, 3 y

(2) w — 4 t, 4 u

(2) z — 4 t, 4 v

(2) x — 4 t, 4 u

(3) y — 4 t, 4 u, 3 v

(4) a ⟷ u (4)

b ~~⟷~~ t

As degree of adjacent vertices of b & t are not same

∴ No one-one mapping & hence G3 & G4 are not isomorphic

Graph Isomorphisms

Solution:

- First note that both graphs have seven vertices and ten edges, with two vertices each of degrees 4 and 3, and three vertices of degree 2.
- we know corresponding vertices must have the same degree, and so the vertices of degree 4 in G3, a and b, must map to the vertices of degree 4 in G4, namely t and u.
- However, in G3 the degree 4 vertices (a and b) are adjacent, whereas in G4 there is no edge between the degree 4 vertices (t and u).
- Thus G3 and G4 are not isomorphic.

Isomorphisms for pseudo graphs(self loops)

Definition: Two pseudo graphs G and H are isomorphic, denoted G ~= H, if there exists a structure-preserving vertex bijection $f : V_G \rightarrow V_H$ if

(i) the number of edges (even if 0) between every pair of distinct vertices u and v in graph G equals the number of edges between their images f(u) and f(v) in graph H,

(ii) the number of vertices are same in both graphs.

(iii) The degree of vertices should be same and

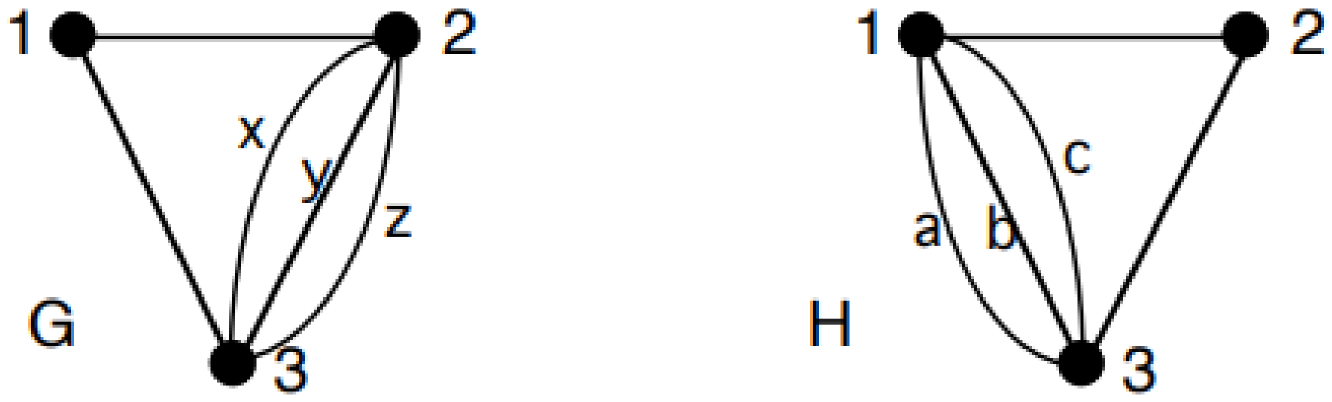
(iv) the number of self-loops at each vertex x in G equals the number of self-loops at the vertex f(x) in H.

(v) One to one mapping of vertices

(vi) Same adjacency matrix

Isomorphisms for multi graphs(multiple edges)

Definition: If G and H are graphs with multi-edges, then an isomorphism from G to H is specified by giving a vertex bijection $f_V : V_G \rightarrow V_H$ and an edge bijection $f_E : E_G \rightarrow E_H$ that are consistent.



Isomorphisms for Diagraphs(directional edges)

Definition: Two digraphs are isomorphic if there is an isomorphism f between their underlying graphs that preserves the direction of each edge. That is, e is directed from u to v if and only if f(e) is directed from f(u) to f(v).

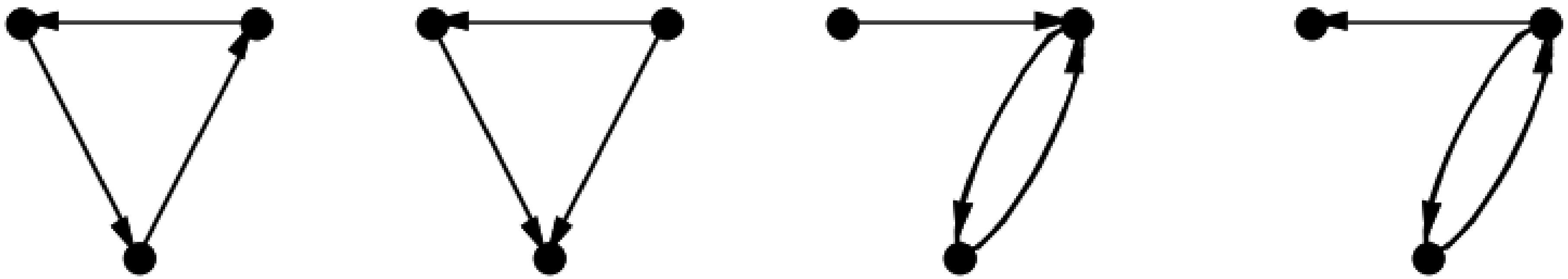
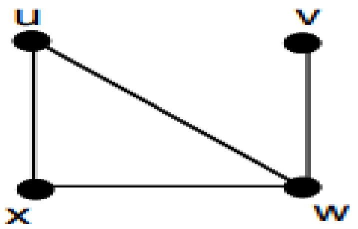
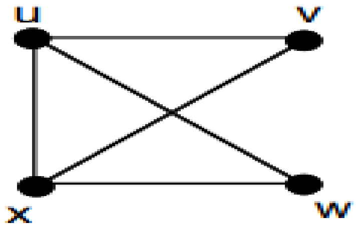


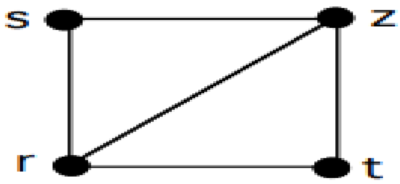
Figure 2.1.12 Four non-isomorphic digraphs.

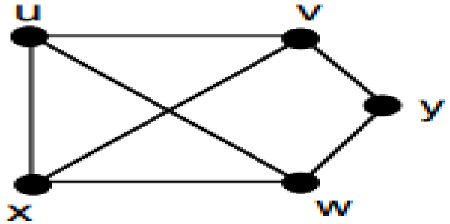
Graph Isomorphisms

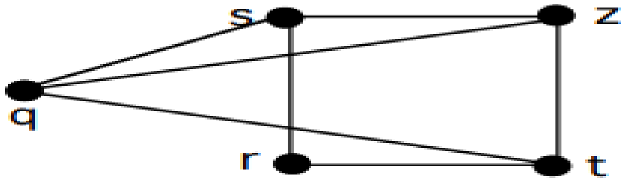
Q. find an isomorphism between the two graphs or digraphs shown

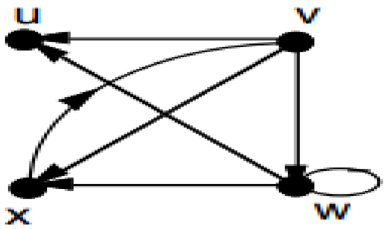


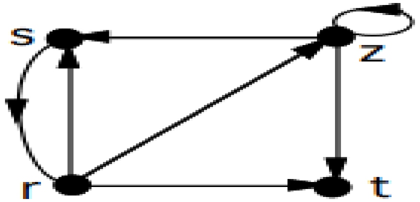












Matrix representations of graphs

- Representing graphs by matrices has conceptual and theoretical importance. It helps bring the power of linear algebra to graph theory.
- There are two principal ways to represent a graph G with the matrix, i.e., adjacency matrix and incidence matrix representation.

Matrix representations of graphs

adjacency matrix

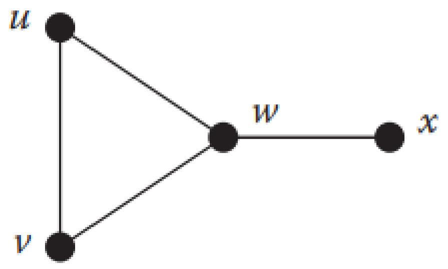
definition: The adjacency matrix of a simple graph G, denoted AG, is the symmetric matrix whose rows and columns are both indexed by identical orderings of VG, such that

$$A_G[u,v] = \begin{cases} 1 & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

definition: The adjacency matrix of a simple digraph D, denoted AD, is the matrix whose rows and columns are both indexed by identical orderings of VG, such that

$$A_D[u,v] = \begin{cases} 1 & \text{if there is a edge from } u \text{ to } v \\ 0 & \text{otherwise} \end{cases}$$

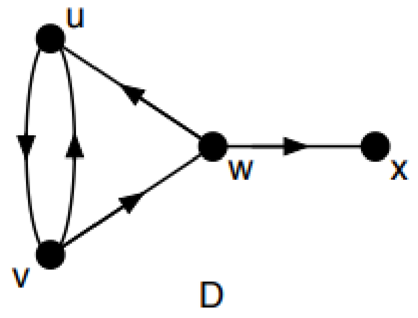
Example : Find the adjacency matrix of a graph G.



Solution :

$$A_G = \begin{array}{c c} & \begin{array}{cccc} u & v & w & x \end{array} \\ \begin{array}{c} u \\ v \\ w \\ x \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Example : Find the adjacency matrix of a graph G.

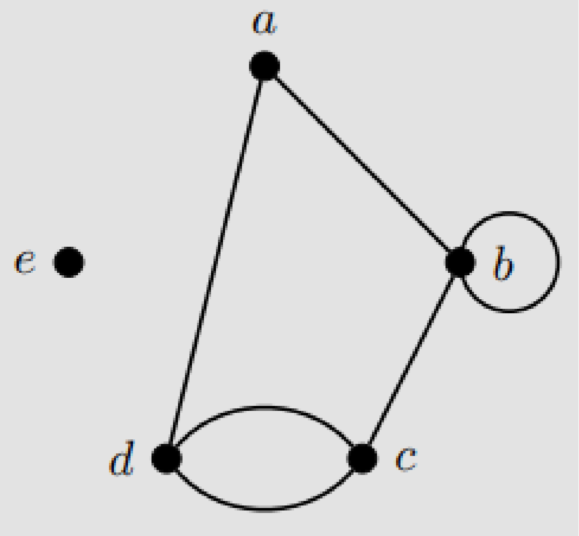


Solution :

$$A_D = \begin{array}{c c} & \begin{array}{cccc} u & v & w & x \end{array} \\ \begin{array}{c} u \\ v \\ w \\ x \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Matrix representations of graphs

Example : Find the adjacency matrix of a graph G.



Solution :

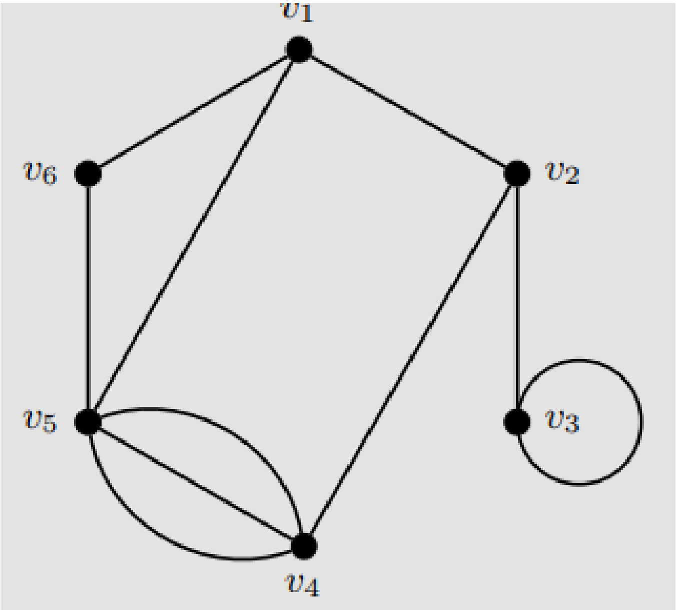
$$\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Matrix representations of graphs

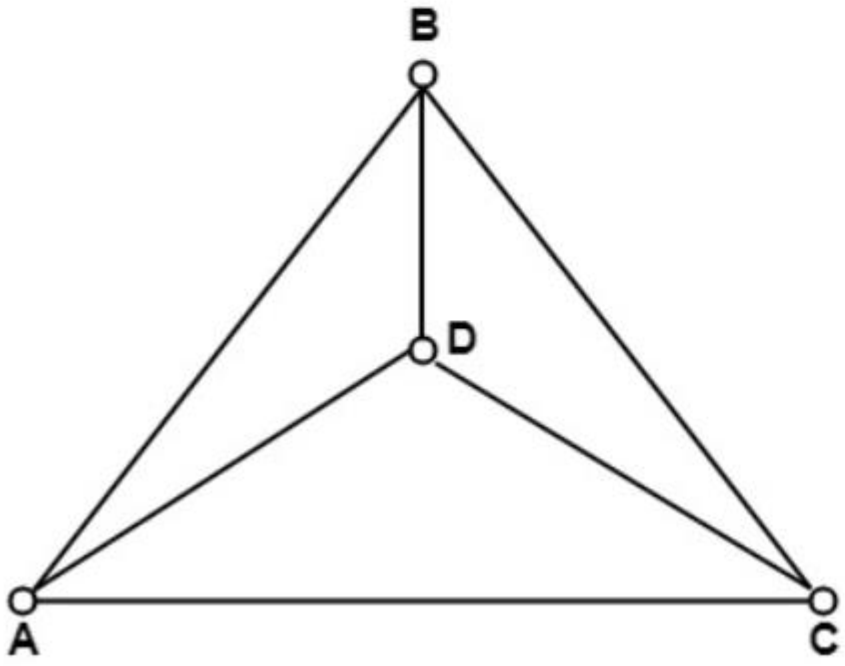
Example : Draw the graph whose adjacency matrix is shown below.

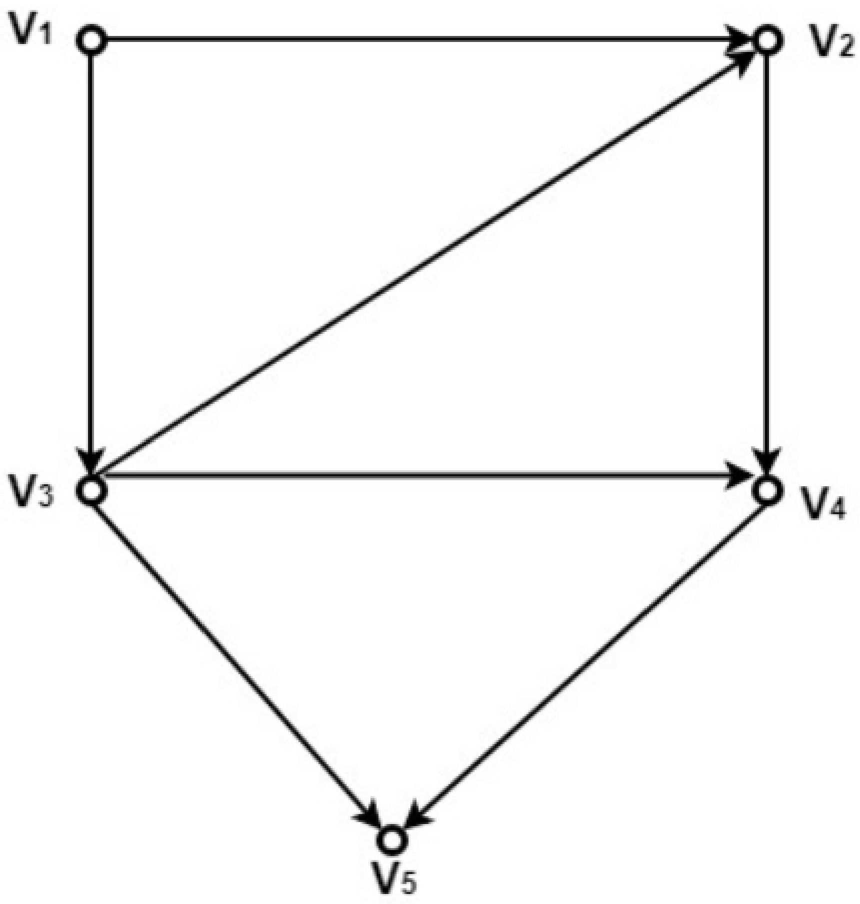
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 3 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Solution :



Example : Find the adjacency matrix of a graph G.





Matrix representations of graphs

Solution :

$$M_A = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 1 & 0 & 1 \\ D & 1 & 1 & 1 & 0 \end{array}$$

$$M_A = \begin{array}{c|ccccc} & V_1 & V_2 & V_3 & V_4 & V_5 \\ \hline V_1 & 0 & 1 & 1 & 0 & 0 \\ V_2 & 0 & 0 & 0 & 1 & 0 \\ V_3 & 0 & 1 & 0 & 1 & 1 \\ V_4 & 0 & 0 & 0 & 0 & 1 \\ V_5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Matrix representations of graphs

incidence matrix

Definition : If an Undirected Graph G consists of n vertices and m edges, then the incidence matrix is an n x m matrix C = $[c_{ij}]$ and defined by

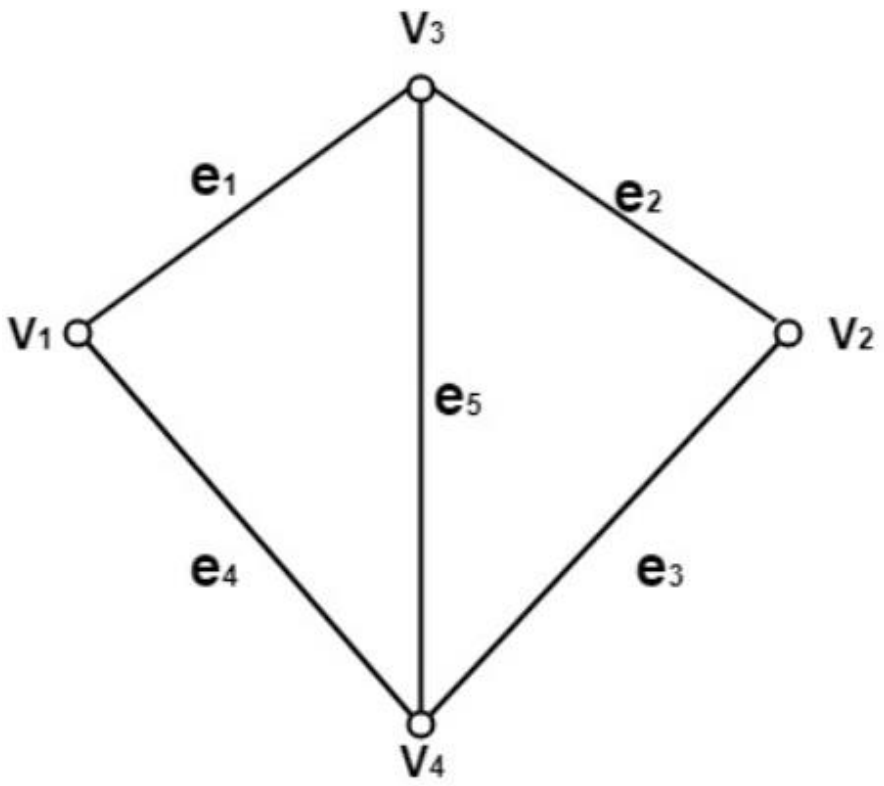
C_{ij}= 1 if vertex v_i incident by edge e_j
= 0 other wise

Definition : If a directed graph G consists of n vertices and m edges, then the incidence matrix is an n x m matrix C = [cij] and defined by

C_{ij}= 1 if vertex v_i is the initial vertex(head or outgoing) of edge e_j
= -1 if vertex v_i is the final vertex(tail or incoming) of edge e_j
= 0 ;other wise

Matrix representations of graphs

Example : Find the incidence matrix of a graph G.



Solution :

$$M_I = \begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline V_1 & 1 & 0 & 0 & 1 & 0 \\ V_2 & 0 & 1 & 1 & 0 & 0 \\ V_3 & 1 & 1 & 0 & 0 & 1 \\ V_4 & 0 & 0 & 1 & 1 & 1 \end{array}$$

Degree sequence

degree sequence:

Definition: Degree sequence of a graph is the list of degree of all the vertices of the graph. Usually we list the degrees in nonincreasing order, that is from largest degree to smallest degree.

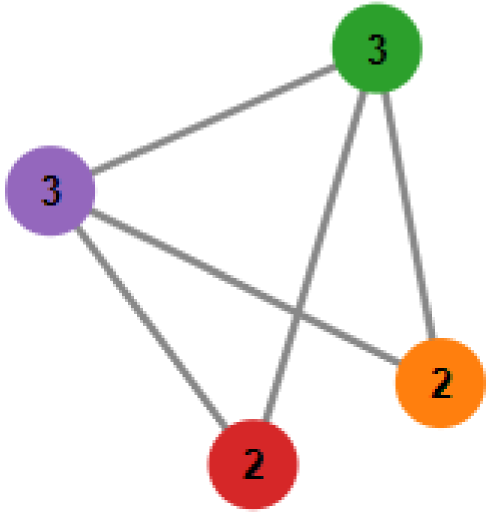
Theorem: The sum of degree of all vertices of a graph is twice the size of graph. Mathematically,

$$\sum deg(v_i) = 2|E|$$

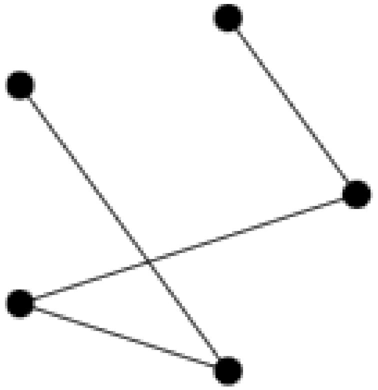
where, |E| stands for the number of edges in the graph (size of graph).

The reasoning behind this result is quite simple. An edge is a link between two vertices. So each edge contributes exactly 2 to the degree sum. And hence, the degree sum must be twice the number of edges.

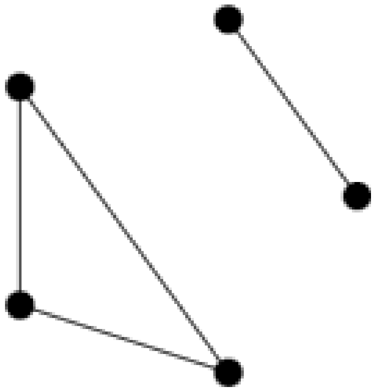
Degree sequence



Degree Sequence=(3,3,2,2)



{2, 2, 2, 1, 1}



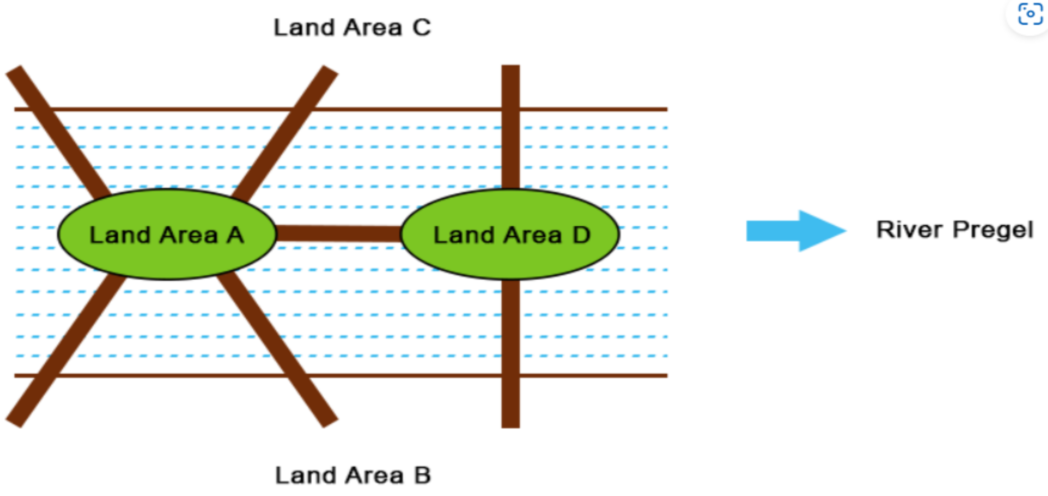
{2, 2, 2, 1, 1}

MODULE - II GRAPH ROUTES

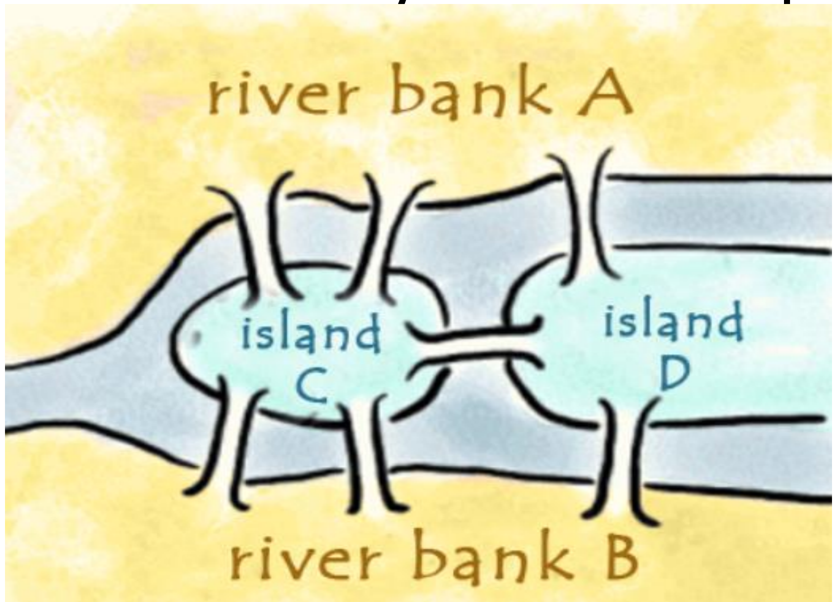
Eulerian circuit: Konigsberg bridge problem, touring a graph; Eulerian graphs, Hamiltonian cycles, the traveling salesman problem; Shortest paths: Dijkstra’s algorithm, walks using matrices

Konigsberg Bridge Problem

- The Konigsberg is the name of the German city, but this city is now in Russia.
- In the below image, we can see the inner city of Konigsberg with the river Pregel.
- There are a total of four land areas in which this river Pregel is divided, i.e., A, B, C and D.
- There are total 7 bridges to travel from one part of the city to another part of the city.



Konigsberg Bridge Problem



Konigsberg Bridge problem

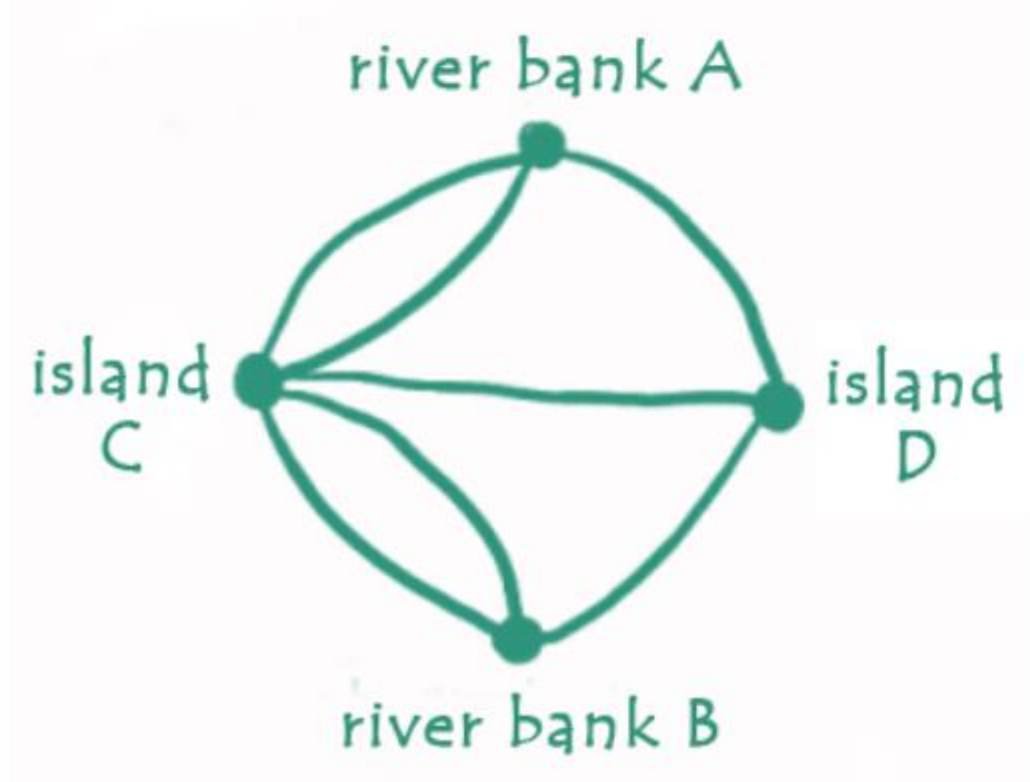
The Konigsberg Bridge contains the following problem which says:

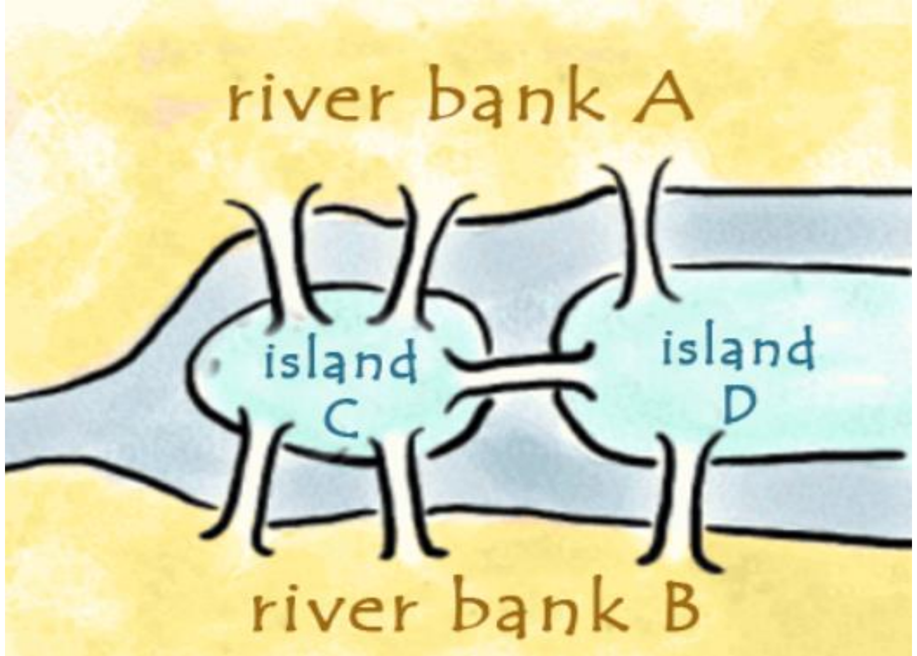
Is it possible for anyone to cross each of the seven bridges only a single time and come back to the beginning point without swimming across the river if we begin this process from any of the four land areas that are A, B, C, and D?.

Solution of Konigsberg Bridge problem

In 1735, this problem was solved by Swiss mathematician Leon hard Euler.
According to the solution to this problem, these types of walks are not possible.
With the help of following graph, Euler shows the given solution.

Konigsberg Bridge Problem





- The vertices of this graph are used to show the landmasses.
- The edges are used to show the bridges.

Konigsberg Bridge Problem

- The problem of Konigsberg was to travel around the town by crossing each bridge only a single time.
- On Euler's network, it meant that all the vertices had to be visited and traced over each arc only a single time.
- Since there is a total of 4 odd vertices in the bridge, so it would not be possible to do.

Konigsberg Bridge Problem

So Euler observed that at the time of tracing the graph, if they try to visit a vertex, then the following things happen:

- The graph will contain an edge that enters into the vertex.
- It will contain one more edge that leaves the vertex.
- Therefore, there must be an even number in the order of vertex.

On the basis of the above observation, Euler also discovers one thing about a network, i.e., with the help of a number of odd vertices which exists in the network is used to determine whether any network is traversable or not.

- Since there are total 4 odd vertices in the Konigsberg network, therefore Euler concluded that the network is not traversable.
- Thus, Euler finally concluded that it is not possible to traverse the desired walking tour of Konigsberg.

Touring a graph

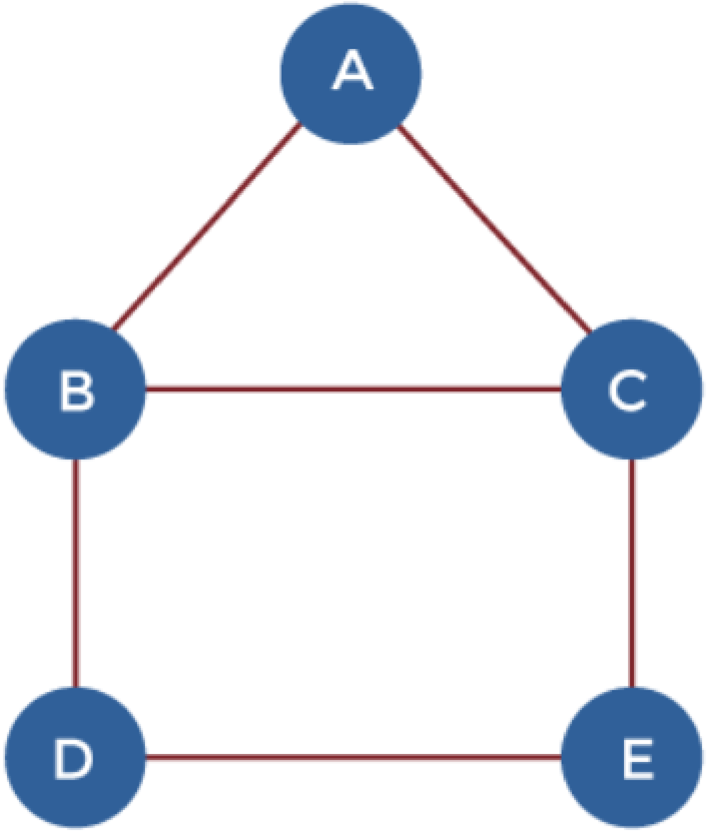
Walk:

Definition: A walk is a sequence of vertices so that there is an edge between consecutive vertices. A walk can repeat vertices and edges.

- When we have a graph and traverse it, then that traverse will be known as a walk.
- The number of edges which is covered in a walk will be known as the Length of the walk. In a graph, there can be more than one walk.

Touring a graph

example



In this graph, there can be many walks, but some of them are described as follows:

1. A, B, C, E, D (Number of length = 4)
2. E, C, B, A, C, E, D (Number of length = 6)

Touring a graph

There are two types of the walk, which are described as follows:

1. Open walk
2. Closed walk

Open Walk:

A walk will be known as an open walk in the graph theory if the vertices at which the walk starts and ends are different. That means for an open walk, the starting vertex and ending vertex must be different. In an open walk, the length of the walk must be more than 0.

Closed Walk:

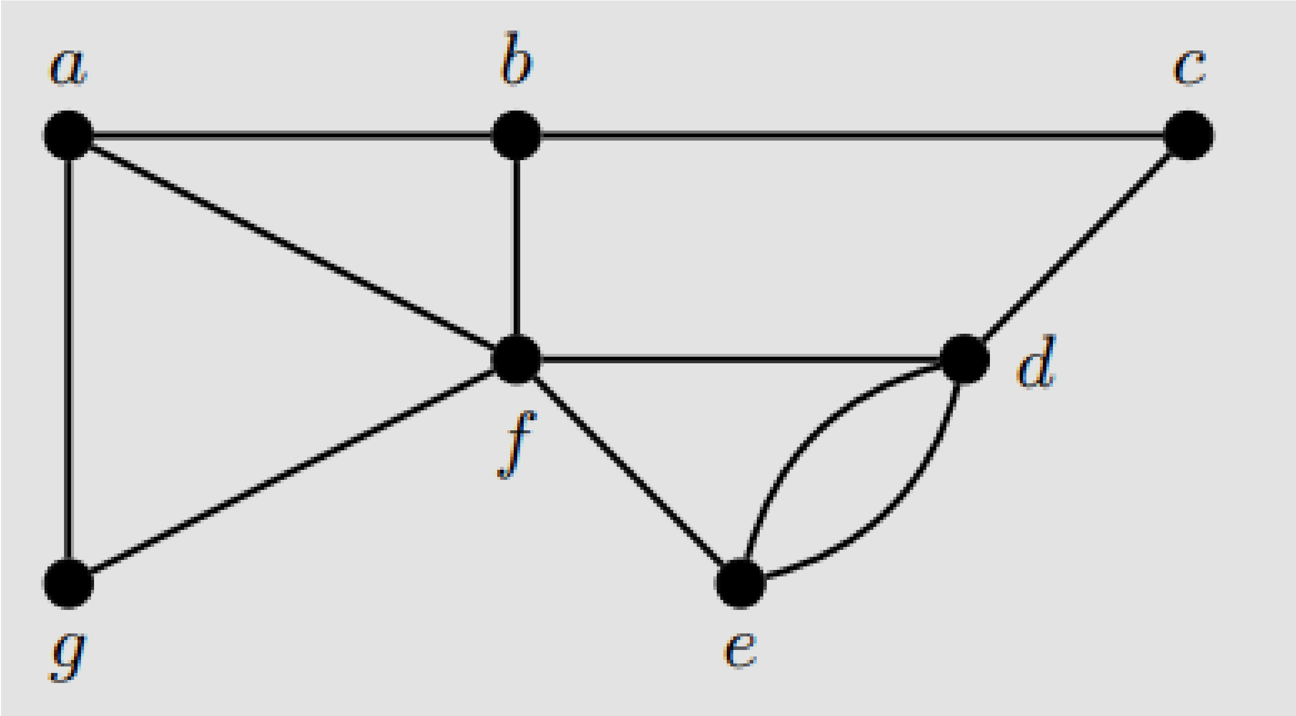
A walk will be known as a closed walk in the graph theory if the vertices at which the walk starts and ends are identical. That means for a closed walk, the starting vertex and ending vertex must be the same. In a closed walk, the length of the walk must be more than 0.

Touring a graph

- A **trail** is a walk with no repeated edges. A trail can repeat vertices but not edges.
- A **path** is a trail with no repeated vertex. A path on n vertices is denoted Pn.
- A **circuit** is a closed trail; that is, a trail that starts and ends at the same vertex with no repeated edges though vertices may be repeated.
- A **cycle** is a closed path; that is, a path that starts and ends at the same vertex. Thus cycles cannot repeat edges or vertices. Note: we do not consider the starting and ending vertex as being repeated since each vertex is entered and exited exactly once. A cycle on n vertices is denoted Cn.

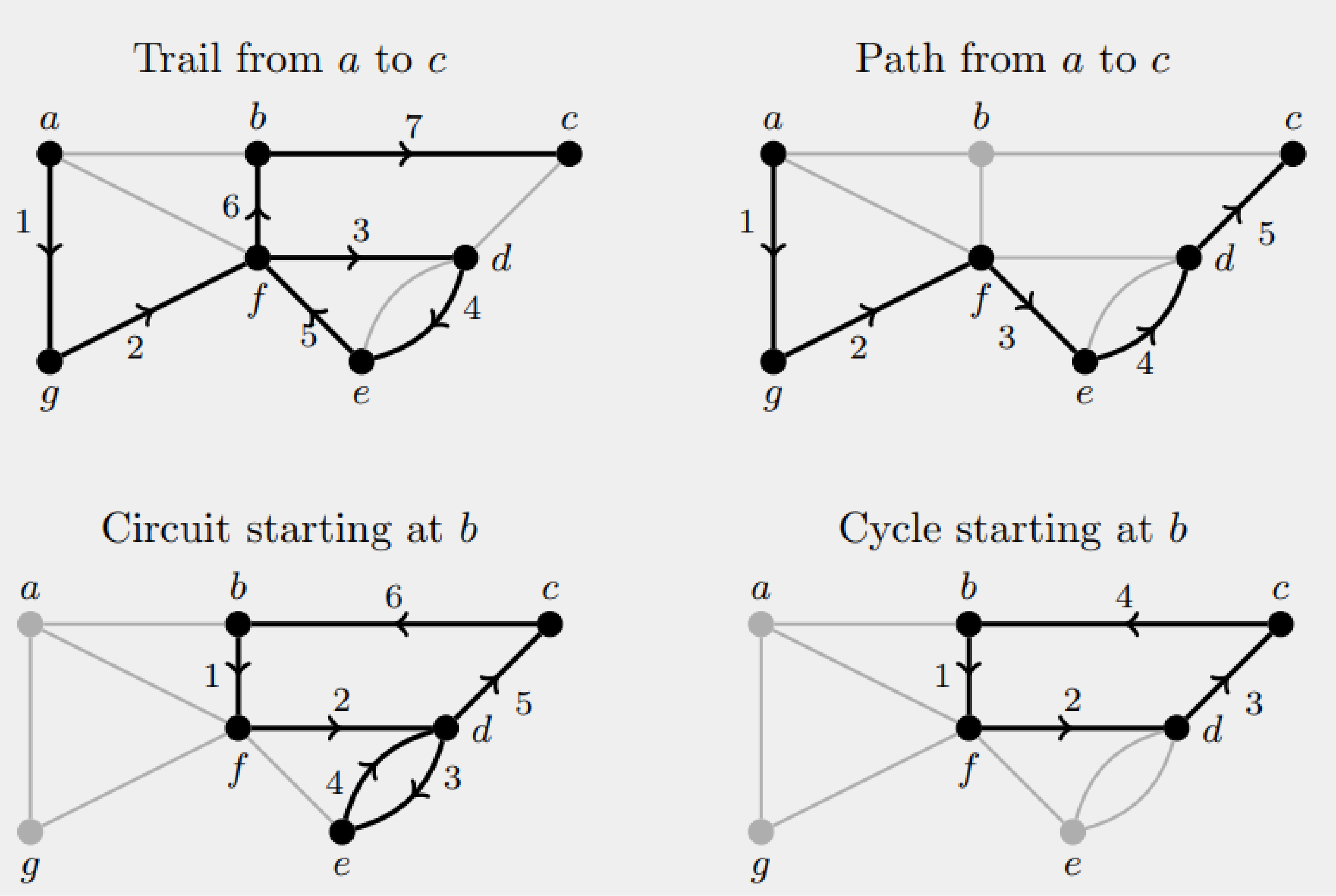
Example :

Given the graph below, find a trail (that is not a path) from a to c, a path from a to c, a circuit (that is not a cycle) starting at b, and a cycle starting at b.



Touring a graph

Solution: The order in which the edges will be traveled are noted in the following routes.



Q. Show that every $x - y$ walk contains an $x - y$ path

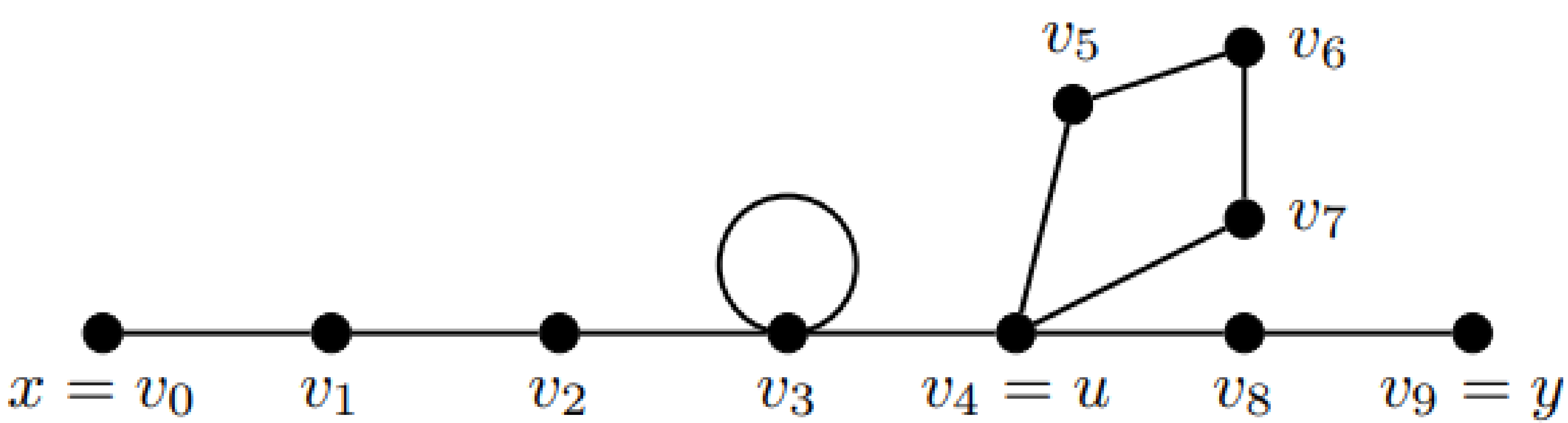
Proof:

- Let be the length l of an $x - y$ walk W. For the base case, if $l = 0$ then the walk does not have any edges and so contains a singe vertex, that is $x = y$ and so W is a path of length 0.
- Next assume $l \geq 1$ and suppose the claim holds for all walks of length $k<l$. If W has no repeated vertex, then it cannot have any repeated edges and so W is an $x-y$ path.
- Otherwise W has a repeated vertex, call it u. We can remove all edges and vertices between two appearances of u in the walk, and leave only one copy of u. This will produce a shorter $x - y$ walk W which is contained in W. Thus, by the induction hypothesis we know W contains an $x - y$ path P, which is also contained in W. Thus W contains an $x - y$ path P.

Touring a graph

Consider a walk as shown below. Using the technique from the proof above, we begin by identifying a repeated vertex, namely v4.

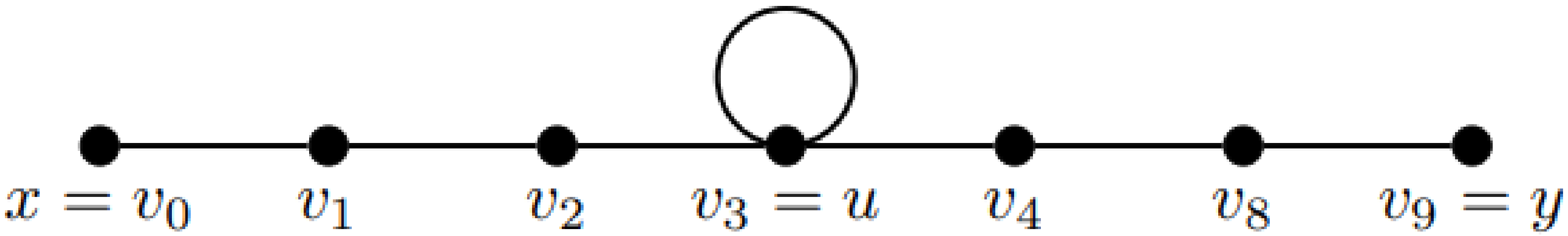
$$W : v_0\ v_1\ v_2\ v_3\ v_3\ \boxed{v_4\ v_5\ v_6\ v_7\ v_4}\ v_8\ v_9$$



Touring a graph

Notice that these selected edges and vertices can be removed without destroying the walk from v0 to v9, as shown below. Also, we find another repeated vertex, namely v3, and can repeat the process.

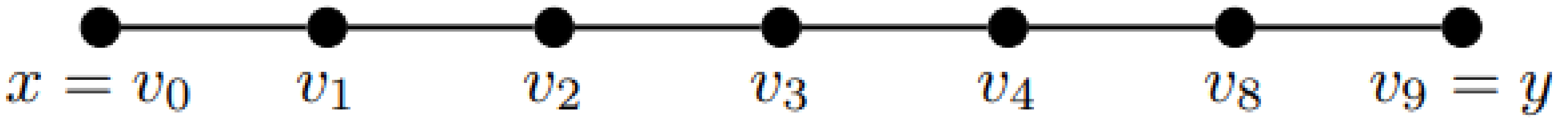
$$W' : v_0\ v_1\ v_2\ \boxed{v_3\ v_3}\ v_4\ v_8\ v_9$$



Touring a graph

The final graph below does not contain any repeated vertices and so can be considered a path from v0 to v9, all of whose vertices and edges were contained in the original walk W.

$$W'' : v_0\ v_1\ v_2\ v_3\ v_4\ v_8\ v_9$$



Eulerian Graph

Definition :Let G be a graph. An eulerian circuit (or trail) is a circuit (or trail) that contains every edge and every vertex of G.

If G contains an eulerian circuit it is called eulerian and if G contains an eulerian trail but not an eulerian circuit it is called semi-eulerian

A graph G is eulerian if and only if

(i) G is connected and

(ii) every vertex has even degree

A graph G is semi-eulerian if and only if

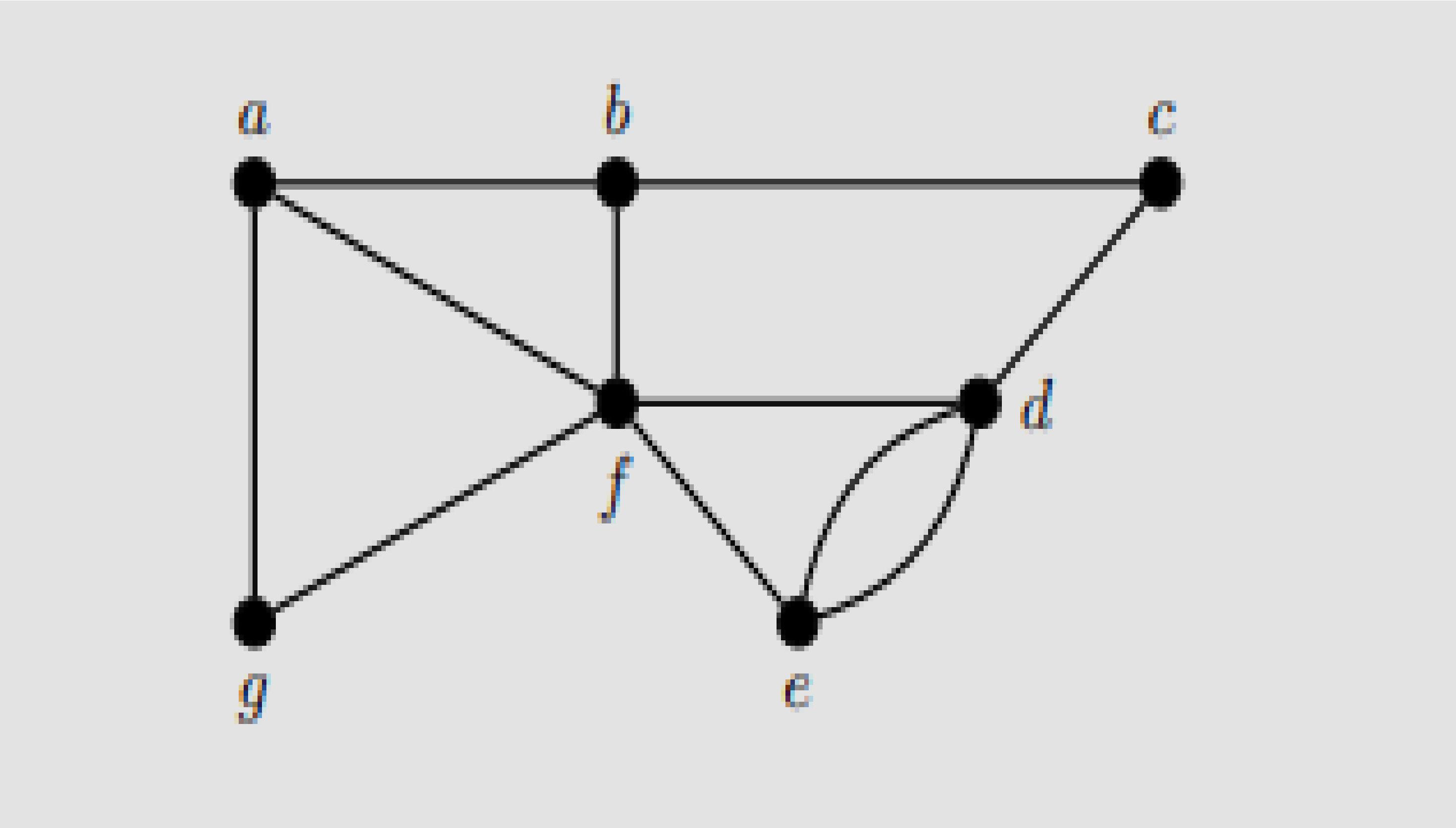
(i) G is connected and

(ii) exactly two vertices have odd degree.

Don’t write or place any image in this area

Eulerian Graph

Consider the graphs ,Which ones are eulerian? semi-eulerian?

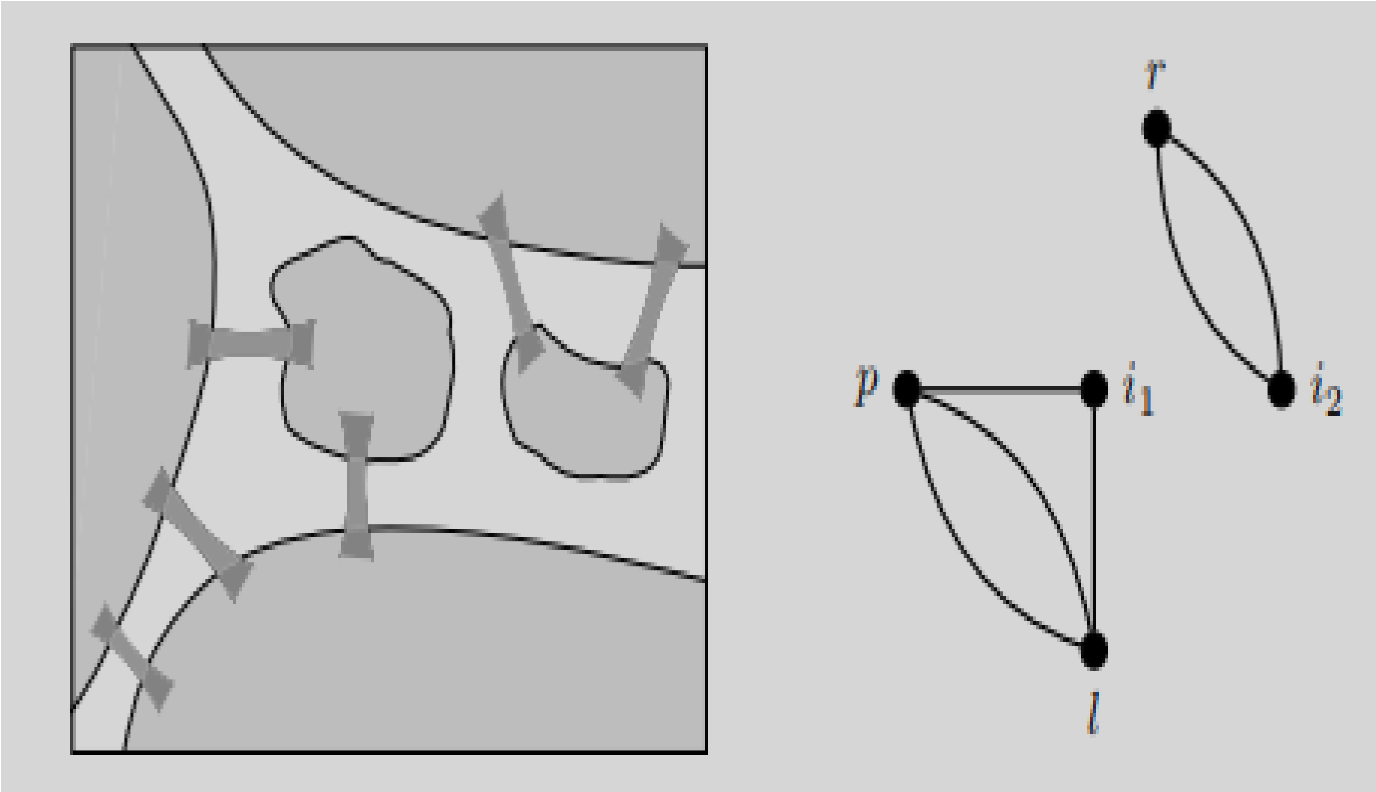


It is neither eulerian nor semi-eulerian since it has more than two odd vertices (namely, a, b, e, and f).

Don’t write or place any image in this area

Eulerian Graph

Consider the graphs ,Which ones are eulerian? semi-eulerian?

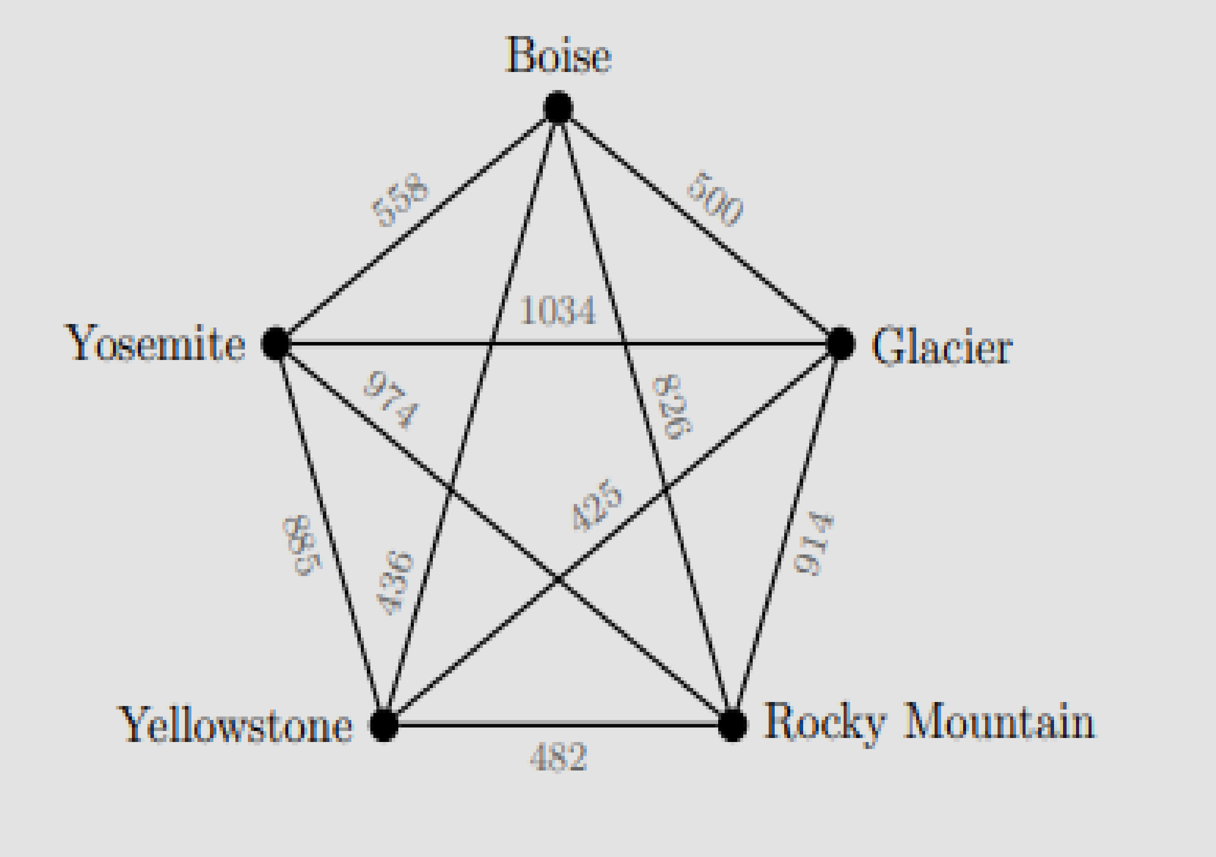


The graph representing Island City in Example 2.2 is not connected, so it is neither eulerian nor semi-eulerian.

Don’t write or place any image in this area

Eulerian Graph

Consider the graphs ,Which ones are eulerian? semi-eulerian?

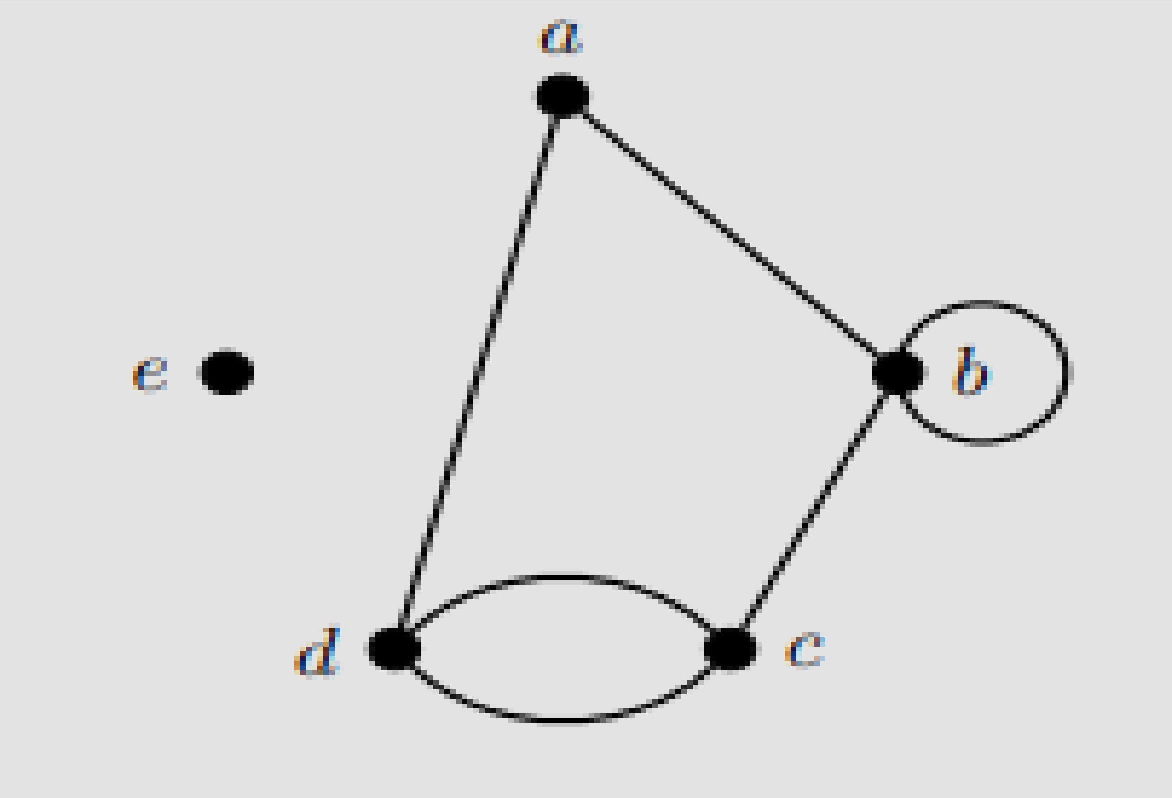


The graph in Example 1.7 is eulerian since it is connected and all the vertices have degree 4.

Don’t write or place any image in this area

Eulerian Graph

Consider the graphs ,Which ones are eulerian? semi-eulerian?

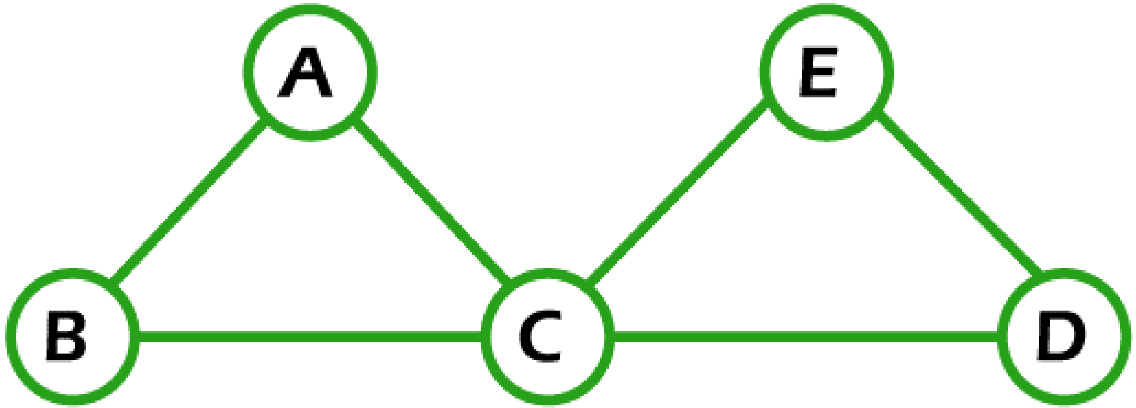


The graph in is neither eulerian nor semi-eulerian since it is not connected

Don’t write or place any image in this area

Euler graph

If all the vertices of any connected graph have an even degree, then this type of graph will be known as the Euler graph. In other words, we can say that an Euler graph is a type of connected graph which have the Euler circuit. The simple example of Euler graph is described as follows:



Example of Euler Graph

Don't write or place any image in this area

The above graph is a connected graph, and the vertices of this graph contain the even degree. Hence we can say that this graph is an Euler graph.
In other words, we can say that this graph is an Euler graph because it has the Euler circuit as BACEDCB.

Euler Path

We can also call the Euler path as Euler walk or Euler Trail. The definition of Euler trail and Euler walk is described as follows:

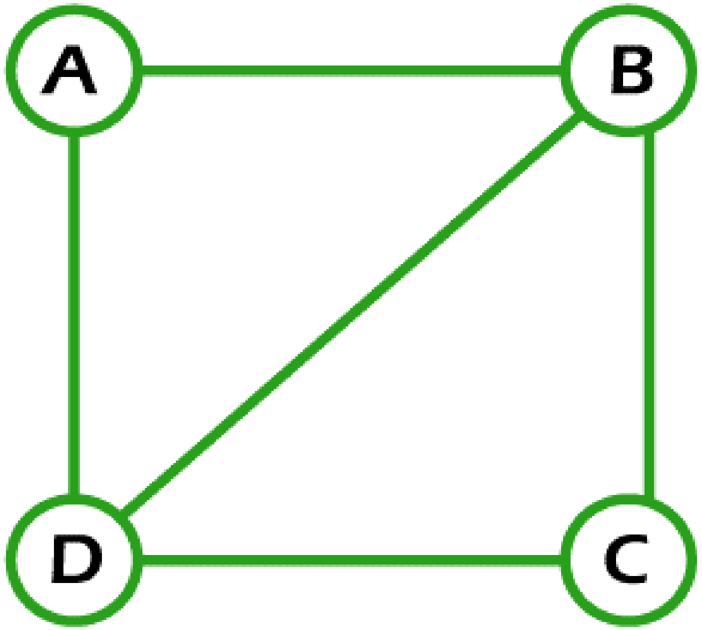
- If there is a connected graph with a trail that has all the edges of the graph, then that type of trail will be known as the Euler trail.

- If there is a connected graph, which has a walk that passes through each and every edge of the graph only once, then that type of walk will be known as the Euler walk or path.

Examples of Euler path:

There are a lot of examples of the Euler path, and some of them are described as follows:

Example 1: In the following image, we have a graph with 4 nodes. Now we have to determine whether this graph contains an Euler path.

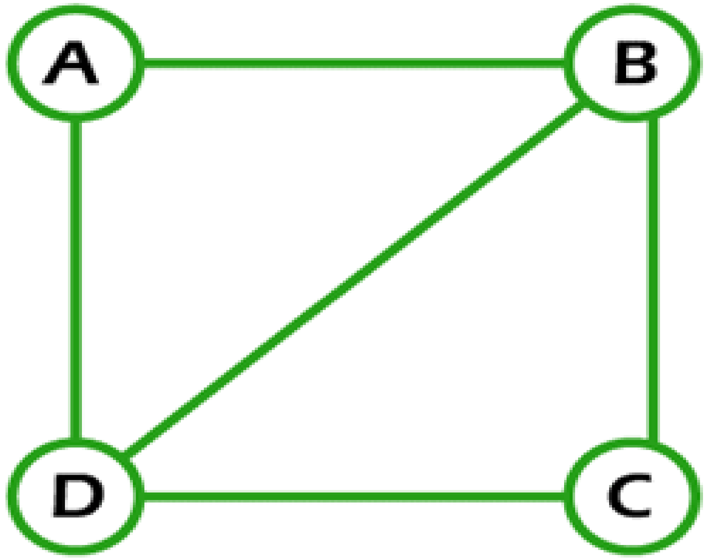


Don't write or place any image in this area

Solution:

The above graph will contain the Euler path if each edge of this graph must be visited exactly once, and the vertex of this can be repeated. So if we begin our path from vertex B and then go to vertices C, D, B, A, and D, then in this process, each and every edge is visited exactly once, and it also contains repeated vertex. So the above graph contains an Euler path, which is described as follows:

Euler path = BCDBAD

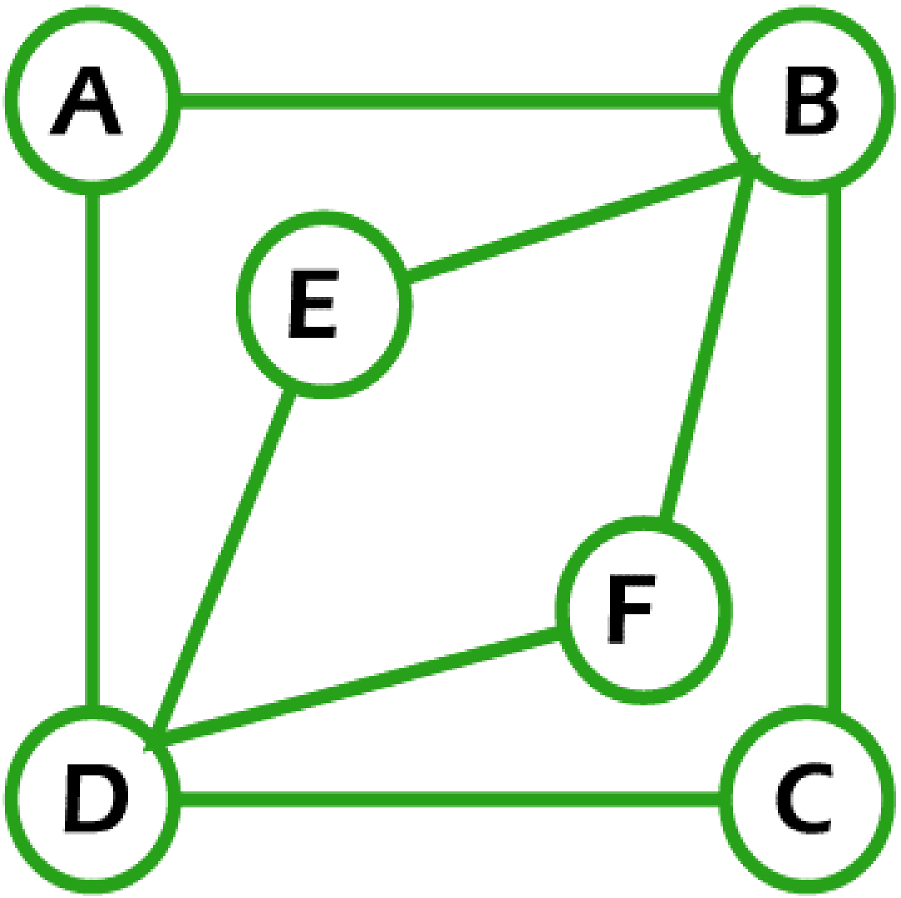


Example 2: In the following image, we have a graph with 6 nodes. Now we have to determine whether this graph contains an Euler path.

Solution:

Graph will contain the Euler path if each edge of this graph must be visited exactly once, and the vertex of this can be repeated. So if we begin our path from vertex B and then go to vertices C, D, F, B, E, D, A, and B, then in this process, each and every edge is visited exactly once, and it also contains repeated vertex. So the above graph contains an Euler path, which is described as follows:

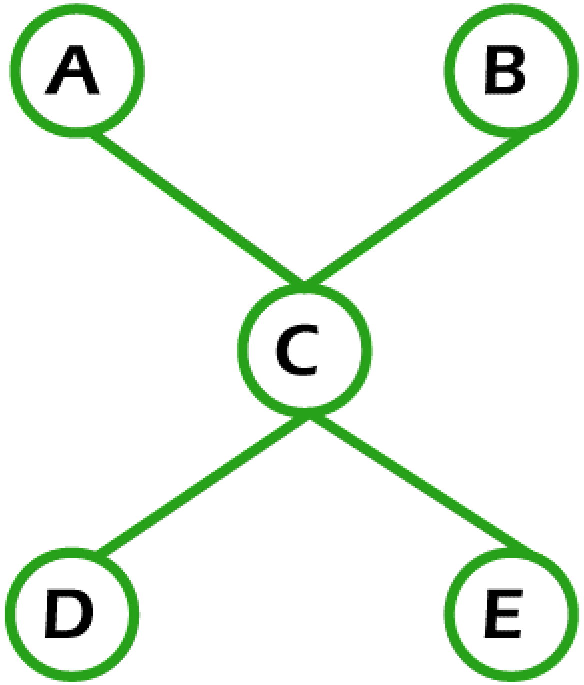
Euler path = BCDFBEDAB



any

a

Example 3: In the following image, we have a graph with 5 nodes. Now we have to determine whether this graph contains an Euler path.



Euler Circuit:

There are various definitions of the Euler circuit, which are described as follows:

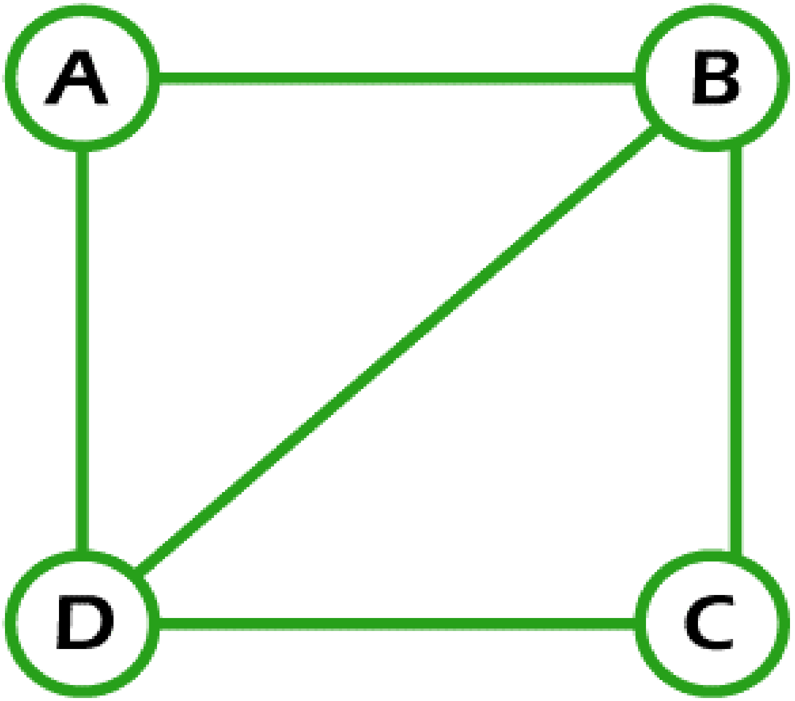
- If there is a connected graph with a circuit that has all the edges of the graph, then that type of circuit will be known as the Euler circuit.

- If there is a connected graph, which has a walk that passes through each and every edge of the graph only once, then that type of walk will be known as the Euler circuit. In this walk, the starting vertex and ending vertex must be the same, and this walk can contain the repeated vertex, but it is not compulsory.

- If an Euler trail contains the same vertex at the start and end of the trail, then that type of trail will be known as the Euler Circuit.

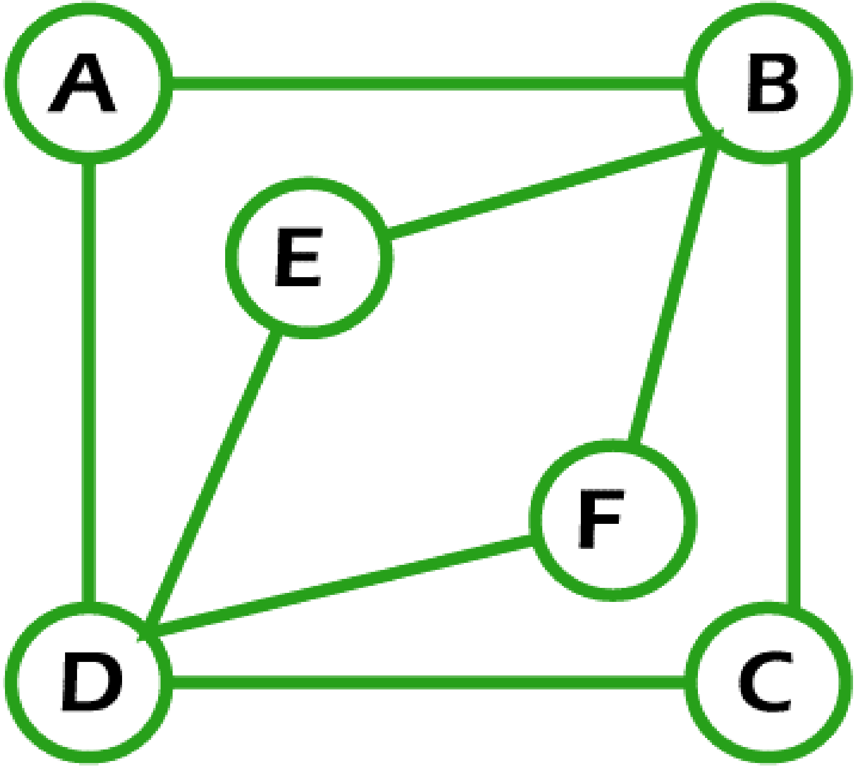
- A closed Euler trail will be known as the Euler Circuit.

Example 1: In the following image, we have a graph with 4 nodes. Now we have to determine whether this graph contains an Euler circuit.



Don’t write or place any image in this area

Example 2: In the following image, we have a graph with 6 nodes. Now we have to determine whether this graph contains an Euler circuit.



Euler circuit = ABCDFBEDA

Don’t write or place any image in this area

Fleury’s Algorithm Input

.

Fleury’s Algorithm Input: Connected graph G where zero or two vertices are odd.

Steps: 1. Choose a starting vertex, call it v. If G has no odd vertices, then any vertex can be the starting point. If G has exactly two odd vertices, then v must be one of the odd vertices.

2. Choose an edge incident to v that is unlabeled and label it with the number in which it was chosen, ensuring that the graph consisting of unlabeled edges remains connected.

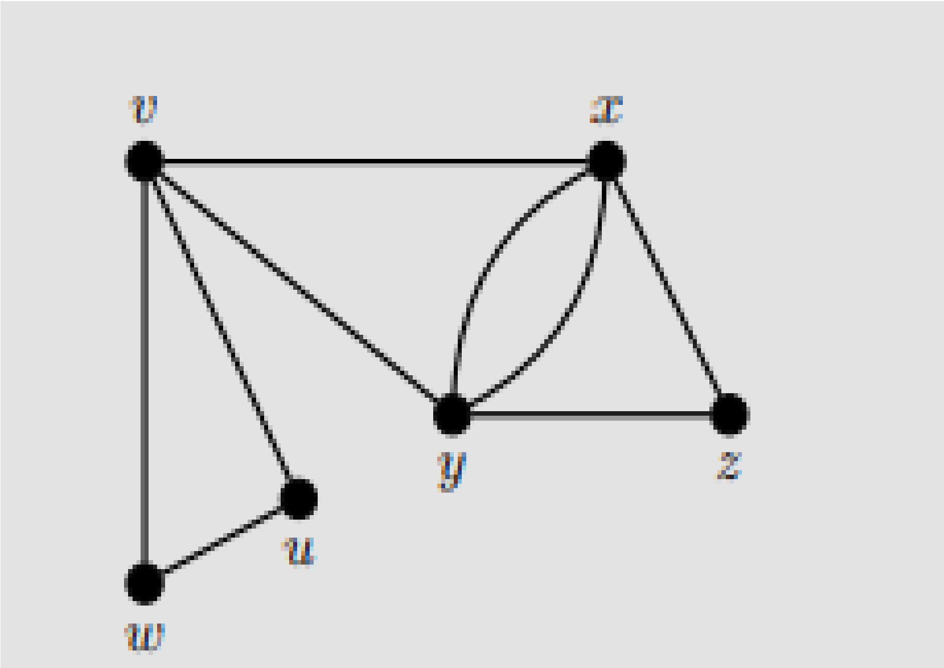
3. Travel along the edge to its other endpoint.

4. Repeat Steps (2) and (3) until all edges have been labeled.

Output: Labeled eulerian circuit or trail

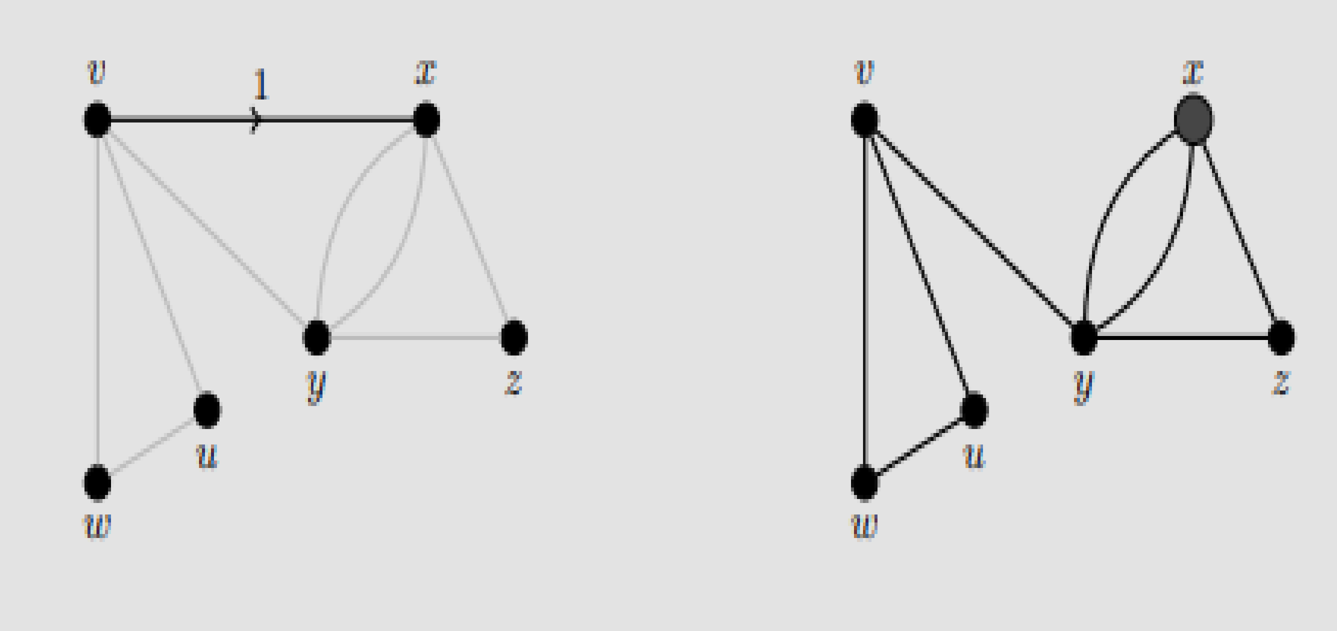
Example 2.4 Input: A connected graph (shown below) where every vertex has even degree. We are looking for an eulerian circuit.

Step 1: Since no starting vertex is explicitly stated, we choose vertex v to be the starting vertex

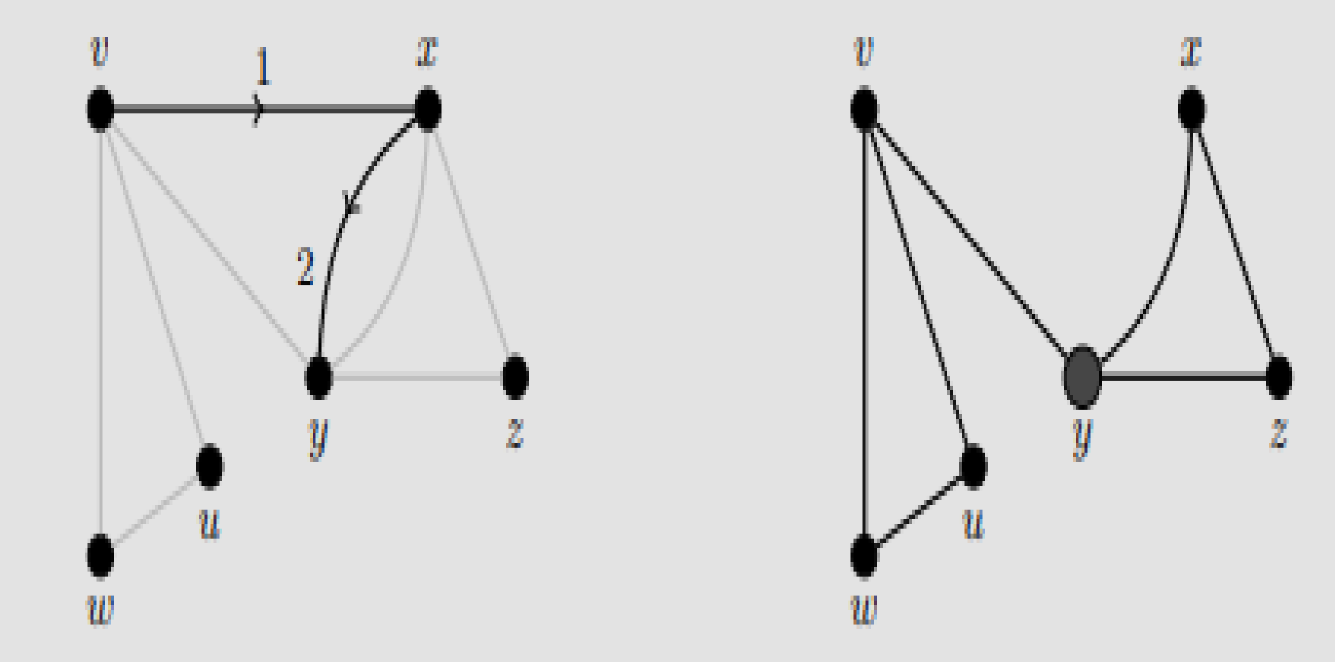


Don’t write or place any image in this area

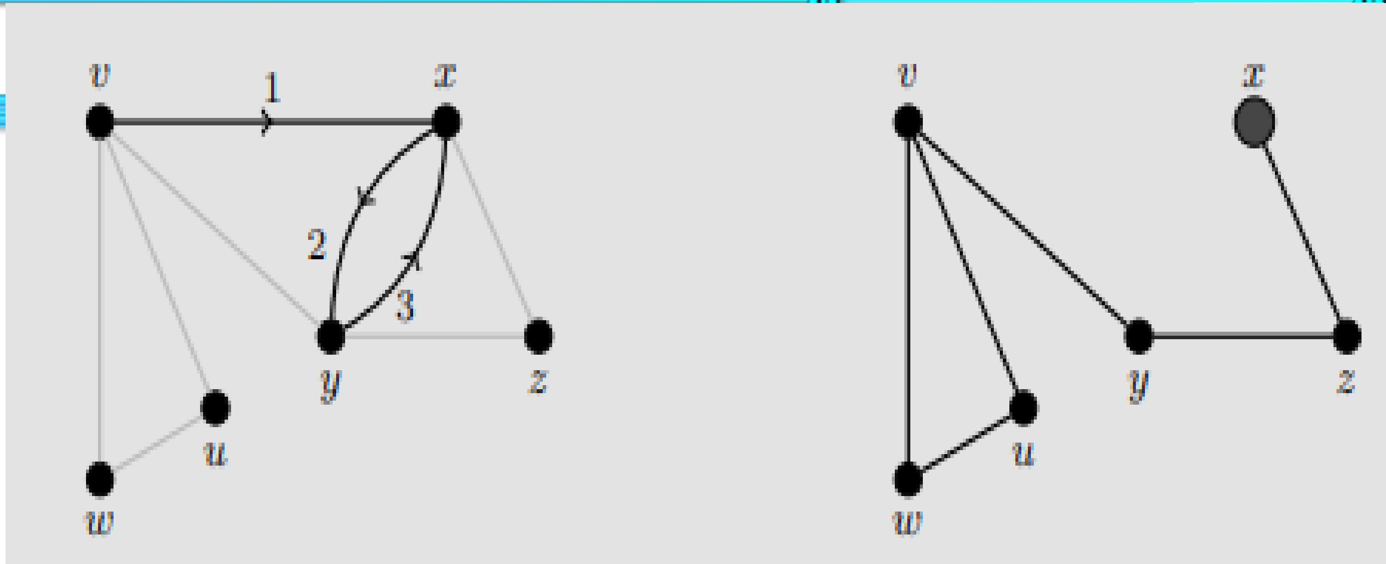
Step 2: We can choose any edge incident to v. Here we chose vx. The labeled graph is on the left and the unlabeled portions are shown on the right with edges removed that have already been chosen.



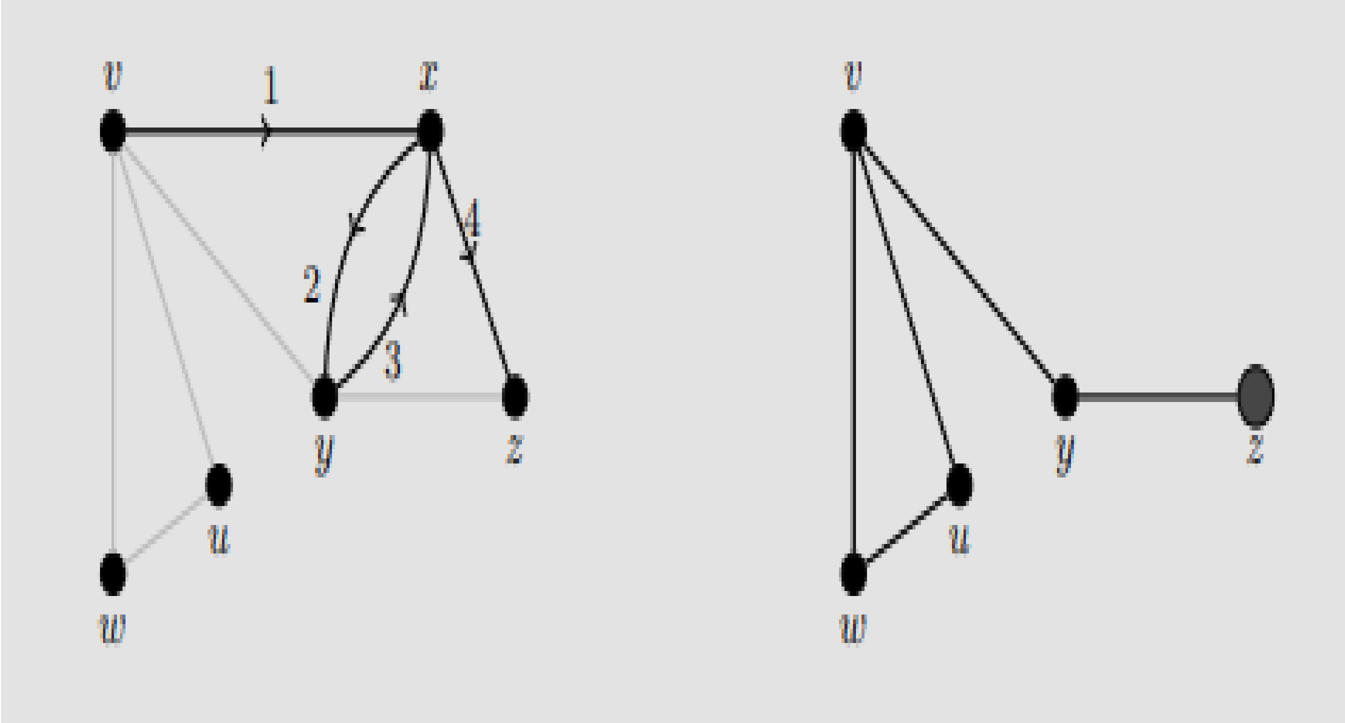
Step 3: Looking at the graph to the right, we can choose any edge out of x. Here we chose xy. The labeled and unlabeled graphs have been updated below.



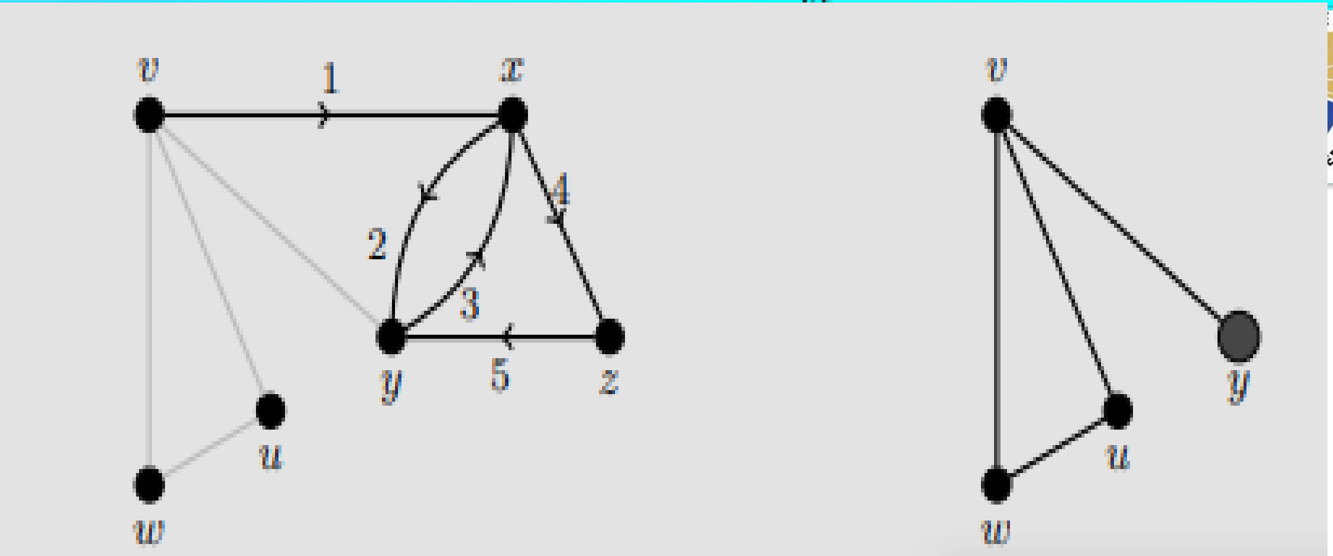
Step 4: At this point we cannot choose yv, as its removal would disconnect the unlabeled graph shown on the right in Step 3. However, yx and yz are both valid choices. Here we chose yx



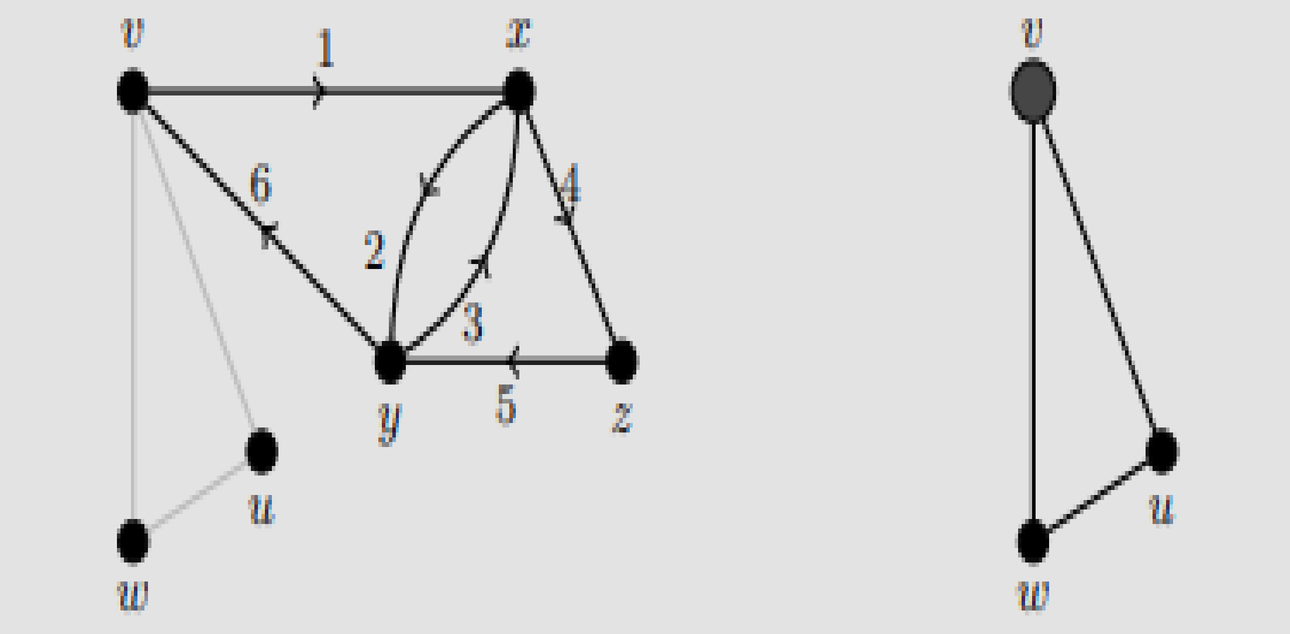
Step 5: There is only one available edge xz.



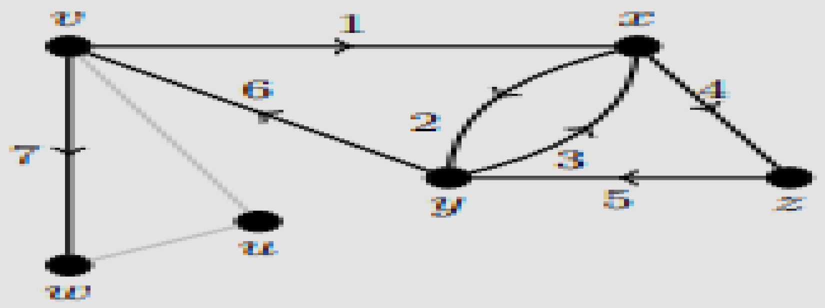
Step 6: There is only one available edge zy.

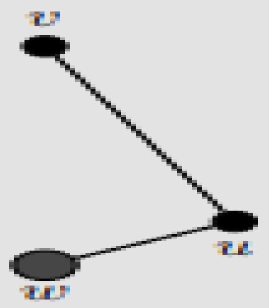


Step 7: There is only one available edge yv

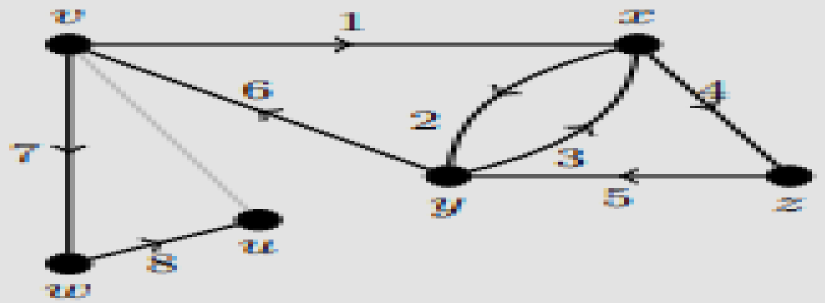


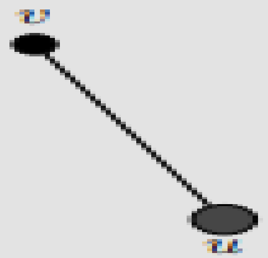
Step 8: Both vw and vu are valid choices for the next edge. Here we chose vw.



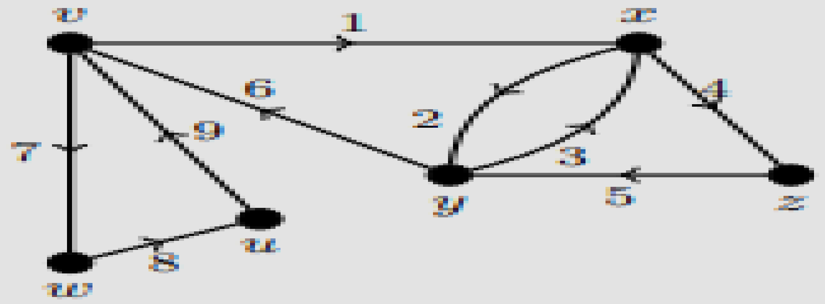


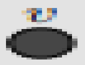
Step 9: There is only one available edge wu.





Step 10: There is only one available edge uv.





Hierholzer's Algorithm

- Hierholzer's Algorithm, is named for the German mathematician mentioned earlier whose paper inspired the procedure to follow.
- This efficient algorithm begins by finding an arbitrary circuit originating from the starting vertex.
- If this circuit contains all the edges of the graph, then an eulerian circuit has been found.
- If not, then we join another circuit to the existing one.

Hierholzer’s Algorithm

- Hierholzer’s Algorithm Input: Connected graph G where all vertices are even.
- Steps:
- 1. Choose a starting vertex, call it v. Find a circuit C originating at v.
- 2. If any vertex x on C has edges not appearing in C, find a circuit C 0 originating at x that uses two of these edges.
- 3. Combine C and C 0 into a single circuit C ∗ .
- 4. Repeat Steps (2) and (3) until all edges of G are used. Output: Labeled eulerian circuit.

ny

Hierholzer's Algorithm

•Hierholzer's Algorithm Input: Connected graph G where all vertices are even.

•Steps: 1. Choose a starting vertex, call it v. Find a circuit C originating at v.

•2. If any vertex x on C has edges not appearing in C, find a circuit C 0 originating at x that uses two of these edges.

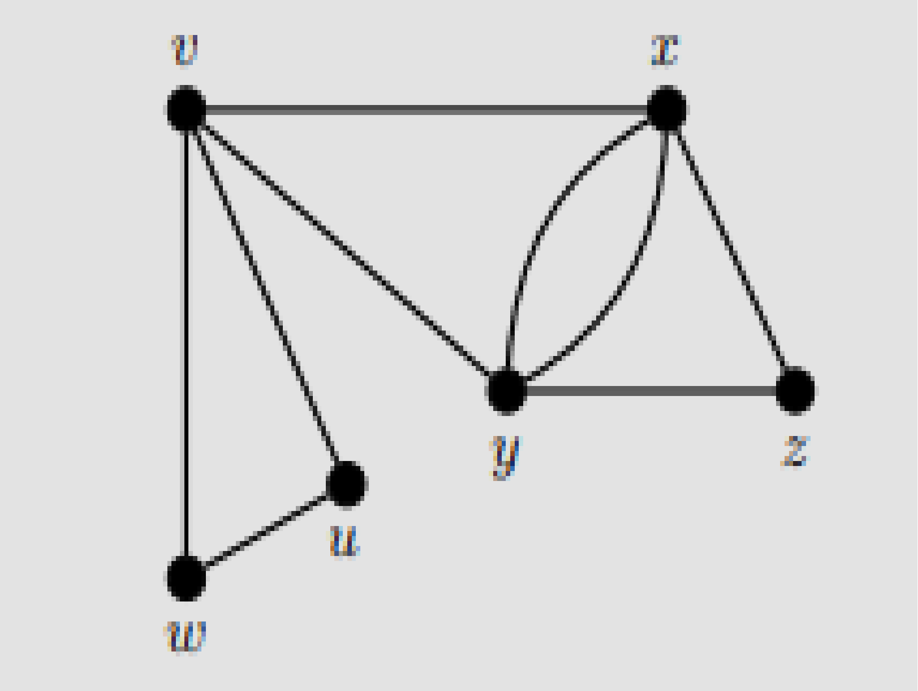
•3. Combine C and C 0 into a single circuit C ∗ .

•4. Repeat Steps (2) and (3) until all edges of G are used. Output: Labeled eulerian circuit.

Note that Hierholzer's Algorithm requires the graph to be eulerian, whereas Fleury's Algorithm allows for the graph to be eulerian or semieulerian. In the implementation of Hierholzer's Algorithm shown below, a new circuit will be highlighted in bold with other edges in gray. As with Fleury's Algorithm, the edges will be labeled in the order in which they are traveled.

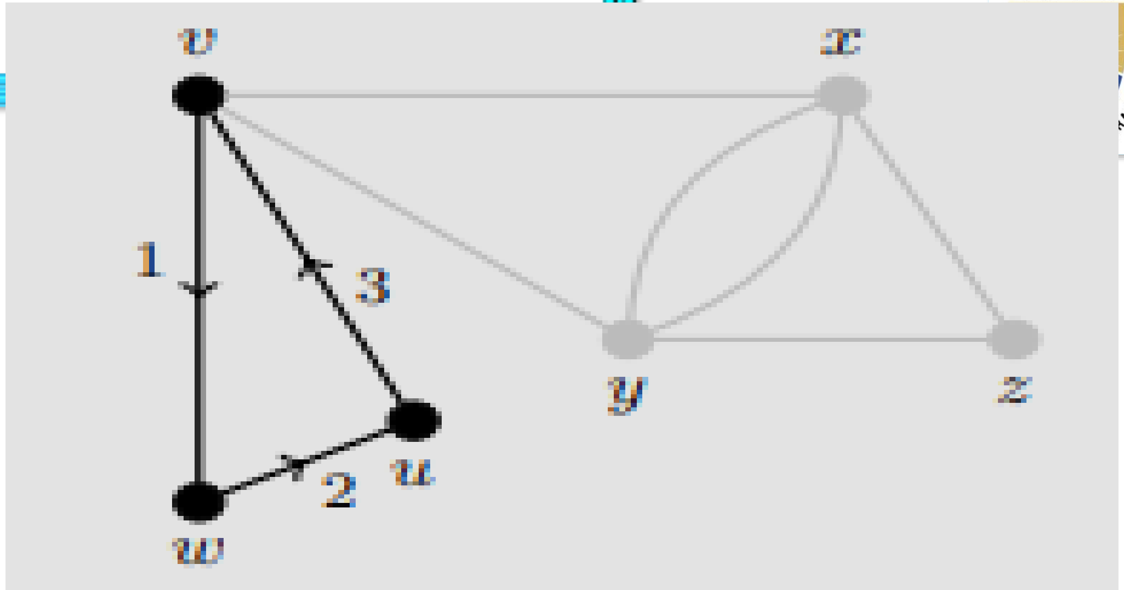
Example 2.5 Input: A connected graph where every vertex has even degree.

We are looking for an eulerian circuit.

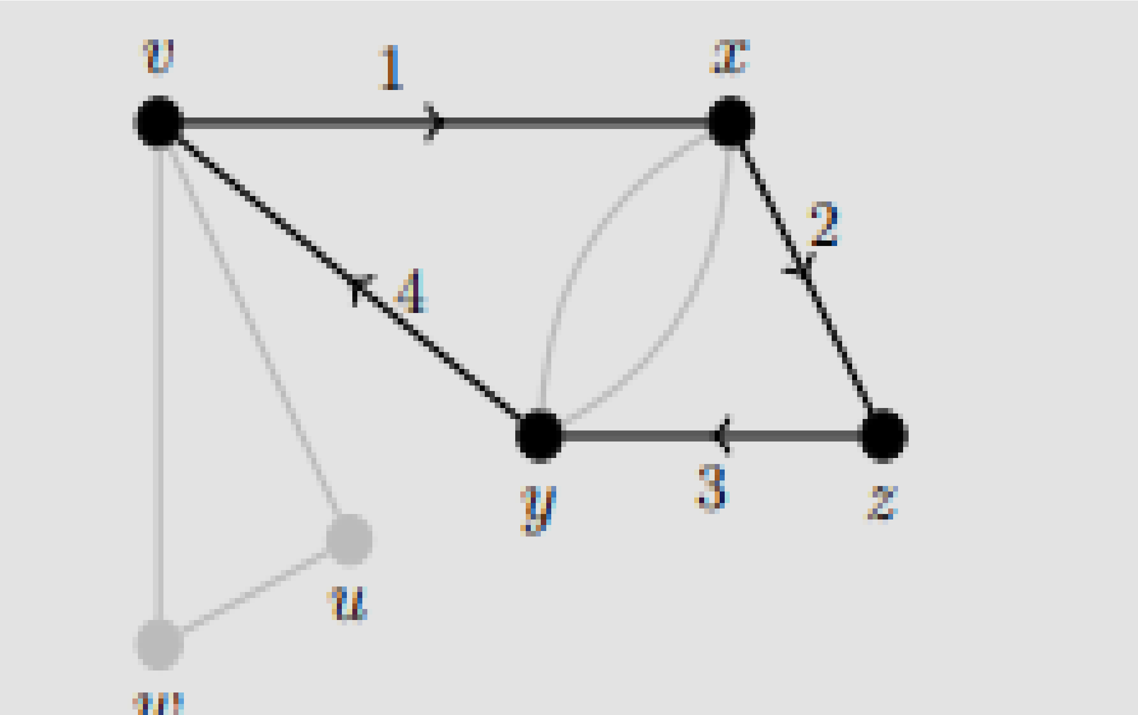


Don’t write or place any image in this area

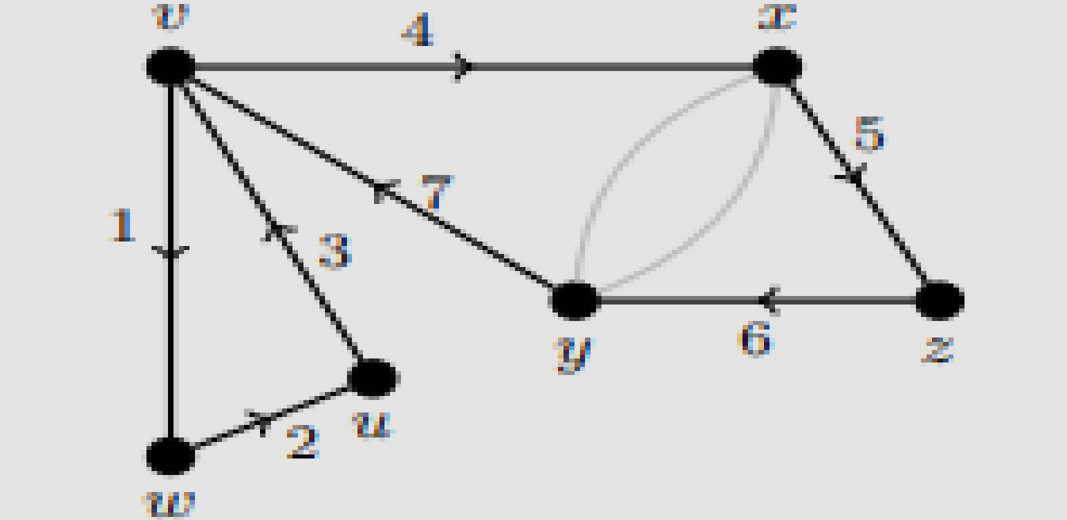
Step 1: Since no starting vertex is explicitly stated, we choose v and find a circuit originating at v. One such option is highlighted below.

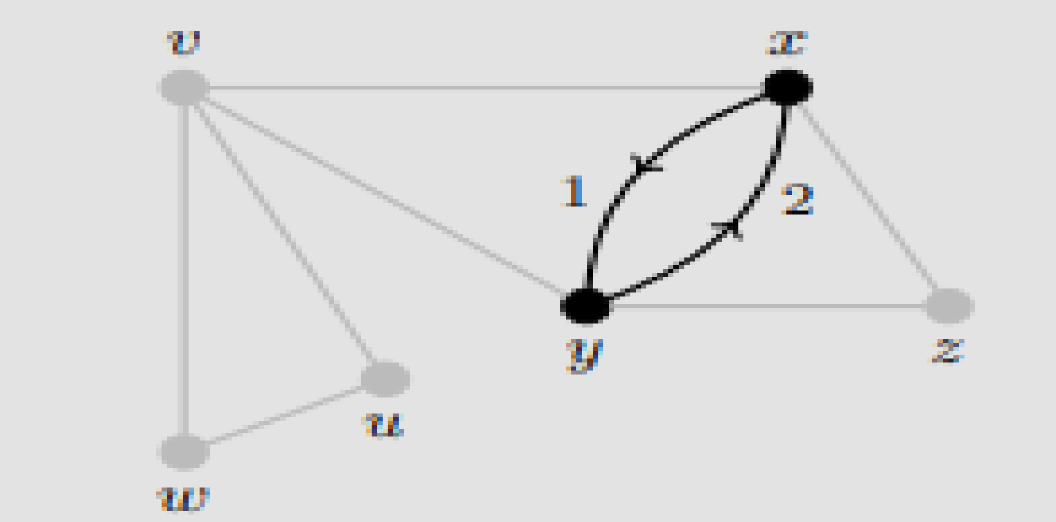


Step 2: As deg(v) = 4 and two edges remain for v (shown in gray in the previous figure), a second circuit starting at v is needed. One option is shown below.

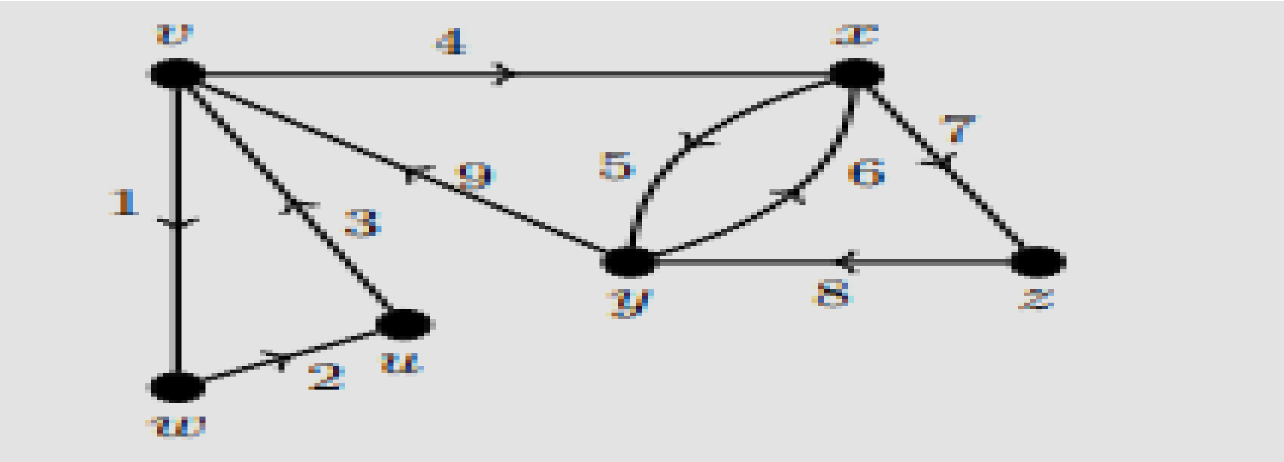


Step 3: Combine the two circuits from Step 1 and Step 2. There are multiple ways to combine two circuits, but it is customary to travel the first circuit created and then travel the second





Step 4: As deg(x) = 4 and two edges remain for x (shown in gray above), a circuit starting at x is needed. It is shown below



Don't write or place any image in this area

Step 5: Combine the two circuits from Step 3 and Step 4.

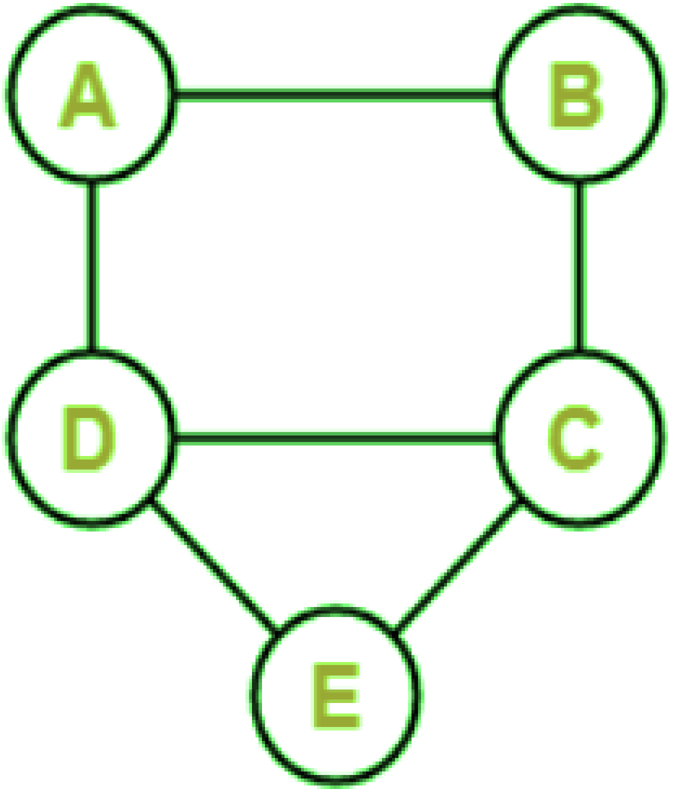
Hamiltonian cycles

Definition :A cycle in a graph G that contains every vertex of G is called a hamiltonian cycle. A path that contains every vertex is called a hamiltonian path. A graph that contains a hamiltonian cycle is called hamiltonian.

- A cycle or a path can only pass through a vertex once, so the hamiltonian cycles and paths travel through every vertex exactly once.

Example 1: In the following graph, we have 5 nodes. Now we have to determine whether this graph contains a Hamiltonian path.

start from A, then we can go to B, C, D, and then E.
So this is the path that contains all the vertices (A, B, C, D, and E) only once, and there is no repeating edge. That's why we can say that this graph has a Hamiltonian path, which is described as follows:
Hamiltonian path = ABCDE
Hamiltonian cycle = ABCEDA



y

Hamiltonian cycles

Properties of Hamiltonian Graphs

(1) G must be connected.

(2) No vertex of G can have degree less than 2.

(3) G cannot contain a ***cut-vertex***, that is a vertex whose removal disconnects the graph.

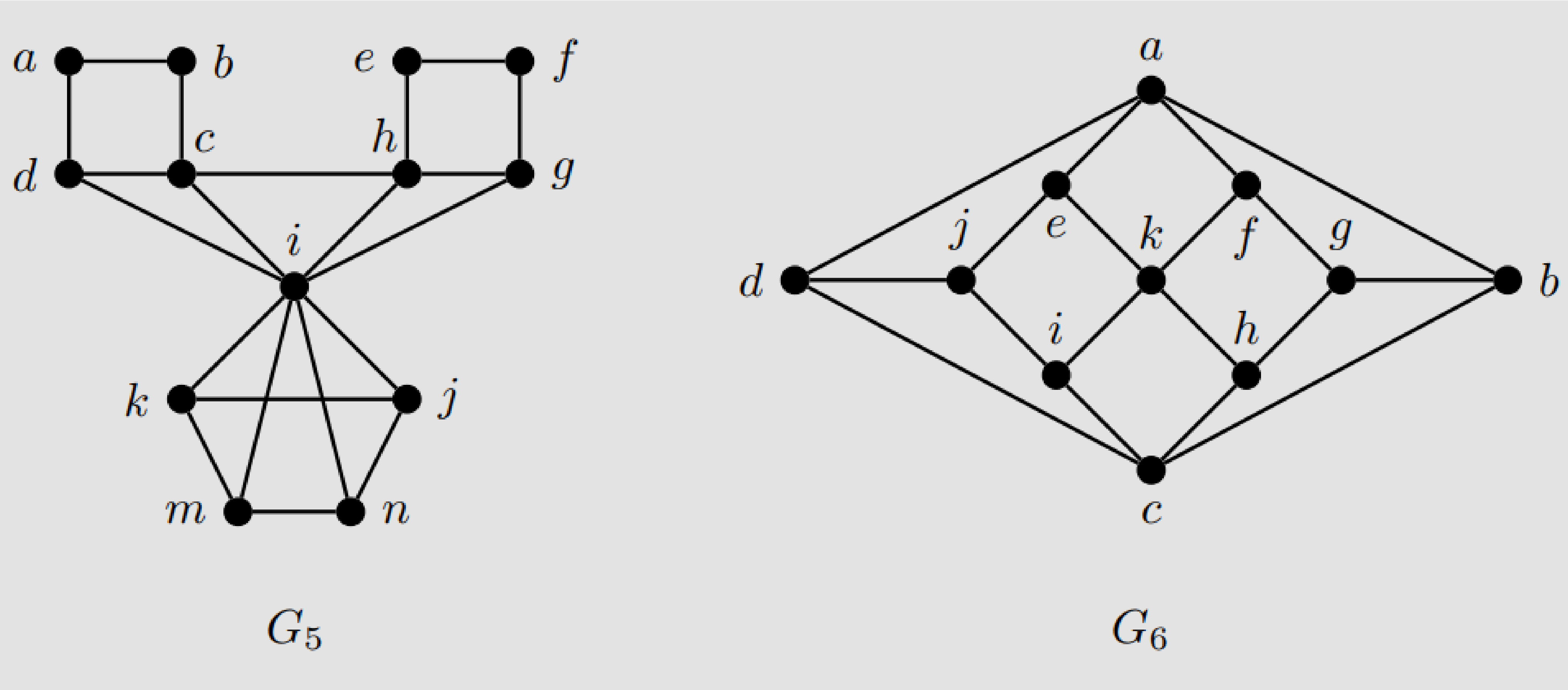
(4) If G contains a vertex x of degree 2 then both edges incident to x must be included in the cycle.

(5) If two edges incident to a vertex x must be included in the cycle, then all other edges incident to x cannot be used in the cycle.

(6) If in the process of attempting to build a hamiltonian cycle, a cycle is formed that does not span G, then G cannot be hamiltonian.

Hamiltonian cycles

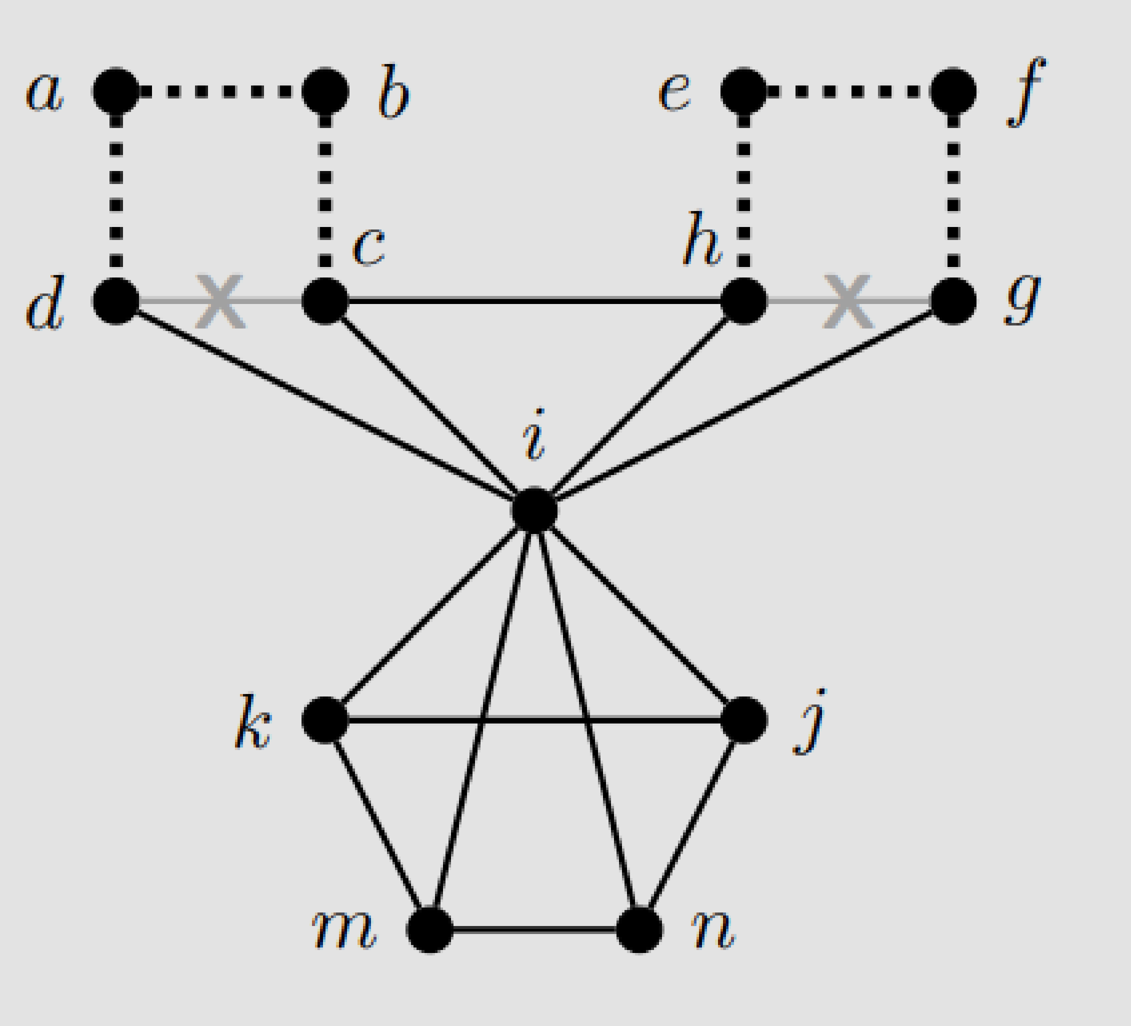
Use the properties listed above to show that the graphs below are not hamiltonian.



Hamiltonian cycles

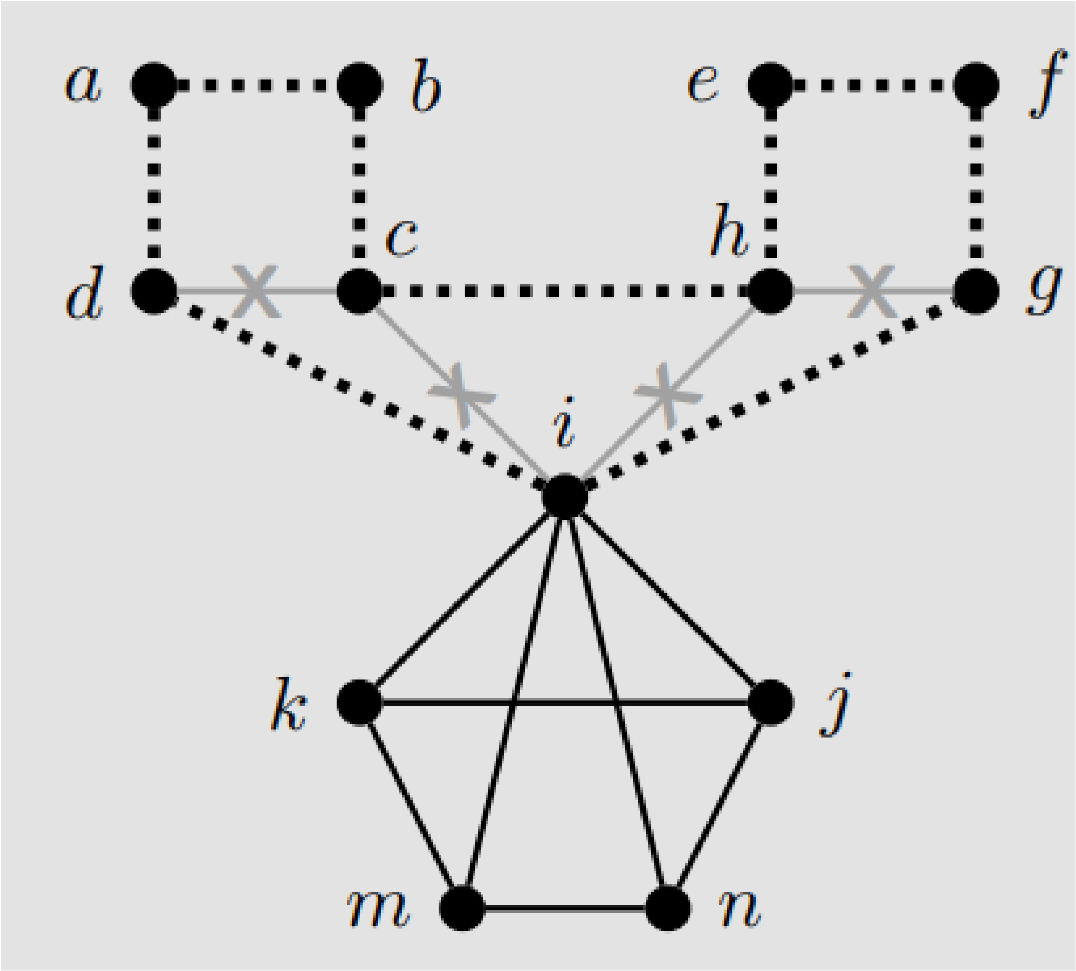
Solution:

- For G5, the vertices a, b, e, and f all have degree 2 and so all the edges incident to these vertices must be included in the cycle.
- But then edges cd and hg cannot be a part of a cycle since they would create smaller cycles that do not include all of the vertices in G5.



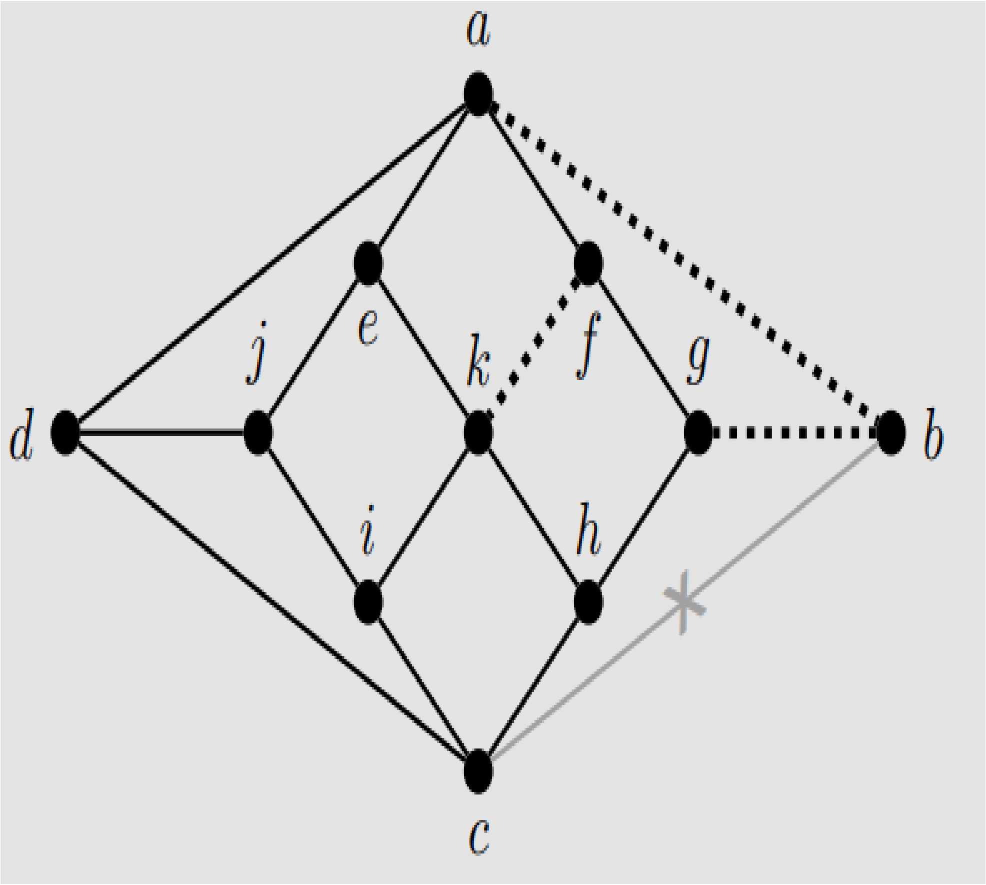
Hamiltonian cycles

- Since only one edge remains incident to d and g, namely di and gi, then these must be a part of the cycle.
- But in doing so, we would be forced to use ch since the other edges incident to i could not be chosen.
- This creates a cycle that does not span G5, and so G5 is not hamiltonian



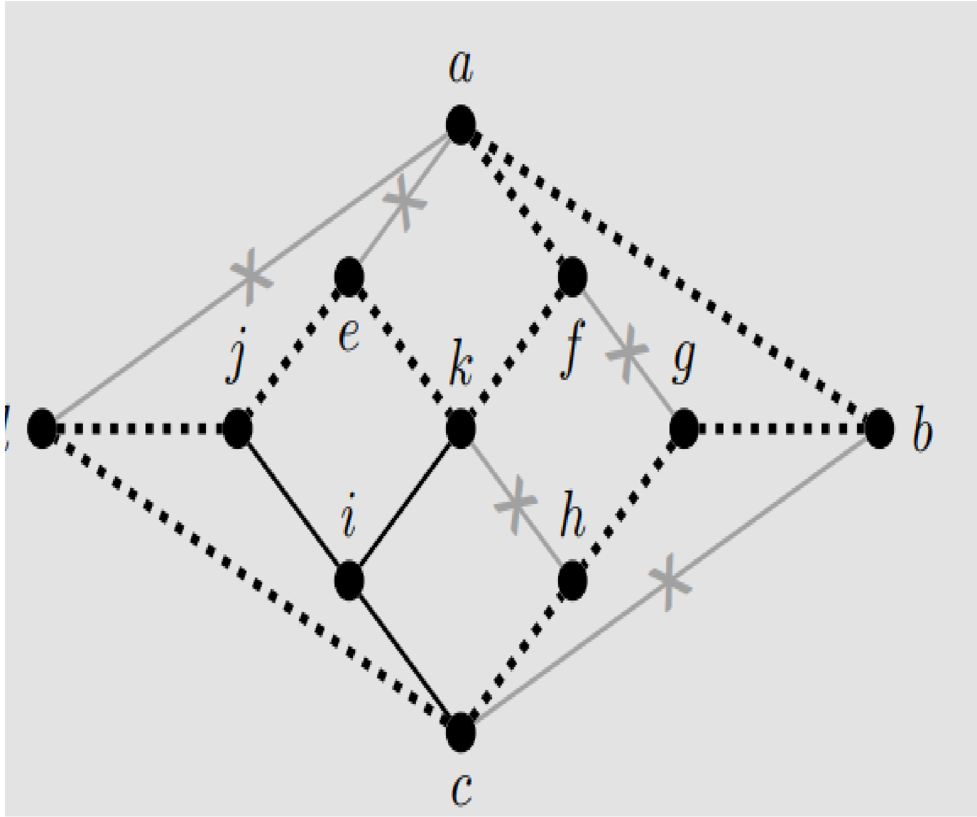
Hamiltonian cycles

- For G6, we know we cannot use all of the edges ab, ad, bc, and bd, as these four edges together create a cycle that does not span the graph.
- Since either b or d must have an edge not from this list, by symmetry we will choose bg to be a part of the hamiltonian cycle C we are attempting to build.
- Then either ab or bc must be the other edge incident to b in the cycle C, and again using symmetry we can choose ab.
- Now, we cannot use both af and fg, since together with ab and bg we would have a cycle that does not span G6. Thus we must choose f k to be a part of the cycle C.



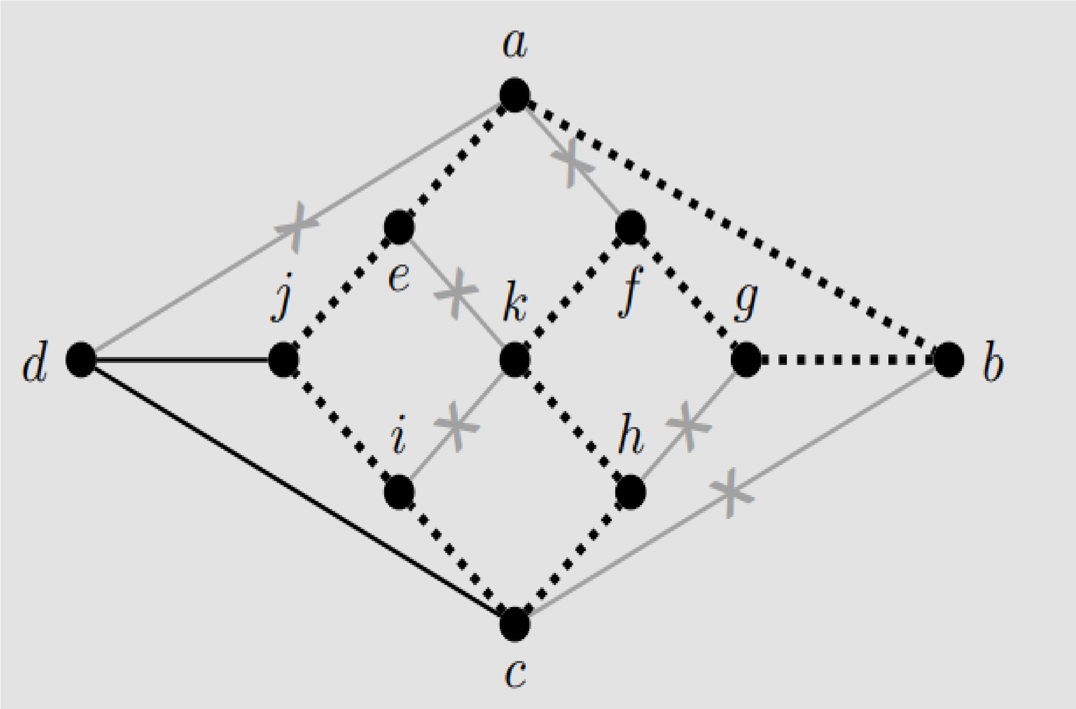
Hamiltonian cycles

- Now, we consider two scenarios based on the last edge incident to f in C.
- For the first case, we choose to add af to C. Then we cannot include fg and so gh must also be a part of the cycle since it is the only edge left incident to g.
- We cannot use hk as it would close the cycle before it spans G6, and so ch must also be a part of the cycle C.
- Since a already has two incident edges in the cycle, we cannot use either of ae or ad. Thus the other edges incident to d and e must be a part of the cycle, namely dj, dc, ej, and ek. But this closes the cycle without including vertex i.

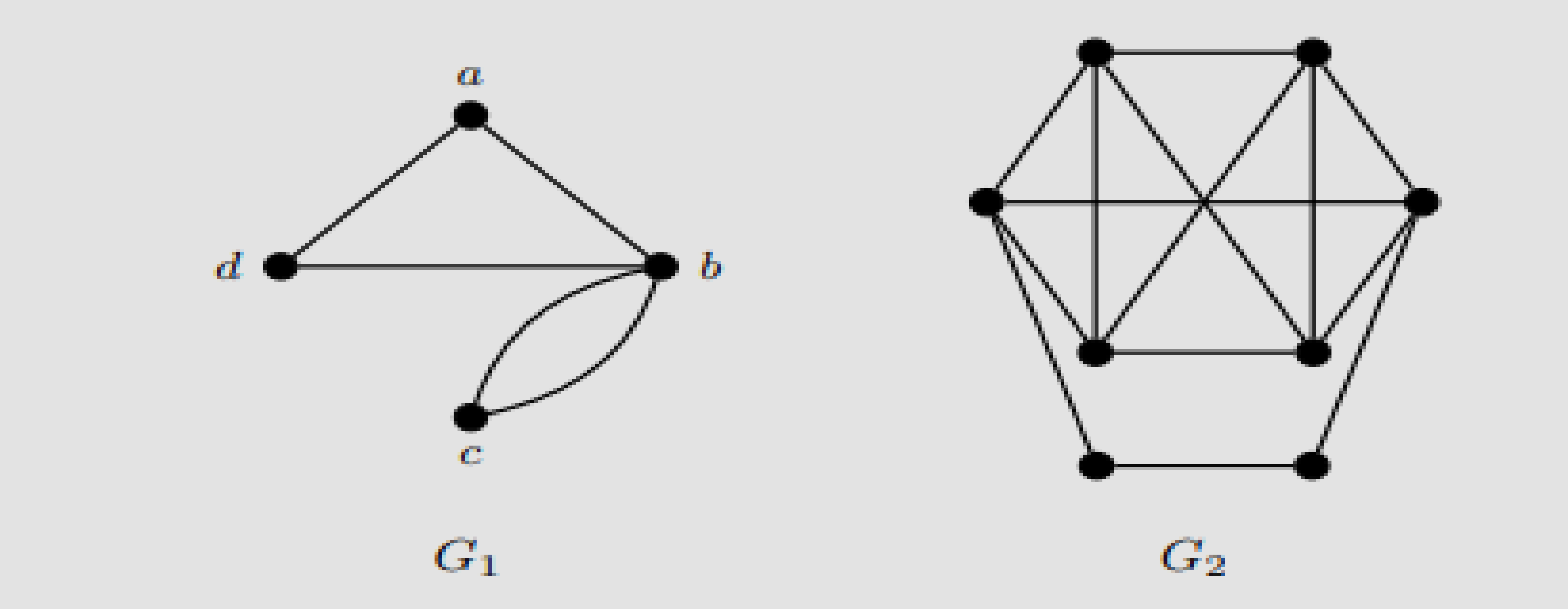


Hamiltonian cycles

- Thus af cannot be a part of a hamiltonian cycle C (if such a cycle exists).
- Thus fg must be chosen and we cannot use gh since g already has two incident edges as a part of the cycle.
- Thus the other edges incident to h must be used, namely hk and hc, and so the other edges incident to k cannot be, namely ke and ki.
- Based on the edges remaining, all of ci, ij, ej, and ea would be required, which closes a cycle that does not include d.
- Since neither option produces a spanning cycle of G6, we know it is not hamiltonian.



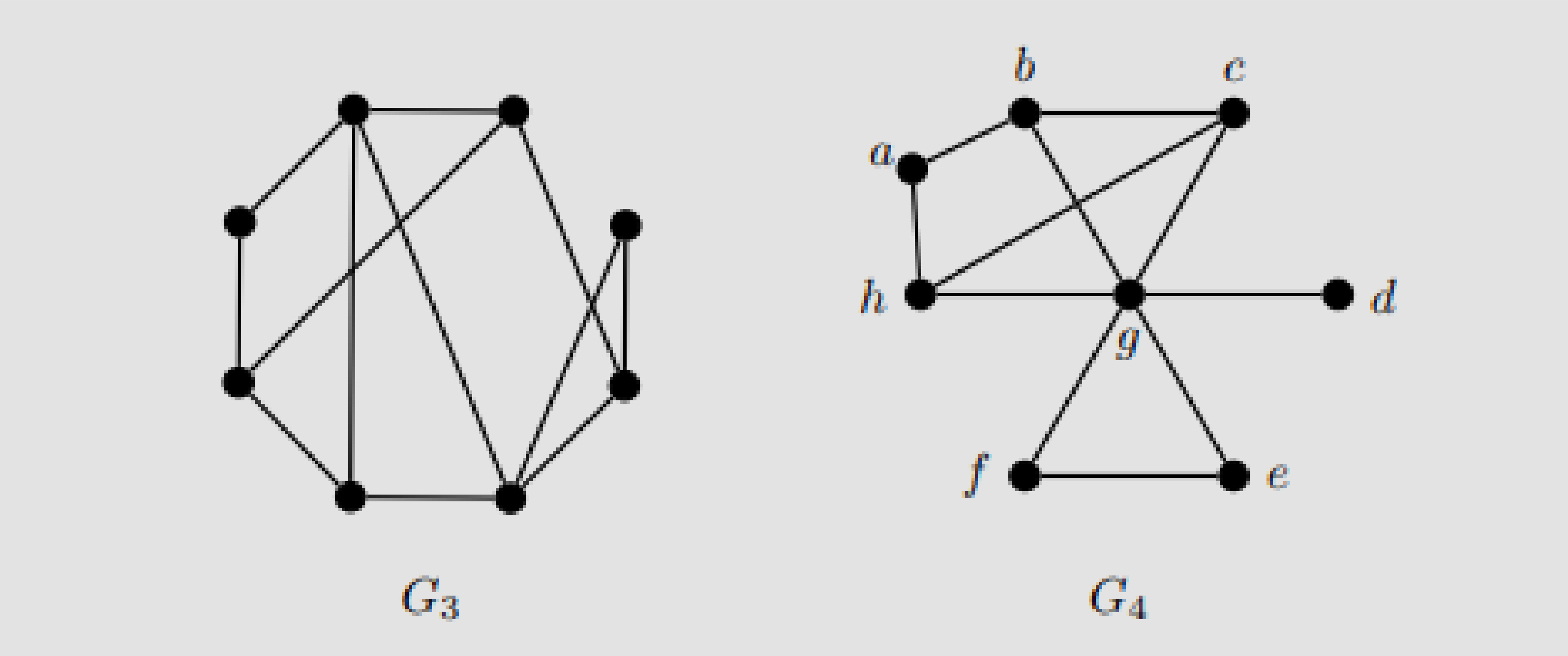
For each of the graphs below, determine if they have hamiltonian cycles (and paths) and eulerian circuits (and trails).



Solution:

- Since G1 is connected and all vertices are even, we know it has an eulerian circuit. There is no hamiltonian cycle since we need to include c in the cycle and by doing so we have already passed through b twice, making it impossible to visit a and d.
- Since G2 is connected and all vertices are even, we know it is eulerian. Hamiltonian cycles and hamiltonian paths also exist. To find one such path, remove any one of edges from a hamiltonian cycle

For each of the graphs below, determine if they have hamiltonian cycles (and paths) and eulerian circuits (and trails).

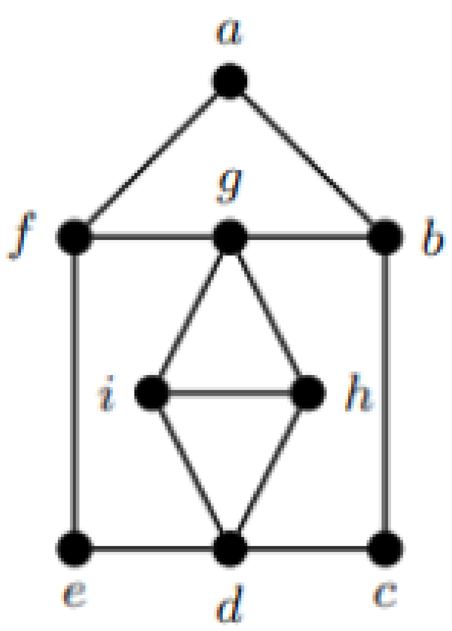


- Some vertices of G3 are odd, so we know it is not eulerian. Moreover, since more than two vertices are odd, the graph is not semi-eulerian. However, this graph does have a hamiltonian cycle (and so also a hamiltonian path).
- Four vertices of G4 are odd, we know it is neither eulerian nor semieulerian. This graph does not have a hamiltonian cycle since d cannot be a part of any cycle. Moreover, this graph does not have a hamiltonian path since any traversal of every vertex would need to travel through g multiple times.

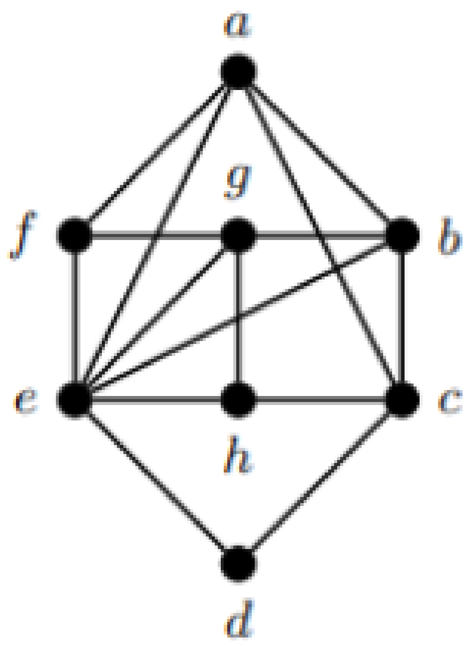
Hamiltonian cycles

Q.Determine if each of the graphs below are Hamiltonian. For those that are, find a Hamiltonian cycle. Otherwise, provide a clear and concise argument as to why the graph is not Hamiltonian.

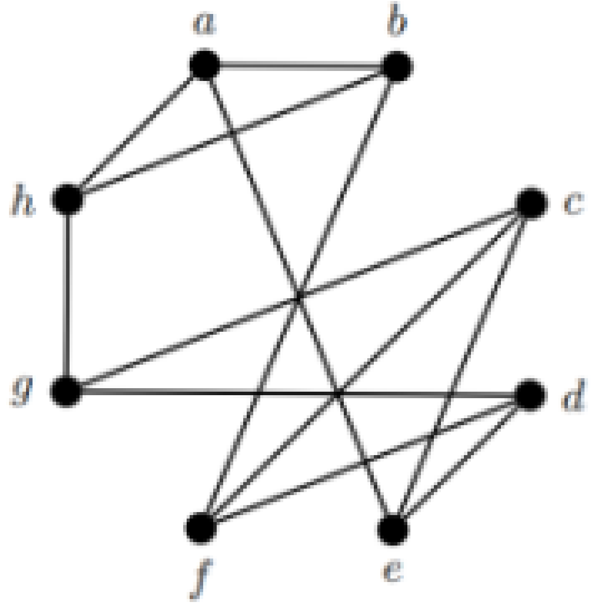
(a)

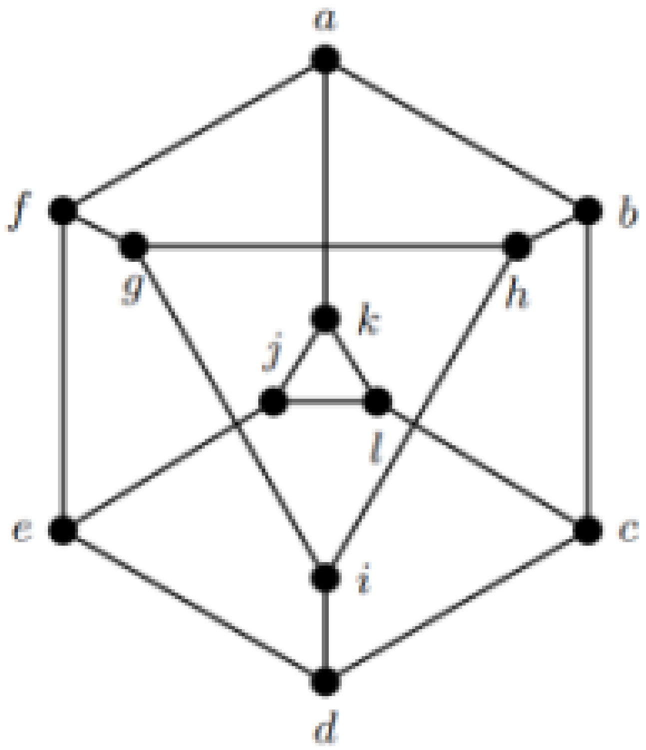


(b)



Q. Determine if each of the graphs below are Hamiltonian. For those that are, find a Hamiltonian cycle.





Understanding the Travelling Salesman Problem

- The Travelling Salesman Problem (also known as the Travelling Salesperson Problem or TSP) is an NP-hard graph computational problem where the salesman must visit all cities (denoted using vertices in a graph) given in a set just once.

- The distances (denoted using edges in the graph) between all these cities are known. We are requested to find the shortest possible route in which the salesman visits all the cities and returns to the origin city.

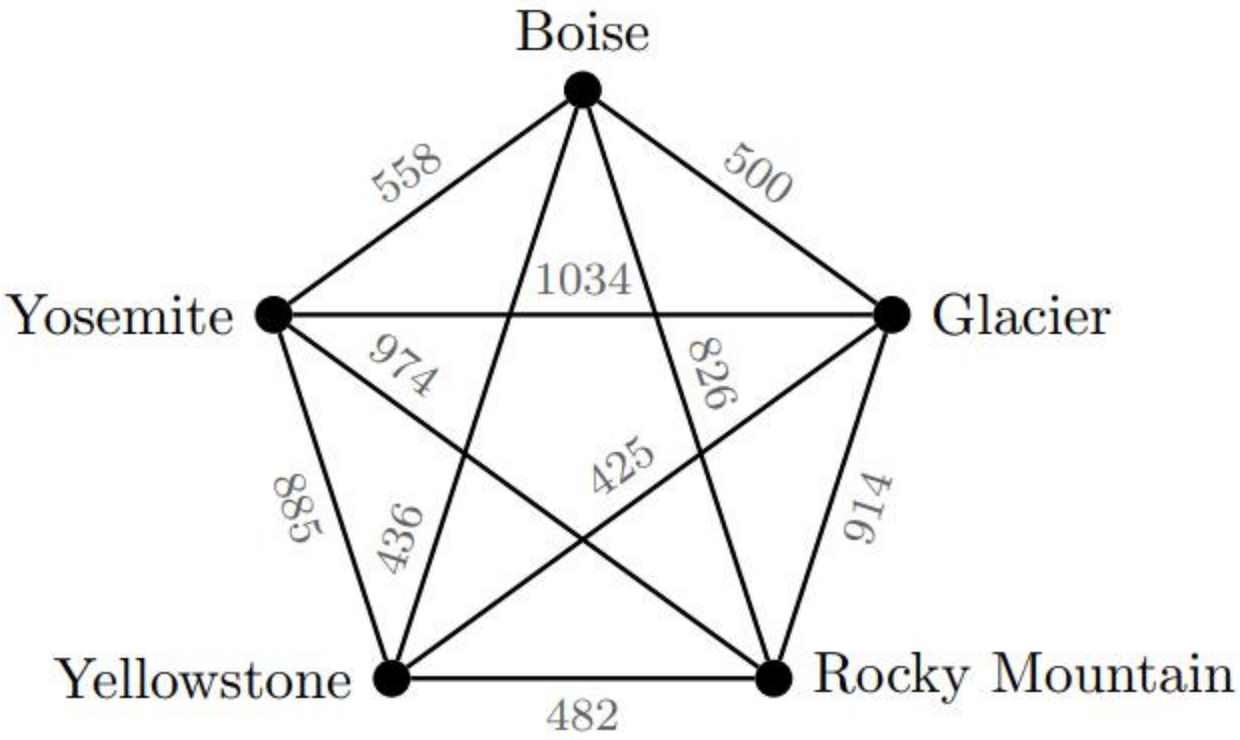
- There is a difference between Hamiltonian Cycle and TSP.

- The Hamiltonian Cycle problem is the problem where we have to find if there exists a tour that visits every city accurately once.

- However, in this problem, we already know that Hamiltonian Tour exists as the graph is complete, and in fact, there exist many such tours.

Understanding the Travelling Salesman Problem

- The graph that models the general Traveling Salesman Problem (TSP) is a weighted complete graph. Eg:



Brute Force Algorithm:

- To find the hamiltonian cycle of least total weight, one obvious method is to find all possible hamiltonian cycles and pick the cycle with the smallest total.
- The method of trying every possibility to find an optimal solution is referred to as an exhaustive search, or use of the Brute Force Algorithm.
- This method can be used for any number of problems, not just the Traveling Salesman Problem.

Understanding the Travelling Salesman Problem

Brute Force Algorithm

Input: Weighted complete graph K_n.

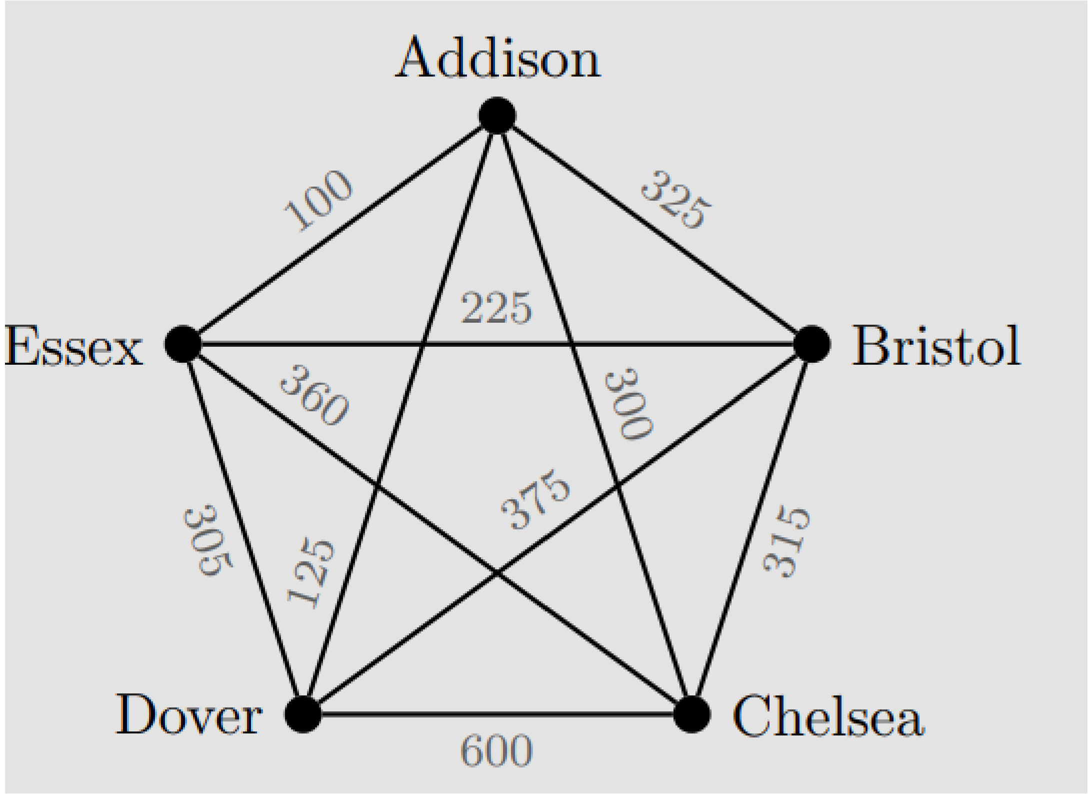
Steps:

1. Choose a starting vertex, call it v.
2. Find all hamiltonian cycles starting at v. Calculate the total weight of each cycle.
3. Compare all $(n-1)!$ cycles. Pick one with the least total weight. (Note: there should be at least two options).

Output: Minimum hamiltonian cycle.

Understanding the Travelling Salesman Problem

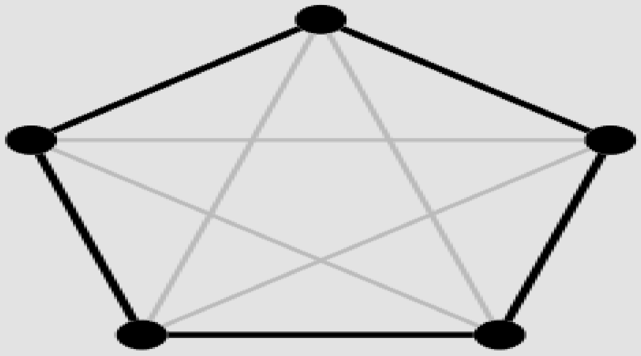
Q. Sam is planning his next business trip from his home-town of Addison and has determined the cost for travel between any ofthe five cities he must visit. This information is modeled in the weightedcomplete graph on the next page, where the weight is given in terms ofdollars. Use Brute Force to find all possible routes for his trip.



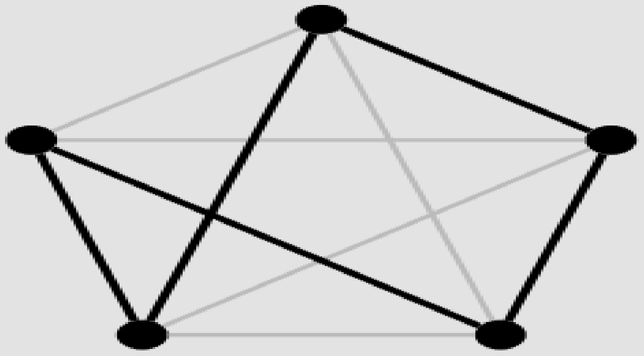
Solution:

- **O**ne method for finding all hamiltonian cycles, and ensuring you indeed have all 24, is to use alphabetical or lexicographic ordering of the cycles.
- Note that all cycles must start and end at Addison and we will abbreviate all cities with their first letter.
- For example, the first cycle is a b c d e a and appears first in the list below. Below each cycle is its reversal and total weight.

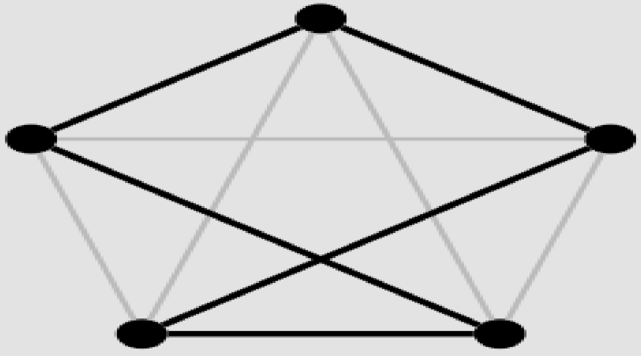
Understanding the Travelling Salesman Problem



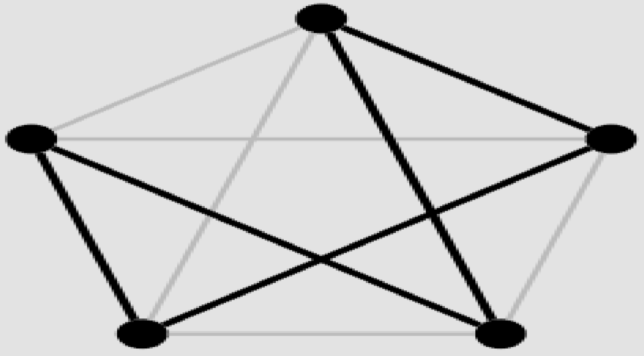
$a\,b\,c\,d\,e\,a$
$a\,e\,d\,c\,b\,a$
1645



$a\,b\,c\,e\,d\,a$
$a\,d\,e\,c\,b\,a$
1430

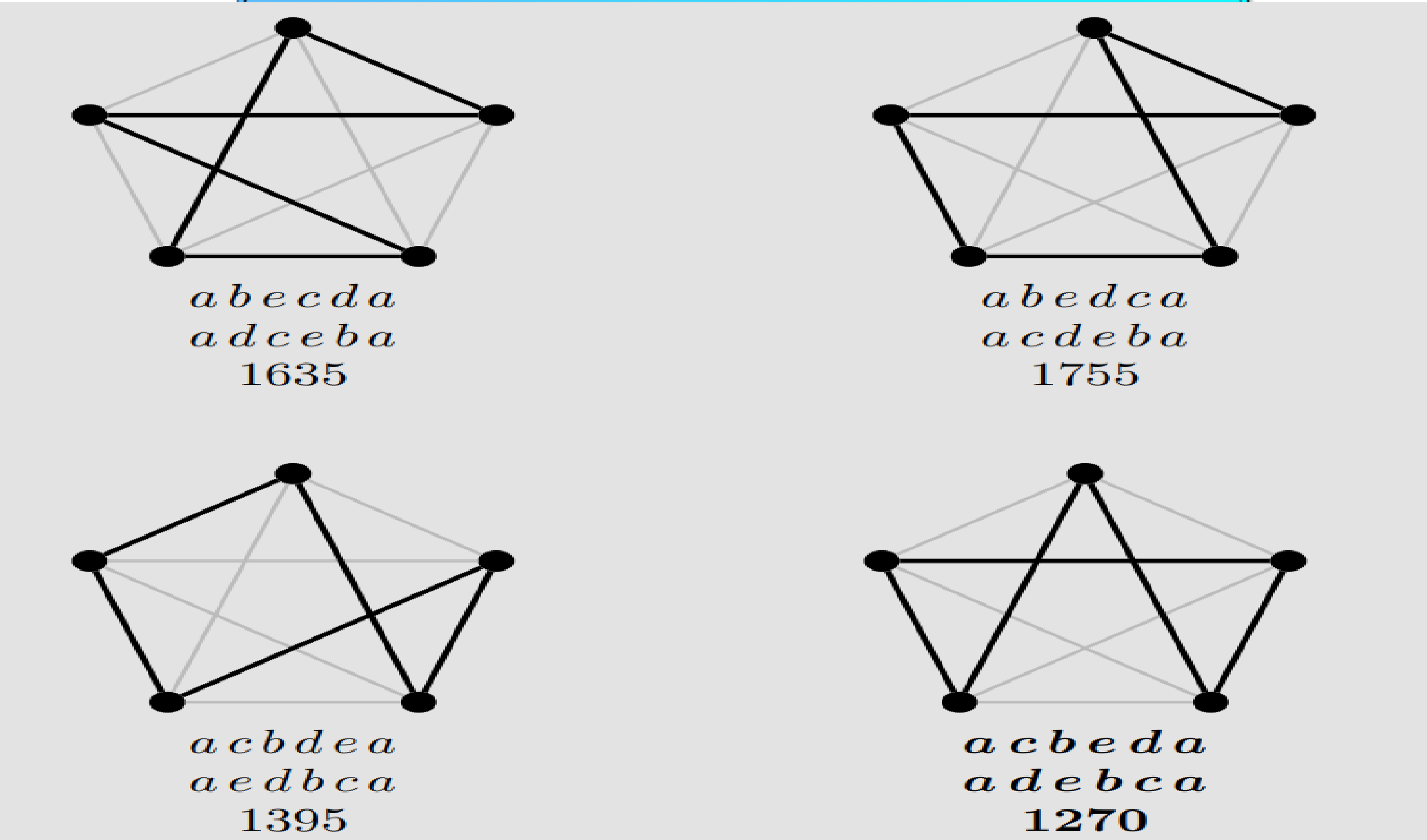


$a\,b\,d\,c\,e\,a$
$a\,e\,c\,d\,b\,a$
1760

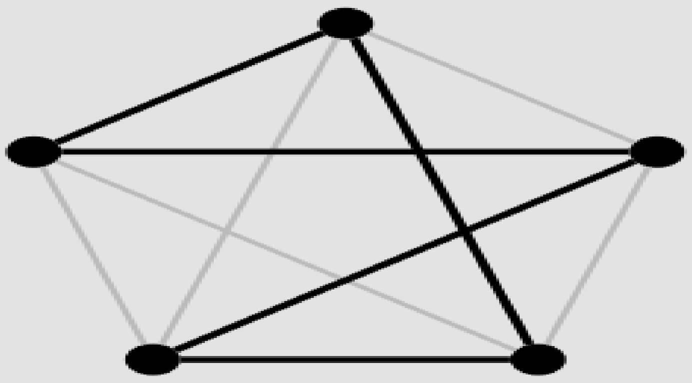


$a\,b\,d\,e\,c\,a$
$a\,c\,e\,d\,b\,a$
1665

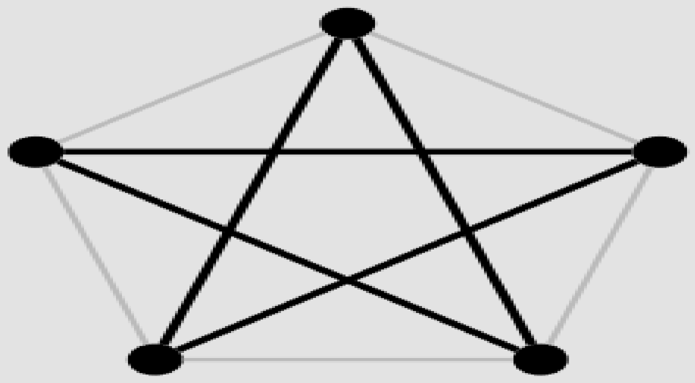
Understanding the Travelling Salesman Problem



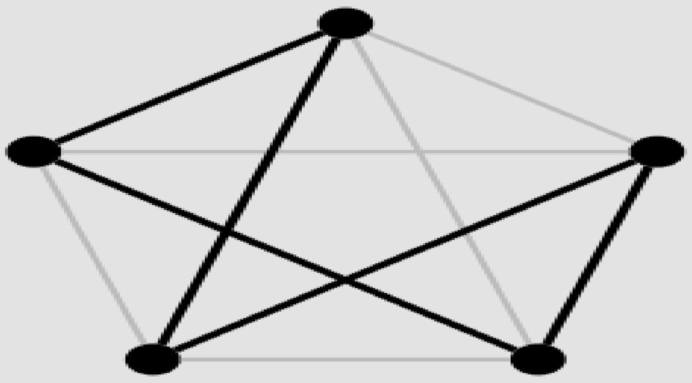
Understanding the Travelling Salesman Problem



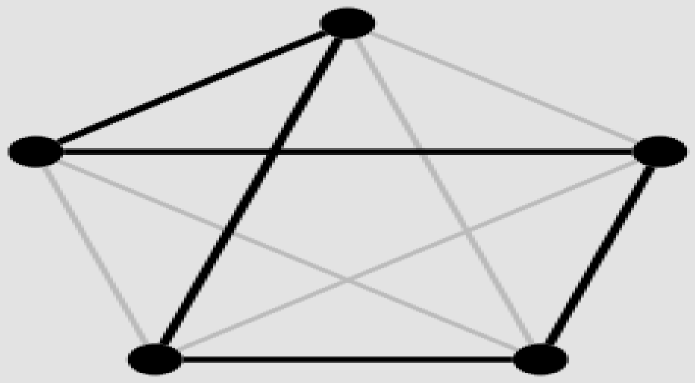
$a\,c\,d\,b\,e\,a$
$a\,e\,b\,d\,c\,a$
1600



$a\,c\,e\,b\,d\,a$
$a\,d\,b\,e\,c\,a$
1385



$a\,d\,b\,c\,e\,a$
$a\,e\,c\,b\,d\,a$
1275



$a\,d\,c\,b\,e\,a$
$a\,e\,b\,c\,d\,a$
1365

Understanding the Travelling Salesman Problem

Q.

The Traveling Salesman Problem (TSP) is a classic optimization problem in which a salesman is tasked with visiting a set of cities exactly once and returning to the starting city, all while minimizing the total distance traveled. Consider a salesman needs to visit four cities (A, Apply B, C, D) and return to the starting city (A).

The distances between the cities are as follows:

Distance from A to B: 10 units

Distance from A to C: 15 units

Distance from A to D: 20 units

Distance from B to C: 35 units

Distance from B to D: 30 units

Distance from C to D: 40 units

Find the shortest possible route that visits each city exactly once and returns to the starting city

Understanding the Travelling Salesman Problem

Nearest Neighbor Algorithm

Input: Weighted complete graph K_n.

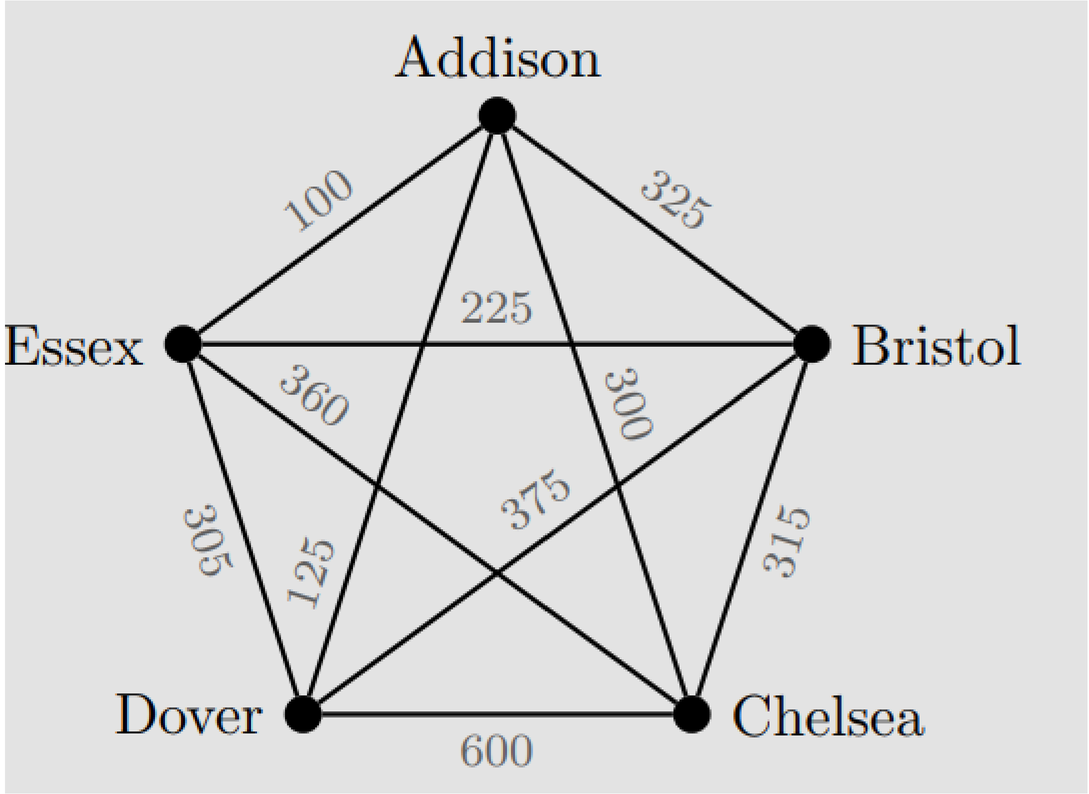
Steps:

1. Choose a starting vertex, call it v. Highlight v.
2. Among all edges incident to v, pick the one with the smallest weight. If two possible choices have the same weight, you may randomly pick one.
3. Highlight the edge and move to its other endpoint u. Highlight u.
4. Repeat Steps (2) and (3), where only edges to unhighlighted vertices are considered.
5. Close the cycle by adding the edge to v from the last vertex highlighted. Calculate the total weight.

Output: hamiltonian cycle.

Understanding the Travelling Salesman Problem

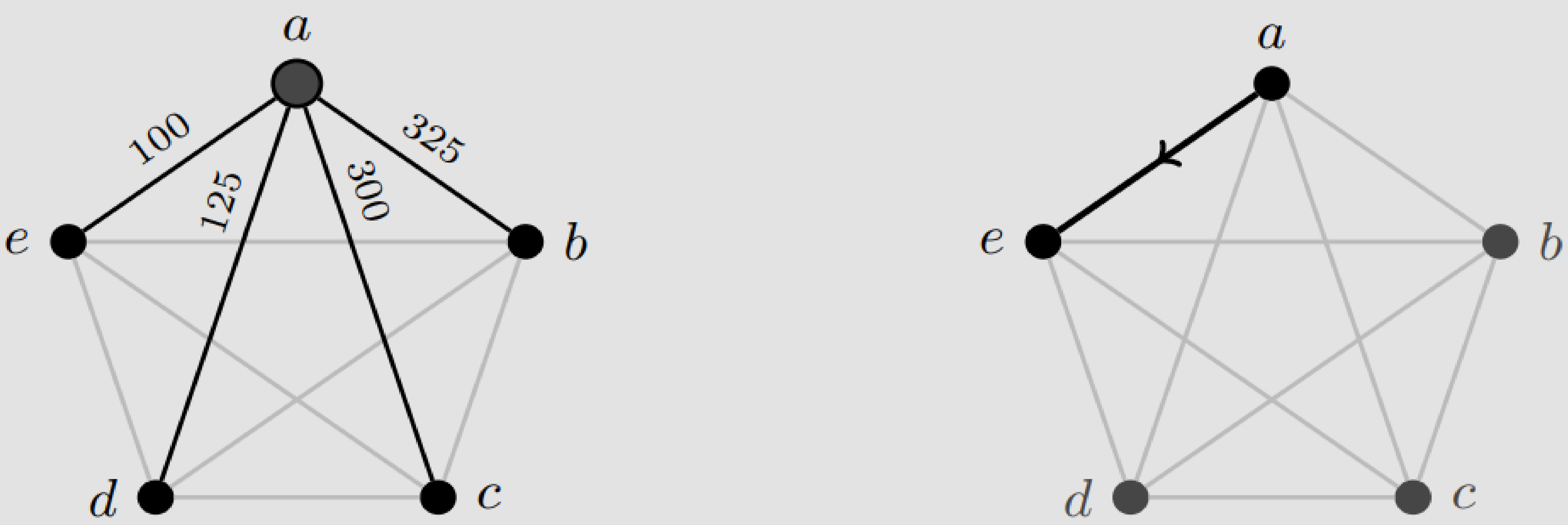
Q. Sam is planning his next business trip from his home-town of Addison and has determined the cost for travel between any ofthe five cities he must visit. This information is modeled in the weightedcomplete graph on the next page, where the weight is given in terms of dollars. Use nearest neighbour to find shortest routes for his trip.



Understanding the Travelling Salesman Problem

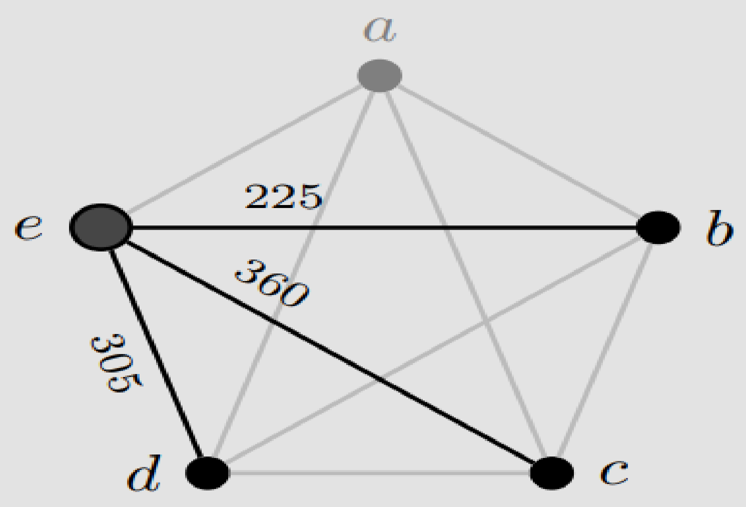
Step 1: The starting vertex is a.

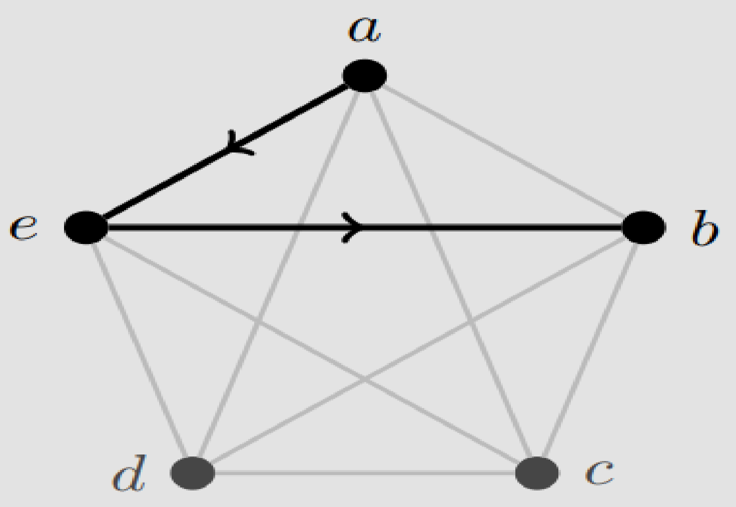
Step 2: The edge of smallest weight incident to a is ae with weight 100.



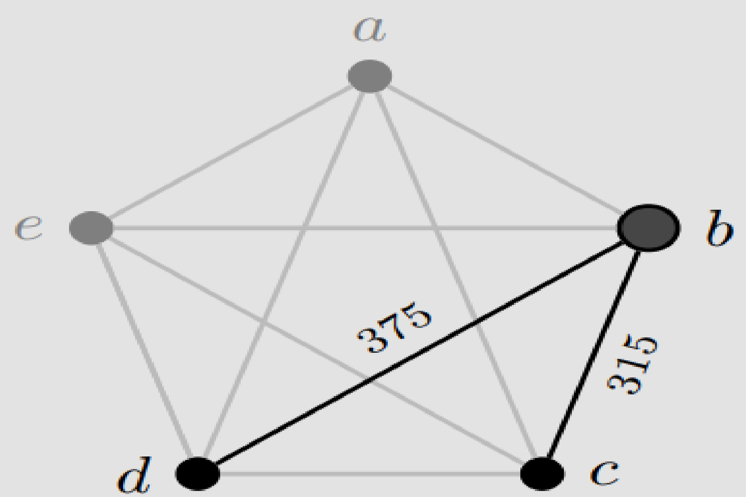
Step 3: From e we only consider edges to b, c, or d. Choose edge eb with weight 225.

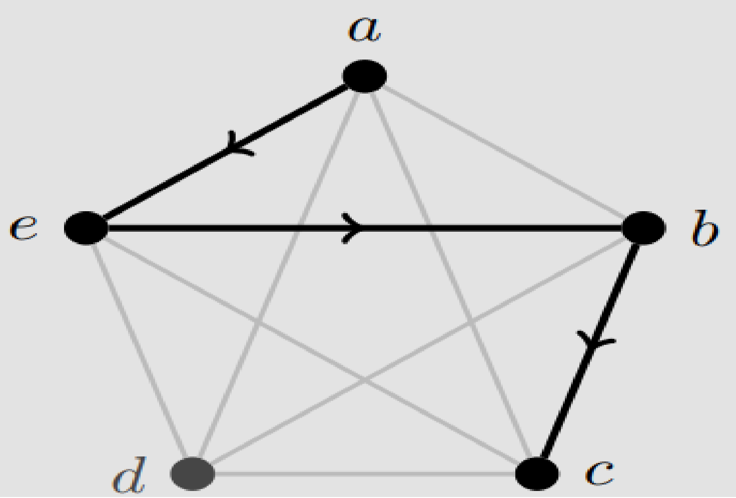
Understanding the Travelling Salesman Problem





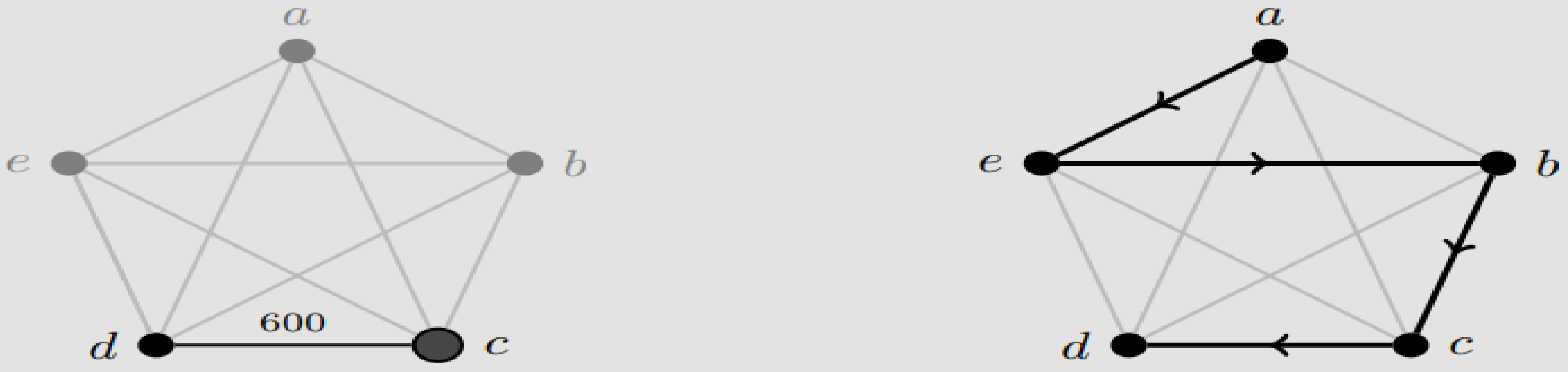
Step 4: From b we consider the edges to c or d. The edge of smallest weig
is bc with weight 315.



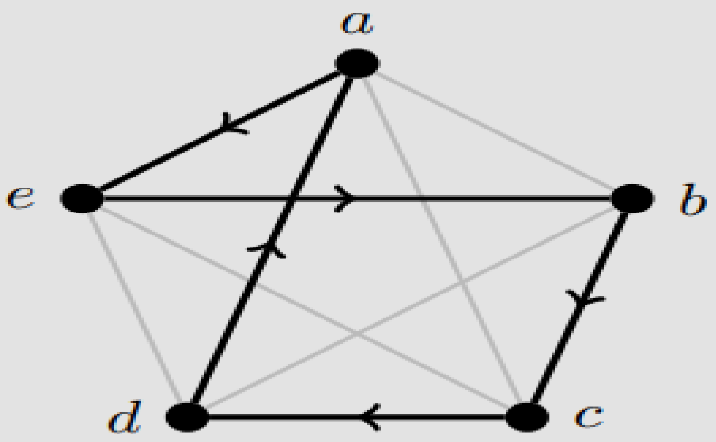


Understanding the Travelling Salesman Problem

Step 5: Even though cd does not have the smallest weight among all edg
incident to c, it is the only choice available.



Step 6: Close the circuit by adding da.



Output: The circuit is $a\,e\,b\,c\,d\,a$ with total weight 1365.

Repetitive Nearest Neighbor Algorithm

Input: Weighted complete graph K_n.

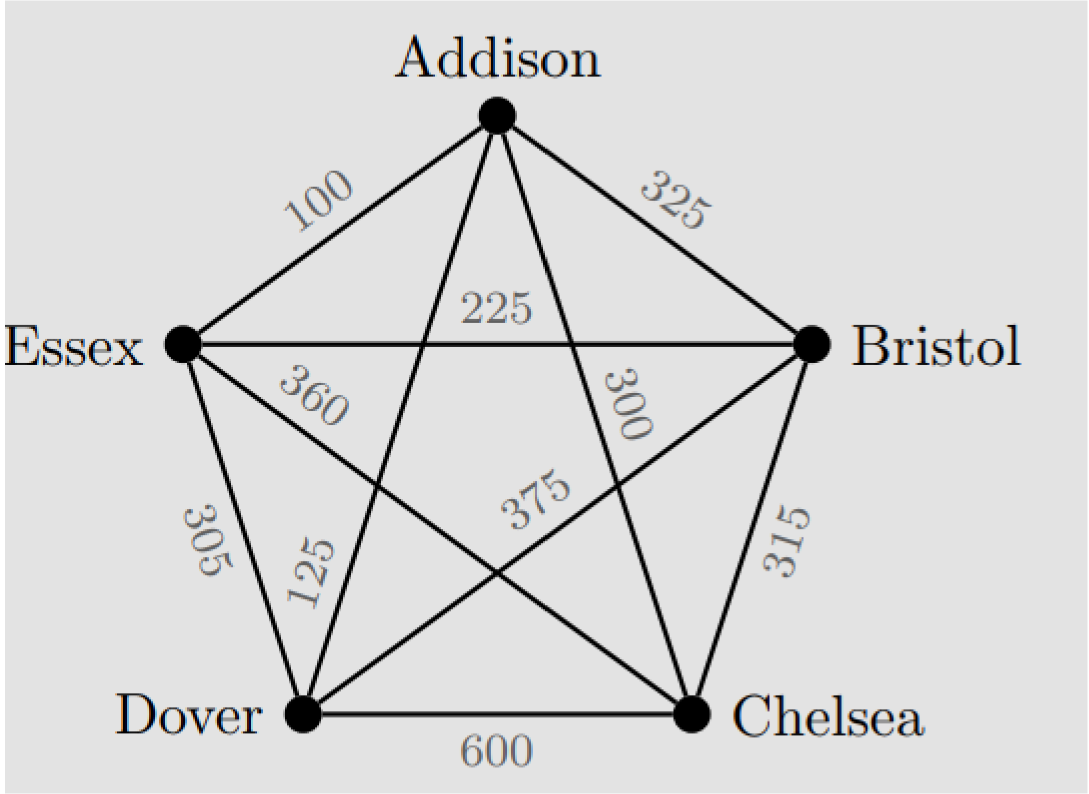
Steps:

1. Choose a starting vertex, call it v.
2. Apply the Nearest Neighbor Algorithm.
3. Repeat Steps (1) and (2) so each vertex of K_n serves as the starting vertex.
4. Choose the cycle of least total weight. Rewrite it with the desired reference point.

Output: hamiltonian cycle.

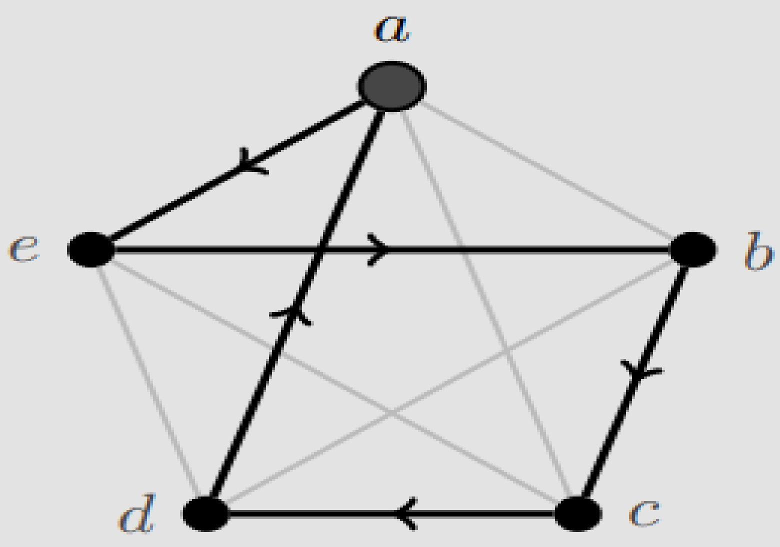
Understanding the Travelling Salesman Problem

Q. Sam is planning his next business trip from his home-town of Addison and has determined the cost for travel between any ofthe five cities he must visit. This information is modeled in the weightedcomplete graph on the next page, where the weight is given in terms of dollars. Use repetitive nearest neighbour to find shortest routes for his trip.



Understanding the Travelling Salesman Problem

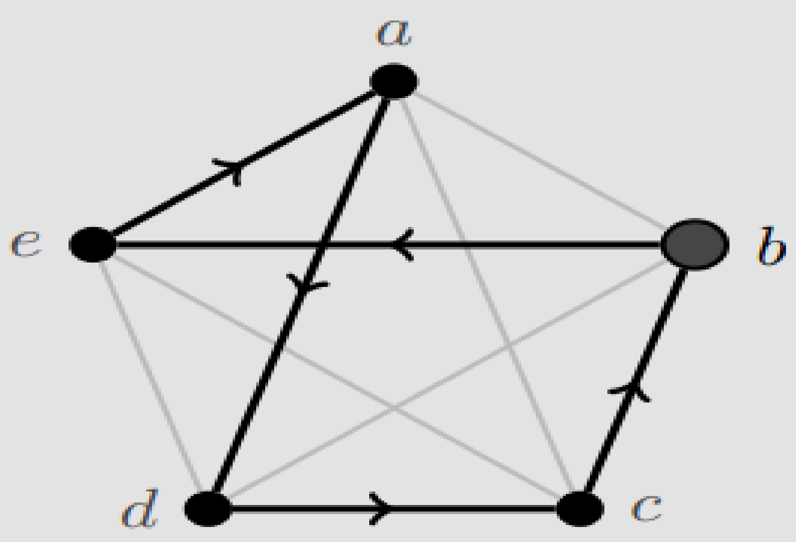
Solution: The five cycles are shown below, with the original name, the rewritten form with a as the reference point, and the total weight of the cycle. You should notice that the cycle starting at d is the same as the one starting at a, and the cycle starting at b is their reversal.



$a\,e\,b\,c\,d\,a$

$a\,e\,b\,c\,d\,a$

1365

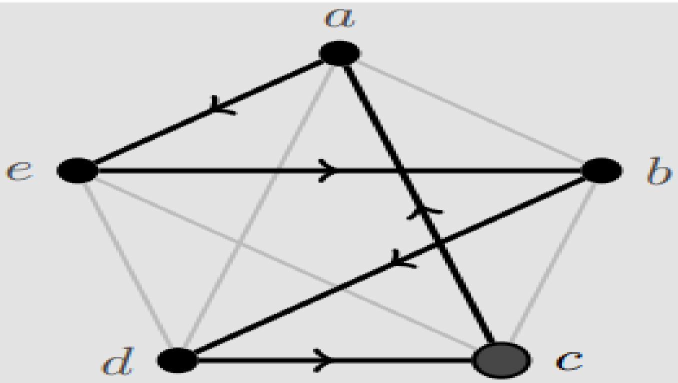


$b\,e\,a\,d\,c\,b$

$a\,d\,c\,b\,e\,a$

1365

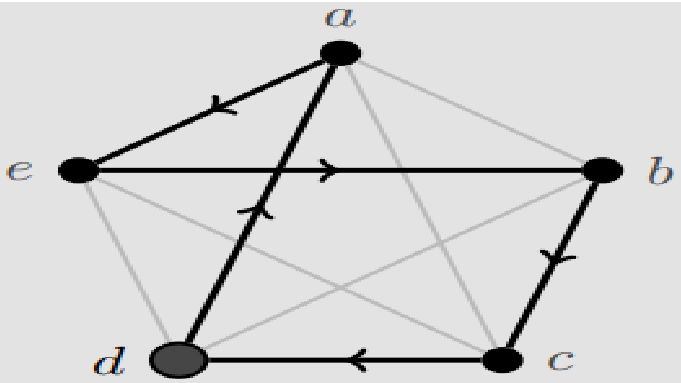
Understanding the Travelling Salesman Problem



c a e b d c

a e b d c a

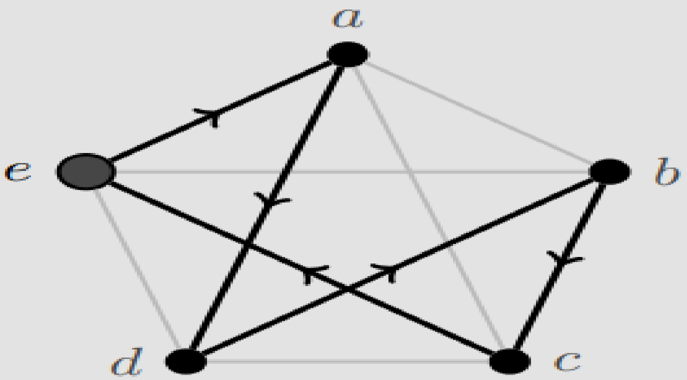
1600



d a e b c d

a e b c d a

1365



e a d b c e

a d b c e a

1275

Cheapest Link Algorithm

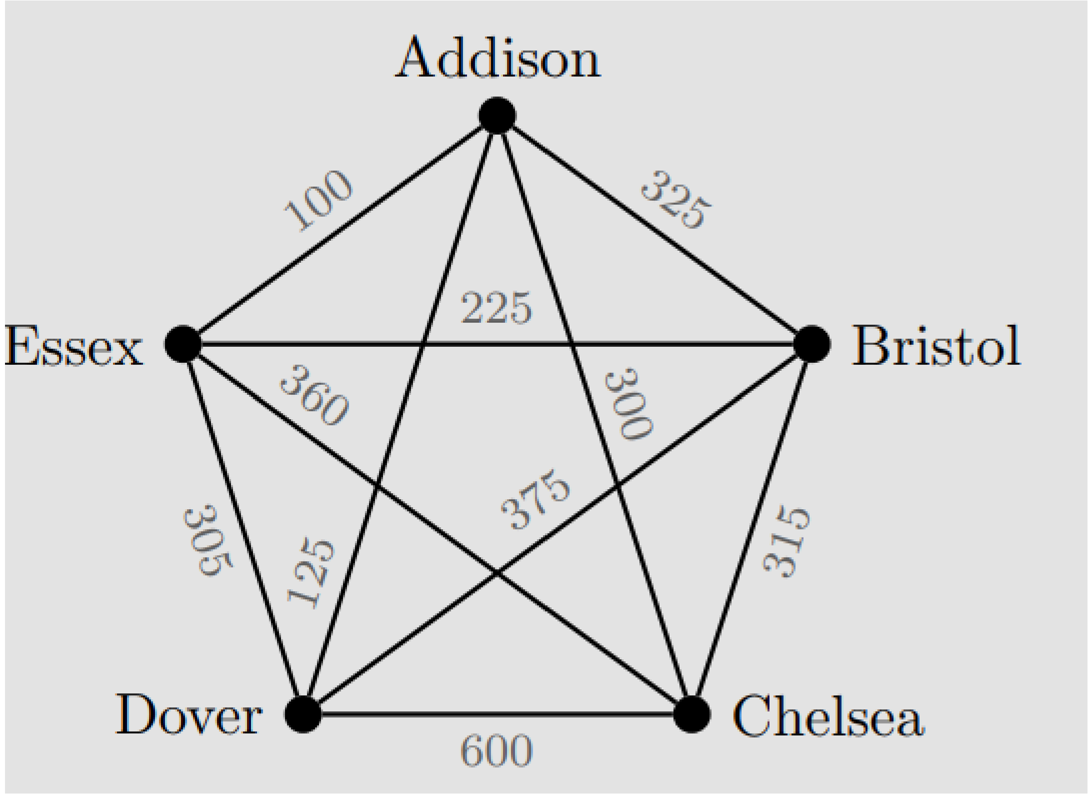
Input: Weighted complete graph K_n.

Steps:

1. Among all edges in the graph pick the one with the smallest weight. If two possible choices have the same weight, you may randomly pick one. Highlight the edge.
2. Repeat Step (1) with the added conditions:
   (a) no vertex has three highlighted edges incident to it; and
   (b) no edge is chosen so that a cycle closes before hitting all the vertices.
3. Calculate the total weight.

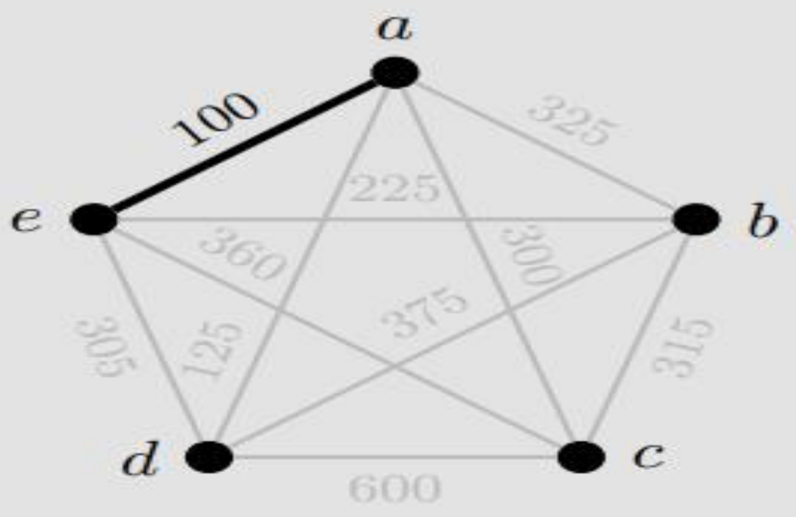
Output: hamiltonian cycle.

Q. Sam is planning his next business trip from his home-town of Addison and has determined the cost for travel between any ofthe five cities he must visit. This information is modeled in the weightedcomplete graph on the next page, where the weight is given in terms of dollars. Use chepest link to find shortest routes for his trip.

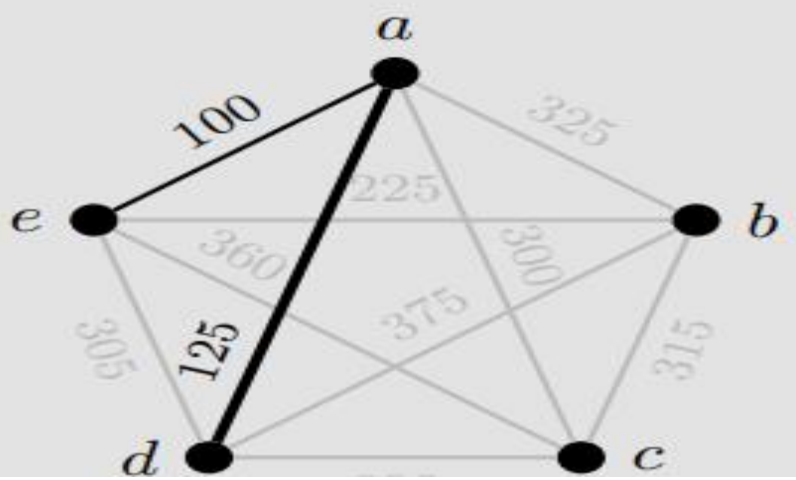


Understanding the Travelling Salesman Problem

Step 1: The smallest weight is 100 for edge ae.

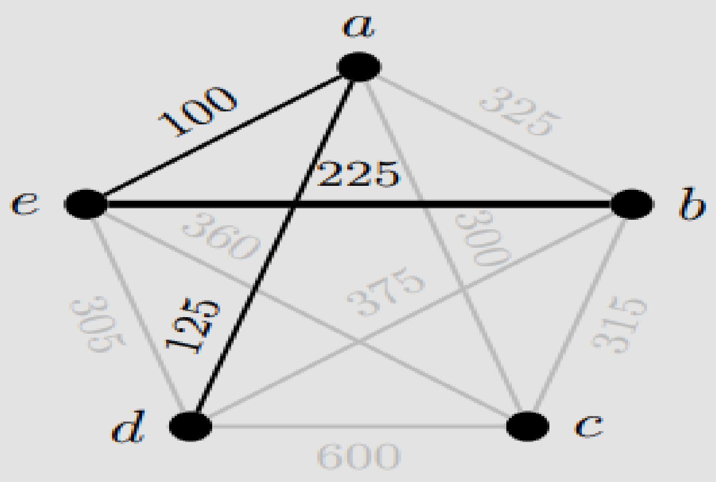


Step 2: The next smallest weight is 125 with edge ad.

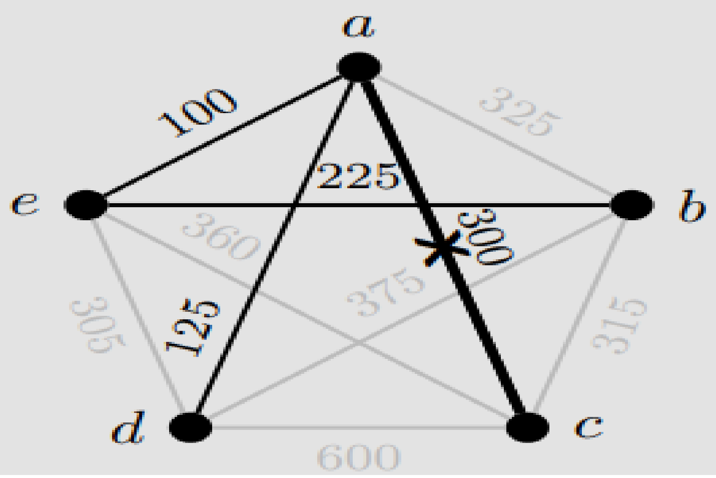


Understanding the Travelling Salesman Problem

Step 3: The next smallest weight is 225 for edge be.

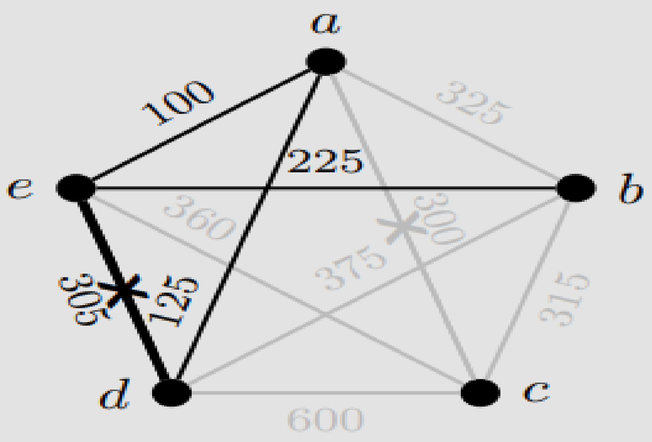


Step 4: Even though ac has weight 300, we must bypass it as it would force a to have three incident edges that are highlighted.

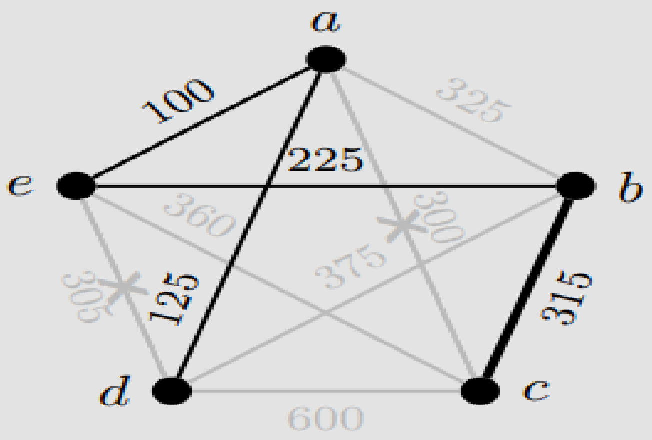


Understanding the Travelling Salesman Problem

Step 5: The next smallest weight is 305 for edge ed, but again we must bypass it as it would close a cycle too early as well as force e to have three incident edges that are highlighted.

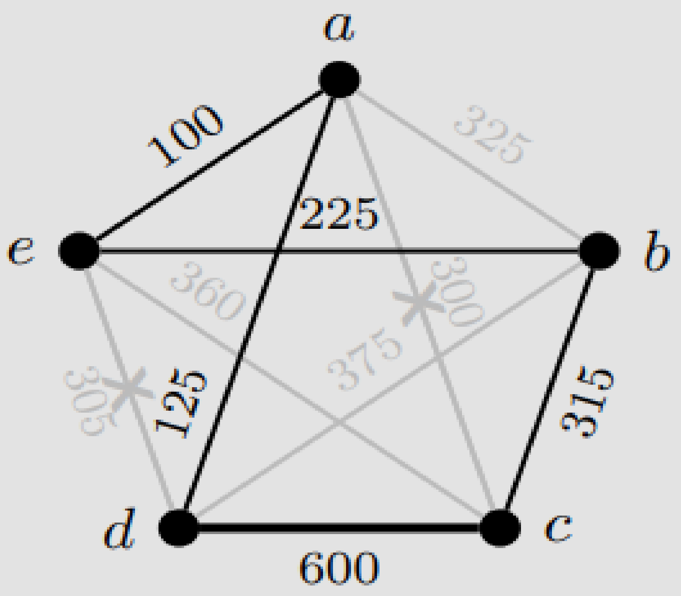


Step 6: The next available is bc with weight 315.



Understanding the Travelling Salesman Problem

Step 7: At this point, we must close the cycle with the only one choice of cd.



Output: The resulting cycle is $a\,e\,b\,c\,d\,a$ with total weight 1365.

Nearest Insertion Algorithm

Input: Weighted complete graph K_n.

Steps:

1. Among all edges in the graph, pick the one with the smallest weight If two possible choices have the same weight, you may randomly pic one. Highlight the edge and its endpoints.
2. Pick a vertex that is closest to one of the two already chosen vertices Highlight the new vertex and its edges to both of the previously chose vertices.
3. Pick a vertex that is closest to one of the three already chosen vertices Calculate the increase in weight obtained by adding two new edge and deleting a previously chosen edge. Choose the scenario with th smallest total. For example, if the cycle obtained from (2) was $a - b -$ $c - a$ and d is the new vertex that is closest to c, we calculate:

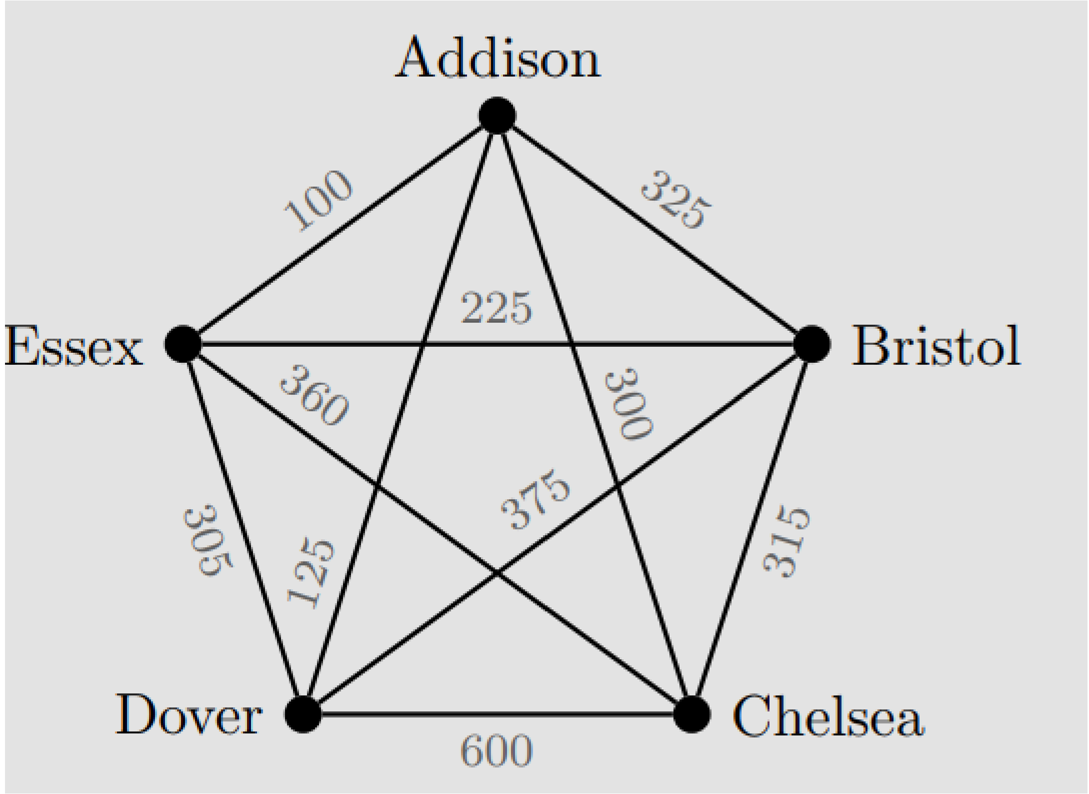
$$w(dc) + w(db) - w(cb) \text{ and } w(dc) + w(da) - w(ca)$$

   and choose the option that produces the smaller total.
4. Repeat Step (3) until all vertices have been included in the cycle.
5. Calculate the total weight.

Output: hamiltonian cycle.

Understanding the Travelling Salesman Problem

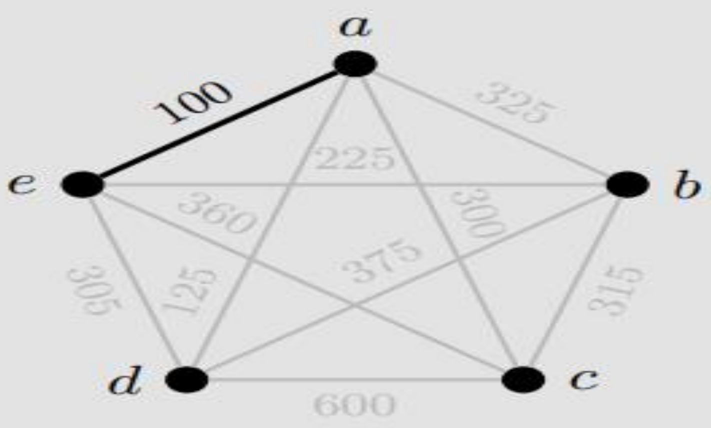
Q. Sam is planning his next business trip from his home-town of Addison and has determined the cost for travel between any ofthe five cities he must visit. This information is modeled in the weightedcomplete graph on the next page, where the weight is given in terms of dollars. Use nearest insertion to find shortest route for his trip.



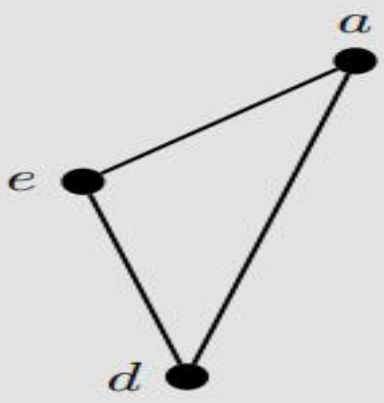
Understanding the Travelling Salesman Problem

Solution: At each step shown, the graph on the left highlights the edge being added and the graph on the right shows how the cycle is built.

Step 1: The smallest weight edge is ae at 100.

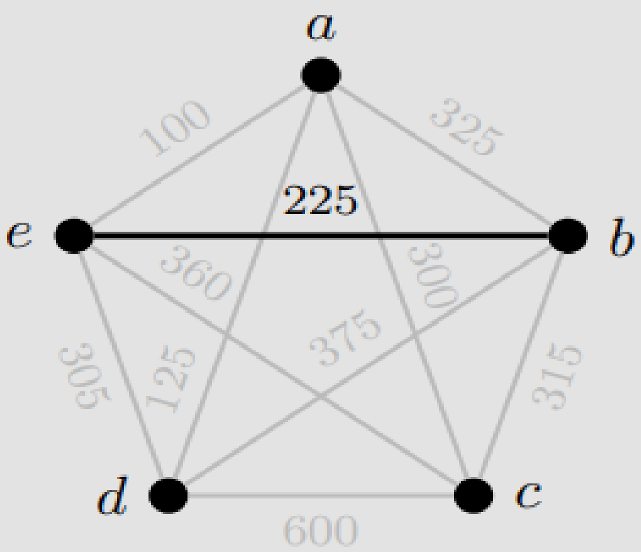


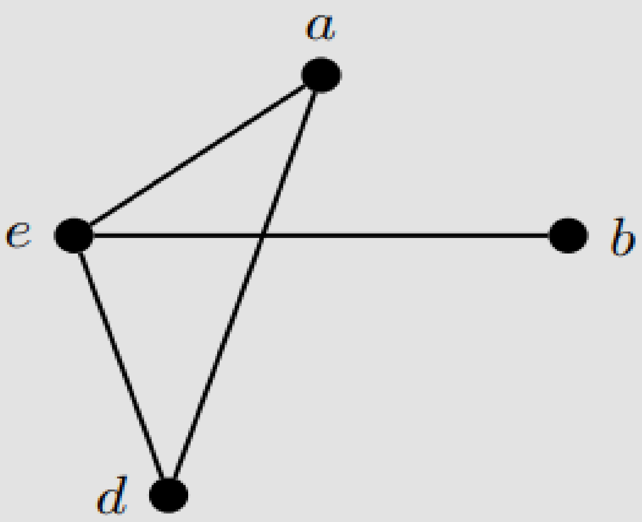
Step 2: The closest vertex to either a or e is d through the edge ad of weight 125. Form a cycle by adding ad and de.



Understanding the Travelling Salesman Problem

Step 3: The closest vertex to any of a, d, or e is b through the edge be with weight 225.





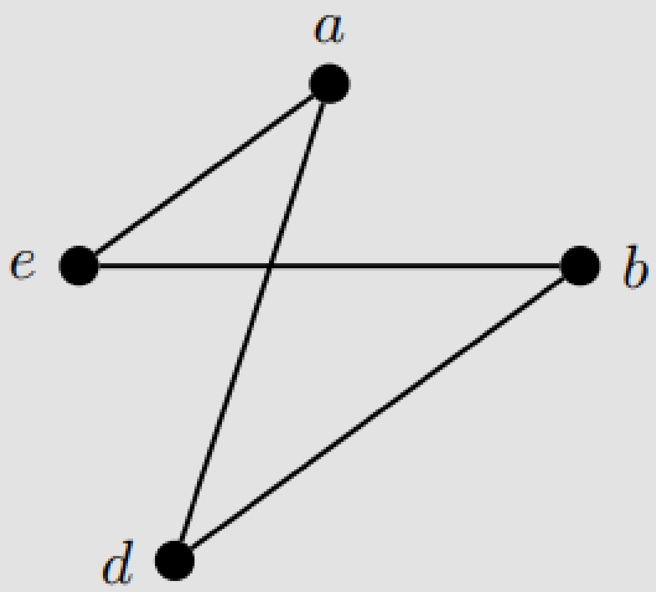
In adding edge be, either ae or de must be removed so that only two edges are incident to e. To determine which is the better choice, compute the following expressions:

$$be + ba - ea = 225 + 325 - 100 = 450$$

$$be + bd - ed = 225 + 375 - 305 = 295$$

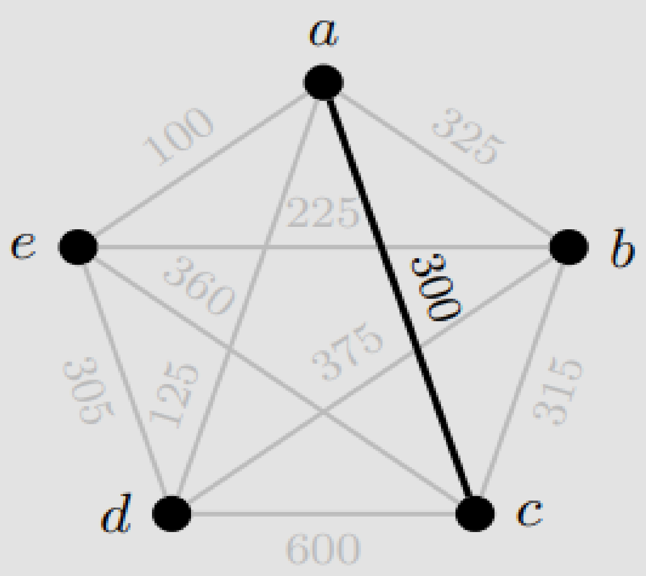
Understanding the Travelling Salesman Problem

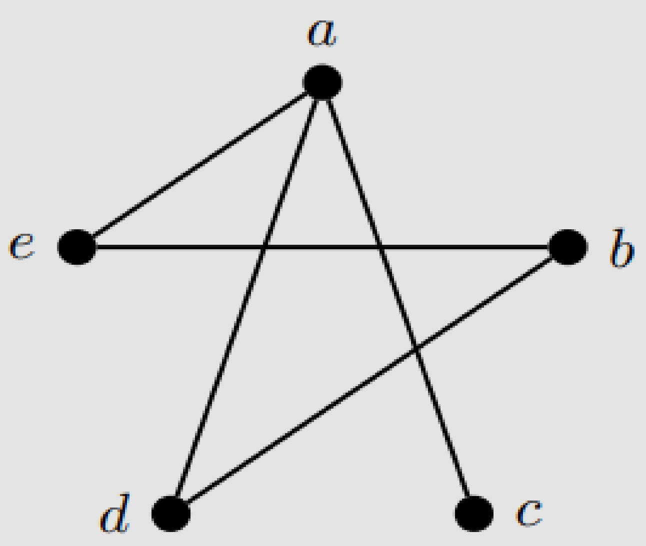
Since the second total is smaller, we create a larger cycle by adding edge bd and removing ed.



Understanding the Travelling Salesman Problem

Step 4: The only vertex remaining is c, and the minimum edge to the other vertices is ac with weight 300.

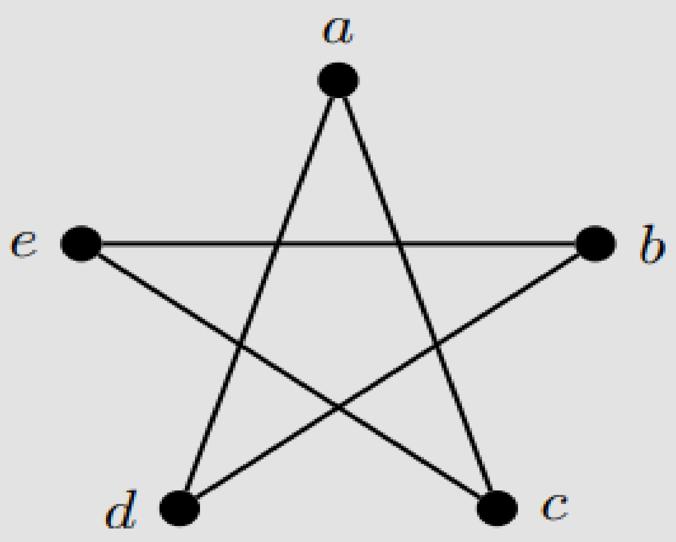




Either ae or ad must be removed. As in the previous step, we compute the following expressions:

$$ca + cd - ad = 300 + 600 - 125 = 775$$
$$ca + ce - ae = 300 + 360 - 100 = 560$$

The second total is again smaller, so we add ce and remove ae.



Output: The cycle is $a\,c\,e\,b\,d\,a$ with total weight 1385.

Definition 2.19 The ***relative error*** for a solution is given by

$$\epsilon_r = \frac{Solution - Optimal}{Optimal}$$

Dijkstra’s Algorithm

Dijkstra’s Algorithm is used to find a shortest path from your chosen starting and ending vertex .

Dijkstra’s Algorithm

Input: Weighted connected simple graph $G = (V, E, w)$ and designated $Start$ vertex.

Steps:

1. For each vertex x of G, assign a label $L(x)$ so that $L(x) = (-, 0)$ if $x = Start$ and $L(x) = (-, \infty)$ otherwise. Highlight $Start$.
2. Let $u = Start$ and define F to be the neighbors of u. Update the labels for each vertex v in F as follows:

$$\text{if } w(u) + w(uv) < w(v), \text{ then redefine } L(v) = (u, w(u) + w(uv))$$

$$\text{otherwise do not change } L(v)$$

Dijkstra’s Algorithm

3. Highlight the vertex with lowest weight as well as the edge uv used to update the label. Redefine $u = v$.

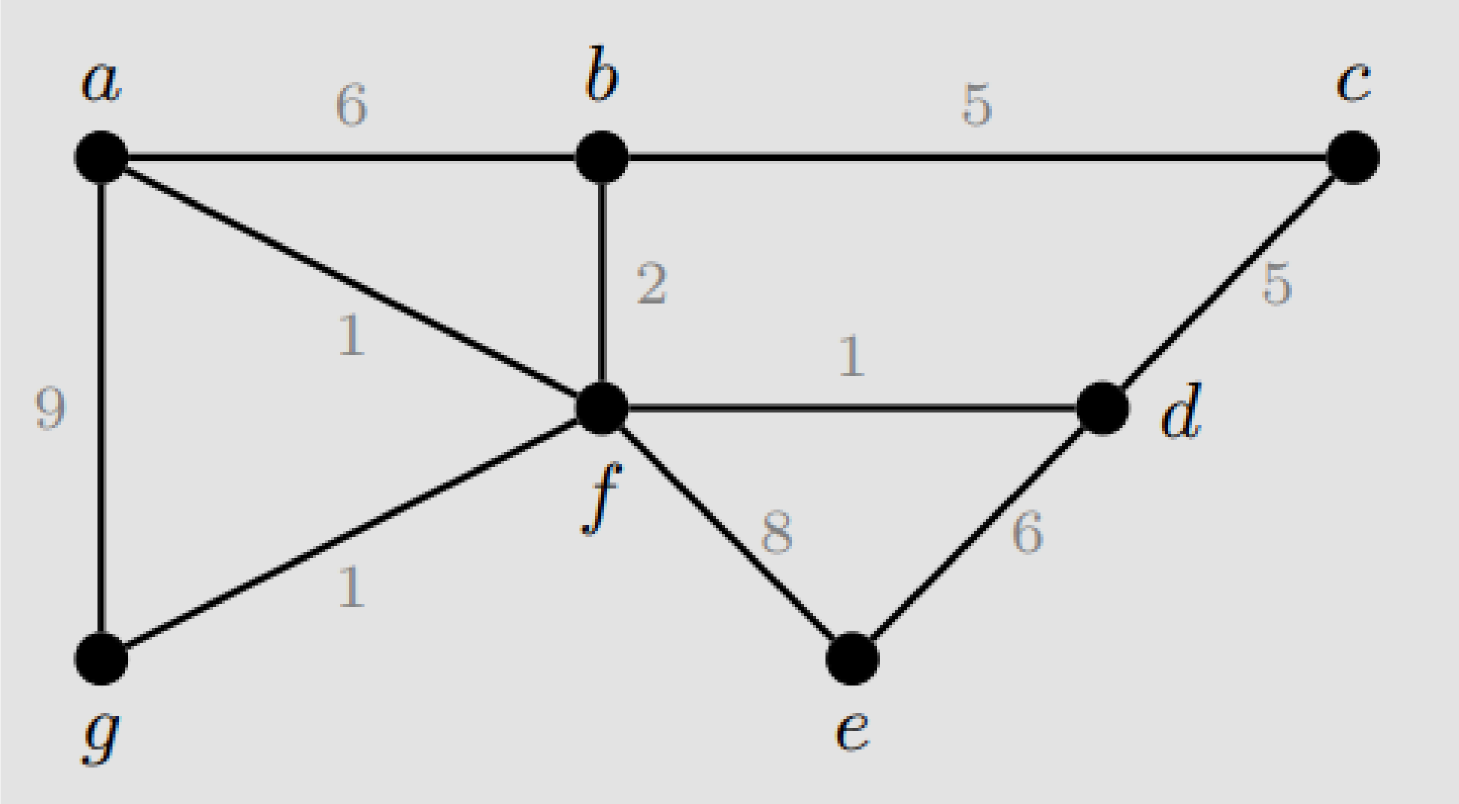
4. Repeat Steps (2) and (3) until each vertex has been reached. In all future iterations, F consists of the un-highlighted neighbors of all previously highlighted vertices and the labels are updated only for those vertices that are adjacent to the last vertex that was highlighted.

5. The shortest path from $Start$ to any other vertex is found by tracing back using the first component of the labels. The total weight of the path is the weight given in the second component of the ending vertex.

<u>Output:</u> Highlighted path from $Start$ to any vertex x of weight $w(x)$.

Dijkstra’s Algorithm

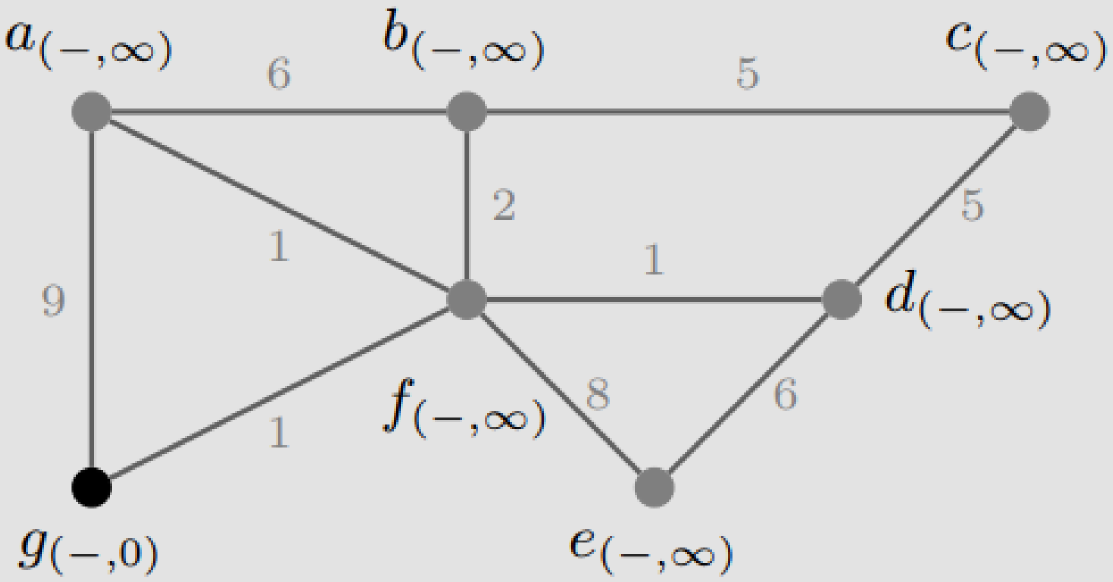
Q. Apply Dijkstra’s Algorithm to the graph below where Start = g.



Dijkstra's Algorithm

Solution: In each step, the label of a vertex will be shown in the table on the right.

Step 1: Highlight g. Define $L(g) = (-,0)$ and $L(x) = (-,\infty)$ for all $x = a, \cdots, f$.



$$F = \{\}$$

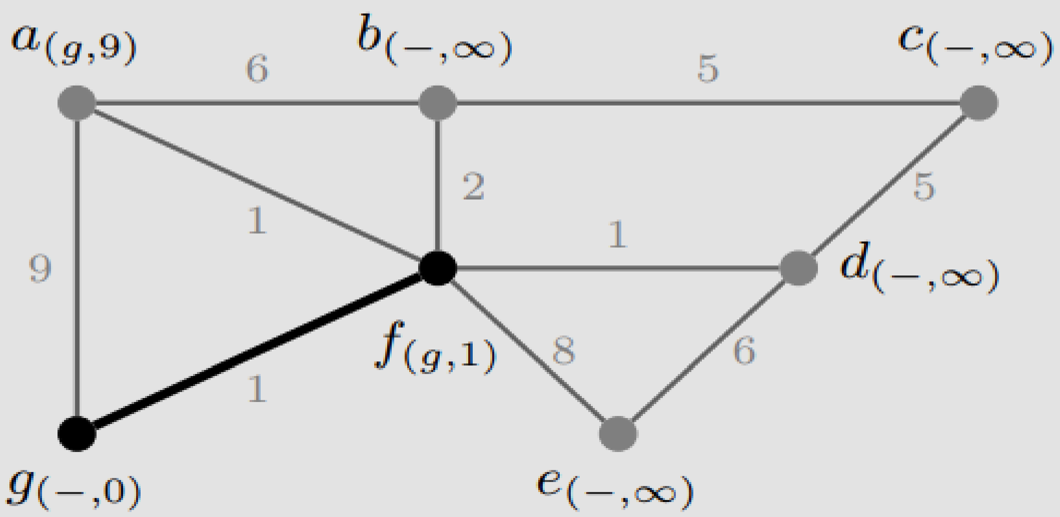
$$\begin{array}{c|l} a & (-,\infty) \\ b & (-,\infty) \\ c & (-,\infty) \\ d & (-,\infty) \\ e & (-,\infty) \\ f & (-,\infty) \\ g & (-,0) \end{array}$$

Dijkstra's Algorithm

Step 2: Let $u = g$. Then the neighbors of g comprise $F = \{a, f\}$. We compute

$$w(g) + w(ga) = 0 + 9 = 9 < \infty = w(a)$$
$$w(g) + w(gf) = 0 + 1 = 1 < \infty = w(f)$$

Update $L(a) = (g, 9)$ and $L(f) = (g, 1)$. Since the minimum weight for all vertices in F is that of f, we highlight the edge gf and the vertex f.



$$F = \{a, f\}$$

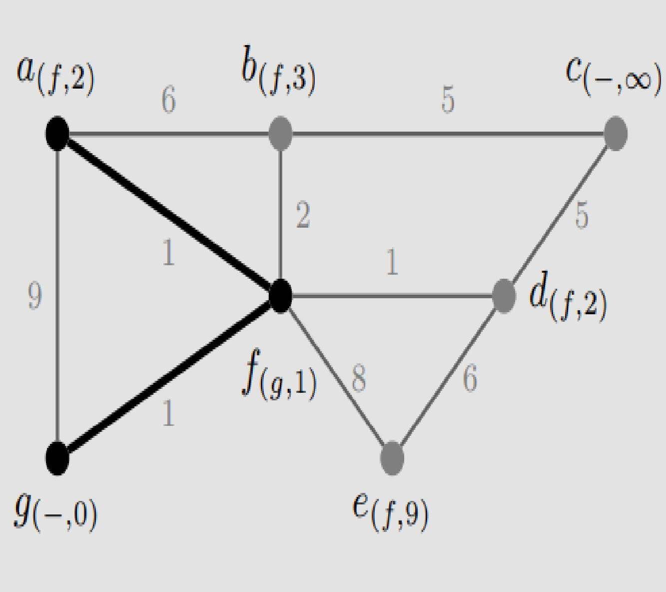
$$\begin{array}{c|l} a & (-,\infty) \to (g,9) \\ b & (-,\infty) \\ c & (-,\infty) \\ d & (-,\infty) \\ e & (-,\infty) \\ f & (-,\infty) \to (g,1) \\ g & (-,0) \end{array}$$

Dijkstra's Algorithm

Step 3: Let $u = f$. Then the neighbors of all highlighted vertices are $F = \{a, b, d, e\}$. We compute

$$w(f) + w(fa) = 1 + 1 = 2 < 9 = w(a)$$
$$w(f) + w(fb) = 1 + 2 = 3 < \infty = w(b)$$
$$w(f) + w(fd) = 1 + 1 = 2 < \infty = w(d)$$
$$w(f) + w(fe) = 1 + 8 = 9 < \infty = w(e)$$

Update $L(a) = (f, 2)$, $L(b) = (f, 3)$, $L(d) = (f, 2)$ and $L(e) = (f, 9)$. Since the minimum weight for all vertices in F is that of a or d, we choose to highlight the edge fa and the vertex a.



$F = \{a, b, d, e\}$

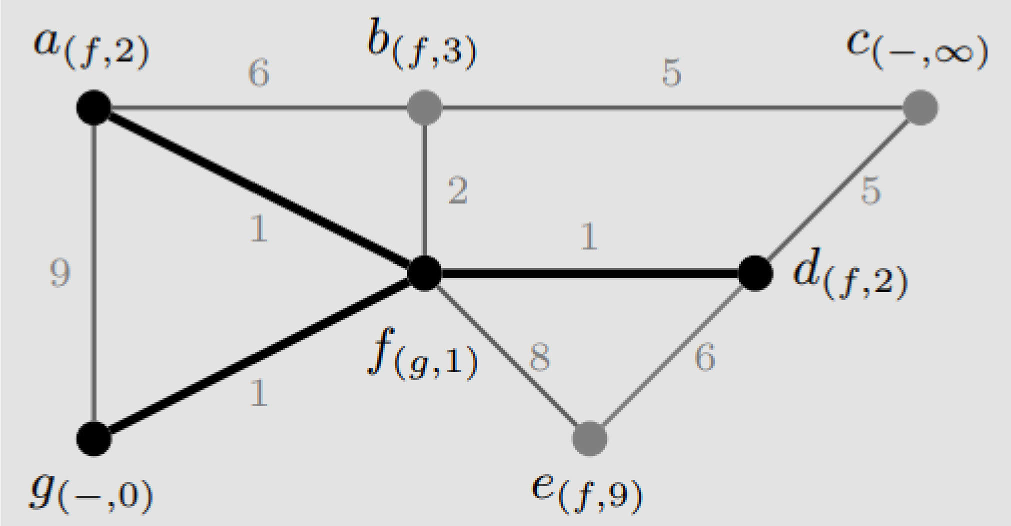
$$\begin{array}{c|l} a & (g, 9) \rightarrow (f, 2) \\ b & (-, \infty) \rightarrow (f, 3) \\ c & (-, \infty) \\ d & (-, \infty) \rightarrow (f, 2) \\ e & (-, \infty) \rightarrow (f, 9) \\ f & (g, 1) \\ g & (-, 0) \end{array}$$

Dijkstra’s Algorithm

Step 4: Let $u = a$. Then the neighbors of all highlighted vertices are $F = \{b, d, e\}$. Note, we only consider updating the label for b since this is the only vertex adjacent to a, the vertex highlighted in the previous step.

$$w(a) + w(ba) = 2 + 6 = 8 \not< 2 = w(b)$$

We do not update the label for b since the computation above is not less than the current weight of b. The minimum weight for all vertices in F is that of d, and so we highlight the edge fd and the vertex d.



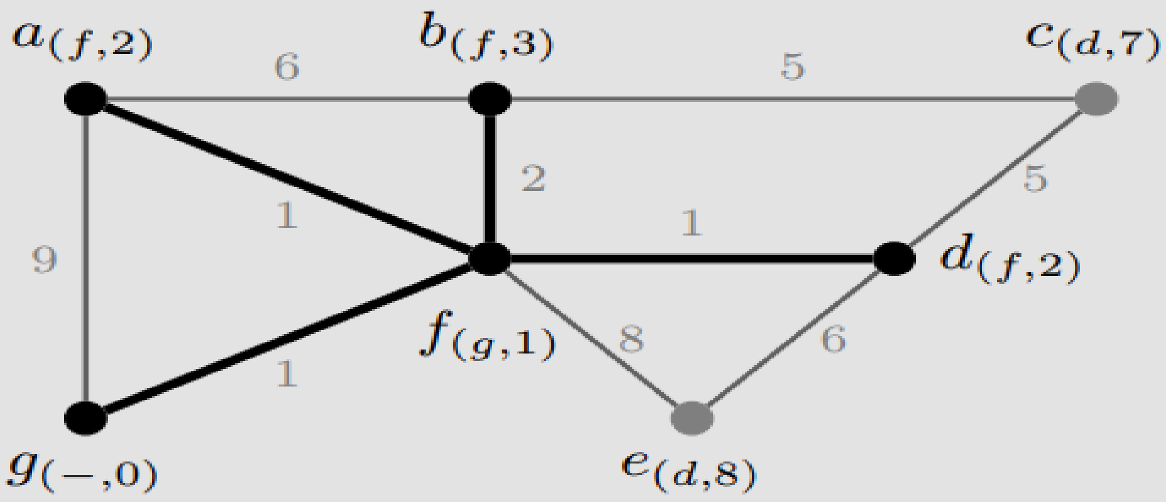
$F = \{b, d, e\}$	
a	$(f, 2)$
b	$(f, 3)$
c	$(-, \infty)$
d	$(f, 2)$
e	$(f, 9)$
f	$(g, 1)$
g	$(-, 0)$

Dijkstra's Algorithm

Step 5: Let $u = d$. Then the neighbors of all highlighted vertices are $F = \{b, c, e\}$. We compute

$$w(d) + w(dc) = 2 + 5 = 7 < \infty = w(c)$$
$$w(d) + w(de) = 2 + 6 = 8 < 9 = w(e)$$

Update $L(c) = (d, 7)$ and $L(e) = (d, 8)$. Since the minimum weight for all vertices in F is that of b, we highlight the edge bf and the vertex b.



$F = \{b, c, e\}$

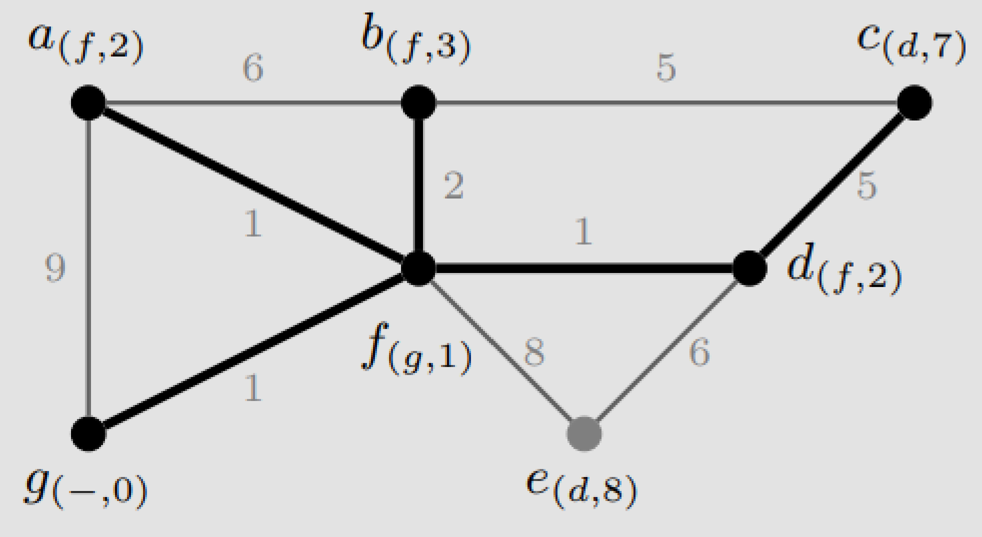
a	$(f, 2)$
b	$(f, 3)$
c	$(-, \infty) \rightarrow (d, 7)$
d	$(f, 2)$
e	$(f, 9) \rightarrow (d, 8)$
f	$(g, 1)$
g	$(-, 0)$

Dijkstra's Algorithm

Step 6: Let $u = b$. Then the neighbors of all highlighted vertices are $F = \{c, e\}$. However, we only consider updating the label of c since e is not adjacent to b. Since

$$w(b) + w(bc) = 3 + 5 = 8 \not< 7 = w(c)$$

we do not update the labels of any vertices. Since the minimum weight for all vertices in F is that of c we highlight the edge dc and the vertex c. This terminates the iterations of the algorithm since our ending vertex has been reached.

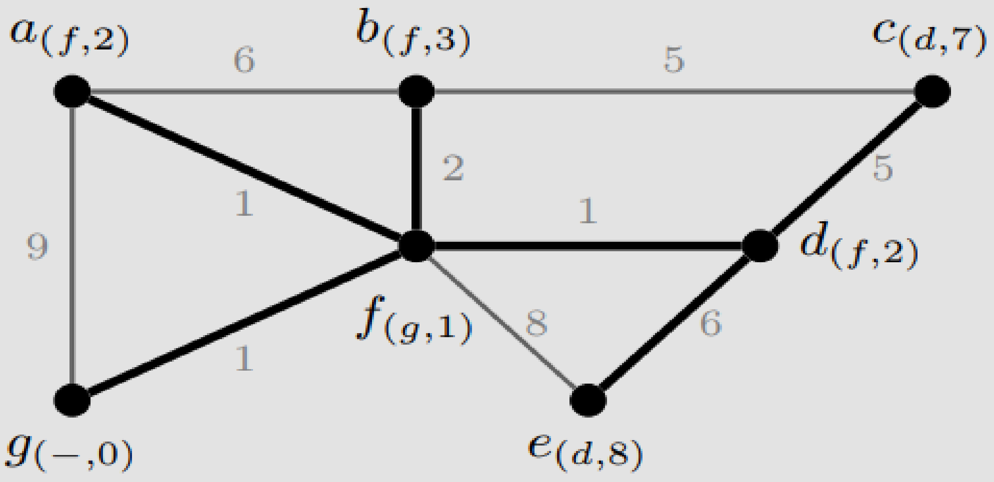


$F = \{c, e\}$

a	$(f, 2)$
b	$(f, 3)$
c	$(d, 7)$
d	$(f, 2)$
e	$(d, 8)$
f	$(g, 1)$
g	$(-, 0)$

Dijkstra's Algorithm

Step 7: Let $u = c$. Then the neighbors of all highlighted vertices are $F = \{e\}$. However, we do not need to update any labels since c and e are not adjacent. Thus we highlight the edge de and the vertex e. This terminates the iterations of the algorithm since all vertices are now highlighted.

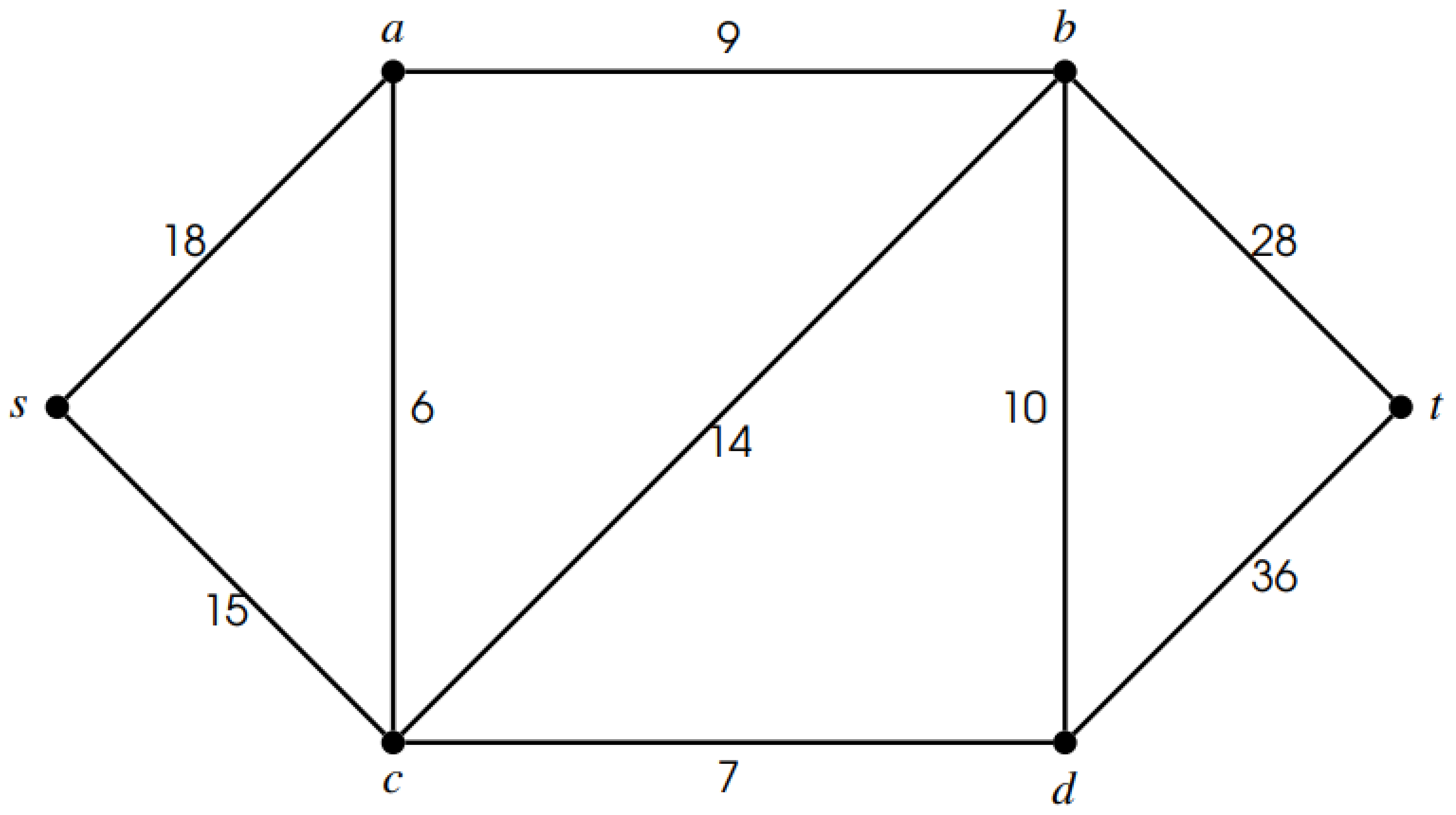


$F = \{e\}$

a	$(f, 2)$
b	$(f, 3)$
c	$(d, 7)$
d	$(f, 2)$
e	$(d, 8)$
f	$(g, 1)$
g	$(-, 0)$

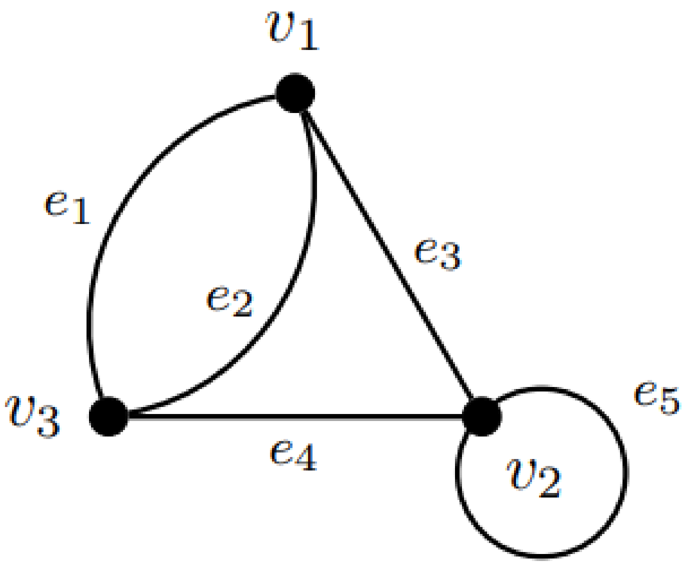
Output: The shortest paths from g to all other vertices can be found highlighted above. For example the shortest path from g to c is $g\,f\,d\,c$ and has a total weight 7, as shown by the label of c.

Dijkstra's Algorithm



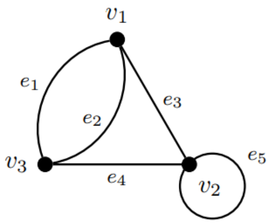
Walks using matrices

- The adjacency matrix is to count the number of walks between two vertices within a graph.
- Consider the graph shown below with its adjacency matrix A on the right.



$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

Walks using matrices



$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

- If we want a walk of length 1, we are in essence asking for an edge between two vertices.
- So to count the number of walks of length 1 from v1 to v3, we need only to count the number of edges between these vertices.

$$v_1 \underset{e_3}{\rightarrow} v_2 \underset{e_4}{\rightarrow} v_3$$

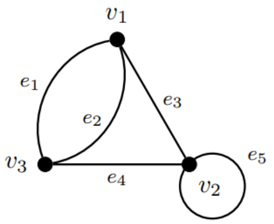
- Now consider the walks from v1 to v2. There is only one walk of length 1, and yet three of length 2

$$v_1 \underset{e_3}{\rightarrow} v_2 \underset{e_5}{\rightarrow} v_2$$

$$v_1 \underset{e_1}{\rightarrow} v_3 \underset{e_4}{\rightarrow} v_2$$

$$v_1 \underset{e_2}{\rightarrow} v_3 \underset{e_4}{\rightarrow} v_2$$

Walks using matrices



$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

v_1 to v_1 :-

$$\left.\begin{array}{l}
v_1 \xrightarrow{e_1} v_3 \xrightarrow{e_2} v_1 \\
v_1 \xrightarrow{e_2} v_3 \xrightarrow{e_1} v_1 \\
v_1 \xrightarrow{e_3} v_2 \xrightarrow{e_3} v_1 \\
v_1 \xrightarrow{e_1} v_3 \xrightarrow{e_1} v_1 \\
v_1 \xrightarrow{e_2} v_3 \xrightarrow{e_2} v_1
\end{array}\right\} 5$$

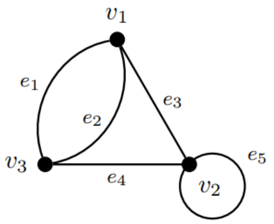
v_2 to v_1 :-

$$\left.\begin{array}{l}
v_2 \xrightarrow{e_5} v_2 \xrightarrow{e_3} v_1 \\
v_2 \xrightarrow{e_4} v_3 \xrightarrow{e_1} v_1 \\
v_2 \xrightarrow{e_4} v_3 \xrightarrow{e_2} v_1
\end{array}\right\} 3$$

v_2 to v_2 :-

$$\left.\begin{array}{l}
v_2 \xrightarrow{e_3} v_1 \xrightarrow{e_3} v_2 \\
v_2 \xrightarrow{e_4} v_3 \xrightarrow{e_4} v_2 \\
v_2 \xrightarrow{e_5} v_2 \xrightarrow{e_5} v_2
\end{array}\right\} 3$$

Walks using matrices



$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

v_2 to v_3 :-

$$\left.\begin{array}{l} v_2 \xrightarrow{e_5} v_2 \xrightarrow{e_4} v_3 \\ v_2 \xrightarrow{e_3} v_1 \xrightarrow{e_2} v_3 \\ v_2 \xrightarrow{e_3} v_1 \xrightarrow{e_1} v_3 \end{array}\right\} 3$$

v_3 to v_1 :-

$$v_3 \xrightarrow{e_4} v_2 \xrightarrow{e_3} v_1 \Big\} \; 1$$

v_3 to v_2 :-

$$\left.\begin{array}{l} v_3 \xrightarrow{e_1} v_1 \xrightarrow{e_3} v_2 \\ v_3 \xrightarrow{e_2} v_1 \xrightarrow{e_3} v_2 \\ v_3 \xrightarrow{e_4} v_2 \xrightarrow{e_5} v_2 \end{array}\right\} 3$$

v_3 to v_3 :-

$$\left.\begin{array}{l} v_3 \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_3 \\ v_3 \xrightarrow{e_2} v_1 \xrightarrow{e_1} v_3 \\ v_3 \xrightarrow{e_4} v_2 \xrightarrow{e_4} v_3 \\ v_3 \xrightarrow{e_1} v_1 \xrightarrow{e_1} v_3 \\ v_3 \xrightarrow{e_2} v_1 \xrightarrow{e_1} v_3 \end{array}\right\} 5$$

for length-2

$$\begin{array}{c c} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{bmatrix} 5 & 3 & 1 \\ 3 & 3 & 3 \\ 1 & 3 & 5 \end{bmatrix} \end{array}$$

Walks using matrices

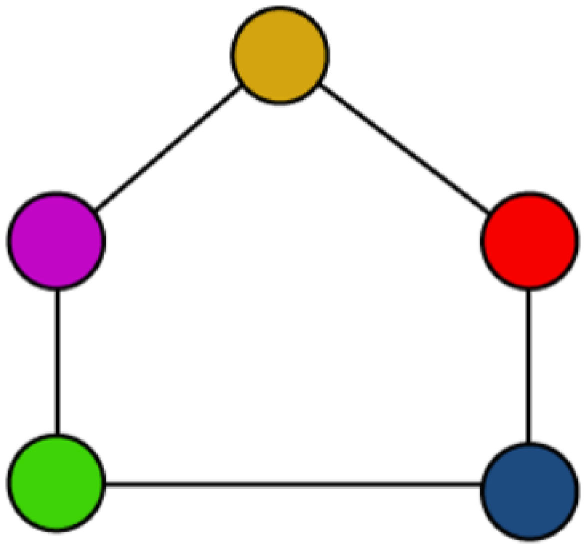
Theorem : Let G be a graph with adjacency matrix A. Then for any integer $n > 0$ the entry a_{ij} in A^n counts the number of walks from v_i to v_j .

- The entry a_{ij} in A^2 represents the number of walks between vertex v_i and v_j of length 2.

$$A^2 = \begin{bmatrix} 5 & 3 & 1 \\ 3 & 3 & 3 \\ 1 & 3 & 5 \end{bmatrix}$$

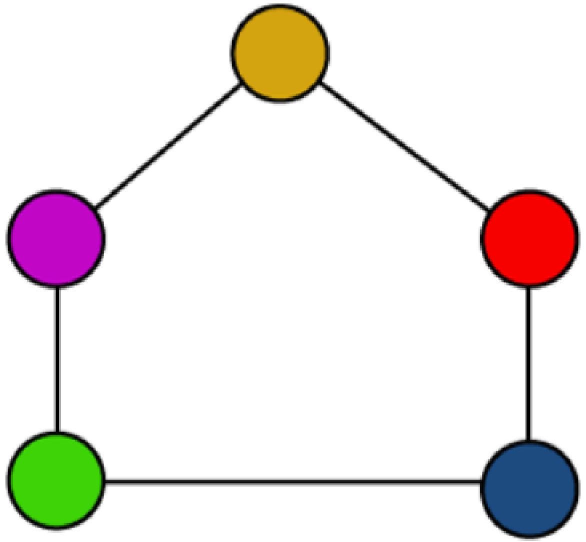
If we multiplied this new matrix by A again, we would simply be counting the number of ways to get from v_i to v_j using 3 edges.

- Graph coloring can be described as a process of assigning colors to the vertices of a graph.
- In this, the same color should not be used to fill the two adjacent vertices. We can also call graph coloring as Vertex Coloring.
- In graph coloring, we have to take care that a graph must not contain any edge whose end vertices are colored by the same color.
- Any graph we consider can be simple or have multi-edges but cannot have loops, since a vertex with a loop could never be assigned a color.
- This type of graph is known as the Properly colored graph.



Example of Graph coloring

In this graph, we are showing the properly colored graph, which is described as follows:



The above graph contains some points, which are described as follows:

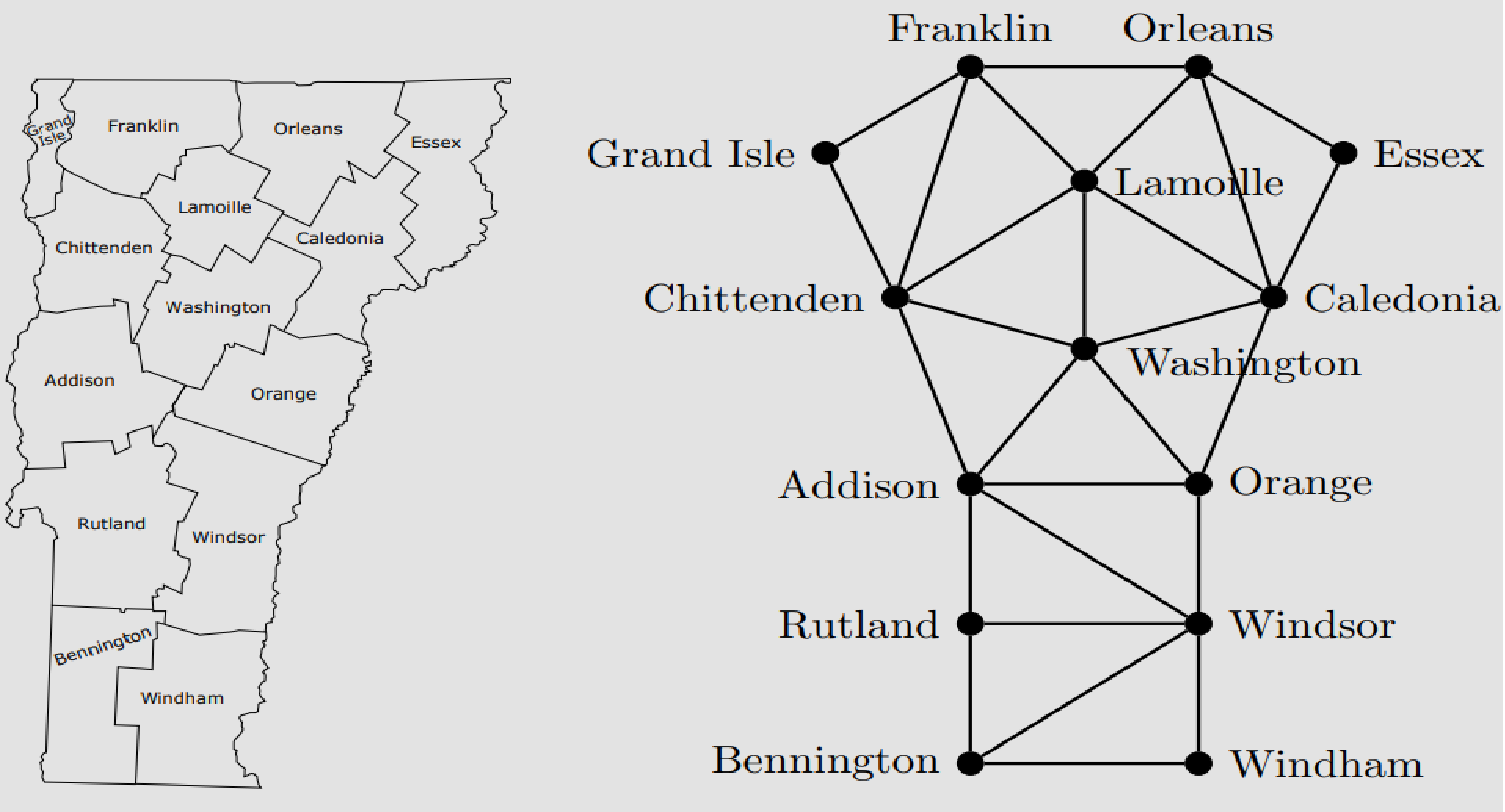
- The same color cannot be used to color the two adjacent vertices.
- Hence, we can call it as a properly colored graph.

Applications of Graph coloring

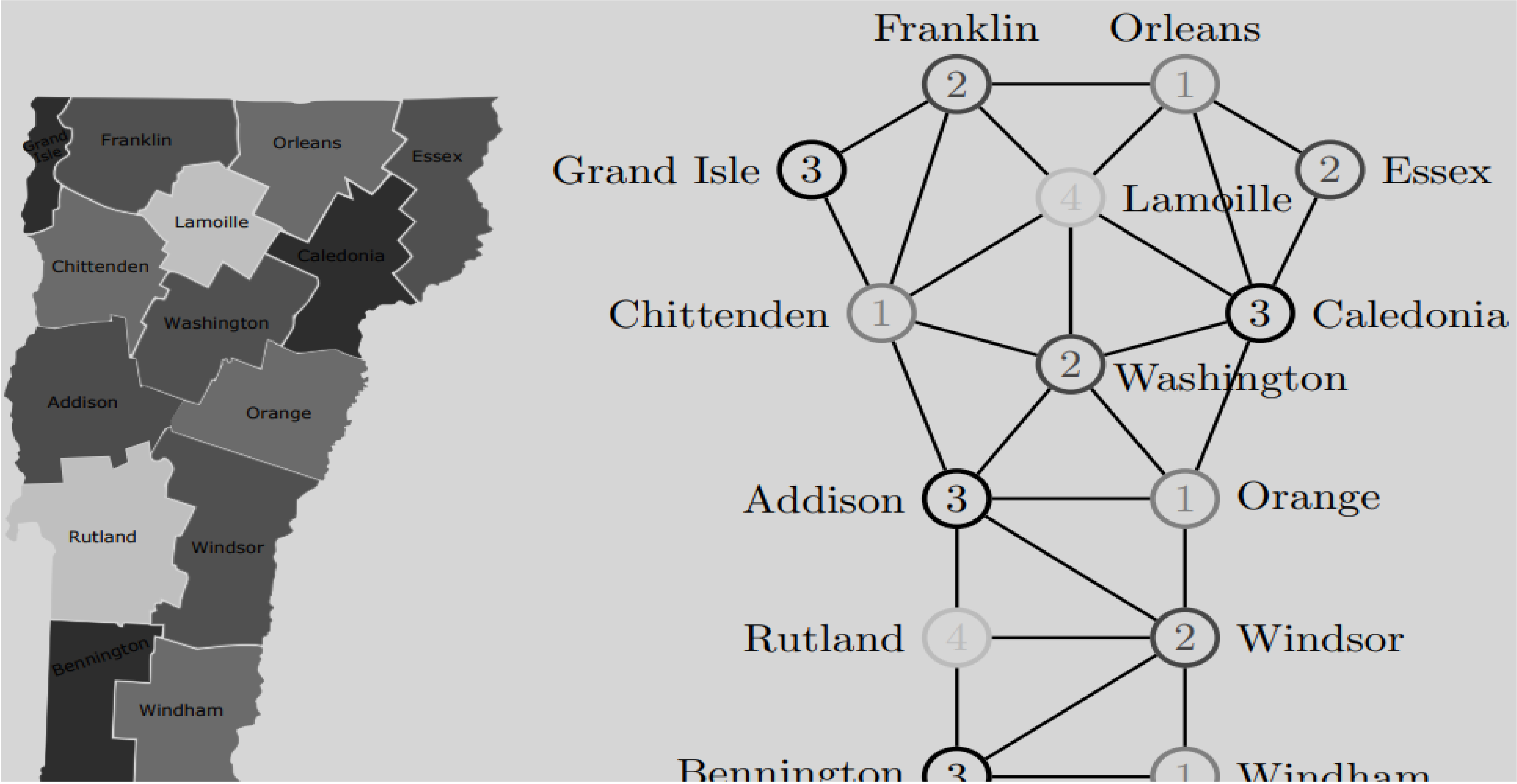
There are various applications of graph coloring. Some of their important applications are described as follows:

- Assignment
- Map coloring
- Scheduling the tasks
- Sudoku
- Prepare time table
- Conflict resolution

Find a coloring of the map of the counties of Vermont and explain why three colors will not suffice



Solution: First note that each county is given a vertex and two vertices are adjacent in the graph when their respective counties share a border. One possible coloring is shown below.



Four Color Theorem

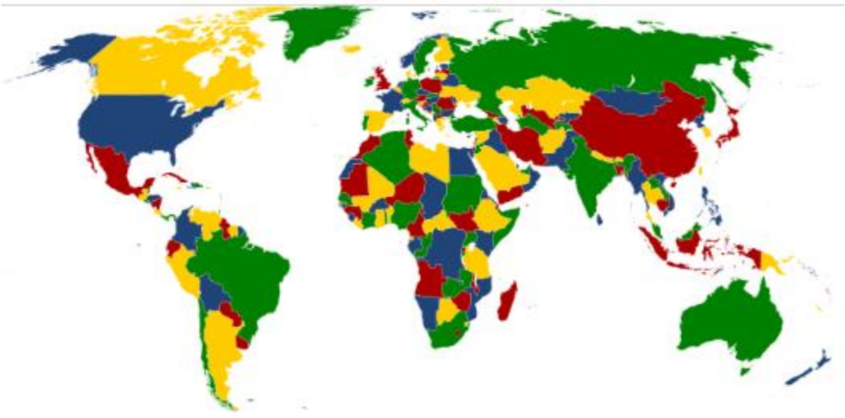
- The Lamoille County is surrounded by five other counties. If we try to alternate colors amongst these five counties, for example Orleans – 1, Franklin – 2, Chittenden – 1, Washington – 2, we still need a third color for the fifth county (Caledonia – 3).
- Since Lamoille touches each of these counties, we know it needs a fourth color.

Four Color Theorem

The **four color theorem** states that any map--a division of the plane into any number of regions--can be colored using no more than four colors in such a way that no two adjacent regions share the same color.

The four color theorem is particularly notable for being the first major theorem proved by a computer.

- Given a proper k-coloring of G, the color classes are sets S1, . . . Sk where S_i consists of all vertices of color i.
- The independence number of a graph G is $\alpha(G) = n$ if there exists a set of n vertices with no edges between them but every set of n + 1 vertices contains at least one edge.



A map of the world, colored using four colors

Chromatic Number

Definition: The chromatic number $\chi(G)$ of a graph is the smallest value k for which G has a proper k-coloring.

- The chromatic number can be described as the minimum number of colors required to properly color any graph.
- In other words, the chromatic number can be described as a minimum number of colors that are needed to color any graph in such a way that no two adjacent vertices of a graph will be assigned the same color.
- In order to determine the chromatic number of a graph, we often need to

complete the following two steps:

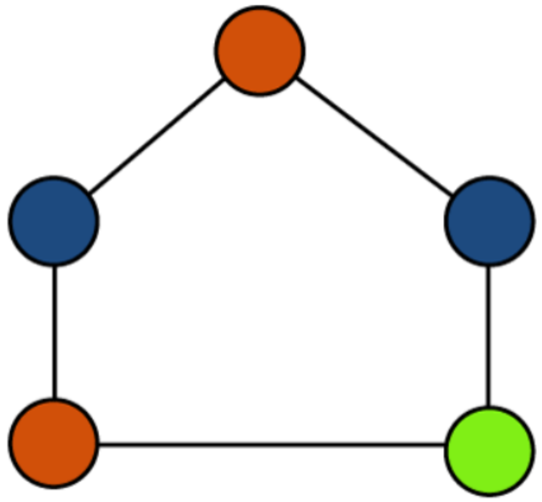
(1) Find a vertex coloring of G using k colors.

(2) Show why fewer colors will not suffice.

Example of Chromatic number:

To understand the chromatic number, we will consider a graph, which is described as follows:

The graph contains some points, which are described as follows:



- The same color is not used to color the two adjacent vertices.
- The minimum number of colors of this graph is 3, which is needed to properly color the vertices.
- Hence, in this graph, the chromatic number = 3
- If we want to properly color this graph, in this case, we are required at least 3 colors.

Types of Chromatic Number of Graphs:

There are various types of chromatic number of graphs, which are described as follows:

Cycle Graph:

A graph will be known as a cycle graph if it contains 'n' edges and 'n' vertices (n >= 3), which form a cycle of length 'n'. There can be only 2 or 3 number of degrees of all the vertices in the cycle graph.

Chromatic number:

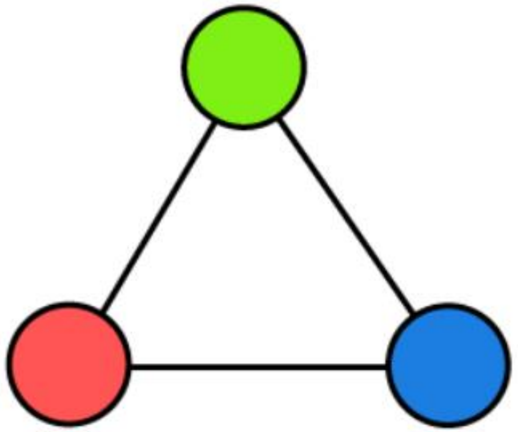
1.The chromatic number in a cycle graph will be 2 if the number of vertices in that graph is even.

2.The chromatic number in a cycle graph will be 3 if the number of vertices in that graph is odd.

Examples of Cycle graph:

There are various examples of cycle graphs. Some of them are described as follows:

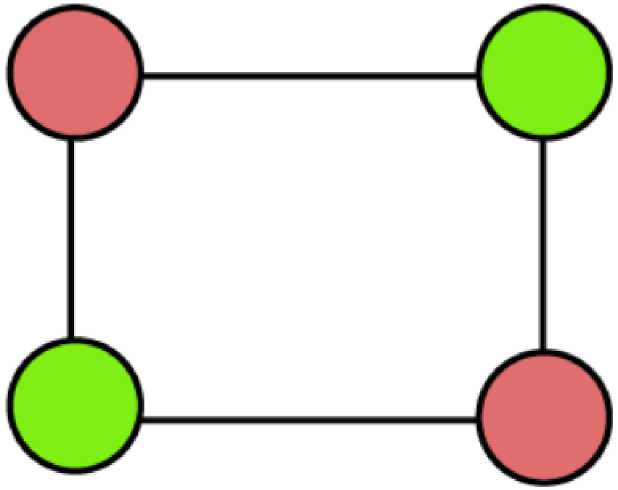
Example 1: In the following graph, we have to determine the chromatic number.



Solution: In the above cycle graph, there are 3 different colors for three vertices, and none of the adjacent vertices are colored with the same color. In this graph, the number of vertices is odd.

So Chromatic number = 3

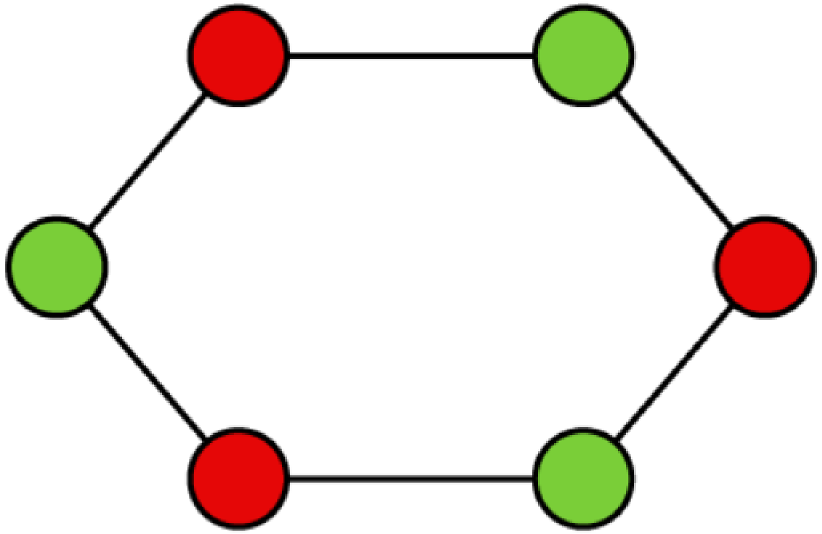
Example 2: In the following graph, we have to determine the chromatic number.



Solution: In the above cycle graph, there are 2 colors for four vertices, and none of the adjacent vertices are colored with the same color. In this graph, the number of vertices is even.

So Chromatic number = 2

Example 4: In the following graph, we have to determine the chromatic number.

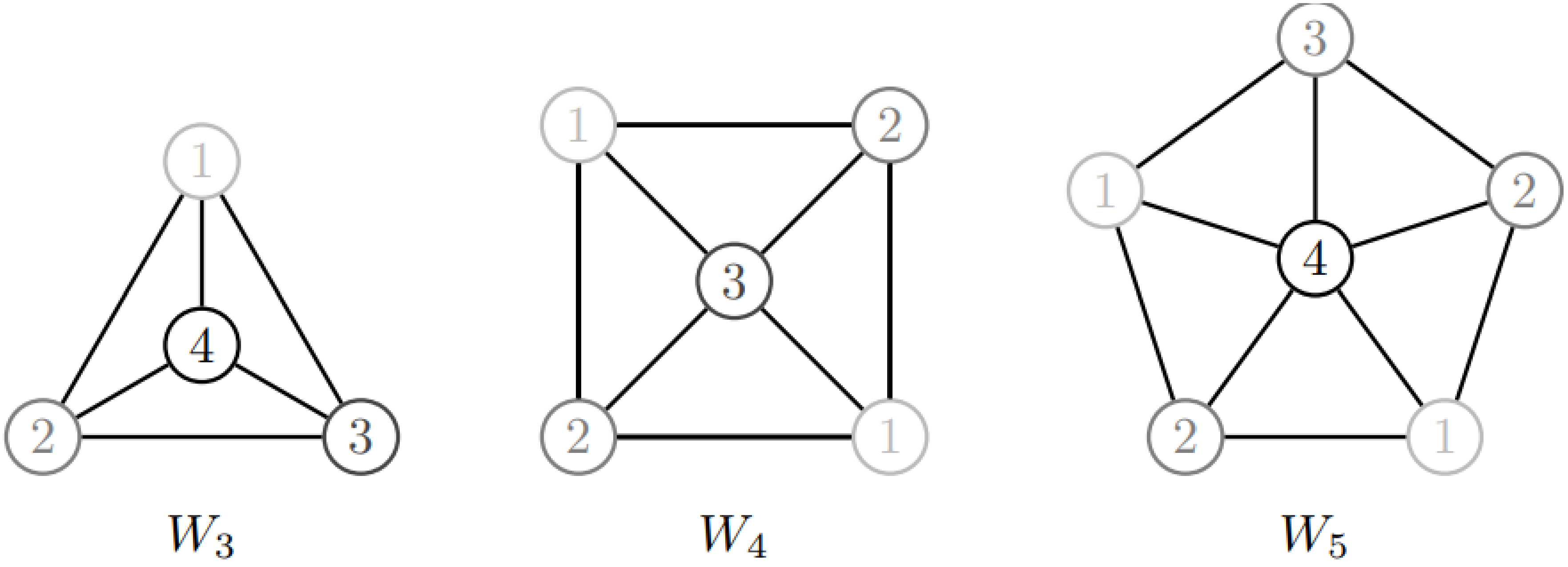


Solution: In the above graph, there are 2 different colors for six vertices, and none of the adjacent vertices are colored with the same color. In this graph, the number of vertices is even.

So Chromatic number = 2

A wheel Wn is a graph in which n vertices form a cycle around a central vertex that is adjacent to each of the vertices in the cycle.

Examples are:



when n is odd, requires 4 colors. W_3, 5 chromatic number is 4 and W_4 is 3

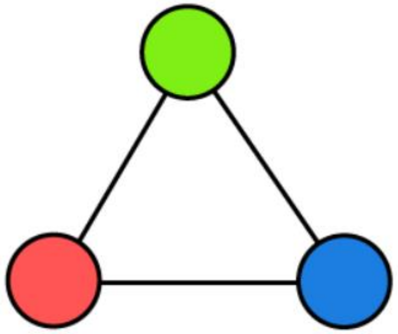
Planar Graph

A graph will be known as a planner graph if it is drawn in a plane. The edges of the planar graph must not cross each other.

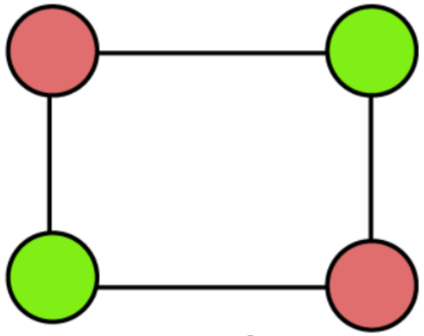
Chromatic Number:

1.In a planner graph, the chromatic Number must be Less than or equal to 4.

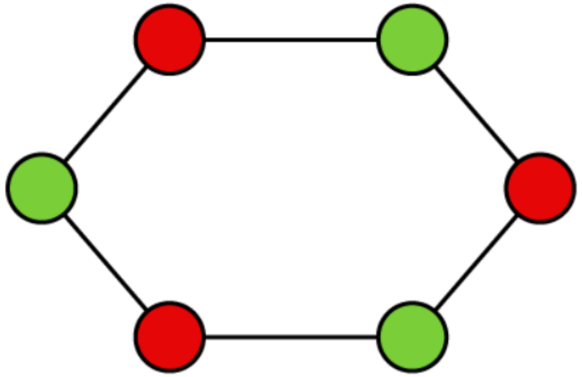
2.The planner graph can also be shown by all the above cycle graphs .



Ex: 1



Ex: 2



Ex: 4

Complete Graph

A graph will be known as a complete graph if only one edge is used to join every two distinct vertices. Every vertex in a complete graph is connected with every other vertex. In this graph, every vertex will be colored with a different color. That means in the complete graph, two vertices do not contain the same color.

Chromatic Number

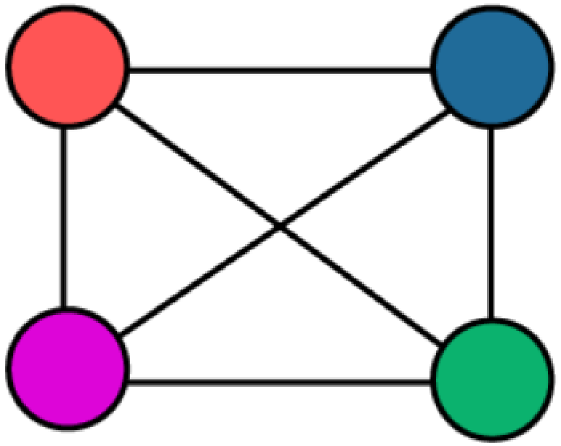
In a complete graph, the chromatic number will be equal to the number of vertices in that graph.

A clique in a graph is a subgraph that is itself a complete graph. The clique size of a graph G, denoted $\omega(G)$, is the largest value of n for which G contains Kn as a subgraph.

Examples of Complete graph:

There are various examples of complete graphs. Some of them are described as follows:

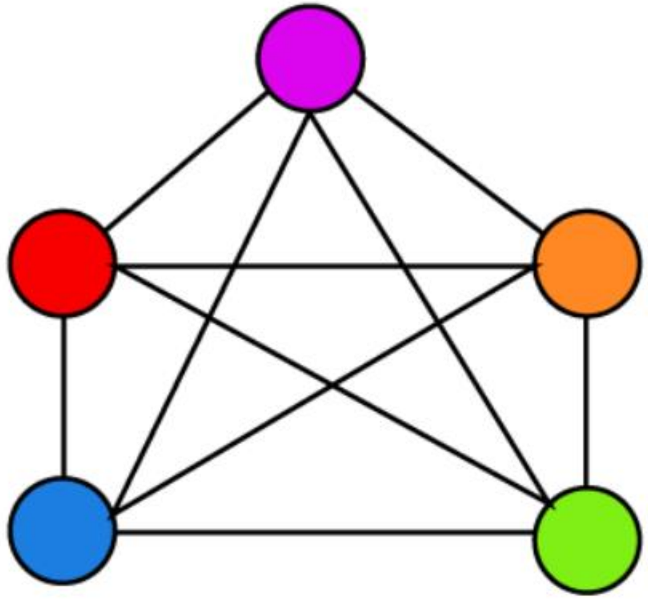
Example 1: In the following graph, we have to determine the chromatic number.



Solution: There are 4 different colors for 4 different vertices, and none of the colors are the same in the above graph. According to the definition, a chromatic number is the number of vertices.

So, Chromatic number = 4

Example 2: In the following graph, we have to determine the chromatic number.



Solution: There are 5 different colors for 5 different vertices, and none of the colors are the same in the above graph. According to the definition, a chromatic number is the number of vertices.

So, Chromatic number = 5

Bipartite Graph

A graph will be known as a bipartite graph if it contains two sets of vertices, A and B. The vertex of A can only join with the vertices of B. That means the edges cannot join the vertices with a set.

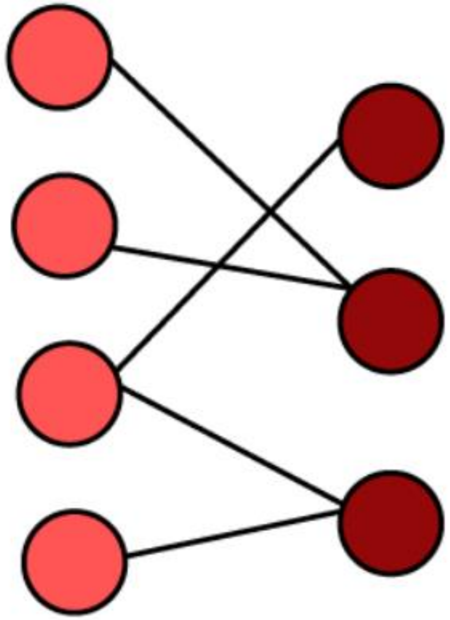
Chromatic Number

In any bipartite graph, the chromatic number is always equal to 2.

Example of Bipartite graph:

There are various examples of bipartite graphs. Some of them are described as follows:

Example: In the following graph, we have to determine the chromatic number.



Solution: There are 2 different sets of vertices in the above graph. So the chromatic number of all bipartite graphs will always be 2.

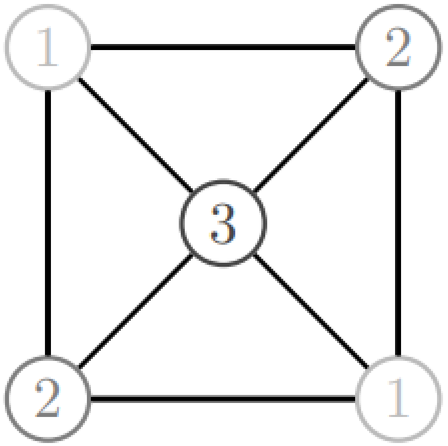
So, Chromatic number = 2

Chromatic Numbers, we have learned the following things:

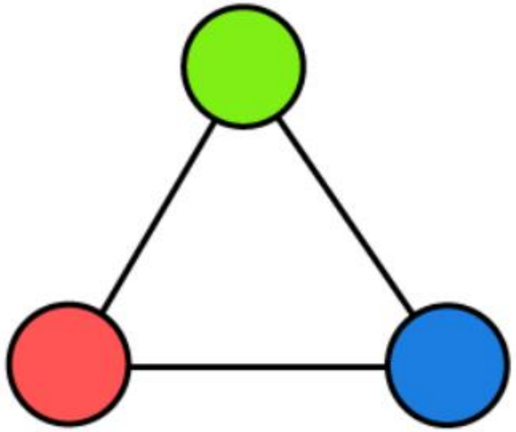
Special Classes of Graphs with known $\chi(G)$

- $\chi(C_n) = 2$ if n is even $(n \geq 2)$
- $\chi(C_n) = 3$ if n is odd $(n \geq 3)$
- $\chi(K_n) = n$
- $\chi(W_n) = 4$ if n is odd $(n \geq 3)$

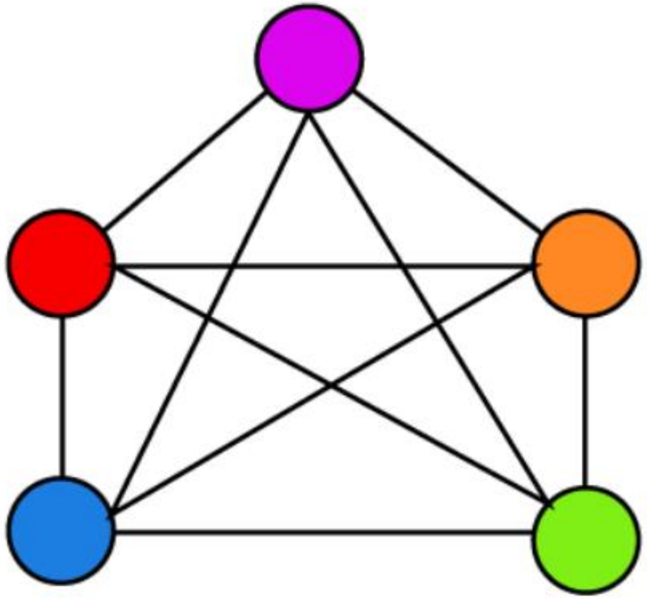
Brooks' Theorem : Let G be a connected graph and Δ denote the maximum degree among all vertices in G. Then χ(G) ≤ Δ as long as G is not a complete graph or an odd cycle. If G is a complete graph or an odd cycle then χ(G) = Δ + 1.



Wheel graph with 4 vertices



odd cycle graph



complete graph

Vertex coloring

Basic Coloring Strategies

• Begin with vertices of high degree.

• Look for locations where colors are forced (cliques, wheels, odd cycles) rather than chosen.

• When these strategies have been exhausted, color the remaining vertices while trying to avoid using any additional **colors.**

.

Graph Coloring Algorithm

- If we want to color a graph with the help of a minimum number of colors, for this, there is no efficient algorithm.
- Graph coloring is also known as the NP-complete algorithm.

However, we can find the chromatic number of the graph with the help of following greedy algorithm.

There are various steps to solve the greedy algorithm, which are described as follows:

Step 1: In the first step, we will color the first vertex with first color.

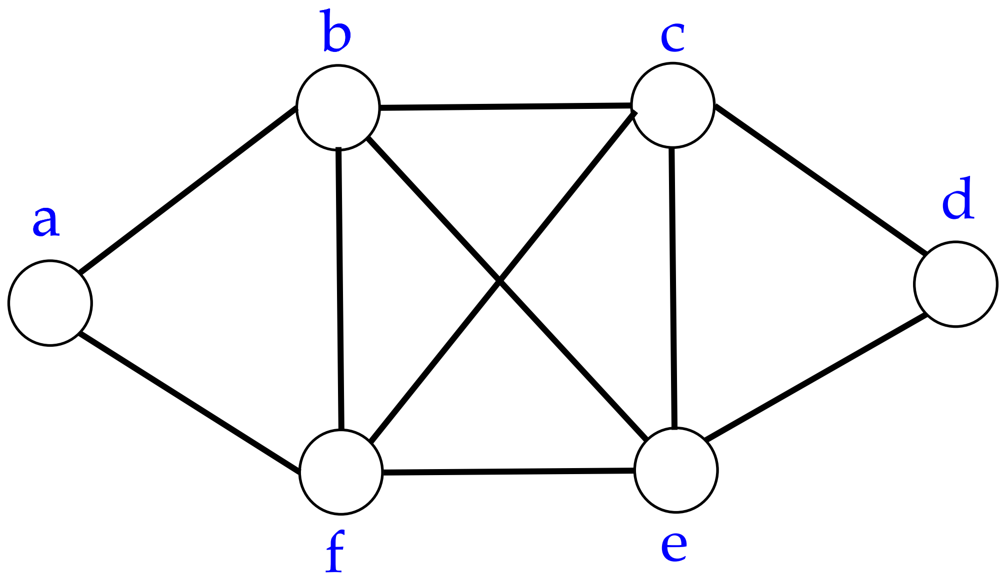
Step 2: Now, we will one by one consider all the remaining vertices (V -1) and do the following:

- We will color the currently picked vertex with the help of lowest number color if and only if the same color is not used to color any of its adjacent vertices.
- If its adjacent vertices are using it, then we will select the next least numbered color.
- If we have already used all the previous colors, then a new color will be used to fill or assign to the currently picked vertex.

Examples of finding Chromatic number of a Graph

There are a lot of examples to find out the chromatic number in a graph. Some of them are described as follows:

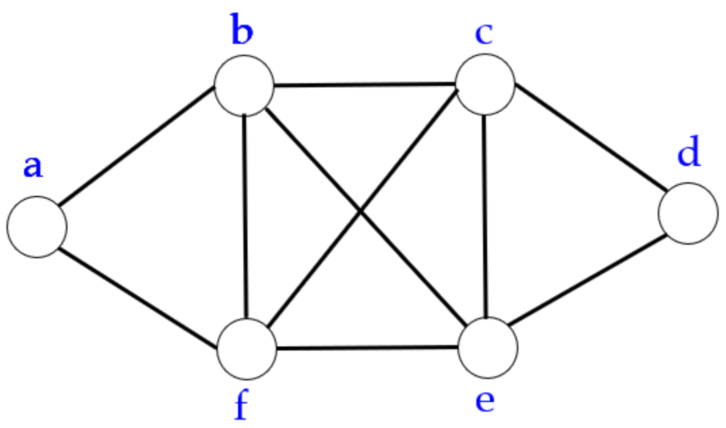
Example 1: In this example, we have a graph, and we have to determine the chromatic number of this graph.



Solution:

When we apply the greedy algorithm, we will have the following:

Vertex	a	b	c	d	e	f
Color	C1	C2	C1	C2	C3	C4

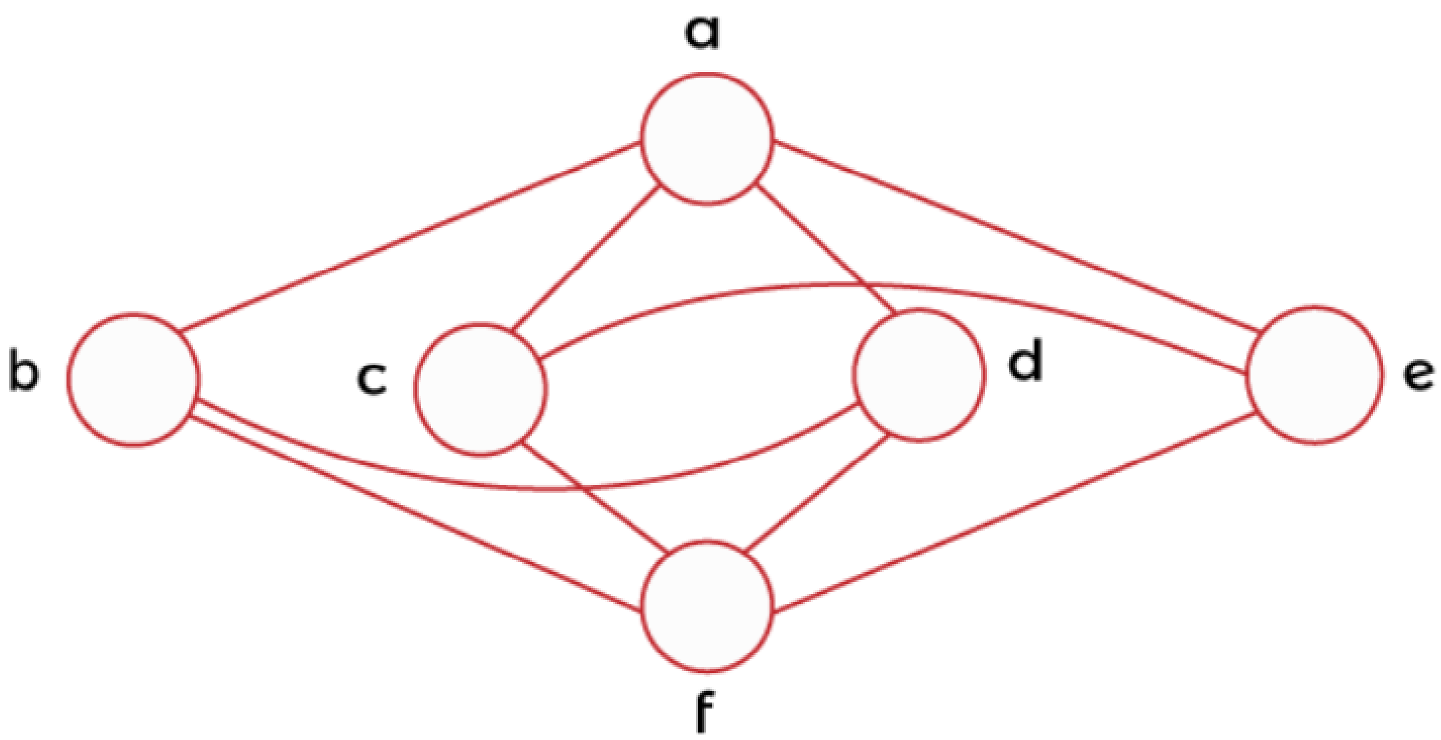


From the above table,

- In the above graph, we are required minimum 4 numbers of colors to color the graph.
- Therefore, we can say that the Chromatic number of above graph = 4

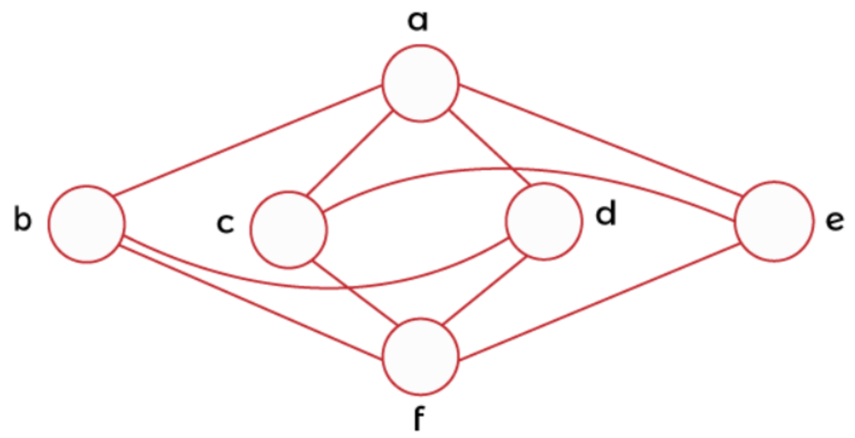
So with the help of 4 colors, the above graph can be properly colored like this:

Example 2: In this example, we have a graph, and we have to determine the chromatic number of this graph.



Solution:

When we apply the greedy algorithm, we will have the following:

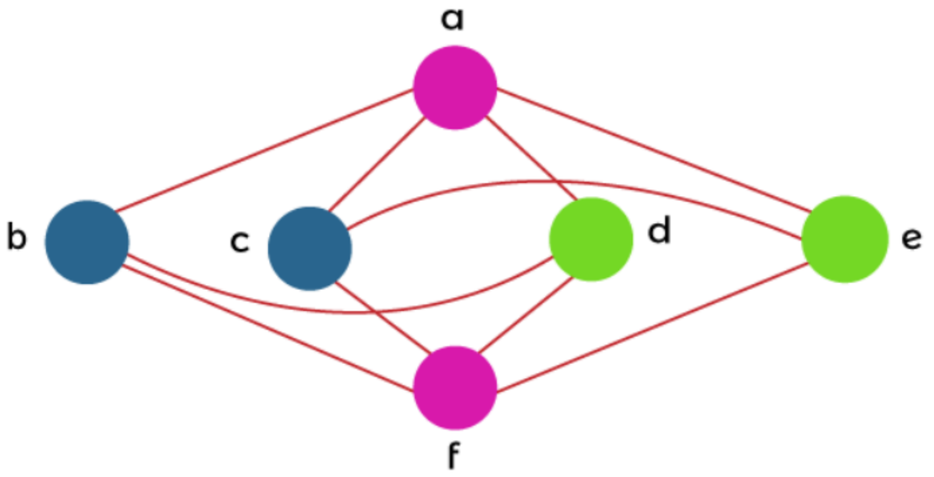


Vertex	a	b	c	d	e	f
Color	C1	C2	C2	C3	C3	C1

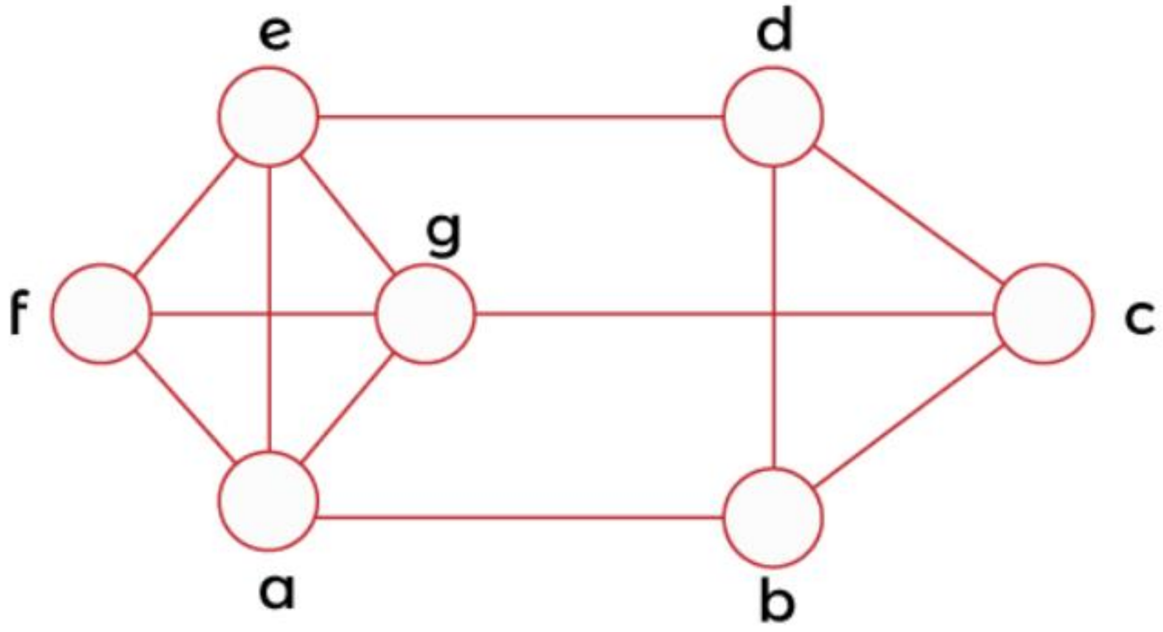
From the above table,

- In the above graph, we are required minimum 3 numbers of colors to color the graph.
- Therefore, we can say that the Chromatic number of above graph = 3

So with the help of 3 colors, the above graph can be properly colored like this:

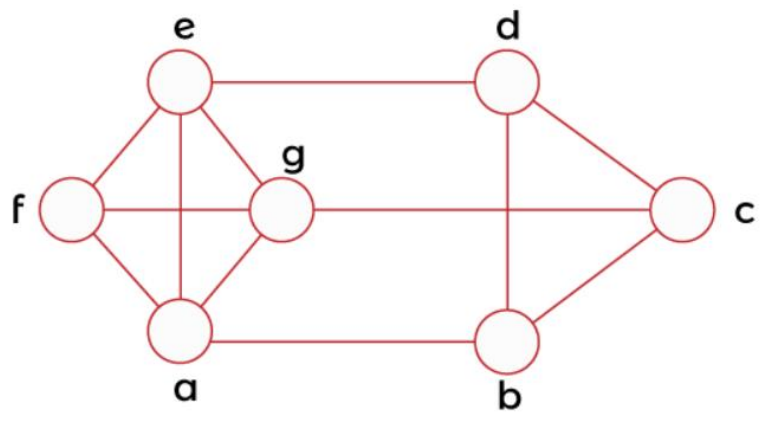


Example 3: In this example, we have a graph, and we have to determine the chromatic number of this graph.



Solution:

When we apply the greedy algorithm, we will have the following:

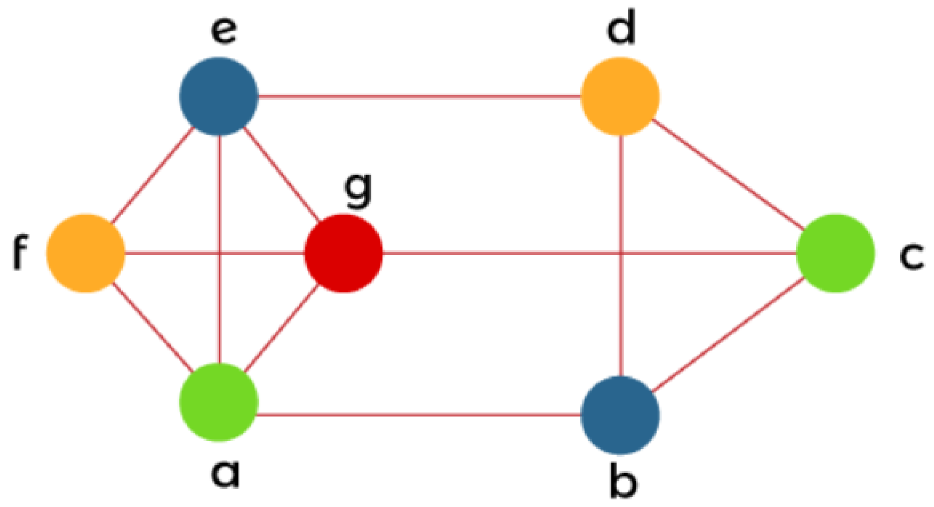


Vertex	a	b	c	d	e	f	g
Color	C1	C2	C1	C3	C2	C3	C4

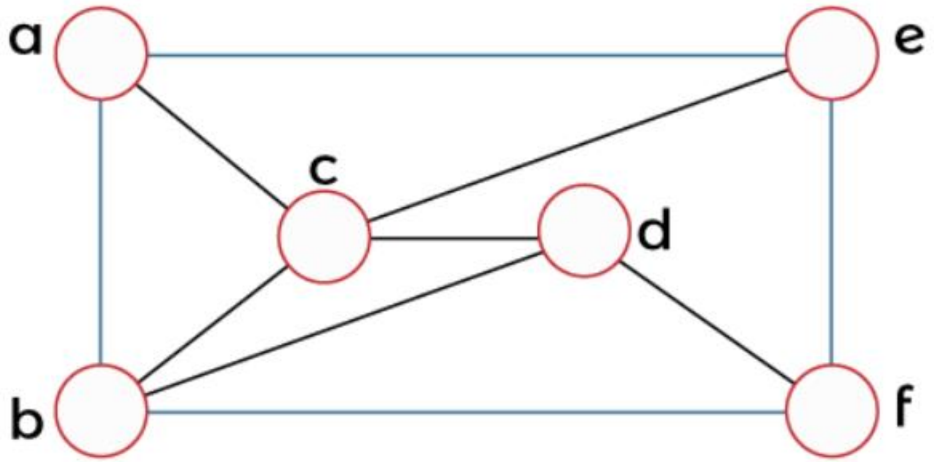
From the above table,

- In the above graph, we are required minimum 4 numbers of colors to color the graph.
- Therefore, we can say that the Chromatic number of above graph = 4

So with the help of 4 colors, the above graph can be properly colored like this:

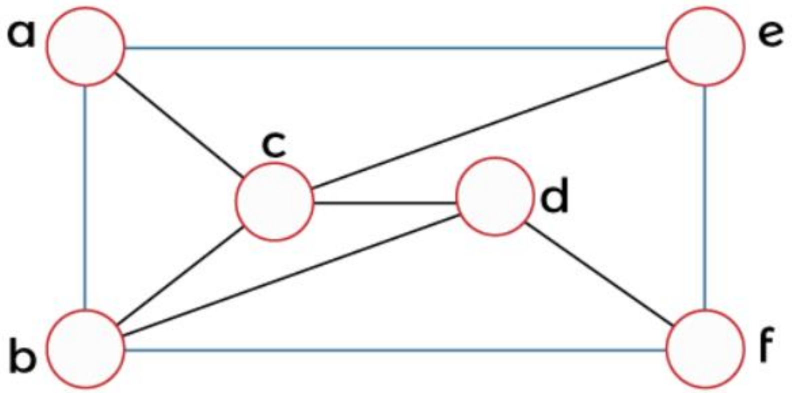


Example 4: In this example, we have a graph, and we have to determine the chromatic number of this graph.



Solution:
When we apply the greedy algorithm, we will have the following:

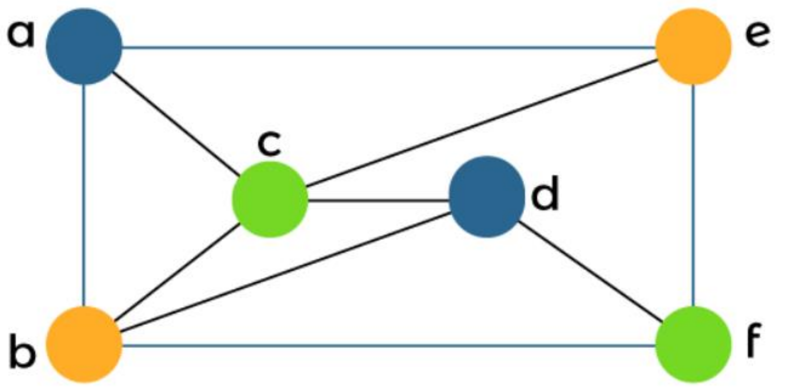
Vertex	a	b	c	d	e	f
Color	C1	C2	C3	C1	C2	C3



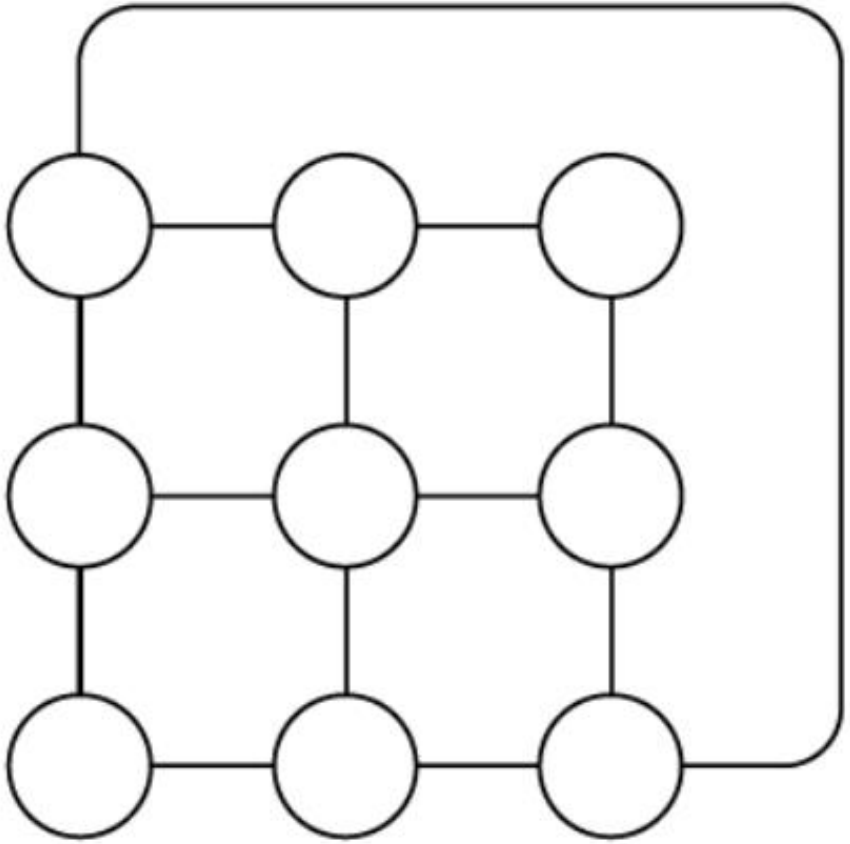
From the above table,

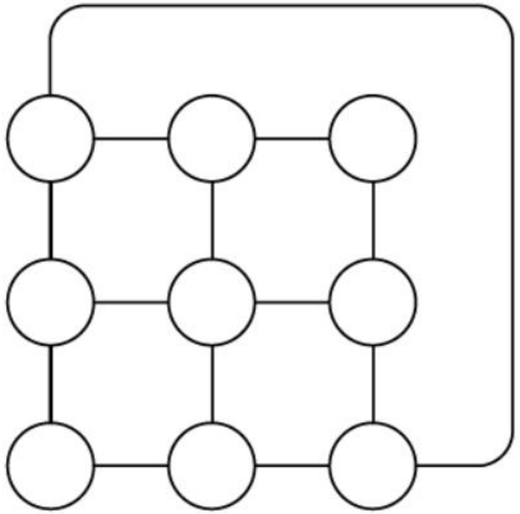
- In the above graph, we are required minimum 3 numbers of colors to color the graph.
- Therefore, we can say that the Chromatic number of above graph = 3

So with the help of 3 colors, the above graph can be properly colored like this:



Example 5: In this example, we have a graph, and we have to determine the chromatic number of this graph.



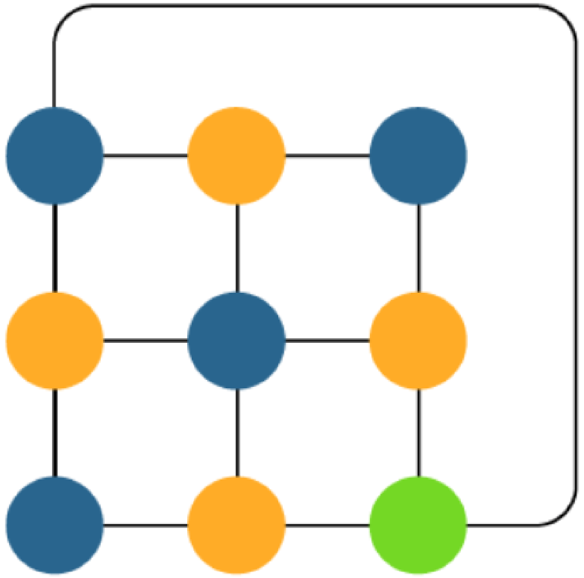


Solution:

When we apply the greedy algorithm, we will have the following:

- In the above graph, we are required minimum 3 numbers of colors to color the graph.
- Therefore, we can say that the Chromatic number of above graph = 3

So with the help of 3 colors, the above graph can be properly colored like this:



Vertex coloring

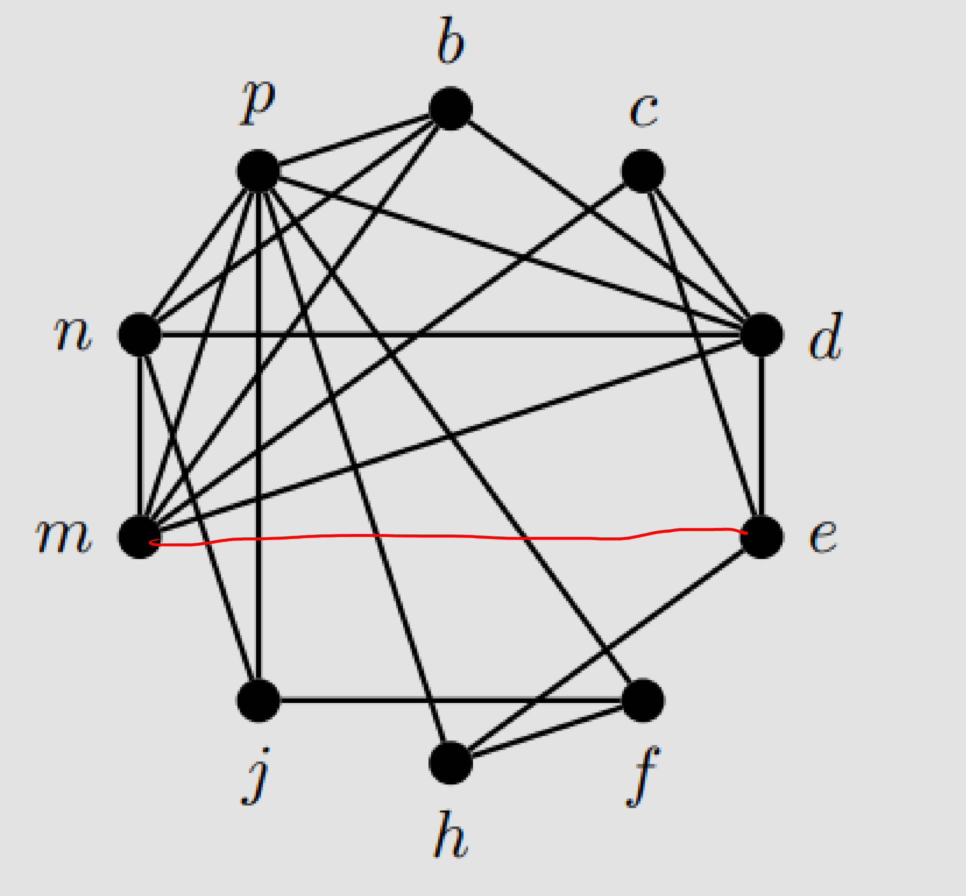
Example: Every year on Christmas Eve, the Petrie family compete in a friendly game of Trivial Pursuit. Unfortunately, due to longstanding disagreements and the outcome of previous years' games, some family members are not allowed on the same team. The table below lists the ten family members competing in this year's Trivial Pursuit game, where an entry of N in the table indicates people who are incompatible. Model the information as a graph and find the minimum number of teams needed to keep the peace this Christmas.

Vertex coloring

	Betty	Carl	Dan	Edith	Frank	Henry	Judy	Marie	Nell	Pete	degree
Betty	·	·	N	·	·	·	·	N	N	N	4
Carl	·	·	N	N	·	·	·	N	·	·	3
Dan	N	N	·	N	·	·	·	N	N	N	6
Edith	·	N	N	·	·	·	·	N	·	·	3
Frank	·	·	·	·	·	N	N	·	·	N	3
Henry	·	·	·	·	N	·	·	·	·	N	2
Judy	·	·	·	·	·	N	·	·	N	N	3
Marie	N	N	N	N	·	·	·	·	N	N	6
Nell	N	·	N	·	·	·	N	N	·	N	5
Pete	N	·	N	·	N	N	N	N	N	·	7

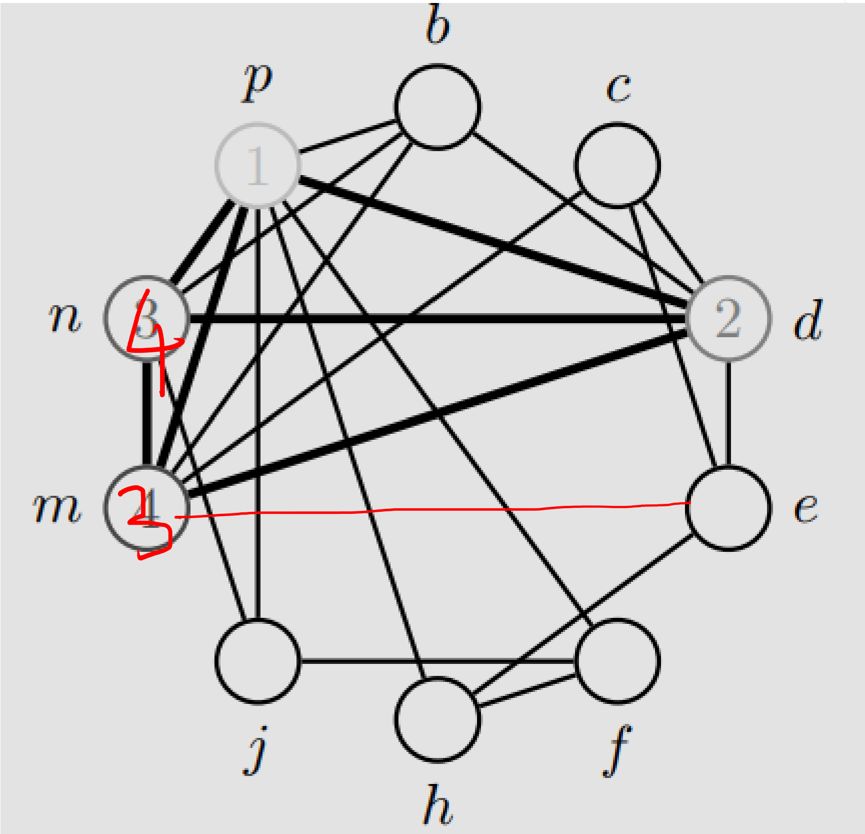
Vertex coloring

Solution: Each person will be represented by a vertex in the graph and an edge indicates two people who are incompatible. Colors will be assigned to the vertices, where each color represents a Trivial Pursuit team.



Vertex coloring

- At our initial step, we want to find a vertex of highest degree (p) and give it color 1.
- Once p has been assigned a color, we look at its neighbors with high degree as well, namely d (degree 6), m and n (both of degree 5). These four vertices are also all adjacent to each other (forming a K4 shown in bold below on the left) and so must use three additional colors.



p_adjacent vertices → {b, d, f, h, j, m, n}

& degrees → {4, 6, 3, 2, 3, 6, 5}

Vertex coloring

b -> { a, m, n, p } -> colors { 2, 3, 4, 1 }
so assign 5

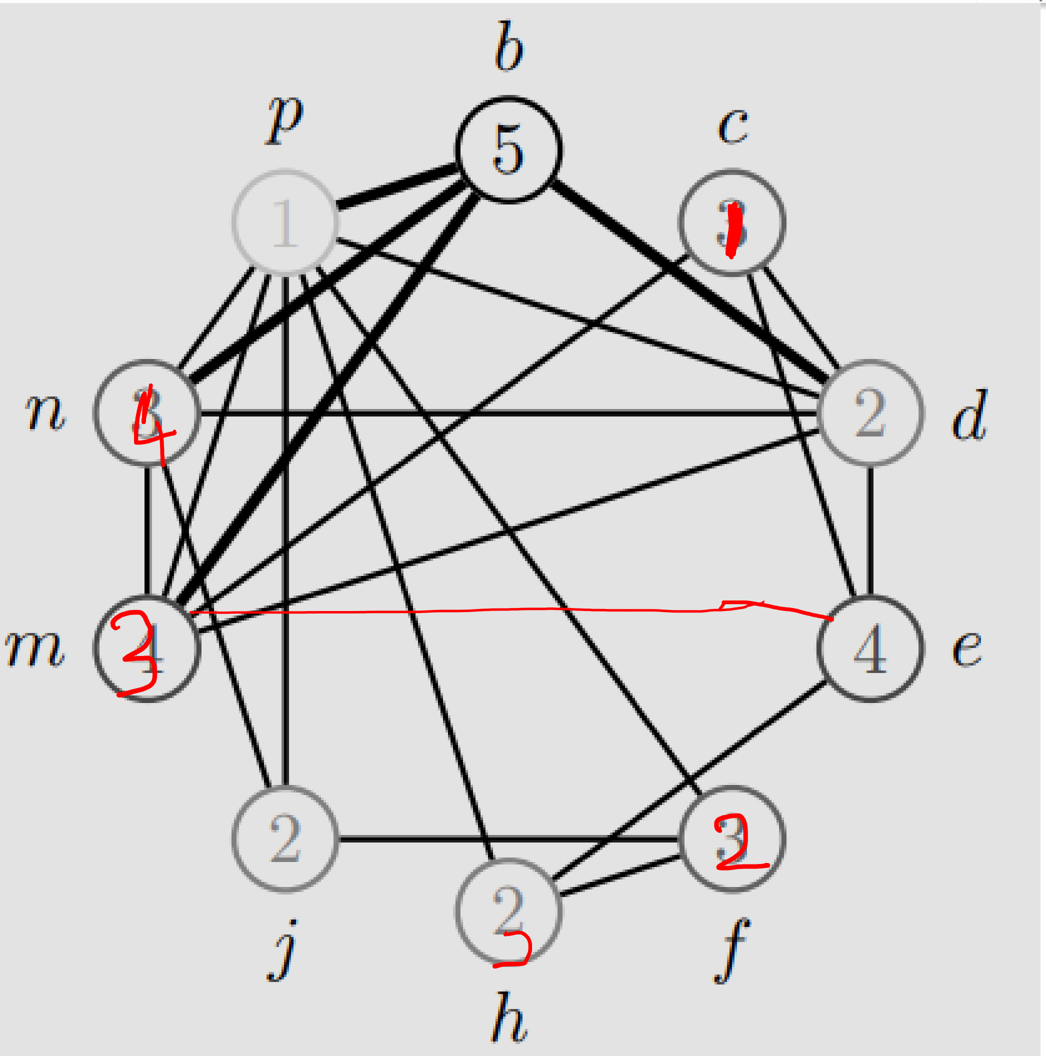
c -> { a, e, m } -> colors { 2, -, 3 }
Assign c = 1

e -> { c, d, m } -> { 1, 2, 3 }
Assign e = 4

f -> { h, j, p } -> { -, -, 1 }
Assign f -> 2

h -> { f, p } -> { 2, 1 }
Assign h -> 3

j -> { h, n, p } -> { 3, 4, 1 }
so assign j -> 2



Vertex coloring

This solution translates into the following teams:

team-1 → p, c

team -2 → d, f, i, j

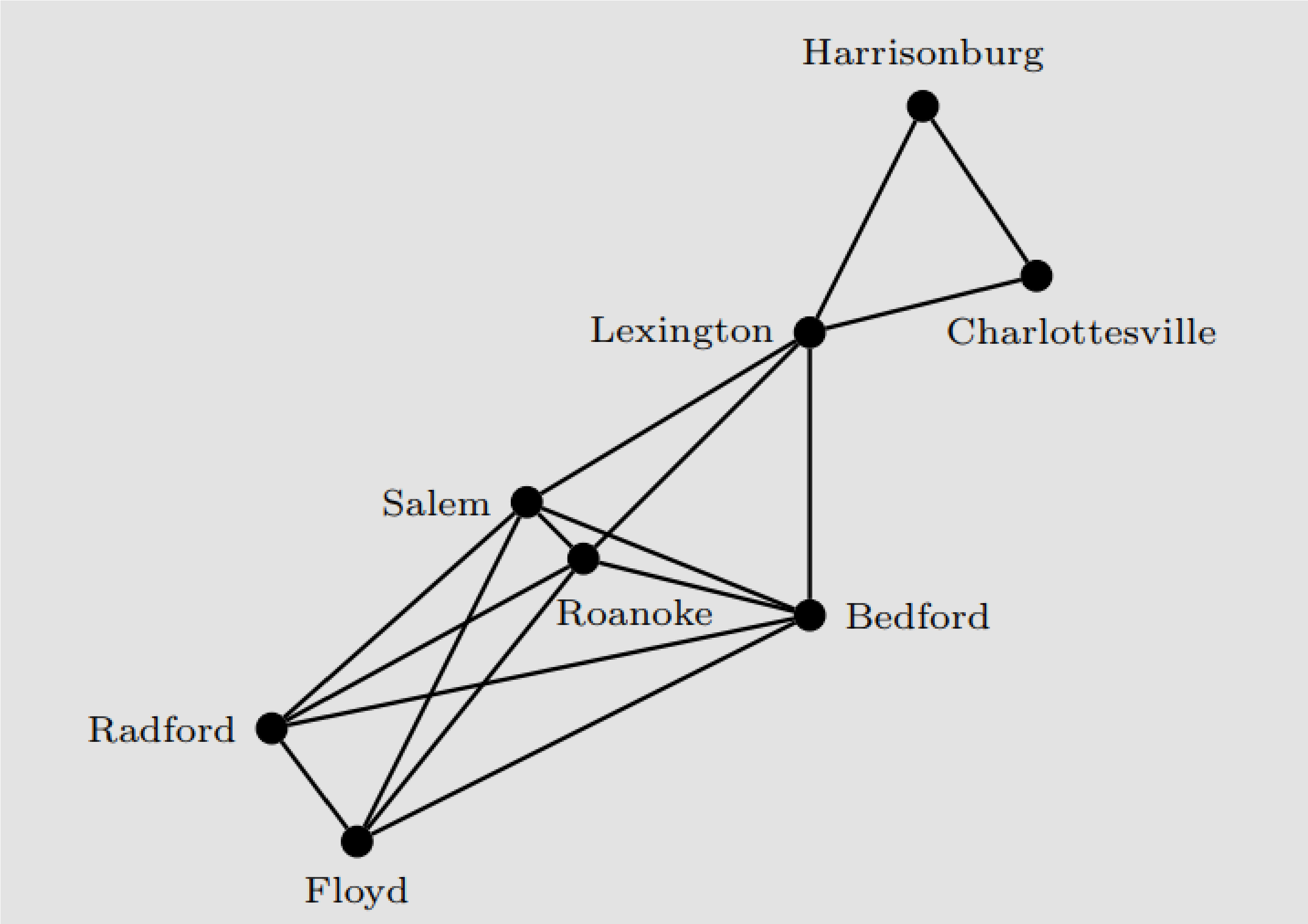
team-3 → m, n

team-4 → r, e

Vertex coloring

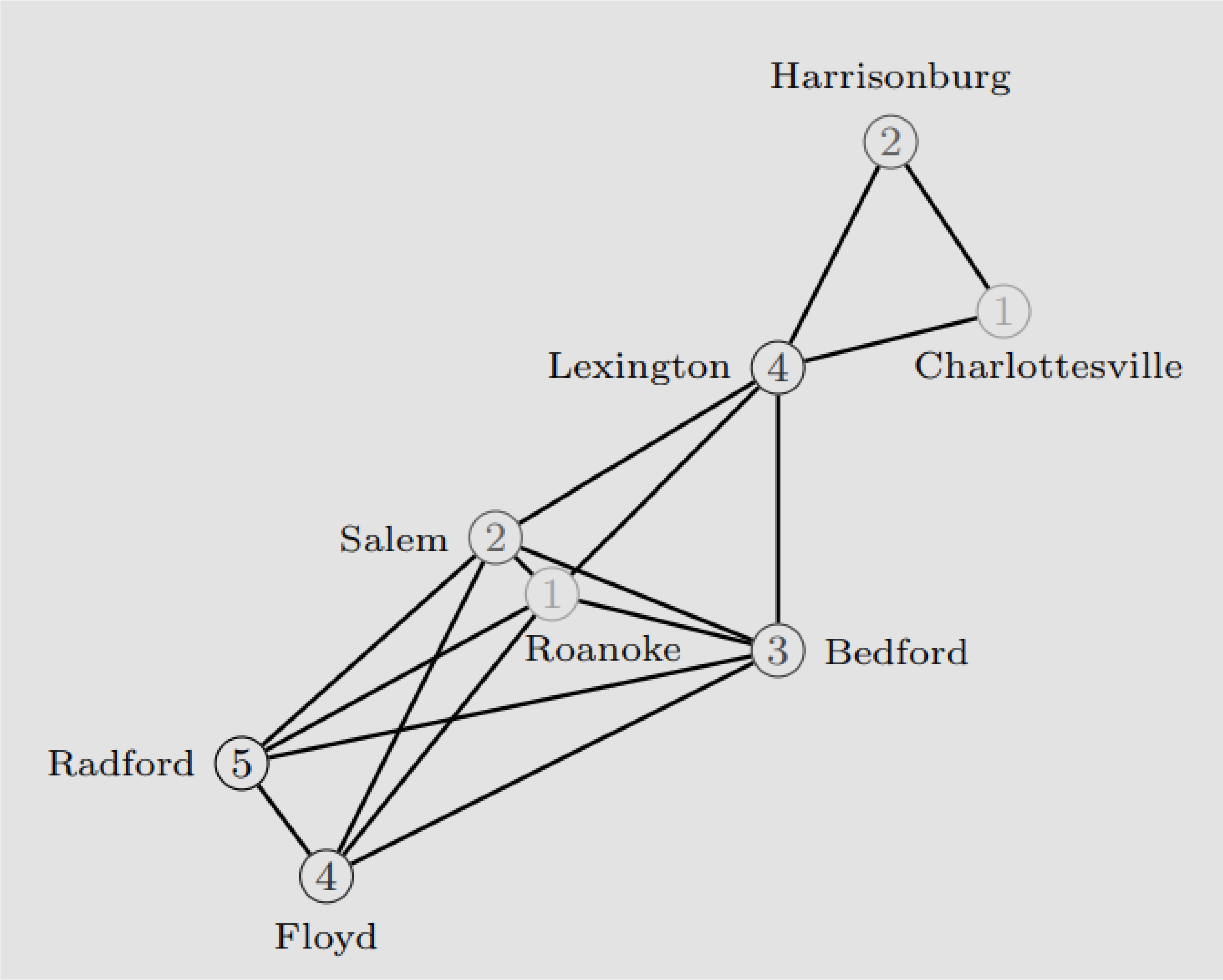
Example : Due to the nature of radio signals, two stations can use the same frequency if they are at least 70 miles apart. An edge in the graph below indicates two cities that are at most 70 miles apart, necessitating different radio stations. Determine the fewest number of frequencies need for each city shown below (not drawn to scale) to have its own municipal radio station.

Vertex coloring



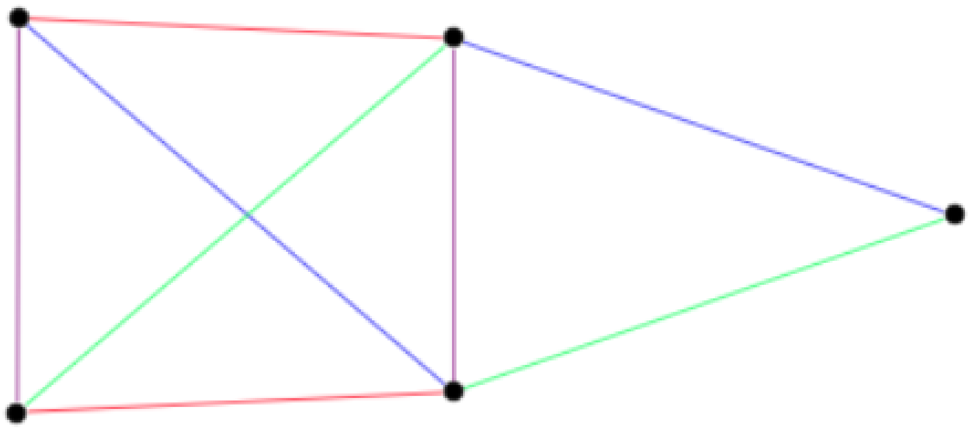
Vertex coloring

Solution: Each vertex will be assigned a color that corresponds to a radio frequency. This graph has $\chi = 5$ since we have a 5-coloring, as shown below, and fewer than 5 colors will not suffice as there is a K5 among the vertices representing the cities of Roanoke, Salem, Bedford, Floyd, and Radford.



Edge Colorings

The most common type of edge coloring is analogous to graph (vertex) colorings. Each edge of a graph has a color assigned to it in such a way that no two adjacent edges are the same color. Such a coloring is a **proper edge coloring**.



A proper edge coloring with 4 colors

Coloring variations-First fit algorithm

First-Fit Coloring Algorithm

Input: Graph G with vertices ordered as $x_1 \prec x_2 \prec \cdots \prec x_n$.

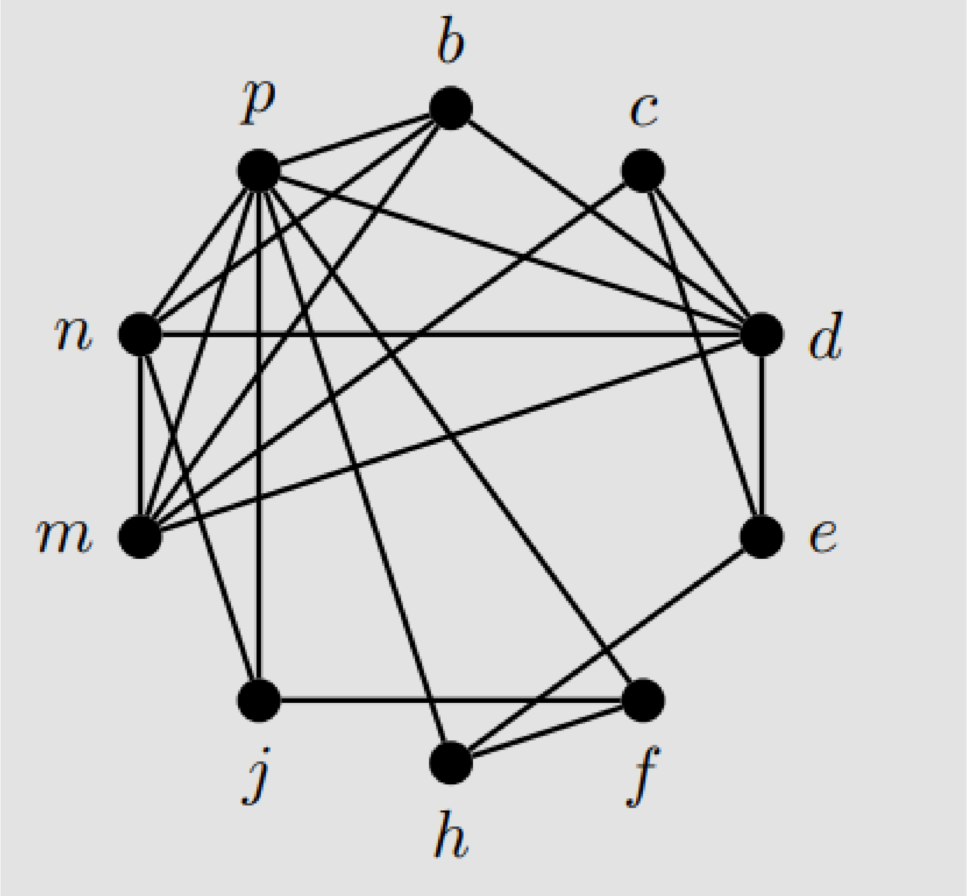
Steps:

1. Assign x_1 color 1.
2. Assign x_2 color 1 if x_1 and x_2 are not adjacent; otherwise, assign x_2 color 2.
3. For all future vertices, assign x_i the least number color available to x_i in $G[x_1, \ldots, x_i]$; that is, give x_i the first color not used by any neighbor of x_i that has already been colored.

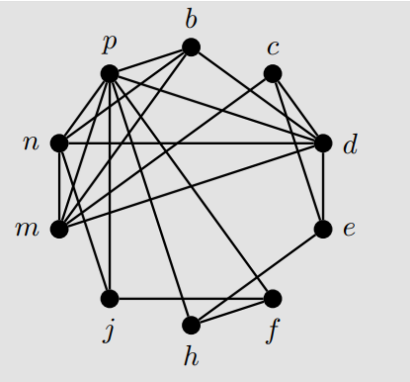
Output: Coloring of G.

Coloring variations-First fit algorithm

Example1: Apply the First-Fit Algorithm to the graph as shown below if the vertices are ordered alphabetically

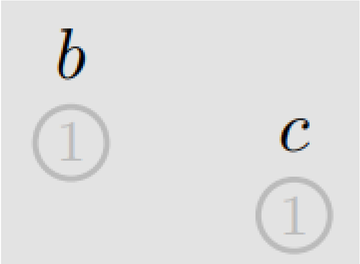


Coloring variations-First fit algorithm

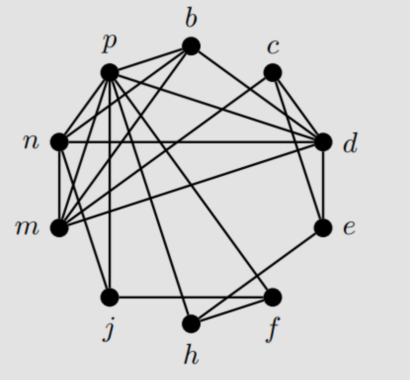


Step 1: Color b with 1.

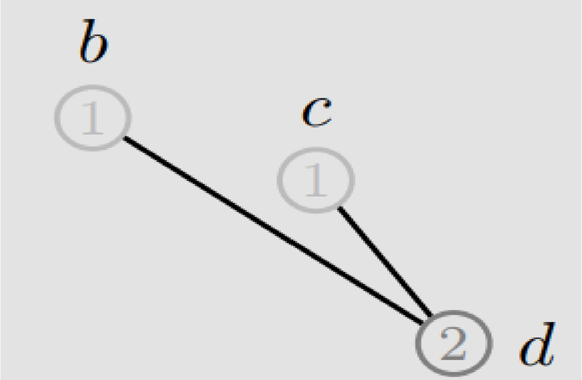
Step 2: Color c with 1 since b and c are not adjacent.



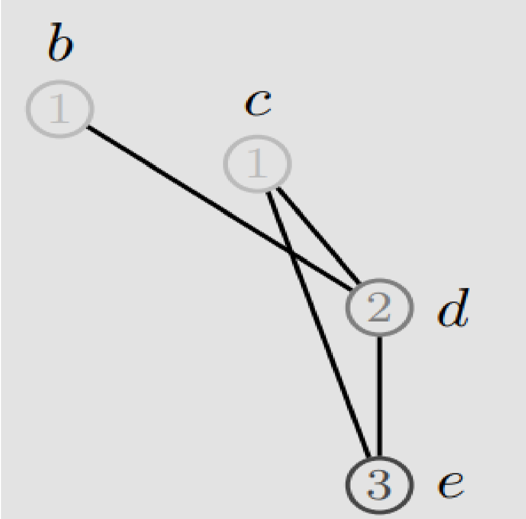
Coloring variations-First fit algorithm



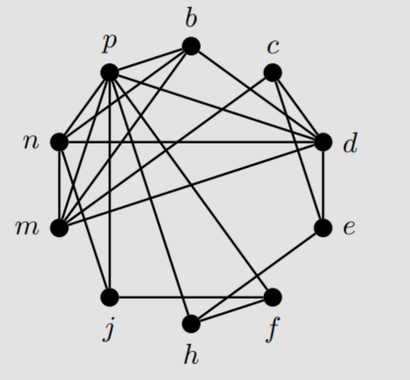
Step 3: Color d with 2 since d is adjacent to a vertex of color 1.



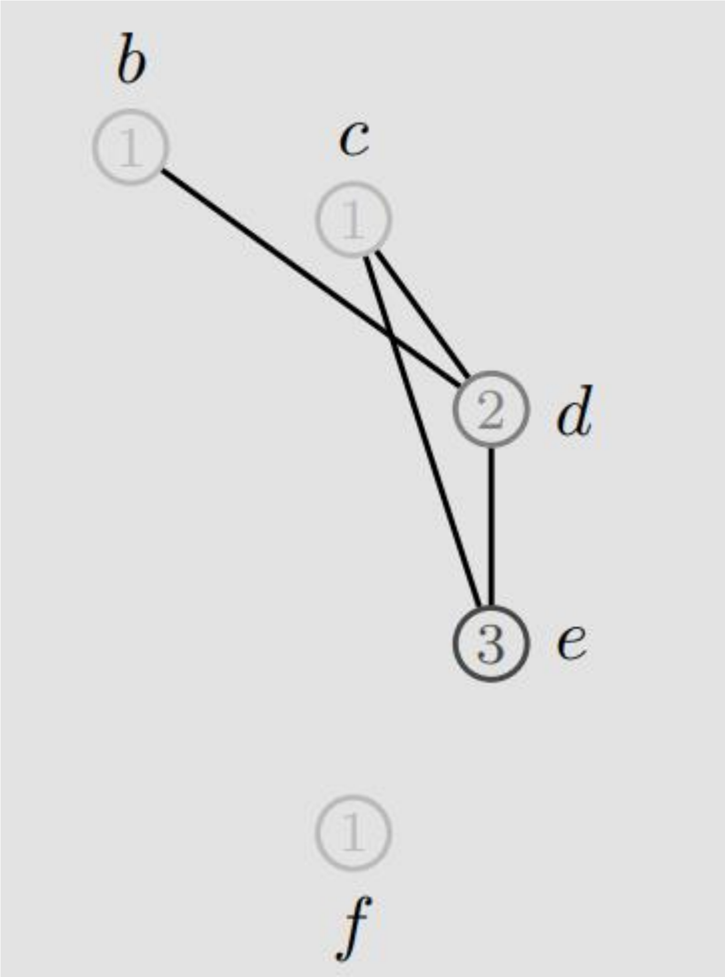
Step 4: Color e with 3 since e is adjacent to a vertex of color 1 (c) and a vertex of color 2 (d).



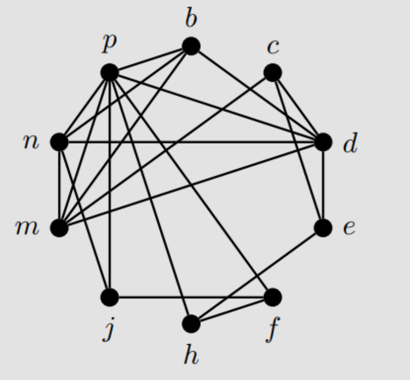
Coloring variations-First fit algorithm



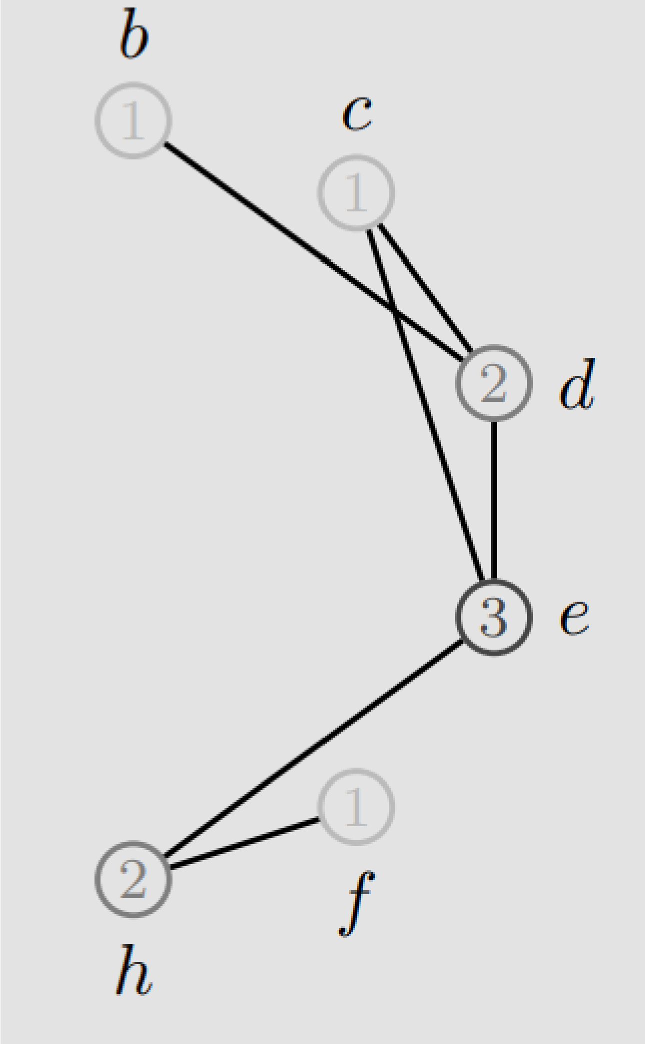
Step 5: Color f with 1 since f is not adjacent to any previous vertices.



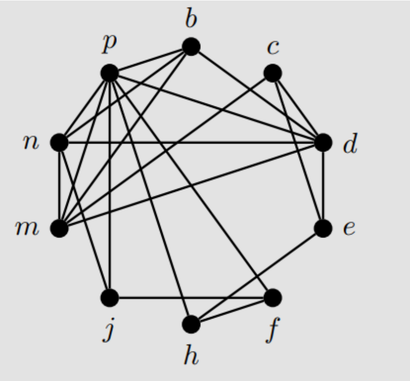
Coloring variations-First fit algorithm



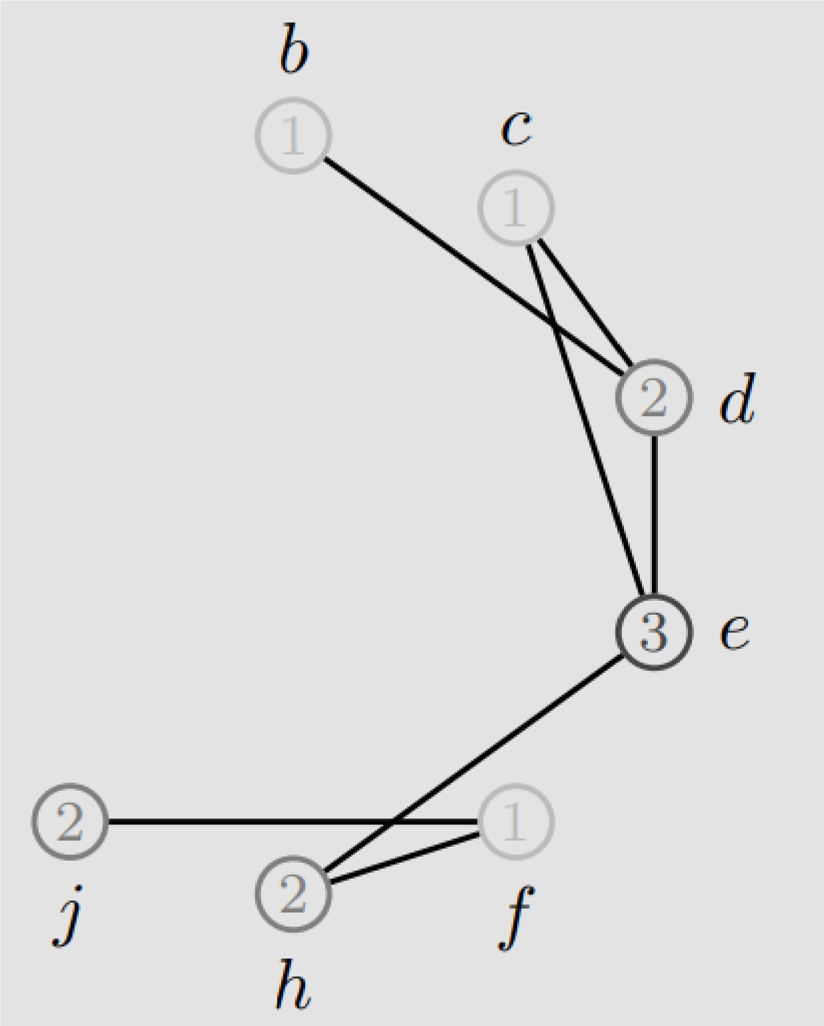
Step 6: Color h with 2 since h is adjacent to a vertex of color 1 (f).



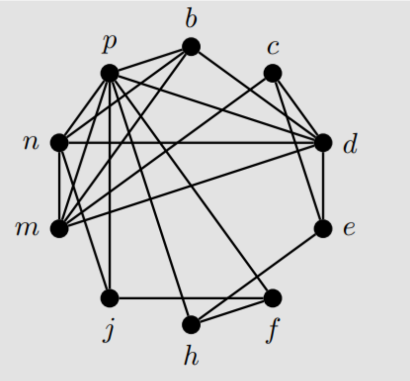
Coloring variations-First fit algorithm



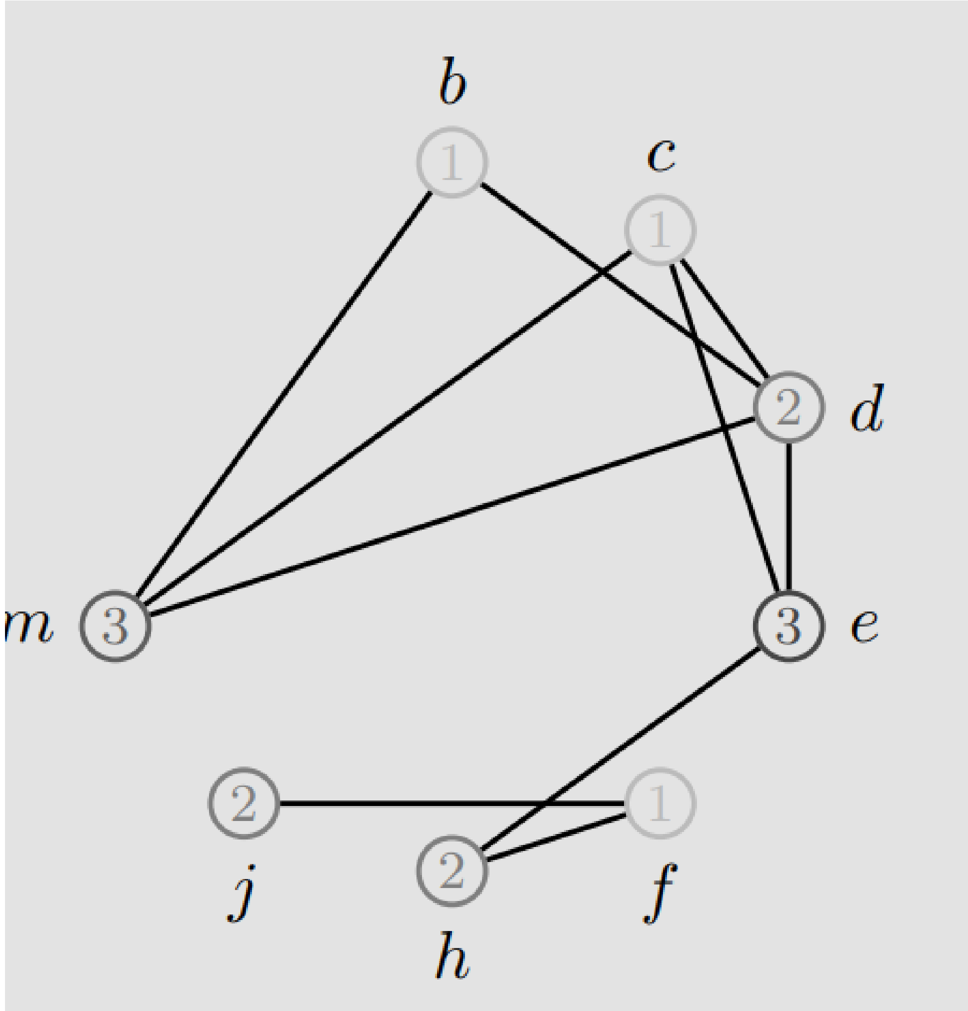
Step 7: Color j with 2 since j is adjacent to a vertex of color 1 (f).



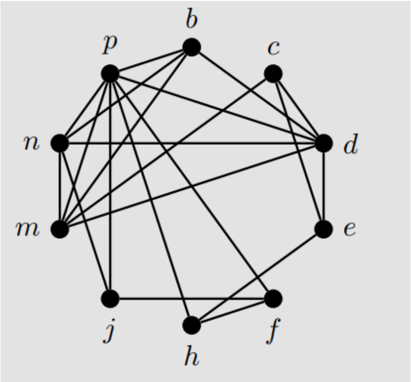
Coloring variations-First fit algorithm



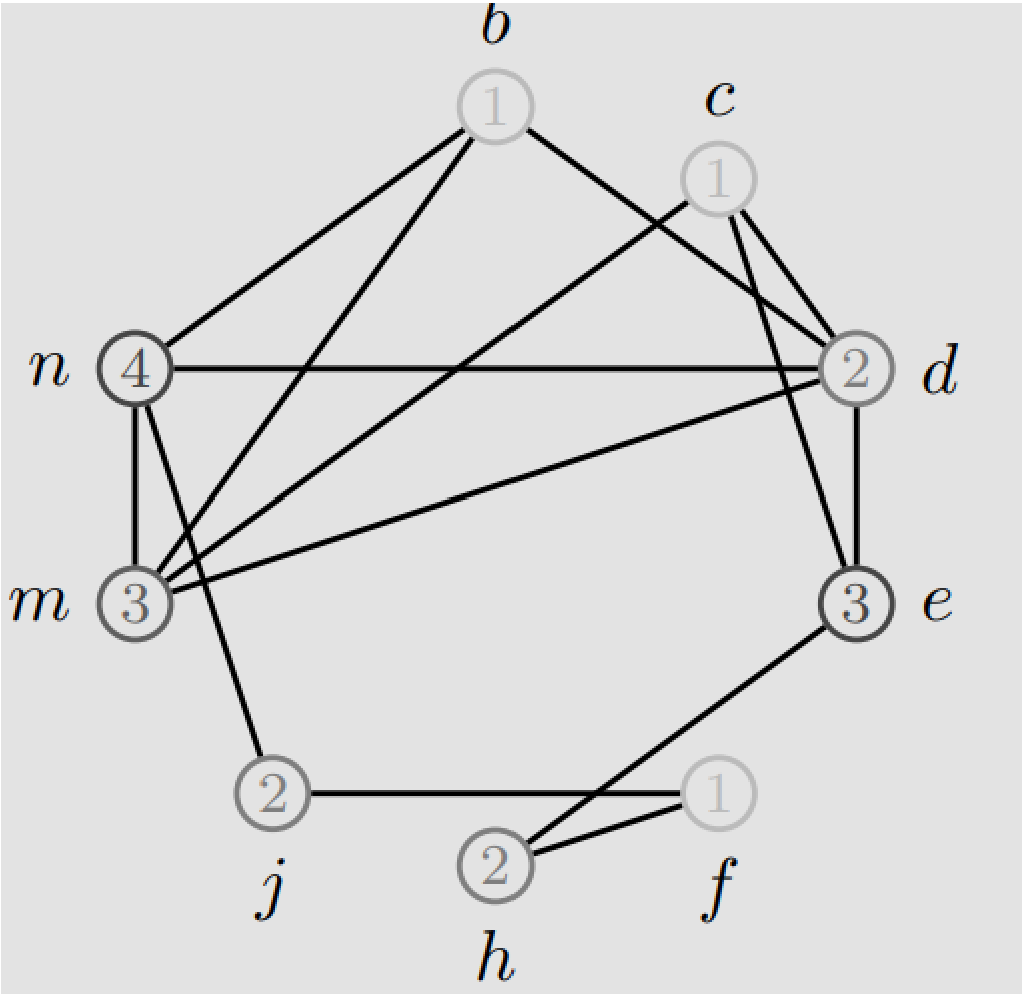
Step 8: Color m with 3 since m is adjacent to vertices of color 1 (b, c) and a vertex of color 2 (d).



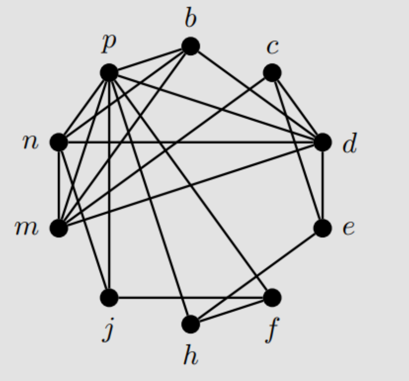
Coloring variations-First fit algorithm



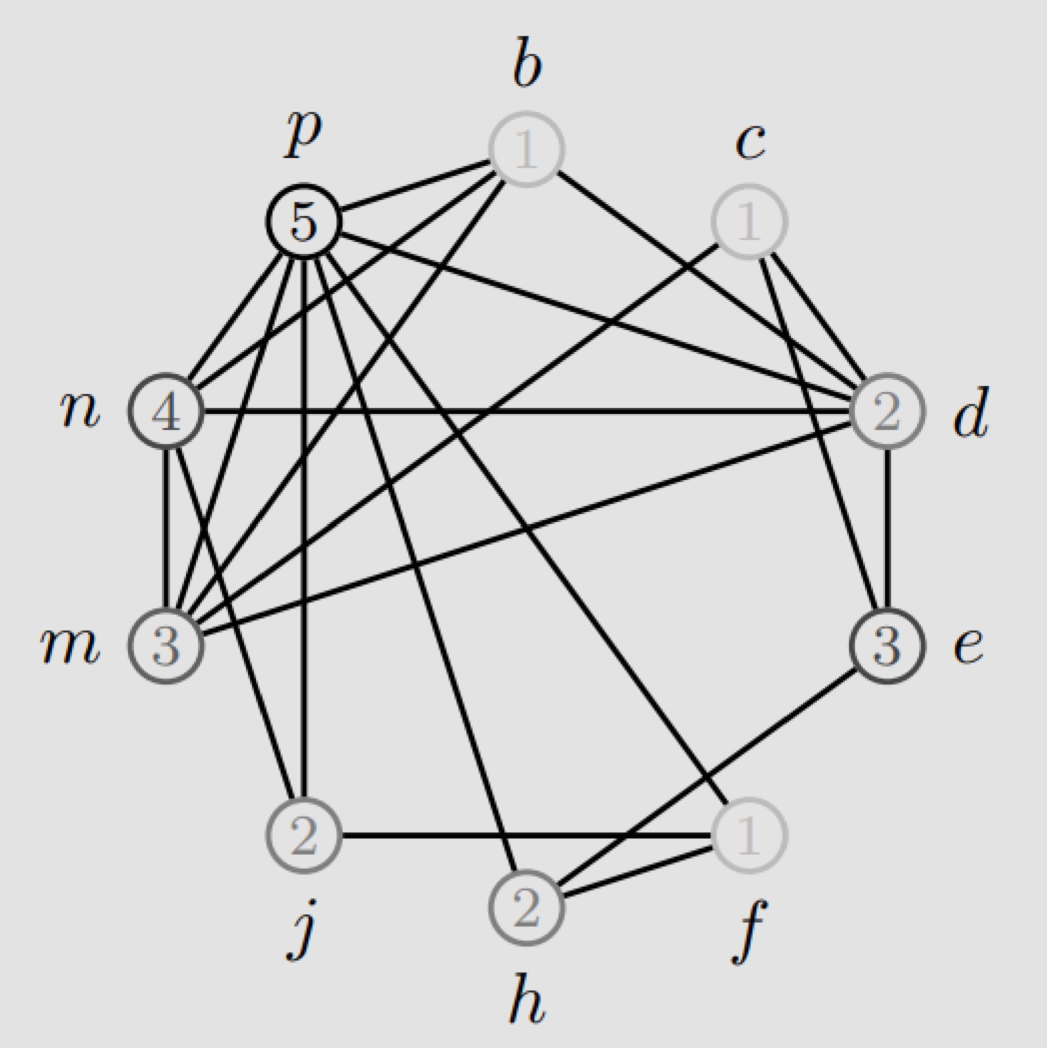
Step 9: Color n with 4 since n is adjacent to vertices of color 1, 2, and 3.



Coloring variations-First fit algorithm



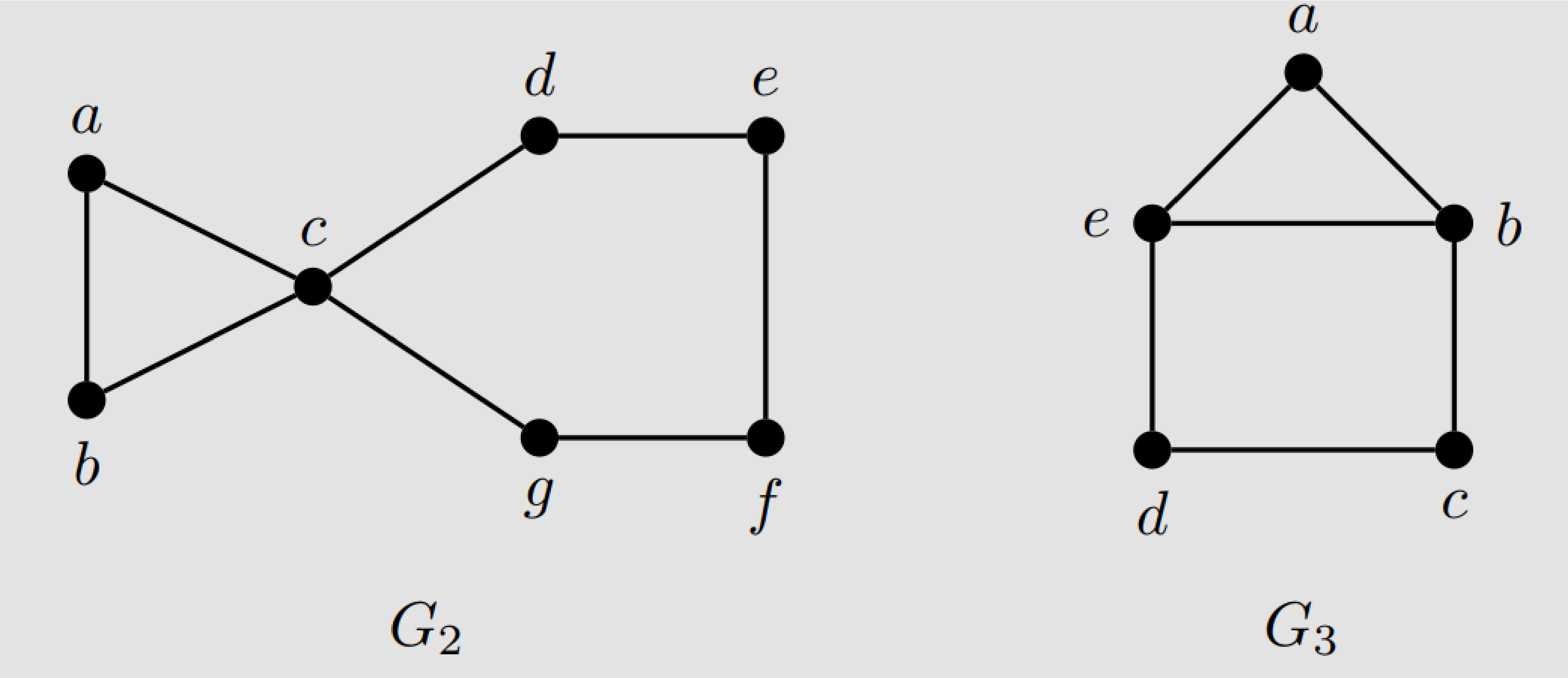
Step 10: Color p with 5 since p is adjacent to vertices of color 1, 2, 3, and 4.



Coloring variations-First fit algorithm

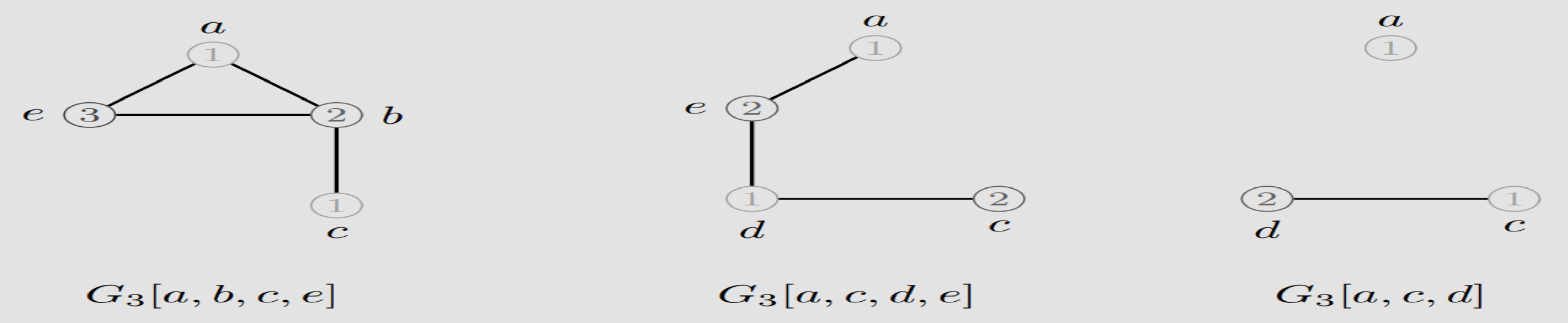
Question: Consider a simple graph with vertices A, B, C, and D connected as follows: A-B, A-C, A-D, B-C, and C-D. Explain first-fit vertex coloring algorithm?

Determine if either of the two graphs below are perfect



Solution:

Without much work we can see that both graphs above satisfy $\chi(G) = \omega(G)$. However, if we look at the subgraph H induced by {c, d, e, f, g} in G2 we see that H is just C5 and so $\chi(H) = 3$ even though $\omega(H) = 2$. Thus G2 is not perfect. However, G3 is in fact perfect. If an induced subgraph H contains {a, b, e}, then $\omega(H) = 3 = \chi(H)$; otherwise one of {a, b, e} will not be in H and so $\omega(H) \leq 2$ and without much difficulty we can show $\omega(H) = \chi(H)$. A few illustrative induced subgraphs are shown below.



A graph G is perfect if and only if $\chi(H) = \omega(H)$ for all induced subgraphs H

Perfect Graphs

The following classes of graphs are known to be perfect:

- Trees
- Bipartite graphs
- Chordal graphs
- Interval graphs

Depth-First Search Tree

Depth-First Search Tree

Input: Simple graph $G = (V, E)$ and a designated root vertex r.

Steps:

1. Choose the first neighbor x of r in G and add it to $T = (V, E')$.
2. Choose the first neighbor of x and add it to T. Continue in this fashion—picking the first neighbor of the previous vertex to create a path P. If P contains all the vertices of G, then P is the depth-first search tree. Otherwise continue to Step (3).
3. Backtrack along P until the first vertex is found that has neighbors not in T. Use this as the root and return to Step (1).

Output: Depth-first search tree T.

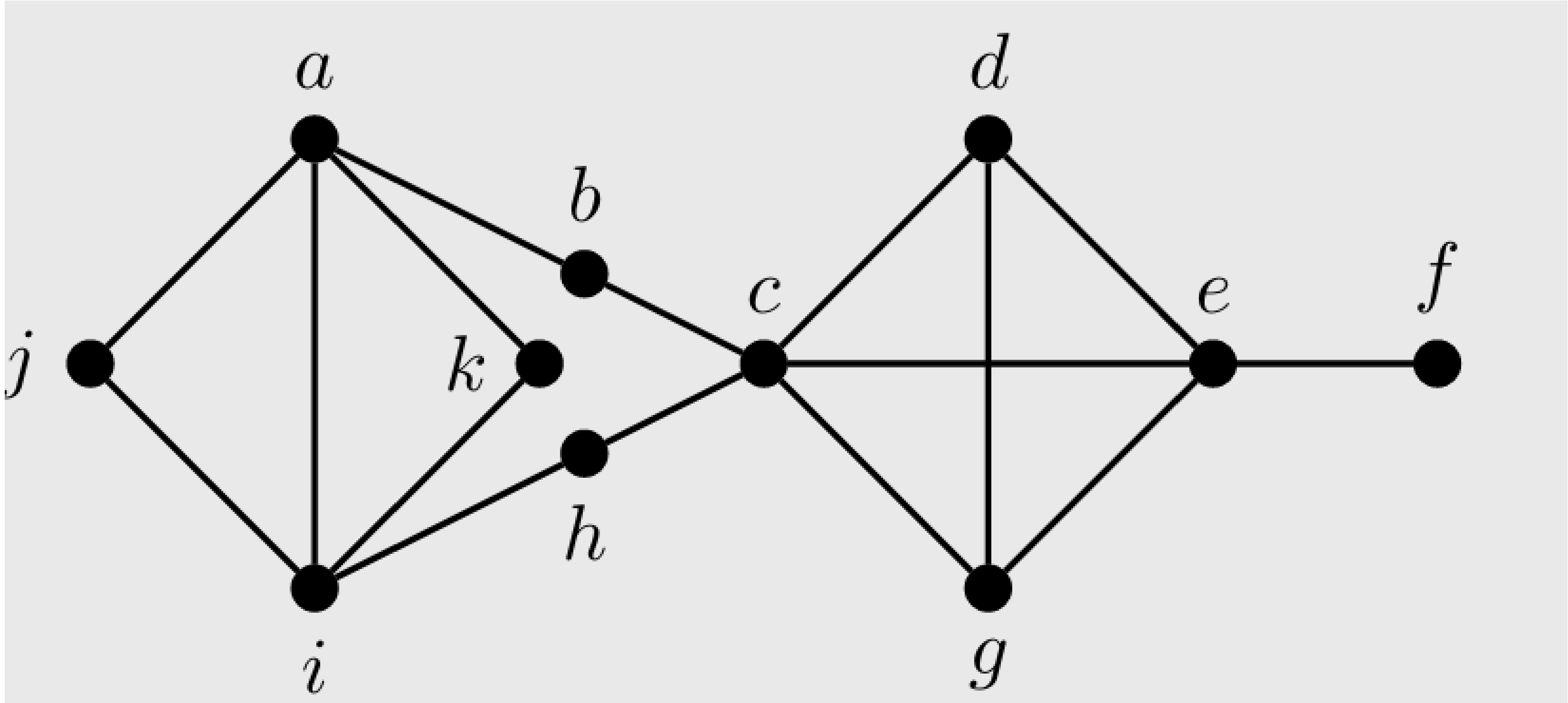
Depth-First Search Tree

In creating a depth- first search tree,

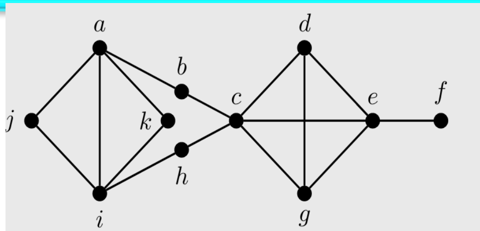
- we begin by building a central spine from which all branches originate.
- These branches are as far down on this path as possible.
- In doing so, the resulting rooted tree is often of large height and is more likely to have more vertices at the lower levels.

Depth-First Search Tree

Find the depth- first search tree for the graph below with the root ‘a’



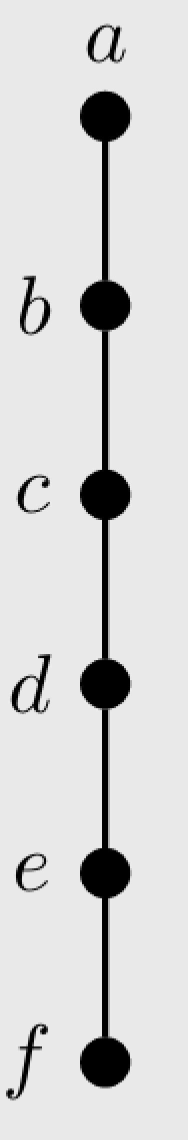
Depth-First Search Tree



Solution:

Step1:

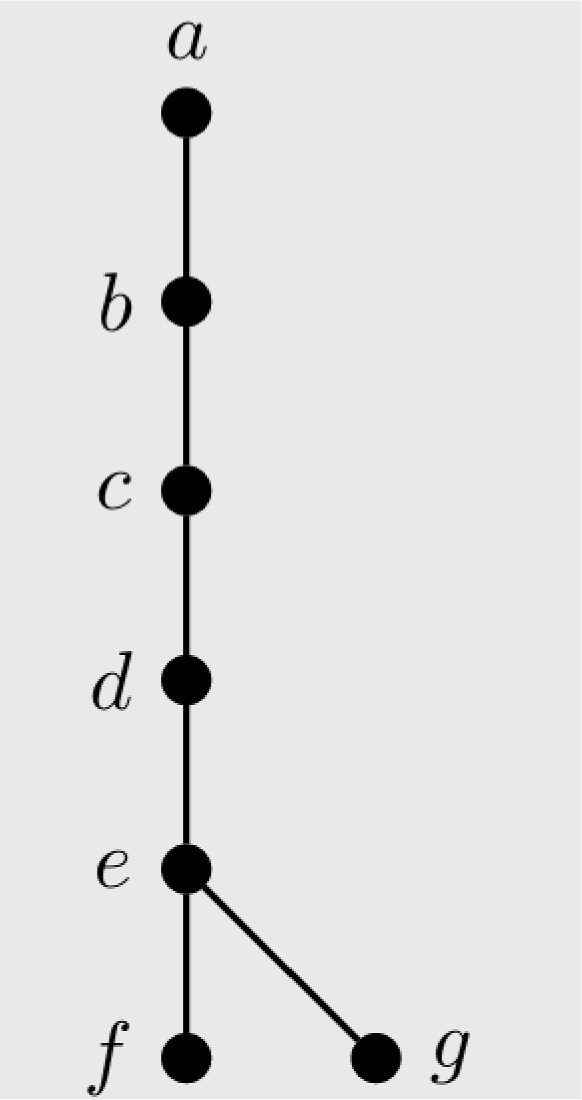
Since a is the root, we add b as it is the first neighbor of a. Continuing in this manner produces the path shown . Note this path stops with f since f has no further neighbors in G.



Depth-First Search Tree

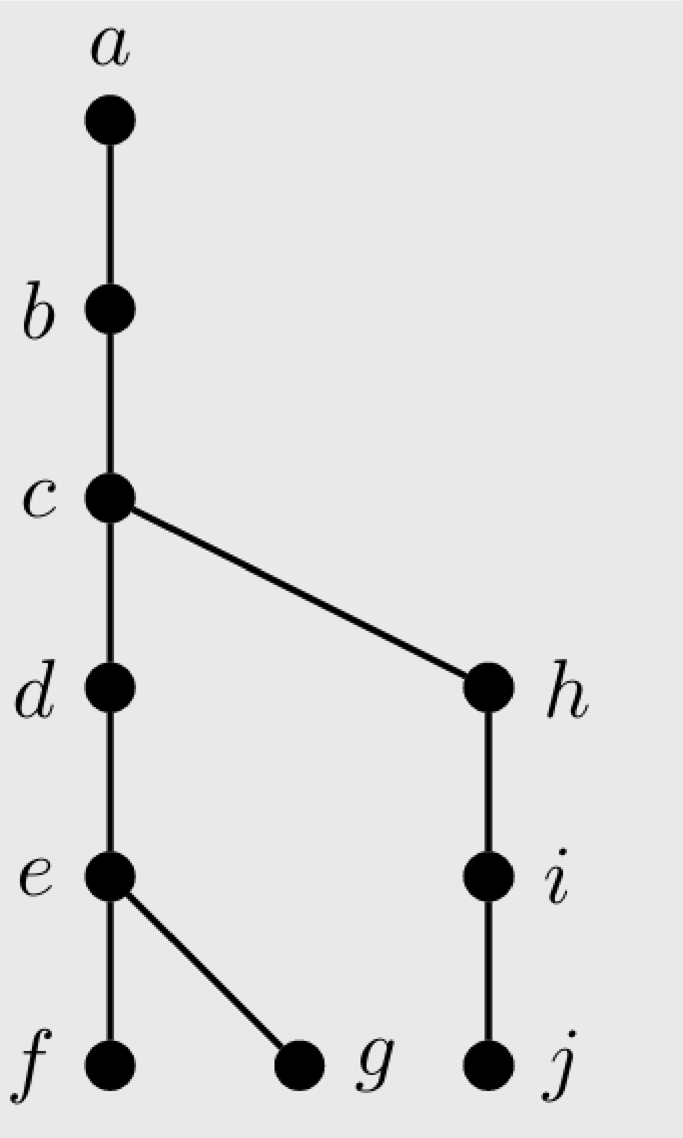
Step2:

Back tracking along the path above, the first vertex with an un-chosen neighbor is e. This adds the edge e g to T. No other edges from g are added since the other neighbor so f g are already part of the tree.



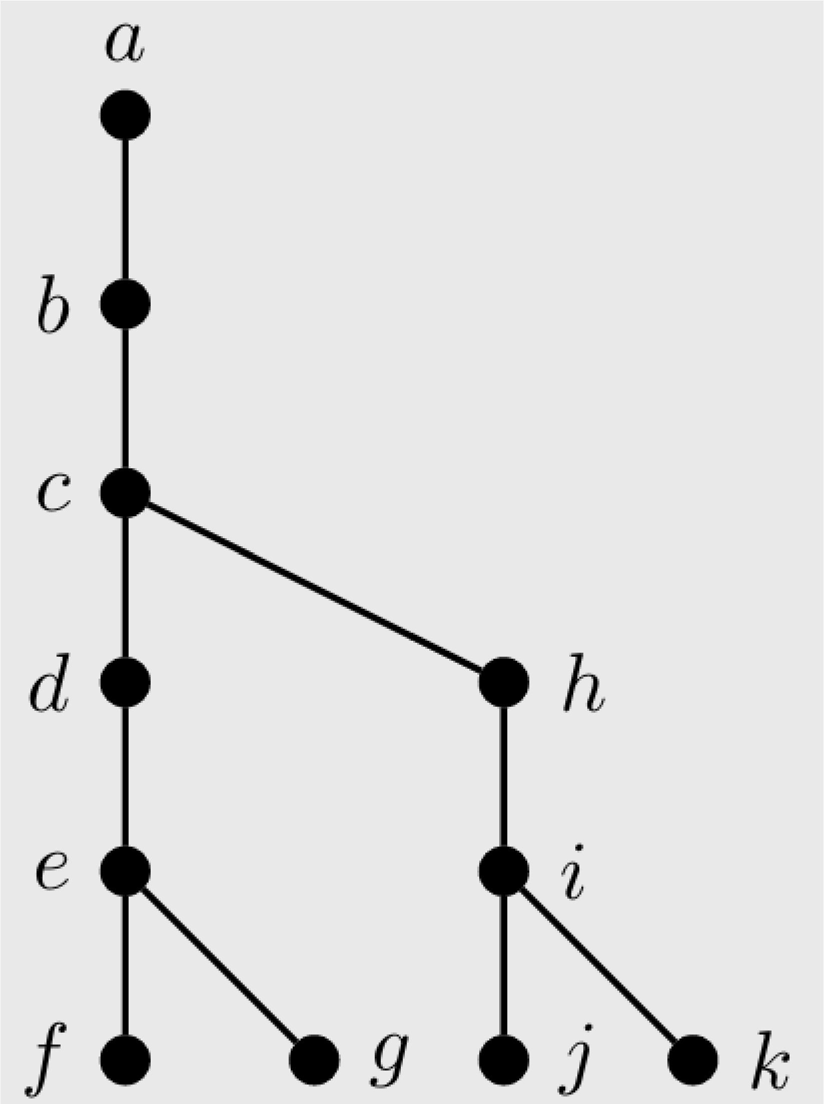
Depth-First Search Tree

Step 3: Backtracking again along the path from Step 1, the next vertex with an unchosen neighbor is c. This adds the path chij to T.



Depth-First Search Tree

Step 4: Backtracking again along the path from Step 3, the next vertex with an unchosen neighbor is i. This adds the edge ik to T and completes the depth- first search tree as all the vertices of G are now included in T.



Output: The tree above is the depth- first search tree.

Prim's Algorithm

Input: Weighted connected graph $G = (V, E)$.

Steps:

1. Let v be the root. If no root is specified, choose a vertex at random. Highlight it and add it to $T = (V', E')$.

2. Among all edges incident to v, choose the one of minimum weight. Highlight it. Add the edge and its other endpoint to T.

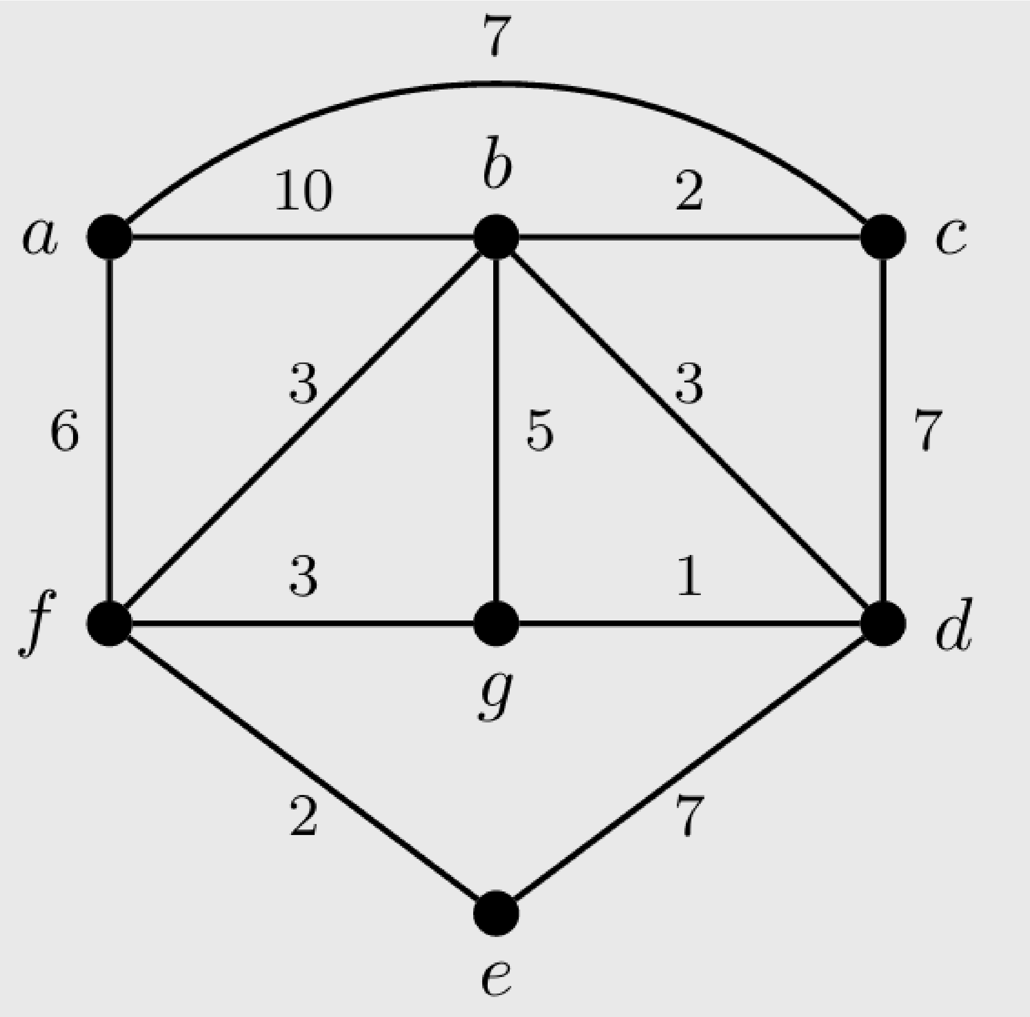
3. Let S be the set of all edges with exactly endpoint from $V(T)$. Choose the edge of minimum weight from S. Add it and its other endpoint to T.

4. Repeat Step (3) until T contains all vertices of G, that is $V(T) = V(G)$.

Output: Minimum spanning tree T of G.

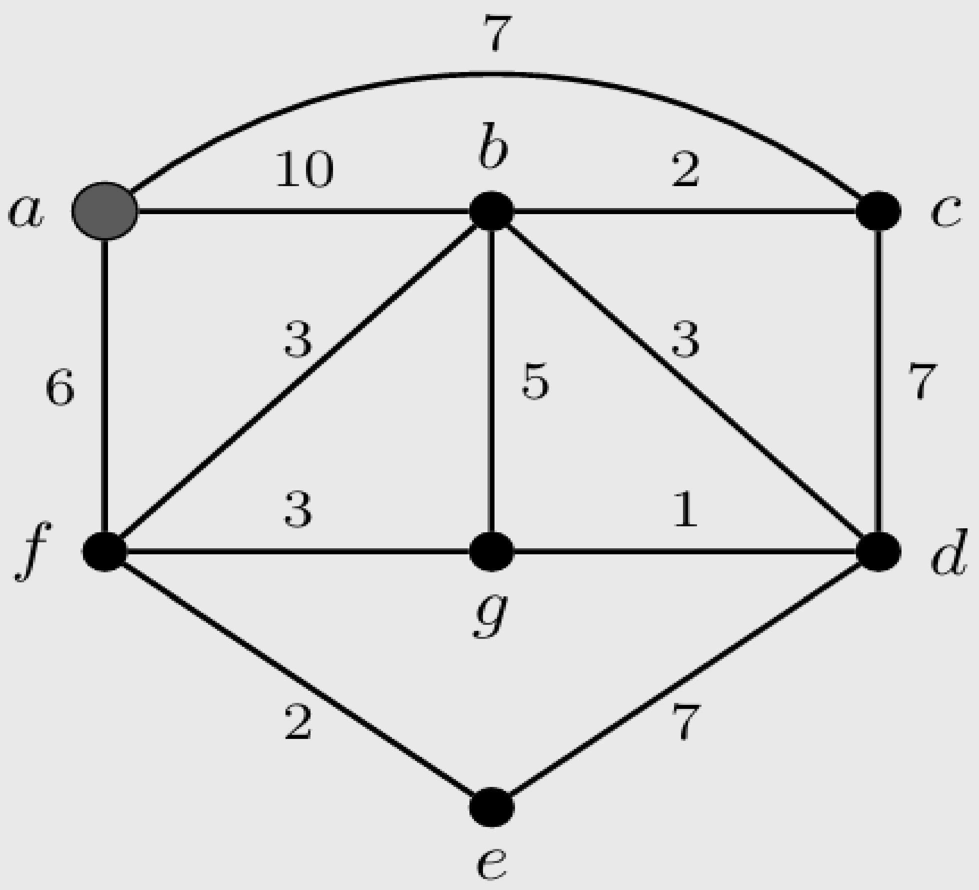
Prim's algorithm

Example: Find the minimum spanning tree of the graph G below using Prim's Algorithm.



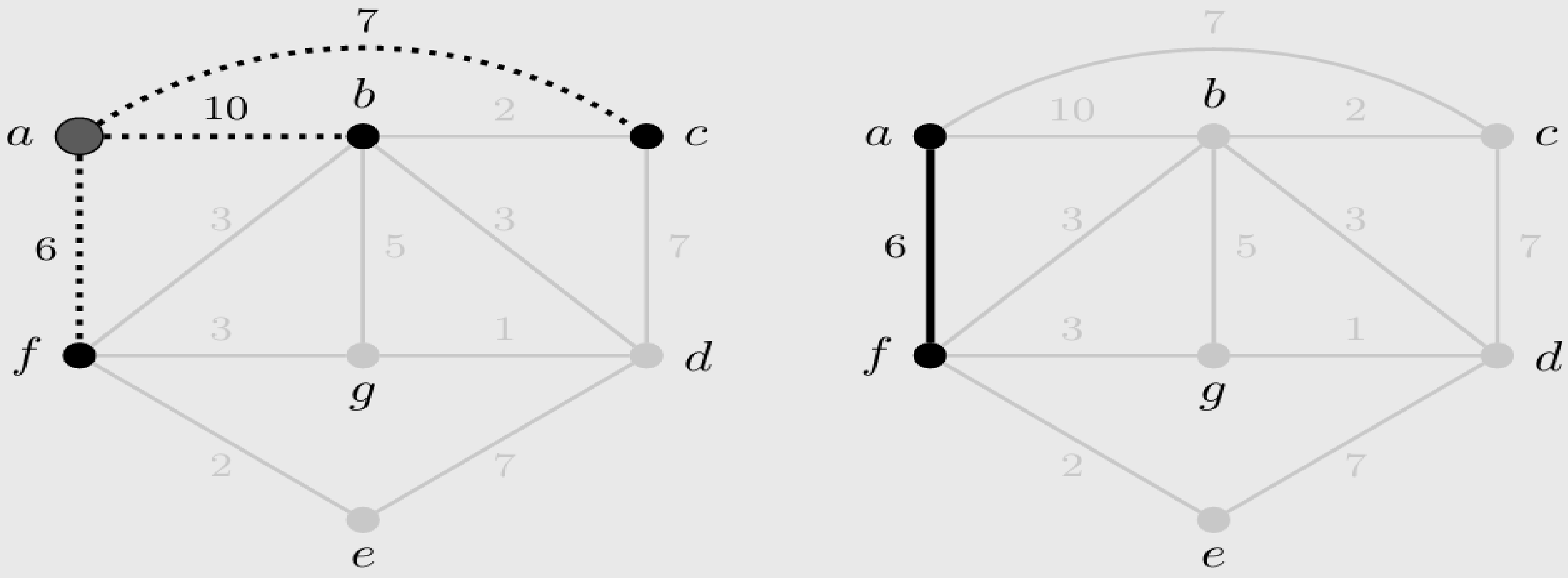
Prims algorithm

Step 1: Since no root was specified, we choose a as the starting vertex.



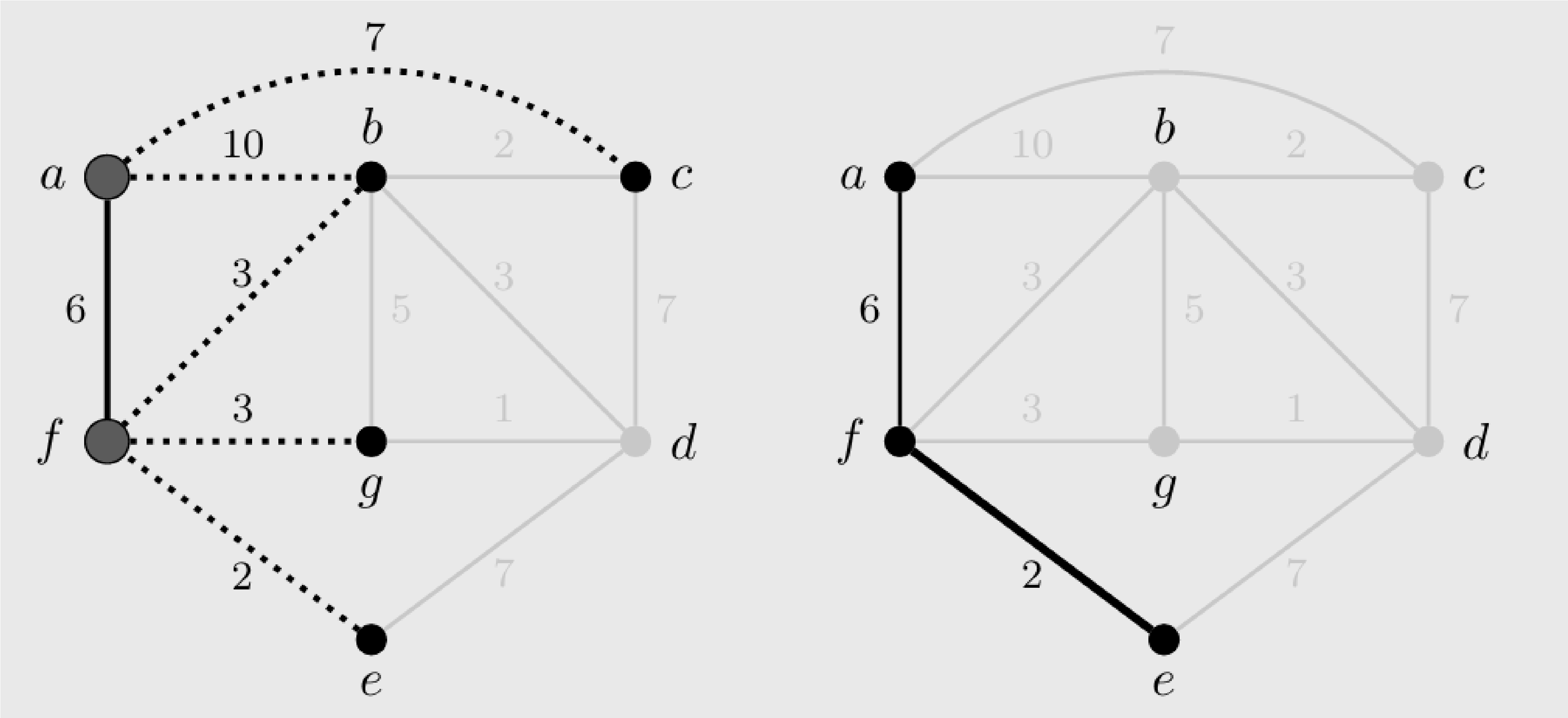
Prims algorithm

Step 2: We consider the edges incident to a, namely ab, ac, and af. These are shown as dotted lines in the graph on the left. The edge of least weight is af. This is added to the tree, shown in bold on the right.



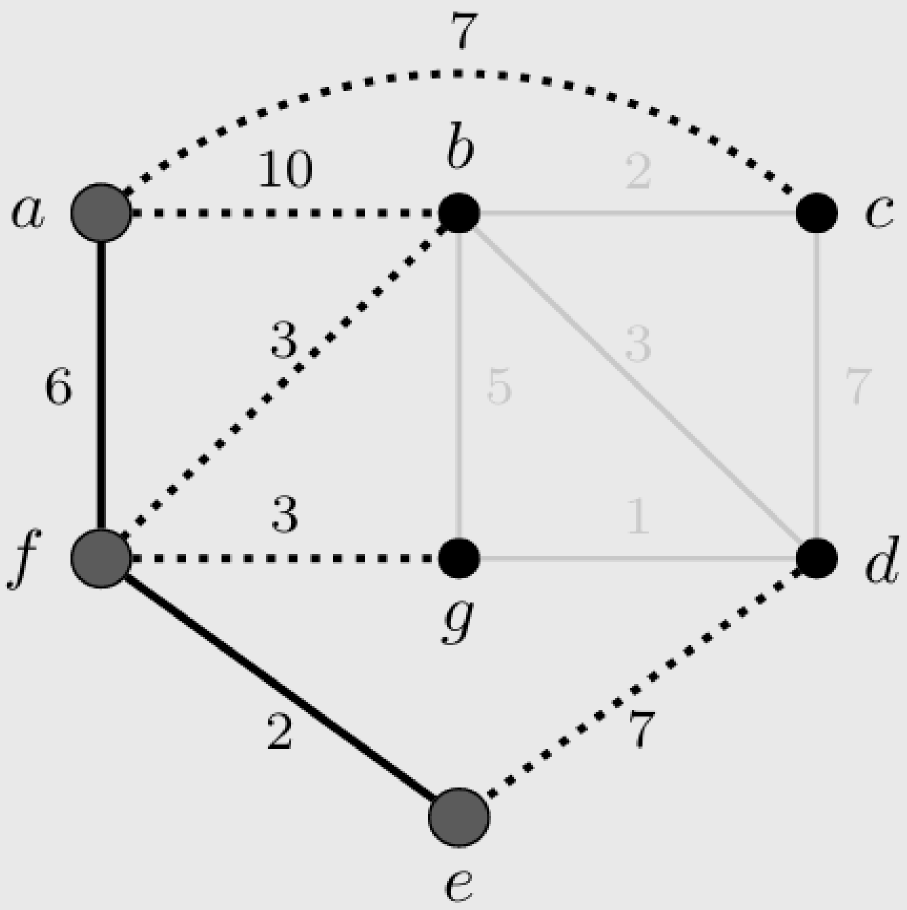
Prims algorithm

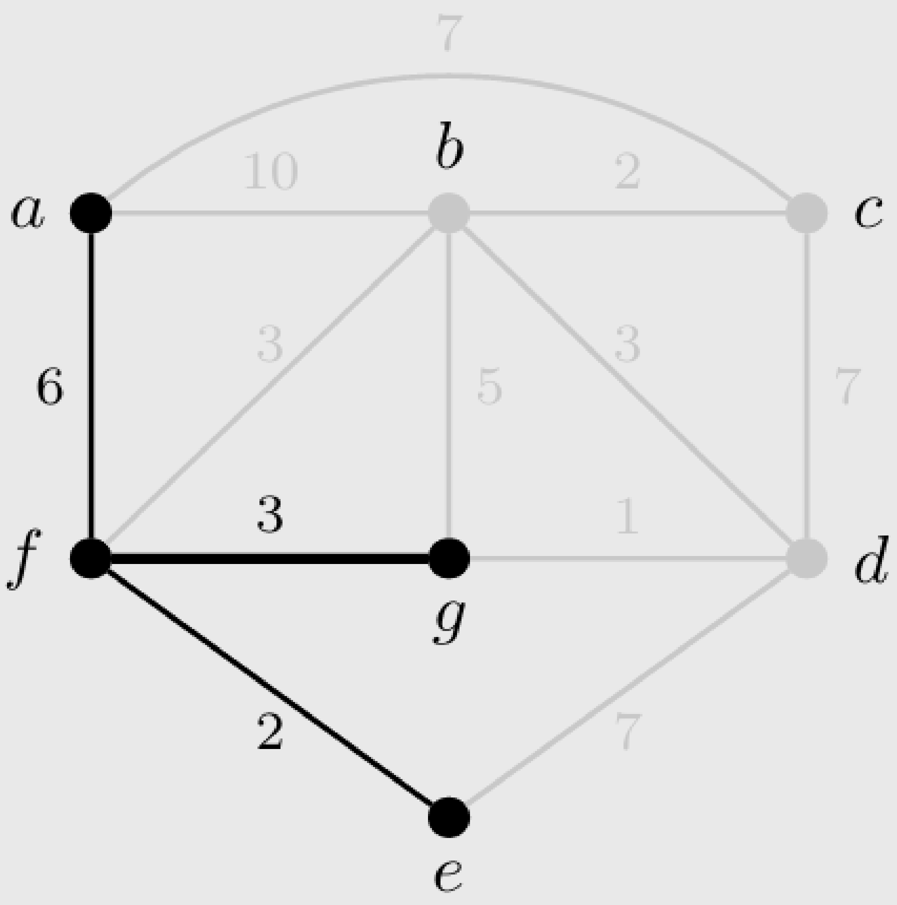
Step 3: The set S consists of edges with one endpoint as a or f, as shown in the graph to the left. The edge of minimum weight from these is ef. This is added to the tree, as shown on the right.



Prims algorithm

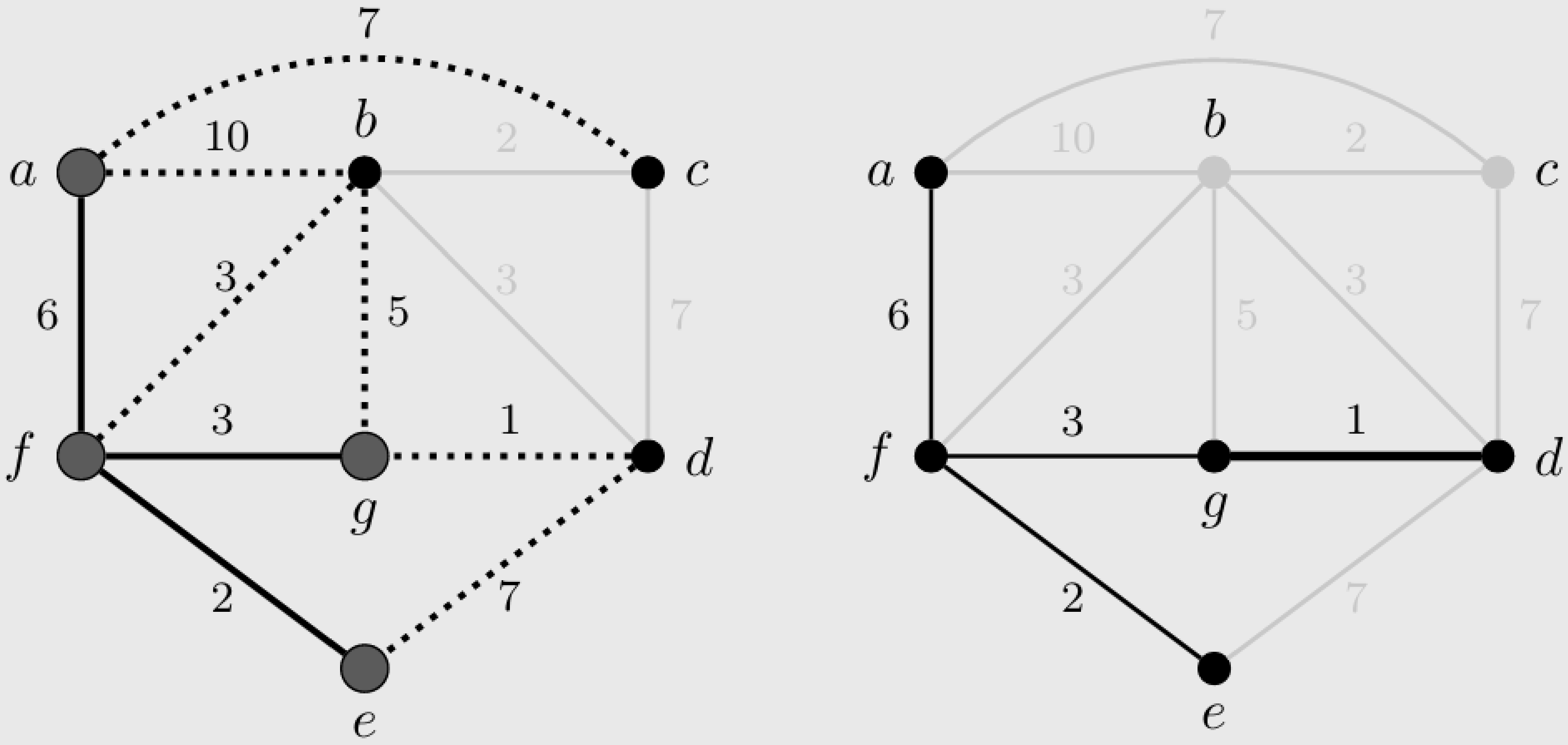
Step 4: The new set S consists of edges with one endpoint as a, e, or f. The next edge added to the tree could either be fg or fb. We choose fg.





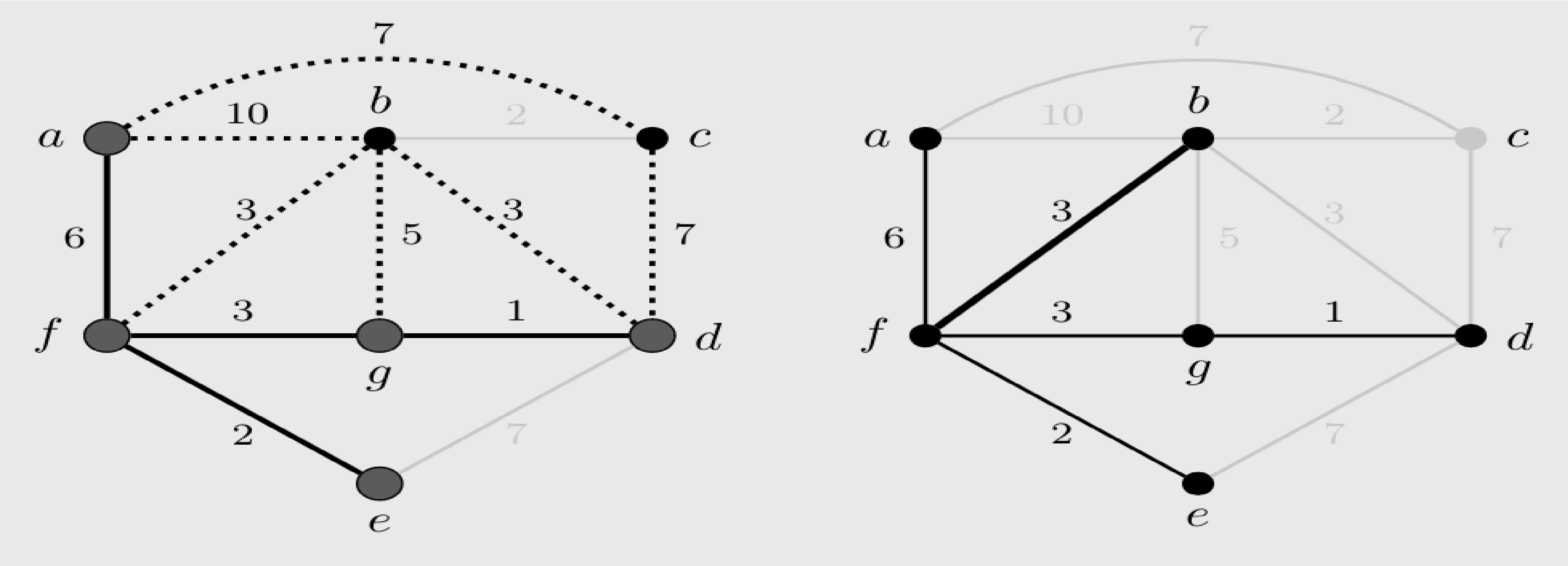
Prims algorithm

Step 5: We consider the edges where exactly one endpoint is from a, e, f, or g. The next edge to add to the tree is dg.



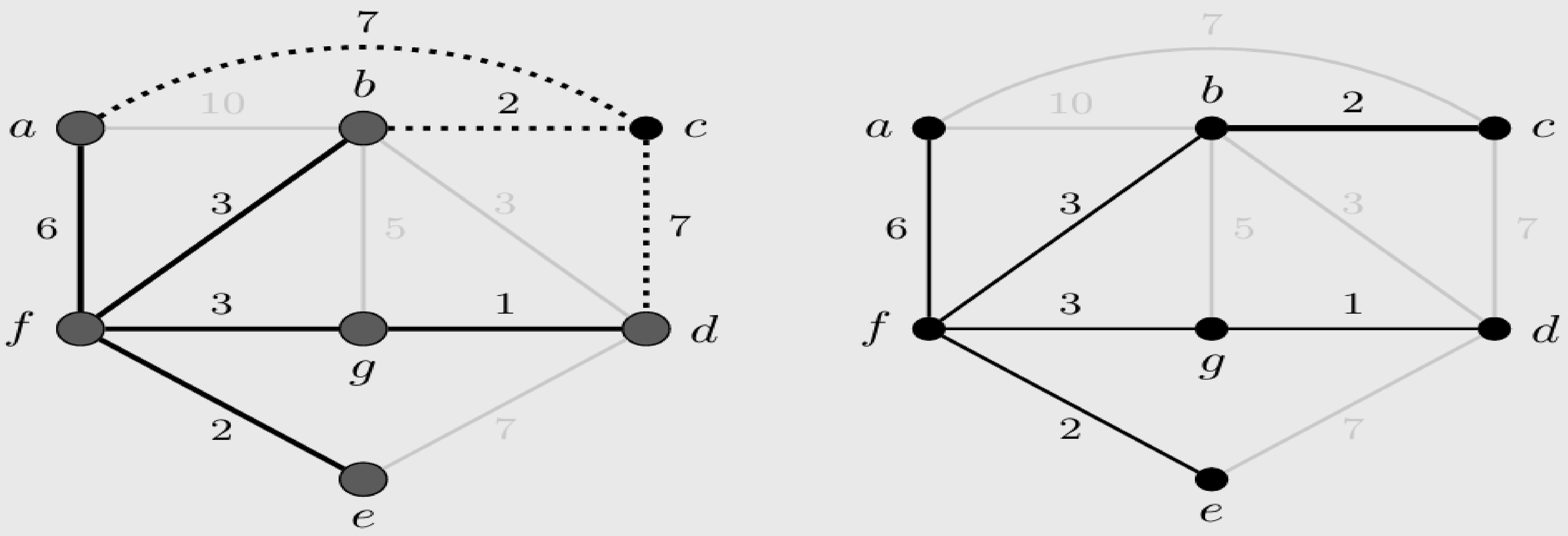
Prims algorithm

Step 6: The edges to consider must have exactly one end point from a,d,e,f, or g. Note that de is no longer available since both end points are already part of the tree(and its addition would create a cycle).There are two possible minimum weight edges,bf or bd. We choose bf.

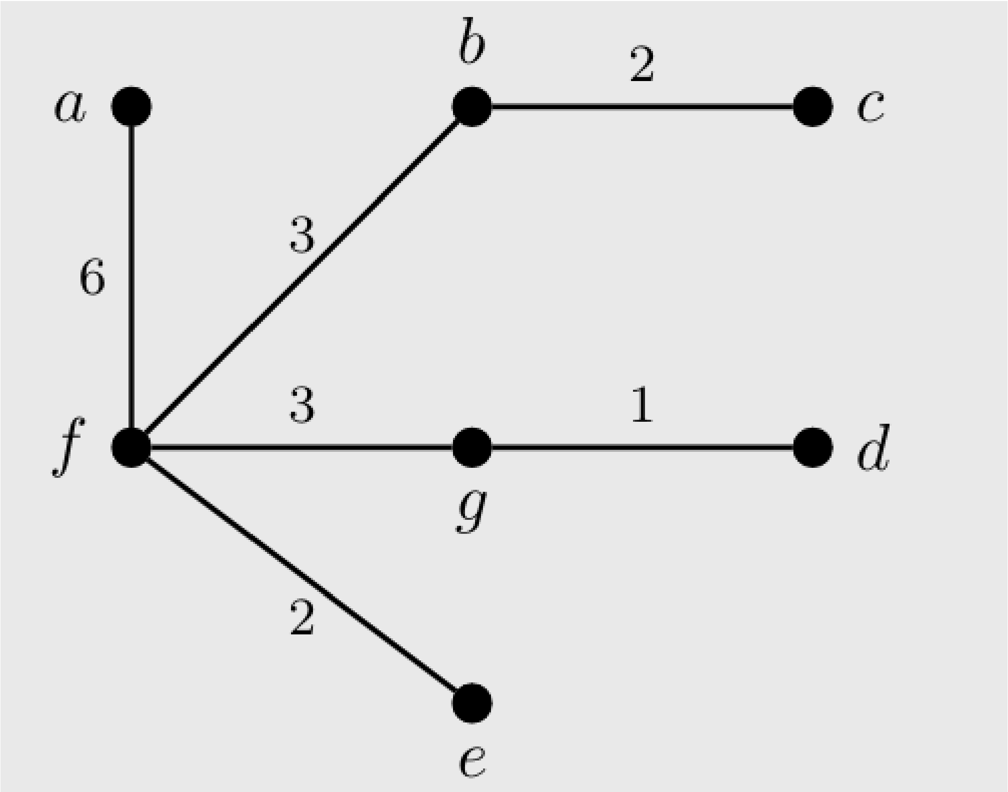


Prims algorithm

Step 7: The only edges we can consider are those with one endpoint of c since this is the only vertex not part of our tree. The edge of minimum weight is bc.



Prims algorithm

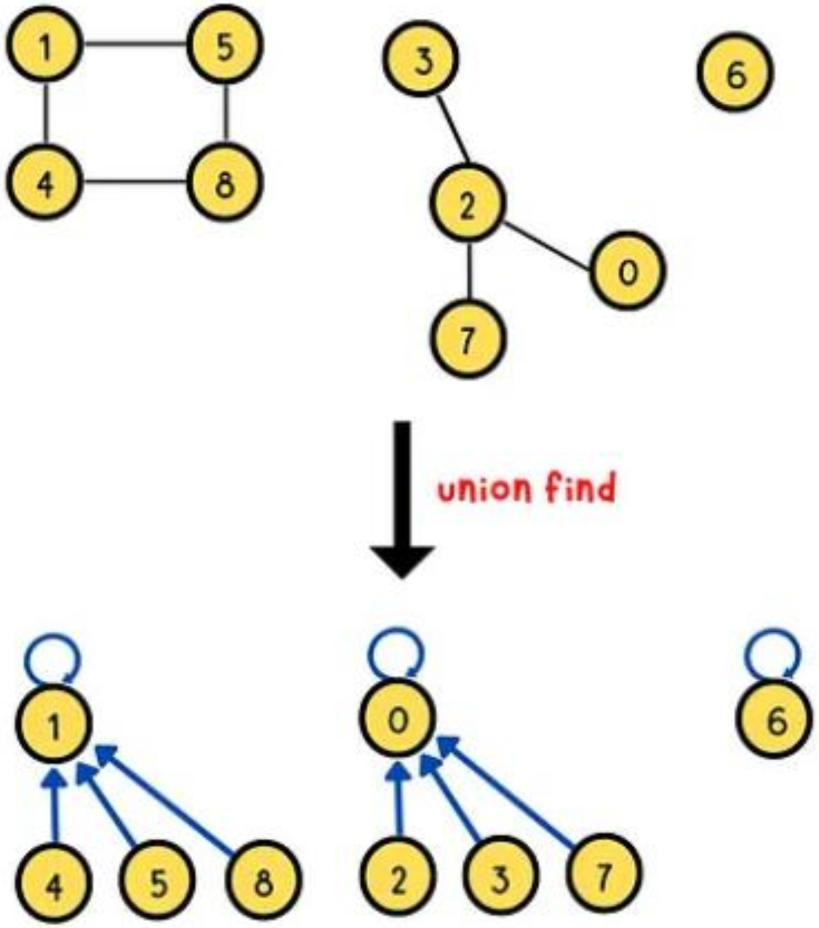


Output:A minimum spanning tree of total weight 17.

An algorithm that implements ***find*** *and* ***union operations*** *on a* ***disjoint set data structure****. It* ***finds*** *the* ***root parent*** *of an element and determines whether if two elements are in the same set or not. If two elements are at different sets,* ***merge*** *the* ***smaller set to the larger set****. At the end, we can get the* ***connected components*** *in a graph.*

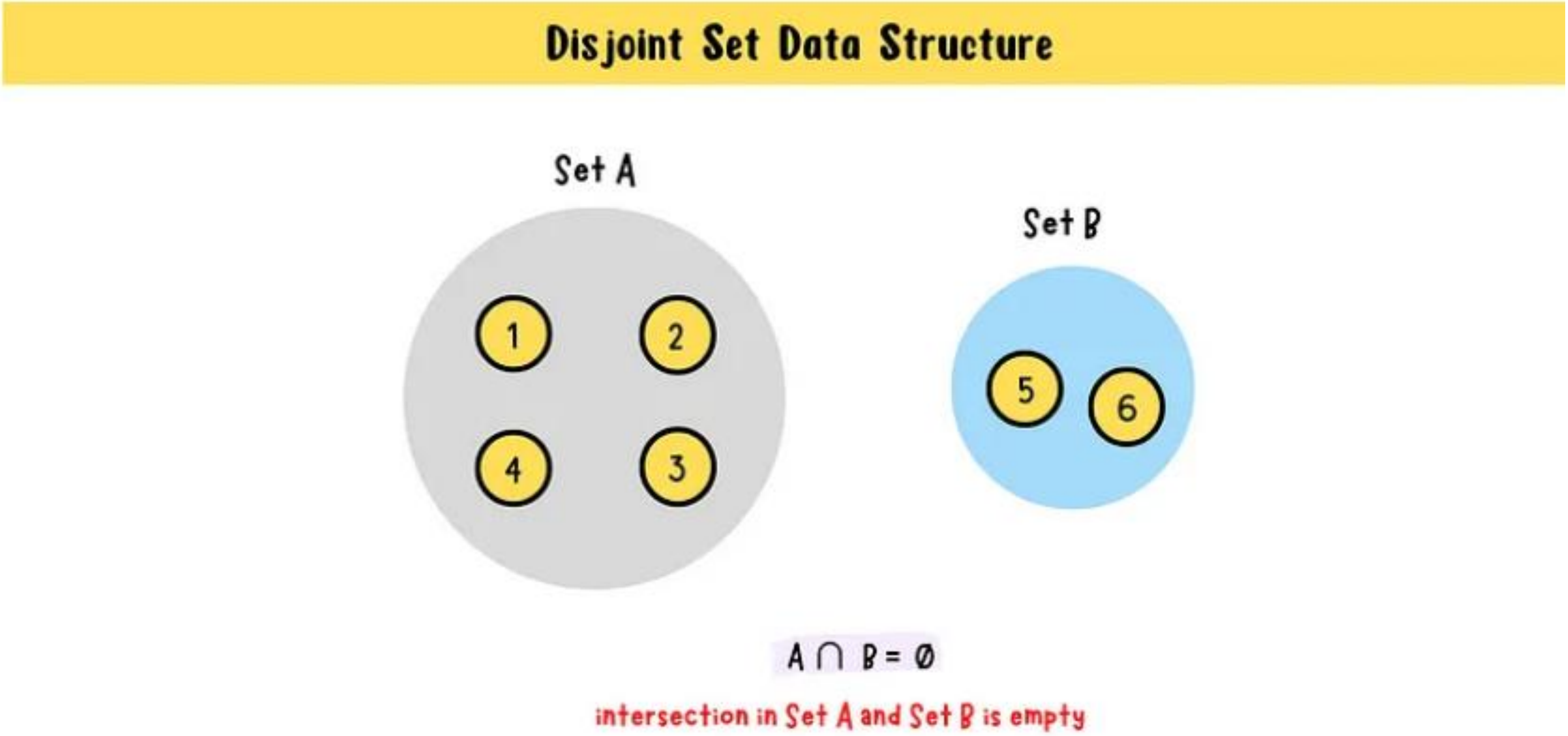
find operation: find root parent and determine if two elements are in the same set

union operation: merge a smaller set to a larger set if two elements are disjoint.



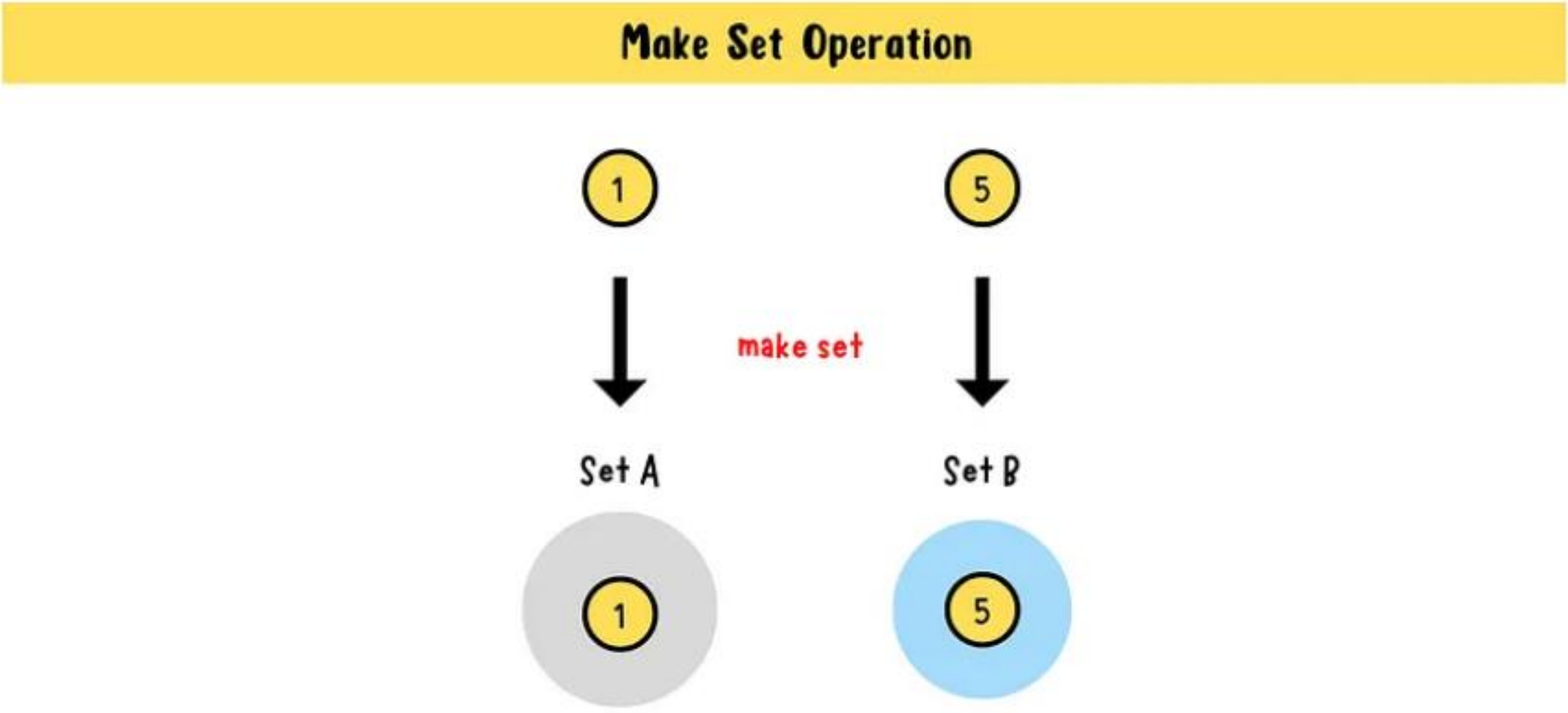
Disjoint Set Data Structure

Two or more sets have **no element in common** are called **disjoint sets.** Disjoint set data structure is also referred to as **union find data structure** because of its union and find operations.



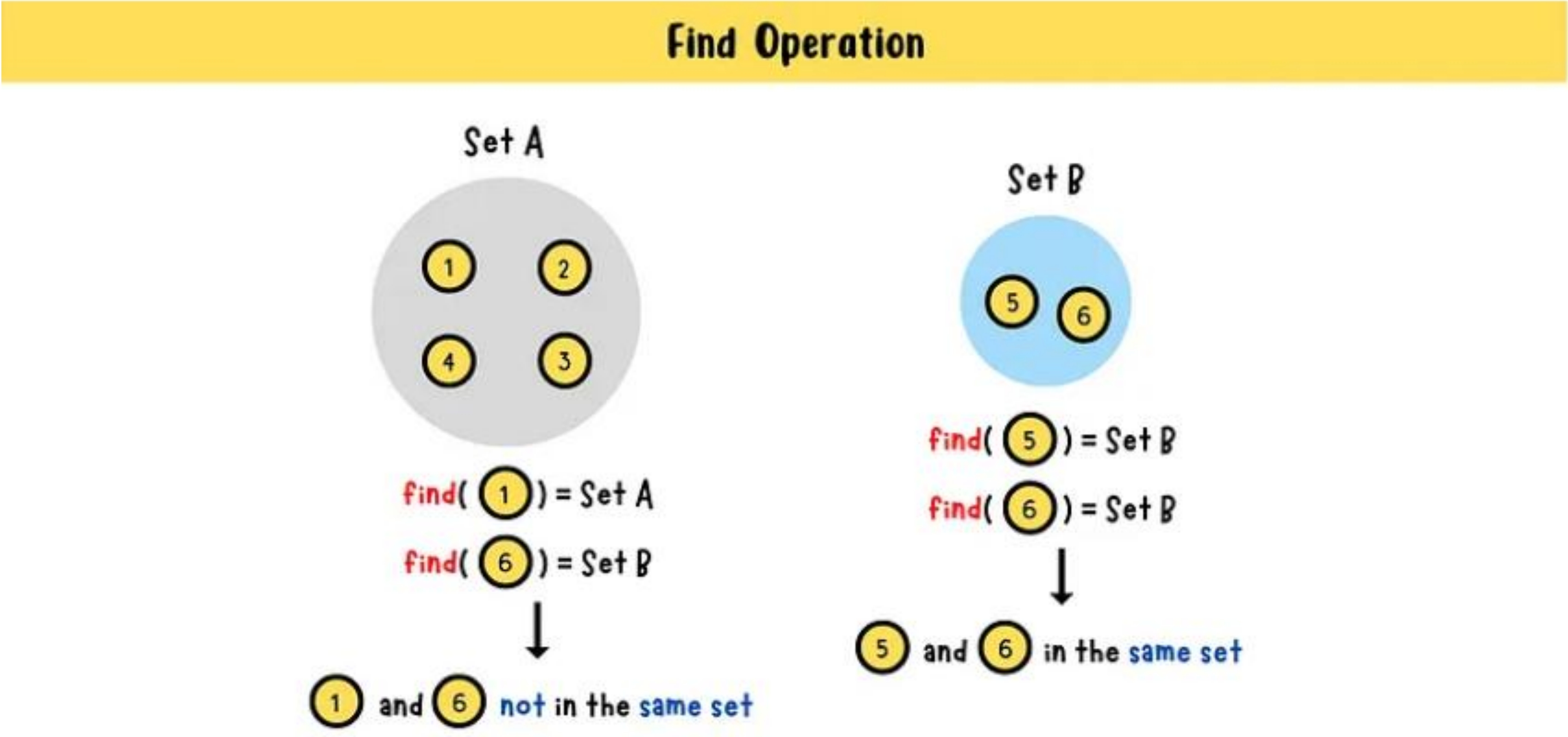
Disjoint set data structure supports three operations:

1.Make Set: create a new disjoint set contains only the given element.



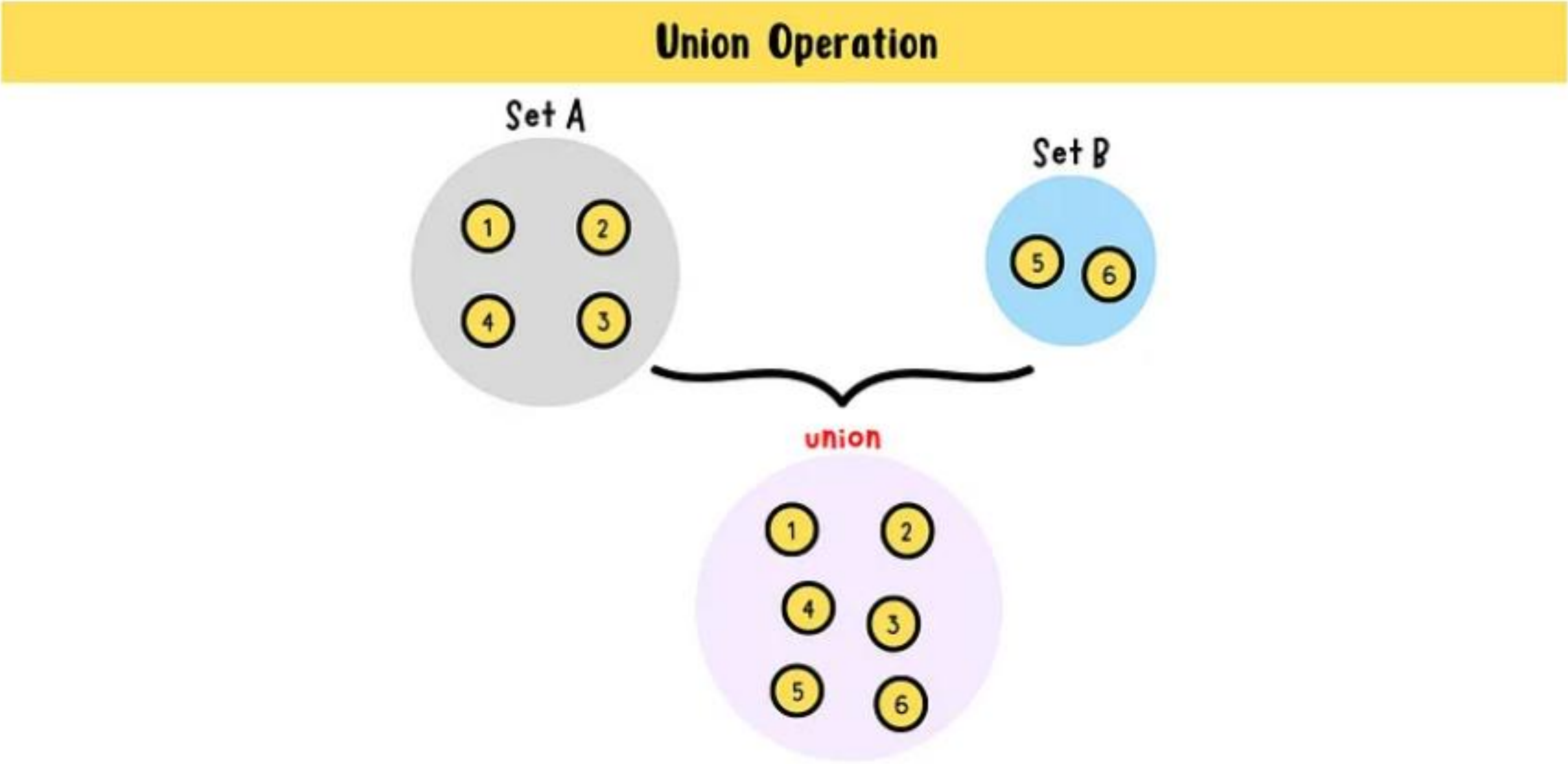
make set

2. Find: determine which subset a given element belongs to. It is used to decide whether if **two elements** are **disjoint or not**.



find

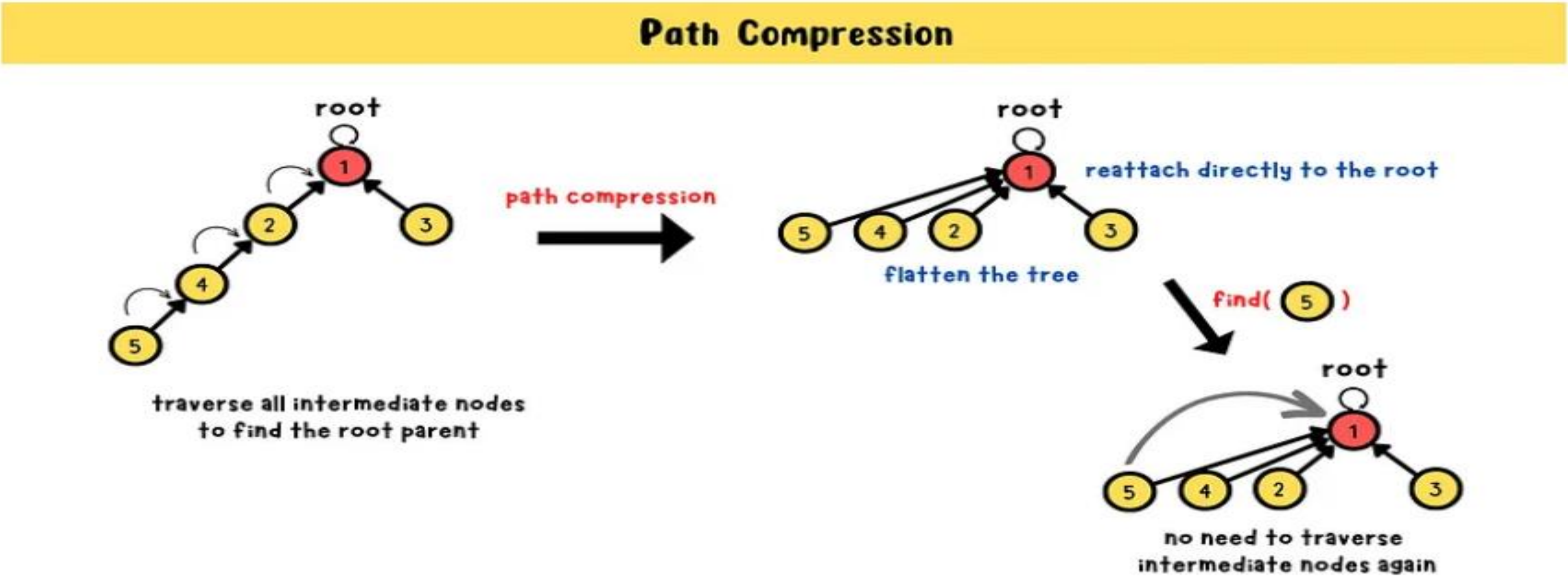
3. Union: **merge two disjoint sets** to a single disjoint set.



union

Path Compression

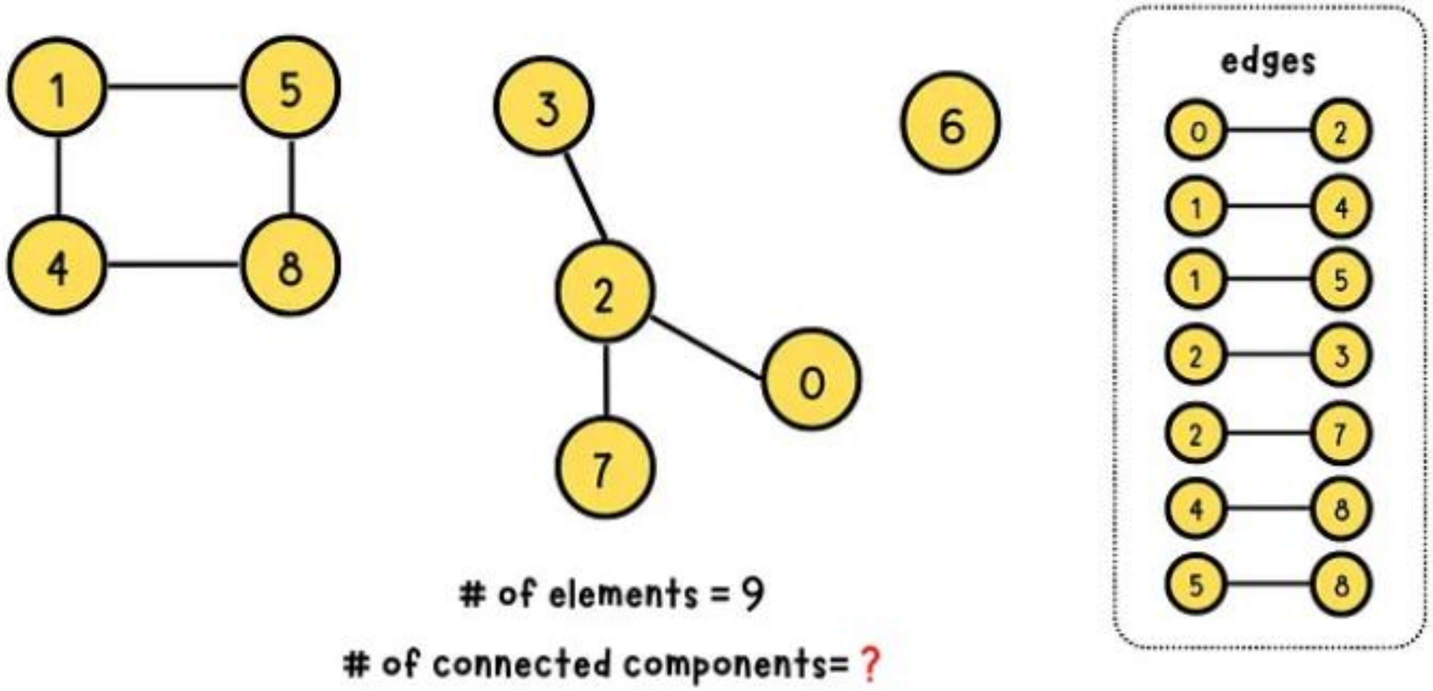
Path compression is a way to **flatten** the structure of the **tree** to make find operation more efficient. Find operation is used to find the **root parent** for a node. Without path compression, we have to travel upward the tree toward the root. The beauty of path compression is that we can find directly the root parent by **reattaching visited nodes to the root**.



How Does Union Find Algorithm Work?

Union find algorithm performs the **find operation** to find a given element's root parent and determine whether if two elements are in the same subset or not. If the two elements are in the same subset, they are already connected. Otherwise, they belong to different sets. Implement the **union operation** to merge the two disjoint sets to a signal set.

Graphical Explanation



•***step1***: Initialize **parent** and **size** arrays with the length of the total number of elements.

parent: store a node's root parent

size: store the total number of elements in a subset

step1.a: originally every node is a root node to itself

    parent[i] = i

step1.b: originally every set contains a single node

    size[i] = 1

•***step2***: **Traversal** through **all edges**

step2.a: find root parent and check if two subsets are in the same set

root1 == root2?

- Yes → in the same set → return

- No → step2.b

step2.b: merge the smaller set to the larger set

parent[root1] = root2 or parent[root2] = root1

step2.c: increment the size of the larger set by 1

size[root2] += 1 or size[root1] += 1

ESSENTIALS OF PROBLEM SOLVING

ALGEBRAIC AND TRANSCENDENTAL EQUATIONS

ESSENTIALS OF PROBLEM SOLVING

1) ALGEBRAIC EQUATION:

- Algebraic equations are equations formed using algebraic expressions, which involve only the operations of addition, subtraction, multiplication, division, and raising to a power. These equations have the general form:

$$P(x)=0$$

where $P(x)$ is a polynomial in x.

- Algebraic equations can be classified based on the degree of the polynomial.

ESSENTIALS OF PROBLEM SOLVING

Examples:

1. **Linear Equations (Degree 1):**

   $$ax + b = 0$$

   Solution: $x = -\frac{b}{a}$

2. **Quadratic Equations (Degree 2):**

   $$ax^2 + bx + c = 0$$

   Solution: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

3. **Cubic Equations (Degree 3):**

   $$ax^3 + bx^2 + cx + d = 0$$

4. **Higher-Degree Polynomial Equations:**

   $$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$$

ESSENTIALS OF PROBLEM SOLVING

2)TRANSCENDENTAL EQUATION:

- Transcendental equations involve transcendental functions, which are functions that cannot be expressed as finite polynomials.
- These functions include exponential, logarithmic, trigonometric, and inverse trigonometric functions.
- Transcendental equations often cannot be solved algebraically and require numerical methods or approximations.

Examples:

1. **Exponential Equations:**

   $a^x = b$

   Solution: $x = \log_a b$

2. **Logarithmic Equations:**

   $\log_a x = b$

   Solution: $x = a^b$

3. **Trigonometric Equations:**

   $\sin(x) = a$

   Solution: $x = \sin^{-1}(a) + 2k\pi$ or $x = \pi - \sin^{-1}(a) + 2k\pi$ for $k \in \mathbb{Z}$

4. **Mixed Equations:**

   $x = \cos(x)$

ESSENTIALS OF PROBLEM SOLVING

Solving Methods:

(1) For Algebraic Equations:

- **Factoring:** Breaking down the polynomial into simpler polynomials.
- **Quadratic Formula:** Specifically for quadratic equations.
- **Synthetic Division:** Useful for higher-degree polynomials.
- **Numerical Methods:** For polynomials that cannot be easily factored, such as Newton's method.

(2) For Transcendental Equations:

- **Graphical Methods:** Plotting the functions and finding points of intersection.
- **Numerical Methods:** Techniques like the bisection method, Newton-Raphson method, or fixed-point iteration.
- **Approximation:** Sometimes solutions are approximated using series expansion or other analytical methods.

ESSENTIALS OF PROBLEM SOLVING

1) Regula Falsi Method
2) Bisection Method
3) Iteration Method
4) Newton- Raphson Method
5) Secant Method
6) Ramanujan’s Method
7) Muller’s Method

Regula Falsi Method

- The Regula Falsi method, also known as the False Position method, is a root-finding algorithm that combines elements of the bisection method and linear interpolation.
- It is used to find the roots of a function $f(x)$ in a given interval $[a,b]$ where $f(a)$ and $f(b)$ have opposite signs (indicating that there is at least one root between them).

Regula Falsi Method

Regula Falsi Method

Algorithm Steps:

1. **Initial Interval Selection**: Choose initial points a and b such that $f(a) \cdot f(b) < 0$. This ensures that a root exists between a and b.

2. **Linear Interpolation**: Compute the point c where the straight line joining $(a, f(a))$ and $(b, f(b))$ crosses the x-axis:

$$c = a - \frac{f(a) \cdot (b - a)}{f(b) - f(a)}$$

$$c = \frac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$$

Regula Falsi Method

3. **Update Interval:**

   - If $f(c) = 0$, then c is the root.
   - If $f(a) \cdot f(c) < 0$, then set $b = c$.
   - If $f(b) \cdot f(c) < 0$, then set $a = c$.

4. **Iteration**: Repeat steps 2 and 3 until the value of c converges to the root within a desired tolerance level or a maximum number of iterations is reached.

Regula Falsi Method

1) Find a real root of $x^3 - 4x + 1 = 0$ using false position method.

Solution:

Put a =0 the f(0) = 1 ; positive

b = 1 then f(1) = -2 ; negative

Then the root lies between 0 and 1

$$c = \frac{a\, f(b) - b\, f(a)}{f(b) - f(a)}$$

c = 0.333

f(c) = f(0.333) = -0.2962= negative

Now the root lies between **0, 0.333**

$$X_2 = \frac{a\, f(b) - b\, f(a)}{f(b) - f(a)}$$

Regula Falsi Method

c = 0.2571 then f(c) = -0.0115 = negative

root lies between **0 and 0.2571**

$$c = \frac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$$

$\mathbf{X_3}$ **= 0.2541** then $f(X_3)$ = 0.00001 = positive

Root lies between **0.2541 and 0.2571**

$$X_4 = \frac{a\,f(b) - b\,f(a)}{f(b) - f(a)}$$

$\mathbf{X_4}$ **= 0.2541**

Required root is **0.2541.**

Regula Falsi Method

2) Find a real root of $3x^3 - 4x - 9 = 0$

Answer:1.7467

Regular Falsi Method

3) Using false-Position method find a real root of the equation $x \log x - 1.2 = 0$ in 3 steps.

Answer: 2.7406

Regular Falsi Method

```
def regula_falsi(f, a, b, tol=1e-6, max_iter=1000):
    """
    Regula Falsi method to find the root of the function f in the interval [a, b].
    Parameters:
    f (function): The function for which to find the root.
    a (float): The starting point of the interval.
    b (float): The ending point of the interval.
    tol (float): The tolerance level.
    max_iter (int): The maximum number of iterations.
    Returns:
    float: The approximate root of the function.
    """
```

Regular Falsi Method

```
if f(a) * f(b) >= 0:
    raise ValueError("The function must have opposite signs at the endpoints a and b.")
    for _ in range(max_iter):
    # Compute the point where the line crosses the x-axis
    c = a - f(a) * (b - a) / (f(b) - f(a))
    # Check for convergence
    if abs(f(c)) < tol:
        return c

    # Update the interval
    if f(a) * f(c) < 0:
        b = c
    else:
        a = c
  raise ValueError("The method did not converge within the maximum number of iterations.")
```

Regular Falsi Method

```
# Example usage:
def f(x):
    return x**3 - x - 2

# Find the root in the interval [1, 2]
root = regula_falsi(f, 1, 2)
print(f"The root is: {root}")
```

BISECTION METHOD

BISECTION METHOD: [BLOZANO METHOD]

- The Bisection Method is a root-finding technique that repeatedly bisects an interval and then selects a subinterval in which a root must lie.
- It's a straightforward and reliable method, especially for continuous functions that change sign over an interval.
- The method guarantees convergence if the initial interval is chosen correctly.

BISECTION METHOD

Bisection Method Algorithm

1. **Initial Interval Selection:**

   - Choose initial points a and b such that $f(a) \cdot f(b) < 0$. This ensures that there is at least one root in the interval $[a, b]$.

2. **Compute the Midpoint:**

   - Compute the midpoint c of the interval $[a, b]$:

$$c = \frac{a+b}{2}$$

BISECTION METHOD

3. **Update the Interval:**

   - Evaluate $f(c)$.
   - If $f(c) = 0$, then c is the root.
   - If $f(a) \cdot f(c) < 0$, then set $b = c$.
   - If $f(b) \cdot f(c) < 0$, then set $a = c$.

4. **Convergence:**

   - Repeat steps 2 and 3 until the interval $[a, b]$ is sufficiently small, or until the function value at c is close enough to zero within a specified tolerance.

BISECTION METHOD

1)Find the root of the equation $x^3 - x - 1 = 0$ using bisection method.

Solution:

X =0 : f(0) = 0-0-1 = -1 = negative.

X=1 : f(1) = 1-1-1 = -1 = negative.

X=2: f(2) = 8-2-1 = 5 = positive.

So, the root lies between **1 and 2.**

Then x_1 = 1+2 / 2 = 1.5

F(x_1) = 1.5^3 – 1.5 -1 = 7/8 positive.

The root lies between **1 and 1.5**

x_2 = 1 + 1.5 / 2 = 1.25

F(x_2) = -19/64 = negative

Root lies between **1.25 and 1.5**

x_3 = 1.25+1.5 / 2 = 1.375

F(x_3) = 2 = positive

BISECTION METHOD

....

Answer : 1.3247

BISECTION METHOD

2) Find the root of the equation $x^3 - 2x - 5 = 0$ by using bisection method.

Answer: $x_{12} = 2.0946$

ITERATION METHOD

- The Iteration Method, often called the Fixed-Point Iteration Method, is a root-finding technique that transforms the equation $f(x)$=0 into a form x=$g(x)$ and then iteratively applies the function $g(x)$g(x) to converge to the fixed point, which is the root of the original equation.

Steps of the Iteration Method

1. **Transform the Equation:**

   Rewrite $f(x) = 0$ into $x = g(x)$.

2. **Initial Guess:**

   Choose an initial guess x_0.

ITERATION METHOD

3. **Iterative Process:**

   Apply the iterative formula:

$$x_{n+1} = g(x_n)$$

   until the sequence $\{x_n\}$ converges to a fixed point x where $x = g(x)$.

4. **Convergence Check:**

   Continue iterating until the difference $|x_{n+1} - x_n|$ is less than a specified tolerance level or the maximum number of iterations is reached.

ITERATION METHOD

1) Find the root of equation $x^3 - x - 11 = 0$ using iteration method.

Solution:

X=0 : f(0) = -11 = negative

X=1 : f(1) = -11 = negative

X= 2: f(2) = -5 = negative

X =3: f(3) = 13 = positive

So the root lies between **2 and 3**

Let $\mathbf{x_0 = 2.5}$ (x_0 = 2 or 3 or (2+3)/2)

From the given equation: $x^3 - x - 11 = 0$

$x^3 = x + 11$

$X = (x+11)^{1/3}$

Let $\phi(x) = (x+11)^{1/3}$

ITERATION METHOD

$x_1 = \phi(x_0) = (x_0 + 1\ 1)^{1/3}$

$x_1 = (2.5 + 11)^{1/3} = 2.3811$

$x_2 = \phi(x_1) = (x_1 + 11)^{1/3}$

$x_2 = 2.3740$

$x_3 = \phi(x_2) = (x_2 + 11)^{1/3}$

$x_3 = 2.3736$

$x_4 = \phi(x_3) = (x_3 + 11)^{1/3}$

$x_4 = 2.3736$

Hence the root is 2.3736

NEWTON RAPHSON

NEWTON RAPHSON METHOD

NEWTON RAPHSON

The Newton-Raphson method is an iterative numerical technique for finding the root of a real-valued function f(x)f(x)f(x). The formula for the method is derived from the Taylor series expansion of the function. The iterative formula is:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

for n = 0,1,2…….

NEWTON RAPHSON

where:

- x_n is the current approximation of the root,
- x_{n+1} is the next approximation,
- $f(x_n)$ is the value of the function at x_n,
- $f'(x_n)$ is the value of the derivative of the function at x_n.

NEWTON RAPHSON

Steps:

1. **Choose an initial guess** x_0.
2. **Evaluate** $f(x_n)$ and $f'(x_n)$.
3. **Compute the next approximation** using $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$.
4. **Check for convergence:** If $|x_{n+1} - x_n|$ is less than a predefined tolerance, then x_{n+1} is accepted as the root. Otherwise, repeat from step 2 using x_{n+1} as the new guess.

NEWTON RAPHSON

1) Using Newton Raphson method solve $x^3 - 3x - 5 = 0$

Solution:

$X = 0 : f(x) = -5$

$X=1 : f(1) = -7$

$X=2 : f(2) = -3$ = negative

$X=3 : f(3) = 13$ = positive

So, the root lies between **2 and 3.**

Let $x_0 = 2+3 / 2 = 2.5$

Put n=0

$x_1 = x_0 - f(x_0)/f'(x_0)$

$x_1 = 2.3015$

put n=1

$x_2 = x_1 - f(x_1)/f'(x_1)$

$x_2 = 2.2792$

NEWTON RAPHSON

Put n=2

$x_3 = x_2 – f(x_2)/f'(x_2)$

$X_3 = 2.2790$

put n = 3

$x_4 = x_3 – f(x_3)/f'(x_3)$

$x_4 = 2.2790$

Hence required root is 2.2790.

NEWTON RAPHSON

2) Find the root of the equation $x^3 - x - 1 = 0$

Answer: $x_4 = 1.32475$

SECANT METHOD

SECANT METHOD

SECANT METHOD

- The secant method is another iterative numerical technique for finding the root of a function. Unlike the Newton-Raphson method, it does not require the calculation of the derivative.
- Instead, it uses two initial approximations to generate a sequence of increasingly accurate estimates of the root.

SECANT METHOD

Secant Method

The secant method formula is given by:

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

where:

- x_{n+1} is the next approximation,
- x_n and x_{n-1} are the current and previous approximations, respectively,
- $f(x_n)$ and $f(x_{n-1})$ are the values of the function at x_n and x_{n-1}.

SECANT METHOD

Steps:

1. **Choose two initial approximations** x_0 and x_1.
2. **Evaluate** $f(x_n)$ and $f(x_{n-1})$.
3. **Compute the next approximation** using $x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$.
4. **Check for convergence:** If $|x_{n+1} - x_n|$ is less than a predefined tolerance, then x_{n+1} is accepted as the root. Otherwise, repeat from step 2 using x_n and x_{n+1} as the new guesses.

SECANT METHOD

1) Find root of $x^3 - 5x + 1 = 0$ in interval (0,1) using secant method.

Solution:

$X_0 = 0 : f(X_0) = 0-0+1 = 1$

$X_1 = 1: \ f(X_1) = 1-5+1 = -3$

$X_{n+1} = x_{n-1} f(x_n) - x_n \ f(x_{n-1}) / f(x_n) - f(x_{n-1})$

put n=1

$X_2 = x_0 f(x_1) - x_1 \ f(x_0) / f(x_1) - f(x_0)$

$X_2 = 0.25$

$f(X_2) = -0.234$

Put n=2

$X_3 = x_1 f(x_2) - x_2 \ f(x_1) / f(x_2) - f(x_1)$

$X_3 = 0.1864$

$f(X_3) = 0.07428$

SECANT METHOD

Put n=3

$X_{4} = x_2 f(x_3) – x_3 \; f(x_2) / f(x_3) – f(x_2)$

$X_4 = 0.20164$

$f(X_4) = -0.00048$

Put n=4

$X_{5} = x_3 f(x_4) – x_4 \; f(x_3) / f(x_4) – f(x_3)$

$X_5 = 0.20081$

$f(X_5) = 0$

Hence required root is 0.20081

RAMANUJAN'S METHOD

RAMANUJAN'S METHOD

RAMANUJAN'S METHOD

RAMANUJAN'S METHOD:

It is used to determine smallest root of the equation $f(x) = 0$

Step 1: $F(x) = 1 - (a_1 x + a_2 x^2 + a_3 x^3 + \ldots\ldots\ldots\ldots + a_n x^n)$

Step 2: For smallest value of x we can write

$[1 - (a_1 x + a_2 x^2 + a_3 x^3 + \ldots\ldots\ldots\ldots + a_n x^n]^{-1} = b_1 + b_2 x + b_3 x^2 + \ldots\ldots\ldots$

Expanding LHS equation by using binomial theorem.

$1 + (a_1 x + a_2 x^2 + \ldots.) + (a_1 x + a_2 x^2 + \ldots..)^2 + \ldots\ldots = b_1 + b_2 x + \ldots\ldots\ldots.$

By comparing the co efficient of x which has like powers on both sides we get

Step 3:

$b_1 = 1$

$b_2 = a_1 b_1$

RAMANUJAN'S METHOD

$b_3 = a_1 b_2 + a_2 b_1$

$b_4 = a_1 b_3 + a_2 b_2 + a_3 b_1$

$b_5 = a_1 b_4 + a_2 b3 + a_3 b_2 + a_4 b_1$

……………………………………………………………….

Step 4:

roots :

b_n / b_{n+1} i.e. b_1 / b_2 ; b_2 / b_3 ; b_3 / b_4; ………………………

called the convergent approach in the limit the smallest root of f(x) = 0

RAMANUJAN'S METHOD

1) Find the smallest root of the equation $x^3 - 9x^2 + 26x - 24 = 0$.

Solution:

$f(x) = x^3 - 9x^2 + 26x - 24$

$f(x) = -1/24\,(x^3 - 9x^2 + 26x - 24)$

$f(x) = 1 - (26x - 9x^2 + x^3) / 24$

$f(x) = 1 - [\,13/12\, x - 3/8\, x^2 + 1/24\, x^3\,]$

$a_1 = 13/12$; $a_2 = -3/8$; $a_3 = 1/24$; $a_4 = a_5 = a_6 = 0$

$1 - [\,13/12\, x - 3/8\, x^2 + 1/24\, x^3\,] = b_1 + b_2 x + b_3 x^2 + b_4 x^4 + \ldots\ldots\ldots$

Now equate co efficient of like powers of x on both sides of equation.

$b_1 = 1$

$b_2 = a_1 b_1 = 13/12 = 1.0833$

$b_3 = a_1 b_2 + a_2 b_1 = 0.7986$

$b_4 = a_1 b_3 + a_2 b_2 + a_3 b_1 = 0.5007$

$b_5 = a_1 b_4 + a_2 b_3 + a_3 b_2 + a_4 b_1 = 0.2880$

$b_6 = a_1 b_5 + a_2 b_4 + a_3 b_3 + a_4 b_2 + a_5 b_1 = 0.1575$

$b_7 = a_1 b_6 + a_2 b_5 + a_3 b_4 + a_4 b_3 + a_5 b_2 + a_6 b_1 = 1.8862$

$b_8 = a_1 b_7 + a_2 b_6 + a_3 b_5 + a_4 b_4 + a_5 b_3 + a_6 b_2 + a_7 b_1 = 1.8862$

$b_9 = a_1 b_8 + a_2 b_7 + a_3 b_6 + a_4 b_5 + a_5 b_4 + a_6 b_3 + a_7 b_2 + a_8 b_1 = 0.223$

RAMANUJAN'S METHOD

Roots:

$b_1 / b_2 = 0.923$

$b_2 / b_3 = 1.356$

$b_3 / b_4 = 1.595$

$b_4 / b_5 = 1.7385$

$b_5 / b_6 = 1.8285$

$b_6 / b_7 = 0.0835$

$b_7 / b_8 = 1$

RAMANUJAN'S METHOD

$b_8 / b_9 = 8.458$

The smallest root of the equation is 2.

RAMANUJAN’S METHOD

2) Find the smallest root of the equation $2x^3 - 6x^2 + 11x - 7 = 0$.

Solution:

$f(x) = 2x^3 - 6x^2 + 11x - 7 = 0$

$f(x) = -1/7\ (-7 + 11x - 6x^2 + 2x^3)$

Step 1: $F(x) = 1- \{(11x - 6x^2 + 2x^3)\} / 7$

We can write this as

Step 2:

$\{1- [(11x - 6x^2 + 2x^3)] / 7\}^{-1} = b_1 + b_2 x + b_3 x^2 + b_4 x^4 + \ldots\ldots\ldots$

Now equate co efficient of like powers of x on both sides of equation.

$a_1 = 11/7$; $a_2 = -6 / 7$; $a_3 = 2/7$; $a_4 = a_5 = a_6 = 0$

Step 3:

$b_1 = 1$

$b_2 = a_1 b_1 = 11/7$

RAMANUJAN'S METHOD

$b_3 = a_1 b_2 + a_2 b_1 = 79/49=1.612$

$b_4 = a_1 b_3 + a_2 b_2 + a_3 b_1 = 505 /345=1.472$

$b_5 = a_1 b_4 + a_2 b3 + a_3 b_2 + a_4 b_1 = 3975 / 2401=1.655$

Step 4:

ROOTS:

$b_1 / b_2 = 1 / 11/7 = 7/11 = 0.6363$

$b_2 / b_3 = 11 * 49 / 7 * 79 = 0.9746$

$b_3 / b_4 = 0.9788$

$b_4 / b_5 = 0.9949$

The smallest root of equation is 1.

MULLER’S METHOD

MULLER’S METHOD

- Muller's method is an iterative algorithm used to find roots of a function.
- It is a generalization of the secant method and uses a quadratic polynomial to approximate the function.

Steps of Muller's Method:

- **Choose Initial Approximations:** Select three initial guesses x_0, x_1 and x_2 that are close to the root.
- **Compute the Differences:** Calculate the differences between these points:

MULLER’S METHOD

Initial approximations x_{i-2} ; x_{i-1} ; x_i

x_{i-2} then $y_{i-2} = f(x_{i-2})$

X_{i-1} then $y_{i-1} = f(x_{i-1})$

X_i then $y_i = f(x_i)$

$\Delta_i = y_i - y_{i-1}$

$\Delta_{i-1} = y_{i-1} - y_{i-2}$

$h_i = x_i - x_{i-1}$

$h_{i-1} = x_{i-1} - x_{i-2}$

MULLER'S METHOD

Step 3:Formula to find iterations:

$$x_{i+1} = x_i - \frac{2y_i}{B \pm \sqrt{B^2 - 4Ay_i}}$$

$$A = \frac{1}{(h_{i-1} + h_i)} \left[\frac{\Delta_i}{h_i} - \frac{\Delta_{i-1}}{h_{i-1}} \right]$$

$$B = \frac{\Delta_i}{h_i} + Ah_i$$

MULLER'S METHOD

Step 4:

Iterate: Use x_1 , x_2 , and x_3 for the next iteration and repeat the process until convergence.

Step 5:

The method converges when the difference between successive approximations is smaller than a pre-defined tolerance level.

MULLER'S METHOD

1) Using Muller's method, find the roots of the equation

$F(x) = x^3 - x - 1 = 0$with the initial approximations 0,1 and 2.

MULLER'S METHOD

1) Using Muller's method, find the roots of the equation

$F(x) = x^3 - x - 1 = 0$with the initial approximations 0,1 and 2.

Solution:

$F(x) = x^3 - x - 1$

Initial approximations $x_{i-2} = 0$; $x_{i-1} = 1$; $x_i = 2$

$x_{i-2} = 0$ then $y_{i-2} = f(0) = 0^3 - 0 - 1 = -1$

$x_{i-1} = 1$ then $y_{i-1} = f(1) = 1^3 - 1 - 1 = -1$

$x_i = 2$ then $y_i = f(2) = 2^3 - 2 - 1 = 5$

$\Delta_i = y_i - y_{i-1} = 5+1 = 6$

$\Delta_{i-1} = y_{i-1} - y_{i-2} = 0$

MULLER'S METHOD

1) Using Muller's method, find the roots of the equation

$F(x) = x^3 – x - 1 = 0$with the initial approximations 0,1 and 2.

Solution:

$F(x) = x^3 – x – 1$

Initial approximations $x_{i-2} = 0$; $x_{i-1} = 1$; $x_i = 2$

$x_{i-2} = 0$ then $y_{i-2} = f(0) = 0^3 – 0 -1 = -1$

$x_{i-1} = 1$ then $y_{i-1} = f(1) = 1^3 – 1- 1 = -1$

$x_i = 2$ then $y_i = f(2) = 2^3 – 2 – 1 = 5$

$\Delta_i = y_i - y_{i-1} = 5+1 = 6$

$\Delta_{i-1} = y_{i-1} - y_{i-2} = 0$

MULLER'S METHOD

$h_i = x_i - x_{i-1} = 2\text{-}1 = 1$

$h_{i-1} = x_{i-1} - x_{i-2} = 1\text{-}0 = 1$

$$A = \frac{1}{(h_{i-1}+h_i)}\left[\frac{\Delta_i}{h_i} - \frac{\Delta_{i-1}}{h_{i-1}}\right]$$

A = 3

$$B = \frac{\Delta_i}{h_i} + Ah_i$$

B = 9 and $\sqrt{B^2-4Ay_i}$ **=** √'21

MULLER'S METHOD

First iteration:

$$x_{i+1} = x_i - \frac{2y_i}{B \pm \sqrt{B^2 - 4Ay_i}}$$

$x_{i+1} = 1.26376$

Let $x_{i-2} = 1$; $x_{i-1} = 2$; $x_i = 1.26376$

$x_{i-2} = 1$ then $y_{i-2} = f(1) = -1$

$x_{i-1} = 2$ then $y_{i-1} = f(2) = 5$

$x_i = 1.26376$ then $y_i = f(1.26376) = -0.24542$

MULLER'S METHOD

$\Delta_i = y_i - y_{i-1} = -5.24542$

$\Delta_{i-1} = y_{i-1} - y_{i-2} = 6$

$h_i = x_i - x_{i-1} = -0.73624$

$h_{i-1} = x_{i-1} - x_{i-2} = 1$

$$A = \frac{1}{(h_{i-1}+h_i)}\left[\frac{\Delta_i}{h_i} - \frac{\Delta_{i-1}}{h_{i-1}}\right]$$

$A = 4.26375$

$$B = \frac{\Delta_i}{h_i} + Ah_i$$

$B = 3.98546$ $\quad \sqrt{B^2 - 4Ay_i} = 4.47990$

MULLER'S METHOD

Second Iteration:

$$x_{i+1} = x_i - \frac{2y_i}{B \pm \sqrt{B^2 - 4Ay_i}}$$

x_{i+1} = 1.32174

Let x_{i-2} = 2 ; x_{i-1} = 1.26376 ; x_i = 1.32174

x_{i-2} = 2 then y_{i-2} = f(2) = 5

x_{i-1} = 1.26376 then y_{i-1} = f(1.26376) = -0.24542

x_i = 1.32174 then y_i = f(1.1.32174) = -0.01266

MULLER'S METHOD

$\Delta_i = y_i - y_{i-1} = 0.23276$

$\Delta_{i-1} = y_{i-1} - y_{i-2} = -5.24542$

$h_i = x_i - x_{i-1} = 0.05798$

$h_{i-1} = x_{i-1} - x_{i-2} = -0.73624$

$$A = \frac{1}{(h_{i-1}+h_i)}\left[\frac{\Delta_i}{h_i} - \frac{\Delta_{i-1}}{h_{i-1}}\right]$$

A = 4.58549

$$B = \frac{\Delta_i}{h_i} + Ah_i$$

B = 4.28035 $\sqrt{B^2 - 4Ay_i}$ = 4.30739

MULLER'S METHOD

Third Iteration:

$$x_{i+1} = x_i - \frac{2y_i}{B \pm \sqrt{B^2 - 4Ay_i}}$$

$x_{i+1} = 1.32469$

Let $x_{i-2} = 1.26376$; $x_{i-1} = 1.32174$; $x_i = 1.32469$

$x_{i-2} = 1.26376$ then $y_{i-2} = -0.24542$

$x_{i-1} = 1.32174$ then $y_{i-1} = -0.01266$

$x_i = 1.32469$ then $y_i = -\ 0.00012$

MULLER'S METHOD

$\Delta_i = y_i - y_{i-1} = 0.01254$

$\Delta_{i-1} = y_{i-1} - y_{i-2} = 0.23276$

$h_i = x_i - x_{i-1} = 0.01266$

$h_{i-1} = x_{i-1} - x_{i-2} = 0.05798$

$$A = \frac{1}{(h_{i-1} + h_i)}\left[\frac{\Delta_i}{h_i} - \frac{\Delta_{i-1}}{h_{i-1}}\right]$$

A = 3.87920

$$B = \frac{\Delta_i}{h_i} + A h_i$$

B = 4.26229 $\qquad \sqrt{B^2 - 4Ay_i} = 4.26472$

MULLER'S METHOD

Fourth Iteration:

$$x_{i+1} = x_i - \frac{2y_i}{B \pm \sqrt{B^2 - 4Ay_i}}$$

$x_{i+1} = 1.32472$

∴The required root is 1.32472, correct to 4 decimal places.

Given N, we want to find x such that $x = \frac{1}{N}$. To apply the Newton-Raphson method, we define the function:

$$f(x) = \frac{1}{x} - N$$

We seek the root of $f(x) = 0$. The derivative of $f(x)$ is:

$$f'(x) = -\frac{1}{x^2}$$

The Newton-Raphson iteration formula is given by:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Substituting $f(x)$ and $f'(x)$ into this formula, we get:

$$x_{n+1} = x_n - \frac{\frac{1}{x_n} - N}{-\frac{1}{x_n^2}} = x_n - \frac{1 - Nx_n}{-\frac{1}{x_n}} = x_n + x_n(1 - Nx_n) = x_n(2 - Nx_n)$$

Thus, the iteration formula for finding the reciprocal of N is:

$$x_{n+1} = x_n(2 - Nx_n)$$

ESSENTIALS OF PROBLEM SOLVING

NUMERICAL INTEGRATION & ORDINARY DIFFERENTIAL EQUATIONS

ESSENTIALS OF PROBLEM SOLVING

NUMERICAL INTEGRATION

a) TRAPEZOIDAL RULE
b) SIMPSON'S 1/3 RULE
c) SIMPSON'S 3/8 RULE

ORDINARY DIFFERENTIAL EQUATIONS

a) TAYLOR'S SERIES
b) EULER'S METHOD
c) RUNGE-KUTTA METHOD

NUMERICAL INTEGRATION

Trapezoidal Rule:

$$\int_a^b f(x)dx = \frac{h}{2}\left[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + \ldots\ldots\ldots\ldots) \right]$$

Where $h = \frac{b-a}{n}$; h = length of subintervals

NUMERICAL INTEGRATION

Example1: Evaluate $\int_0^1 x^3$ dx with 5 sub- intervals by Trapezoidal rule.

Solution: $\int_a^b f(x)dx = \int_0^1 (x^3)$

y = f(x) = x^3 ; h = $\frac{b-a}{n} = \frac{1-0}{5}$ = 0.2

X	0	0.2	0.4	0.6	0.8	1
Y	0	0.008	0.064	0.216	0.512	1

$$\int_0^1 x^3\,dx = \frac{h}{2}[(y_0 + y_n) + 2(y_1 + y_2 + y_3 + \ldots\ldots\ldots\ldots)]$$

$$= \frac{h}{2}[(y_0 + y_5) + 2(y_1 + y_2 + y_3 + y_4)]$$

$$= \frac{0.2}{2}[(0 + 1) + 2(0.008 + 0.064 + 0.216 + 0.512)]$$

$$= 0.26$$

NUMERICAL INTEGRATION

Example2: Evaluate $\int_0^1 (1 + x^3)^{1/2}$ dx taking h = 0.1 using Trapezoidal rule

Answer: 1.11226

NUMERICAL INTEGRATION

Example3: Evaluate $\int_0^{\pi/2} e^{\sin x}\, dx$ using Trapezoidal rule.

Solution: h = $\frac{\pi}{12}$ let n = 6

X	0	$\frac{\pi}{12}$	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{5\pi}{12}$	$\frac{\pi}{2}$
Y	1	1.2954	1.6487	2.0281	2.3774	2.6272	2.7183

$$\int_a^b f(x)dx = \frac{h}{2}[(y_0 + y_6) + 2(y_1 + y_2 + y_3 + y_4 + y_5)]$$

$\int_0^{\pi/2} e^{\sin x}\, dx$ =

NUMERICAL INTEGRATION

Simpson $\frac{1}{3}$ Rule:

$$\int_a^b f(x)dx \quad = \quad \frac{h}{3}\left[(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \dots) + 2(y_2 + y_4 + y_6 + \dots) \right]$$

Where h = $\frac{b-a}{n}$; h = length of subintervals

$y_1 + y_3 + y_5 + \dots\dots$ = Sum of odd terms

$y_2 + y_4 + y_6 + \dots\dots$ = Sum of even terms

NUMERICAL INTEGRATION

Example1: Find solution using Simpson's 1/3 rule.

X	0.0	0.1	0.2	0.3	0.4
Y	1.0	0.9975	0.9900	0.9776	0.8604

Solution:

Given $h = 0.1$

$$\int_a^b f(x)dx = \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots) + 2(y_2 + y_4 + y_6 + \ldots)]$$

$$\int_a^b f(x)dx = 0.39136$$

NUMERICAL INTEGRATION

Example1: Evaluate $\int_0^1 x^3$ dx with 5 sub- intervals by Simpson's 1/3 rule.

Solution: $\int_a^b f(x)dx = \int_0^1 (x^3)$

y = f(x) = x^3 ; h = $\frac{b-a}{n} = \frac{1-0}{5}$ = 0.2

X	0	0.2	0.4	0.6	0.8	1
Y	0	0.008	0.064	0.216	0.512	1

$$\int_0^1 x^3\,dx = \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots) + 2(y_2 + y_4 + y_6 + \ldots)]$$

$$= \frac{h}{3}[(y_0 + y_5) + 4(y_1 + y_3) + 2((y_2 + y_4)]$$

$$= \frac{0.2}{3}[(0 + 1) + 4(0.008 + 0.216) + 2(0.064 + 0.512)]$$

= 0.2011

NUMERICAL INTEGRATION

2) Evaluate $\int_0^6 \frac{1}{1+x^2}$ dx with 6 sub - intervals using simpson's 1/3 rule.

Solution: Given n = 6 ; h = $\frac{b-a}{n}$ = $\frac{6-0}{6}$ = 1

X	0	1	2	3	4	5	6
y	1	0.5	0.2	0.1	0.0588	0.0385	0.027

$\int_a^b f(x)dx = \frac{h}{3}[(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots.) + 2(y_2 + y_4 + y_6 + \ldots.)]$

$\int_0^6 \frac{1}{1+x\,2}$ dx = 1.3662

NUMERICAL INTEGRATION

Simpson $\frac{3}{8}$ Rule:

$$\int_a^b f(x)dx = \frac{3h}{8}\left[(y_0 + y_n) + 3(y_1 + y_2 + y_4 + \ldots.) + 2(y_3 + y_6 + y_9 + \ldots.)\right]$$

Where $h = \frac{b-a}{n}$; h = length of subintervals

$y_3 + y_6 + y_9 + \ldots\ldots\ldots$ = Multiples of 3

$y_1 + y_2 + y_4 + \ldots\ldots\ldots$ = Remaining terms

NUMERICAL INTEGRATION

Example1: Evaluate $\int_0^1 x^3$ dx with 5 sub- intervals by Simpson's 3/8 rule.

Solution: $\int_a^b f(x)dx = \int_0^1 (x^3)$

y = f(x) = x^3 ; h = $\frac{b-a}{n} = \frac{1-0}{5}$ = 0.2

X	0	0.2	0.4	0.6	0.8	1
Y	0	0.008	0.064	0.216	0.512	1

$\int_0^1 x^3$ dx = $\frac{3h}{8}$ [($y_0 + y_n$) + 3 ($y_1 + y_2 + y_4$ + ...)+ 2($y_3 + y_6 + y_9$ + ...)]

= $\frac{3h}{8}$ [($y_0 + y_5$) + 3($y_1 + y_2 + y_4$)+2(y_3)]

= $\frac{0.6}{8}$ [(0 + 1) + 4 (0.008 + +0.064+0.512)+2(0.216)]

= 0.2388

NUMERICAL INTEGRATION

1) Evaluate $\int_0^6 \frac{1}{1+x^2}$ dx with 6 sub - intervals using simpson's 3/8 rule.

Solution: Given n = 6 ; h = $\frac{b-a}{n}$ = $\frac{6-0}{6}$ = 1

X	0	1	2	3	4	5	6
y	1	0.5	0.2	0.1	0.0588	0.0385	0.027

$\int_a^b f(x)dx$ = $\frac{3h}{8}$ [$(y_0 + y_n) + 3(y_1 + y_2 + y_4 + \ldots) + 2(y_3 + y_6 + y_9 + \ldots)$]

$\int_0^6 \frac{1}{1+x\,2}$ dx = 1.3571

ESSENTIALS OF PROBLEM SOLVING

ORDINARY DIFFERENTIAL EQUATIONS:

a) **TAYLOR'S SERIES**
b) **EULER'S METHOD**
c) **RUNGE-KUTTA METHOD**

ORDINARY DIFFERENTIAL EQUATIONS

TAYLOR'S SERIES:

To find numerical solutions of ordinary differential equation

$\frac{dy}{dx}$= f(x, y) given initial condition $y(x_0) = y_0$ we use Taylor series method formula.

$$y_{n+1} = y_n + h\, y_n^{I} + \frac{h^2}{2!} y_n^{II} + \frac{h^3}{3!} y_n^{III} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

For n = 0, 1, 2…………… : and $h = x - x_0$

i.e $$y_1 = y_0 + h\, Y_0^{I} + \frac{h^2}{2!} Y_0^{II} + \frac{h^3}{3!} Y_0^{III} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

ORDINARY DIFFERENTIAL EQUATIONS

Example: Using Taylors series method find the approximate value of Y at $x = 0.2$ for the differential equation $\frac{dy}{dx} - 2y = 3\,e^{x}$; $y(0) = 0$

Solution: Given $y(x_0) = y_0$; $y(0) = 0$ then $x_0 = 0$ and $y_0 = 0$

$Y^{I} = 3\,e^{x} + 2y$; $Y_0^{I} = 3\,e^{0} + 2\,(0) = 3$

$Y^{II} = 3\,e^{x} + 2\,Y^{I}$; $Y_0^{II} = 3\,e^{0} + 2\,Y_0^{I} = 3 + 2 * 3 = 9$

$Y^{III} = 3\,e^{x} + 2\,Y^{II}$; $Y_0^{III} = 3\,e^{0} + 2\,Y_0^{II} = 3 + 2\,(9) = 21$

$Y^{IV} = 3\,e^{x} + 2\,Y^{III}$; $Y_0^{IV} = 3\,e^{0} + 2\,Y_0^{III} = 3 + 2(21) = 45$

ORDINARY DIFFERENTIAL EQUATIONS

$h = x - x_0 = 0.2 - 0 = 0.2$

$$\mathbf{y_1 = y_0 + h}\, Y_0^{\,I} + \frac{\mathbf{h^2}}{\mathbf{2!}} Y_0^{\,II} + \frac{\mathbf{h^3}}{\mathbf{3!}} Y_0^{\,III} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$Y(x_1) = y\,(0.2) = 0 + 0.2 * 3 + (0.2)^2 / 2 * 9 + 0.2^3 / 3! * 21 + 0.2^4 / 4! * 45$

$\mathbf{y_1 = y\,(0.2) = 0.811}$

ORDINARY DIFFERENTIAL EQUATIONS

2) Using Taylor's method $dy / dx = x^2 + y^2$ for $x = 0.4$ given that $y = 0$ when $x = 0$.

Answer: $y(0.4) = 0.0213$

ORDINARY DIFFERENTIAL EQUATIONS

3) Find y (0.2) for $y^{I} = x^2 y - 1$: $y(0) = 1$ with step length 0.1 using Taylor series method.

Solution: $y^{I} = x^2 y - 1$; $y(0) = 1$ then $x_0 = 0$ and $y_0 = 1$

$y^{I} = x^2 y - 1$; $Y_0^{I} = x_0^2 y_0 - 1 = 0 - 1 = -1$

$y^{II} = 2xy + x^2 y^{I}$; $Y_0^{II} = 2x_0 y_0 + x_0^2 Y_0^{I} = 2(0) + 0 = 0$

$y^{III} = 2y + 4 x y^{I} + x^2 y^{II}$; $Y_0^{III} = 2$

$y^{IIII} = 6 y^{I} + 6x y^{II} + x^2 y^{III}$; $Y_0^{IIII} = -6$

ORDINARY DIFFERENTIAL EQUATIONS

$Y_0^{|} = -1$; $Y_0^{||} = 0$; $Y_0^{|||} = 2$; $Y_0^{||||} = -6$ and $h = x - x_0 = 0.1$

$$\mathbf{y_1 = y_0 + h}\, Y_0^{|} + \frac{\mathbf{h^2}}{2!} Y_0^{||} + \frac{\mathbf{h^3}}{2!} Y_0^{|||} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$Y(0.1) = 0.90031$

ORDINARY DIFFERENTIAL EQUATIONS

Again taking (x_1, y_1) in place of (x_0, y_0) and repeat the process.

$Y(0.1) = 0.90031$ then $x_1 = 0.1$ and $y_1 = 0.90031$

$y^{I} = x^2 y - 1$; $Y_1^{I} = x_1^2 y_1 - 1 = -0.991$

$y^{II} = 2xy + x^2 y^{I}$; $Y_1^{II} = 2x_1 y_1 + x_1^2 Y_1^{I} = 0.17015$

$y^{III} = 2y + 4 x y^{I} + x^2 y^{II}$; $Y_1^{III} = 1.40592$

$y^{IIII} = 6 y^{I} + 6x y^{II} + x^2 y^{III}$; $Y_1^{IIII} = -5.82983$

ORDINARY DIFFERENTIAL EQUATIONS

$$\mathbf{y_2 = y_1 + h}\ Y_1^{\ I} + \frac{\mathbf{h^2}}{2!} Y_1^{\ II} + \frac{\mathbf{h^3}}{2!} Y_1^{\ III} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$Y(x_2) = y(0.2) = 0.80227$

ORDINARY DIFFERENTIAL EQUATIONS

EULER'S METHOD:

$$\frac{dy}{dx} = f\,(\,x, y)$$

$Y_1 = y_0 + h\, f(\, x_0\,, y_0)$

$y_{\,2} = y_1 + h\, f(\, x_1\,, y_1\,)$

. . .

. . .

. . .

$Y_n = y_{n-1} + h\, f\,(\, x_{n-1}\,, y_{n-1}\,)$

Euler's method is a straightforward numerical technique for approximating solutions to first-order differential equations.

ORDINARY DIFFERENTIAL EQUATIONS

Example: If dy /dx = x+ y; y (0) = 1 then find y (0.3) by taking step size as 0.1 using Euler's method.

Steps for Euler's Method

1. **Initial Condition:**

   $x_0 = 0, \quad y_0 = 1$

2. **Step Size:**

   $h = 0.1$

3. **Iterations:**

   - **First step (from $x_0 = 0$ to $x_1 = 0.1$):**

$$y_1 = y_0 + h \cdot f(x_0, y_0)$$

$$y_1 = 1 + 0.1 \cdot (0 + 1) = 1 + 0.1 = 1.1$$

ORDINARY DIFFERENTIAL EQUATIONS

- Second step (from $x_1 = 0.1$ to $x_2 = 0.2$):

$$y_2 = y_1 + h \cdot f(x_1, y_1)$$

$$y_2 = 1.1 + 0.1 \cdot (0.1 + 1.1) = 1.1 + 0.1 \cdot 1.2 = 1.1 + 0.12 = 1.22$$

- Third step (from $x_2 = 0.2$ to $x_3 = 0.3$):

$$y_3 = y_2 + h \cdot f(x_2, y_2)$$

$$y_3 = 1.22 + 0.1 \cdot (0.2 + 1.22) = 1.22 + 0.1 \cdot 1.42 = 1.22 + 0.142 = 1.362$$

ORDINARY DIFFERENTIAL EQUATIONS

RUNGE – KUTTA'S SECOND ORDER METHOD:

$$\frac{dy}{dx} = f\,(x, y)$$

$$k_1 = f(x_n, y_n)$$

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2)$$

$$k_2 = f(x_n + h, y_n + hk_1)$$

For n=0

$Y_1 = Y_0 + \frac{h}{2}(K_1 + K_2)$

Where $K_1 = f(x_0, y_0)$ and $K_2 = f((x_0 + h), y_0 + hK_1)$

ORDINARY DIFFERENTIAL EQUATIONS

Q: Solve the given differential equation using the Runge-Kutta second order method (Heun's method) for x=0.2 and x=0.4 for the differential equation given by

$$y' = \frac{y^2 - x^2}{y^2 + x^2}, \quad y(0) = 1$$

ORDINARY DIFFERENTIAL EQUATIONS

The formula for Heun's method (a specific case of the Runge-Kutta second-order method) is giv
by:

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2)$$

where

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + h, y_n + hk_1)$$

ORDINARY DIFFERENTIAL EQUATIONS

Step-by-Step Calculation

Initial Conditions

$$x_0 = 0, \quad y_0 = 1$$

Step size: $h = 0.2$

To find $y(0.2)$

1. **Step 1: Compute k_1**

$$k_1 = f(x_0, y_0) = \frac{y_0^2 - x_0^2}{y_0^2 + x_0^2} = \frac{1^2 - 0^2}{1^2 + 0^2} = 1$$

ORDINARY DIFFERENTIAL EQUATIONS

2. **Step 2: Compute k_2**

$$k_2 = f(x_0 + h, y_0 + hk_1) = f(0 + 0.2, 1 + 0.2 \cdot 1) = f(0.2, 1.2)$$

$$k_2 = \frac{1.2^2 - 0.2^2}{1.2^2 + 0.2^2} = \frac{1.44 - 0.04}{1.44 + 0.04} = \frac{1.4}{1.48} \approx 0.946$$

3. **Step 3: Compute y_1**

$$y_1 = y_0 + \frac{h}{2}(k_1 + k_2) = 1 + \frac{0.2}{2}(1 + 0.946) = 1 + 0.1 \cdot 1.946 = 1.1946$$

So, $y(0.2) \approx 1.1946$.

ORDINARY DIFFERENTIAL EQUATIONS

To find $y(0.4)$

Using the result from $y(0.2)$ as the new initial value:

1. **Step 1: Compute k_1**

$$k_1 = f(0.2, 1.1946) = \frac{1.1946^2 - 0.2^2}{1.1946^2 + 0.2^2} = \frac{1.427 - 0.04}{1.427 + 0.04} \approx 0.968$$

2. **Step 2: Compute k_2**

$$k_2 = f(0.2 + 0.2, 1.1946 + 0.2 \cdot 0.968) = f(0.4, 1.1946 + 0.1936) = f(0.4, 1.3882)$$

$$k_2 = \frac{1.3882^2 - 0.4^2}{1.3882^2 + 0.4^2} = \frac{1.927 - 0.16}{1.927 + 0.16} \approx 0.927$$

3. **Step 3: Compute y_2**

$$y_2 = 1.1946 + \frac{0.2}{2}(0.968 + 0.927) = 1.1946 + 0.1 \cdot 1.895 \approx 1.3841$$

So, $y(0.4) \approx 1.3841$.

ORDINARY DIFFERENTIAL EQUATIONS

Example : **If $dy/dx + 2xy^2 = 0$ with $y(0) = 1$ then find $y(0.2)$ using Runge-Kutta's second order method.**

Solution:

Given $y(0) = 1$ then $x_0 = 0$ and $y_0 = 1$ **$f(x, y) = -2xy^2$**

$h = x_1 - x_0 = 0.2 - 0 = 0.2$

$$Y_1 = Y_0 + \frac{h}{2}(K_1 + K_2)$$

$K_1 = f(x_0, y_0) = f(0, 1) = 0$

$K_2 = f(x_0 + h), y_0 + hK_1) = f(0.2, 1) = -0.4$

$$Y_1 = Y_0 + \frac{h}{2}(K_1 + K_2)$$

$Y_1 = 1 - 0.04 = 0.96$

ORDINARY DIFFERENTIAL EQUATIONS

Solve the initial value problem defined by

$$\frac{dy}{dx}=\frac{3x+y}{x+2y}, \quad y(1)=1$$

And find y(1.2) and y(1.4) by Runge-Kutta fourth order formula.

ORDINARY DIFFERENTIAL EQUATIONS

The Runge-Kutta fourth-order method is given by:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

ORDINARY DIFFERENTIAL EQUATIONS

1. **Initialization:**

$$x_0 = 1,\ y_0 = 1,\ h = 0.1$$

2. **First Step:** $x_0 = 1$ to $x_1 = 1.1$:

$$k_1 = hf(x_0, y_0) = 0.1 \cdot \frac{3 \cdot 1 + 1}{1 + 2 \cdot 1} = 0.1 \cdot \frac{3+1}{1+2} = 0.1 \cdot \frac{4}{3} = \frac{0.4}{3} = 0.1333$$

$$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) = 0.1 \cdot f\,(1 + 0.05, 1 + 0.06665) = 0.1 \cdot f\,(1.05, 1.0666$$

$$k_2 = 0.1 \cdot \frac{3 \cdot 1.05 + 1.06665}{1.05 + 2 \cdot 1.06665} = 0.1 \cdot \frac{3.15 + 1.06665}{1.05 + 2.1333} = 0.1 \cdot \frac{4.21665}{3.1833} \approx 0.1325$$

ORDINARY DIFFERENTIAL EQUATIONS

$$k_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) = 0.1 \cdot f\,(1.05, 1.06625)$$

$$k_3 = 0.1 \cdot \frac{3 \cdot 1.05 + 1.06625}{1.05 + 2 \cdot 1.06625} = 0.1 \cdot \frac{3.15 + 1.06625}{1.05 + 2.1325} = 0.1 \cdot \frac{4.21625}{3.1825} \approx 0.1325$$

$$k_4 = hf(x_0 + h, y_0 + k_3) = 0.1 \cdot f(1.1, 1.1325)$$

$$k_4 = 0.1 \cdot \frac{3 \cdot 1.1 + 1.1325}{1.1 + 2 \cdot 1.1325} = 0.1 \cdot \frac{3.3 + 1.1325}{1.1 + 2.265} = 0.1 \cdot \frac{4.4325}{3.365} \approx 0.1317$$

ORDINARY DIFFERENTIAL EQUATIONS

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1 + \frac{1}{6}(0.1333 + 2 \cdot 0.1325 + 2 \cdot 0.1325 + 0.131$$

$$y_1 = 1 + \frac{1}{6}(0.1333 + 0.265 + 0.265 + 0.1317) = 1 + \frac{1}{6}(0.795) = 1 + 0.1325 = 1.13$$

3. **Second Step:** $x_1 = 1.1$ **to** $x_2 = 1.2$:

$$k_1 = hf(x_1, y_1) = 0.1 \cdot \frac{3 \cdot 1.1 + 1.1325}{1.1 + 2 \cdot 1.1325} = 0.1 \cdot \frac{3.3 + 1.1325}{1.1 + 2.265} = 0.1 \cdot \frac{4.4325}{3.365} = 0.13$$

$$k_2 = hf\left(x_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right) = 0.1 \cdot f\left(1.15, 1.19835\right)$$

ORDINARY DIFFERENTIAL EQUATIONS

$$k_3 = hf\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right) = 0.1 \cdot f\,(1.15, 1.19805)$$

$$k_3 = 0.1 \cdot \frac{3 \cdot 1.15 + 1.19805}{1.15 + 2 \cdot 1.19805} = 0.1 \cdot \frac{3.45 + 1.19805}{1.15 + 2.3961} = 0.1 \cdot \frac{4.64805}{3.5461} \approx 0.1310$$

$$k_4 = hf(x_1 + h, y_1 + k_3) = 0.1 \cdot f(1.2, 1.2635)$$

$$k_4 = 0.1 \cdot \frac{3 \cdot 1.2 + 1.2635}{1.2 + 2 \cdot 1.2635} = 0.1 \cdot \frac{3.6 + 1.2635}{1.2 + 2.527} = 0.1 \cdot \frac{4.8635}{3.727} \approx 0.1305$$

ORDINARY DIFFERENTIAL EQUATIONS

$$y_2 = y_1 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.1325 + \frac{1}{6}(0.1317 + 2 \cdot 0.1311 + 2 \cdot 0.1310 +$$

$$y_2 = 1.1325 + \frac{1}{6}(0.1317 + 0.2622 + 0.262 + 0.1305) = 1.1325 + \frac{1}{6}(0.7864) = 1.132$$

Repeating these steps, we find the values of $y(1.2)$ and $y(1.4)$:

- $y(1.2) \approx 1.1325$
- (y(1.4) \approx 1.

ORDINARY DIFFERENTIAL EQUATIONS

Use the trapezoidal rule to evaluate the double integral

$$\int_{-2}^{2}\int_{0}^{4}(x^2 - xy + y^2)\,dx\,dy.$$

ORDINARY DIFFERENTIAL EQUATIONS

To evaluate the double integral of $f(x, y) = x^2 - xy + y^2$ over the region $D = [-2, 2] \times [0, 4]$ using the trapezoidal rule, we need to apply the trapezoidal rule in both dimensions. Here's a step-by-step guide:

Steps for the Trapezoidal Rule

1. **Define the Integration Region:**

   - $x \in [-2, 2]$
   - $y \in [0, 4]$

2. **Discretize the Integration Region:**

   - Choose n subintervals in the x-direction and m subintervals in the y-direction.
   - Let's choose $n = 2$ and $m = 2$ for simplicity.

3. **Compute the Step Sizes:**

   - $\Delta x = \frac{b-a}{n} = \frac{2-(-2)}{2} = 2$
   - $\Delta y = \frac{d-c}{m} = \frac{4-0}{2} = 2$

4. **Grid Points and Weights:**

   - x values: $x_0 = -2$, $x_1 = 0$, $x_2 = 2$
   - y values: $y_0 = 0$, $y_1 = 2$, $y_2 = 4$

5. **Evaluate the Function at Grid Points and Apply the Trapezoidal Rule:**

$$\iint_D f(x,y)\,dx\,dy \approx \frac{\Delta x \Delta y}{4}\left[\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij} f(x_i, y_j)\right]$$

where w_{ij} are the weights for the trapezoidal rule:

- $w_{ij} = 1$ for interior points,
- $w_{ij} = \frac{1}{2}$ for points on the boundary,
- $w_{ij} = \frac{1}{4}$ for corner points.

ORDINARY DIFFERENTIAL EQUATIONS

Evaluate the Function at Grid Points

$$
\begin{matrix}
f(-2,0) & f(-2,2) & f(-2,4) \\
f(0,0) & f(0,2) & f(0,4) \\
f(2,0) & f(2,2) & f(2,4)
\end{matrix}
$$

$$
\begin{matrix}
4 & 12 & 2\cancel{4}\,8 \\
0 & 4 & 16 \\
4 & \cancel{0} & 12 \\
 & 4 &
\end{matrix}
$$

Apply the Trapezoidal Rule

1. **Compute the Weights and Function Values:**

$f(x,y)$	(x,y)	w_{ij}	$w_{ij} \cdot f(x,y)$
4	$(-2,0)$	$\frac{1}{4}$	1
12	$(-2,2)$	$\frac{1}{2}$	6
$2\cancel{4}$ 8	$(-2,4)$	$\frac{1}{4}$	$\cancel{6}$ 7
0	$(0,0)$	$\frac{1}{2}$	0
4	$(0,2)$	1	4
16	$(0,4)$	$\frac{1}{2}$	8
4	$(2,0)$	$\frac{1}{4}$	1
$\cancel{0}$ 4	$(2,2)$	$\frac{1}{2}$	$\cancel{0}$ 2
12	$(2,4)$	$\frac{1}{4}$	3

2. **Sum the Contributions:**

$$\sum w_{ij} \cdot f(x,y) = 1 + 6 + 6 + 0 + 4 + 8 + 1 + 0 + 3 = \cancel{29}\ 32$$

3. **Multiply by the Factor $\frac{\Delta x \Delta y}{4}$:**

$$\iint_D f(x,y)\,dx\,dy \approx \frac{2 \cdot 2}{4} \cdot 29 = 1 \cdot 29 = 2\cancel{9}\ 32$$

So, the approximate value of the double integ using the trapezoidal rule is 29.

ORDINARY DIFFERENTIAL EQUATIONS

Given the differential equation

$y'' - xy' - y = 0$

with the conditions $y(0) = 1$ and $y'(0) = 0$, use Taylor's series method to determine the value of $y(0.1)$.

ORDINARY DIFFERENTIAL EQUATIONS

Step 1: Compute the Initial Derivatives

1. **Given initial conditions:**

   $y(0) = 1$

   $y'(0) = 0$

2. **Second derivative $y''(0)$:**

   The differential equation is $y'' - xy' - y = 0$.

   At $x = 0$, this simplifies to:

   $y''(0) - 0 \cdot y'(0) - y(0) = 0$

   $y''(0) - 1 = 0$

   $y''(0) = 1$

3. **Third derivative $y'''(0)$:**

   Differentiate the differential equation:

   $y''' - (xy')' - y' = 0$

   $y''' - (xy'' + y') - y' = 0$

   $y''' - xy'' - 2y' = 0$

   At $x = 0$:

   $y'''(0) - 0 \cdot y''(0) - 2y'(0) = 0$

   $y'''(0) = 0$

4. **Fourth derivative $y''''(0)$:**

   Differentiate the equation again:

   $y'''' - (xy'')' - 2y'' = 0$

   $y'''' - (xy''' + y'') - 2y'' = 0$

   $y'''' - xy''' - 3y'' = 0$

   At $x = 0$:

   $y''''(0) - 0 \cdot y'''(0) - 3y''(0) = 0$

   $y''''(0) = 3$

ATIONS

Step 2: Construct the Taylor Series

Using the derivatives calculated, we can write the Taylor series up to the fourth term (which g sufficient accuracy for small x):

$$y(x) = y(0) + y'(0)x + \frac{y''(0)}{2!}x^2 + \frac{y'''(0)}{3!}x^3 + \frac{y''''(0)}{4!}x^4 + \cdots$$

Substituting the values we found:

$$y(x) = 1 + 0 \cdot x + \frac{1}{2!}x^2 + \frac{0}{3!}x^3 + \frac{3}{4!}x^4$$
$$y(x) = 1 + \frac{x^2}{2} + \frac{3x^4}{24}$$
$$y(x) = 1 + \frac{x^2}{2} + \frac{x^4}{8}$$

Step 3: Calculate $y(0.1)$

Substitute $x = 0.1$ into the Taylor series:

$$y(0.1) = 1 + \frac{(0.1)^2}{2} + \frac{(0.1)^4}{8}$$
$$y(0.1) = 1 + \frac{0.01}{2} + \frac{0.0001}{8}$$
$$y(0.1) = 1 + 0.005 + 0.0000125$$
$$y(0.1) = 1.0050125$$

Therefore, the value of $y(0.1)$ using Taylor's s…s method is approximately 1.0050125.