Programs Offered

Post Graduate Programmes (PG)

- Master of Business Administration
- Master of Computer Applications
- Master of Commerce (Financial Management / Financial Technology)
- Master of Arts (Journalism and Mass Communication)
- Master of Arts (Economics)
- Master of Arts (Public Policy and Governance)
- Master of Social Work
- Master of Arts (English)
- Master of Science (Information Technology) (ODL)
- Master of Science (Environmental Science) (ODL)

Diploma Programmes

- Post Graduate Diploma (Management)
- Post Graduate Diploma (Logistics)
- Post Graduate Diploma (Machine Learning and Artificial Intelligence)
- Post Graduate Diploma (Data Science)

Undergraduate Programmes (UG)

- Bachelor of Business Administration
- Bachelor of Computer Applications
- Bachelor of Commerce
- Bachelor of Arts (Journalism and Mass Communication)
- Bachelor of Arts (General / Political Science / Economics / English / Sociology)
- Bachelor of Social Work
- Bachelor of Science (Information Technology) (ODL)

AMITY UNIVERSITY ONLINE
Amity Helpline: (Toll free) 18001023434
For Student Support: +91 - 8826334455
Support Email id: studentsupport@amityonline.com | https://amityonline.com

Product code
81030112000593

Graph Theory and Combinatorics

AMITY

Graph Theory and Combinatorics

AMITY UNIVERSITY ONLINE

Graph Theory and Combinatorics

AMITY UNIVERSITY ONLINE

Published by Amity University Press for exclusive use of Amity Directorate of Distance and Online Education, Amity University, Noida-201313

Contents

Module -I: Introduction to Graph Theory

Notes

Learning Objectives:

At the end of this module, you will be able to:

- Aware About the History of Graphs
- Learn Basic Terminologies of Graph Theory
- Describe Graph and Its Types with Examples
- Discuss Network and Weighted Graphs
- Infer Basic Terminologies Like Walk, Path, Cycle, etc.
- Learn About the Types of Connectedness in Graphs
- Analyse Handshaking Lemma Theorem
- Understand the Isomorphism of Graphs
- Summarise the Subgraphs and Induced Graphs
- Explore Planar Graphs and Spanning Graphs
- Learn Basic Operations on Graph: Union and Intersection

Introduction

Graph theory is a significant branch of mathematics that is employed in the fields of computer science and engineering. It addresses the analysis of graphs and the relationship between edges and vertices. This subject is of great importance and has a wide range of applications in the fields of information technology, computer science, and biomathematics. Furthermore, it is fully utilised by the circuit connections. However, it also facilitates the examination of algorithms such as Dijkstra's, Prim's, and Kruskal's algorithms. The primary relationship between graph theory and topology is employed in a variety of topologies, including network, star, bridge, series, and parallel topologies, among others. The interconnection of the devices in the network is determined by the interaction between the graph theory concepts. A graph can be classified based on its structural characteristics.

Connectedness in graphs refers to the property of a graph wherein there is a path between every pair of vertices. In a connected graph, all vertices are reachable from any other vertex. If a graph is not connected, it consists of multiple disconnected subgraphs, known as components.

Graph isomorphism is the concept of mapping one graph onto another in such a way that there is a one-to-one correspondence between their vertex sets and edge sets.

A subgraph is a graph formed from a subset of the vertices and edges of a larger graph. An induced subgraph is a subgraph formed by a subset of vertices of the original graph along with all the edges connecting pairs of vertices in this subset. Induced subgraphs maintain the edge relationships of the original graph.

Planar graphs are graphs that can be drawn on a plane without any edges crossing each other. A spanning graph, or spanning tree (in the context of connected graphs), is a subgraph that includes all the vertices of the original graph and is a tree (i.e., it is connected and acyclic). A spanning graph spans all vertices of the original graph, ensuring that there is a path between any two vertices.

Notes

1.1 Fundamentals of Graph Theory

Graphs are numerical designs that address pairwise connections between objects. A chart is a stream structure that addresses the connection between different items. It tends to be pictured by utilising the accompanying two essential segments: Nodes and Edges. The vertices, or nodes, of the graph are joined by the edges, or lines. In addition to mathematics, other disciplines like computer science, physics and chemistry, linguistics, biology, etc. also employ the linear graph. GPS, which allows you to monitor a path or determine the direction of a road, is also the best illustration of a graph structure in real life.

1.1.1 History of Graphs

According to its history, the well-known Swiss mathematician Leonhard Euler developed graph theory in order to create graphs from provided data or a group of points in order to solve a variety of mathematical issues. The graphical representation uses bar graphs, frequency tables, line graphs, circle graphs, line plots and other visual aids to display various data types.

A graphic representation of any data that is arranged is called a graph in mathematics. The graph illustrates how different quantities relate to one another. According to graph theory, the collection of things that are connected to one another in some way is represented by the graph. The vertices or nodes represent the objects, which are essentially mathematical notions, and the edges represent the relationship between a pair of nodes.

1.1.2 Basic Terminology in Graph Theory

A graph consists of:

- A set, V, of **vertices** (nodes)
- A collection, E, of pairs of vertices from V called edges (arcs)
- **Edges**, also called arcs, are represented by (u, v) and are either:
  - ❖ **Directed** if the pairs are ordered (u, v)
- Where u the origin
- v the destination
  - ❖ **Undirected** if the pairs are unordered

Some basic terminologies are-

1. **Degree**- The degree of a node is its total number of connected edges. Notch One common way to write i’s degree is deg(i).
2. **Walk**- A collection of edges connected in a certain order to provide a continuous path on a network.
3. **Trail**- Route a walk without passing through any edges more than once.
4. **Path-** Way a stroll through which no node (and hence, no edge) is encountered more than once.
5. **Cycle-** A walk with a single starting point and destination point, passing by each node exactly once.
6. **Subgraph-** Part of the graph.

7. **Connected Graph-** A graph where there is a path connecting every two nodes.
8. **Connected Component-** A subgraph of a graph that has connections within it but not with other graph elements.

Notes

1.1.3 Graphs and Examples

Definition:

A Graph is said to be a pair of vertices and edges which consisting a set of objects (V, E) where V={v_1 , v_2, v_3,…., v_n} are called **vertices** and E={e_1, e_2, e_3,….,e_n }are called **edges** . The vertex v_1 is called the **initial vertex** and the vertex v_n is called the **terminal vertex**.

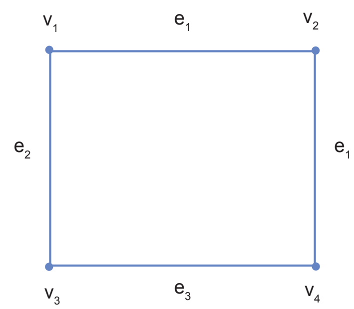


Fig (1.1)

In the above figure (1.1), there are four vertices and four edges namely V={v_1, v_2, v_3, v_4} and E={ e_1, e_2, e_3, e_4}

Simple Graph:

A graph is considered simple if it has parallel edges and no self-loop. A simple graph is, equivalently, one that lacks both parallel edges and a self-loop. Take a look at this basic graph example.

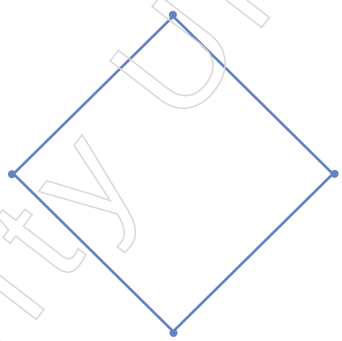


Fig (1.2)

The graph in Fig. (1.2) above demonstrates that it is a simple graph with parallel edges and no self-loop.

Parallel edges: They are any two edges that share the same end vertex as well as the same initial vertex.

Self-Loop: An edge is referred to as a self-loop if its vertex matches both the beginning and end vertices.

The graph with parallel edges and a self-loop is shown in the example below.

Notes

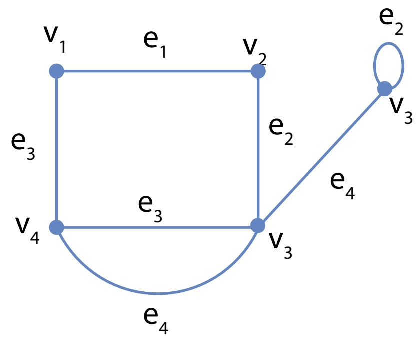


Fig (1.3)

There are five vertices and seven edges in the graph shown in Figure (1.3).

1. The two edges in this case, 3 and 4, are parallel to one another and share the same beginning vertex (c) and ending vertex (d).
2. An edge 7 is considered to be a self-loop if it has the identical initial and end vertices, 'e'.

Finite Graph: A graph is referred to as finite if it has a finite number of vertices; otherwise, it is referred to as endless.

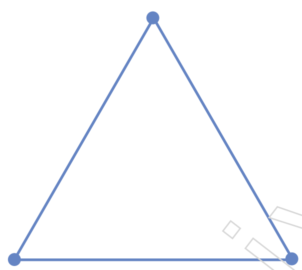


Fig (1.4a)

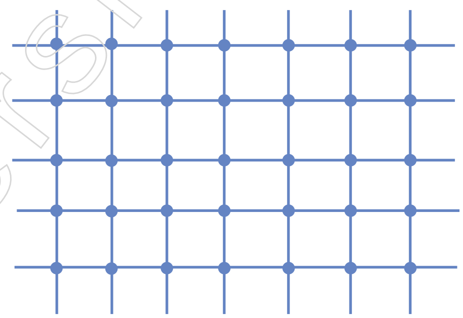


Fig (1.4b)

An example of a finite graph is shown in figure 1.4a above, which has three vertices and three edges. An example of an infinite graph is shown in figure 1.4b, which has an unlimited number of vertices.

Definition:

If there is an edge connecting the two vertices, then they are said to be neighbouring. One of the two vertices that joins the edges if a vertex is incident to an edge is that vertex. The vertices "a" and "b" as well as "d" are adjacent to each other in the above picture (1.1).

Degree: The number of edges linking a vertex is said to be its degree. Take a look at the sample below.

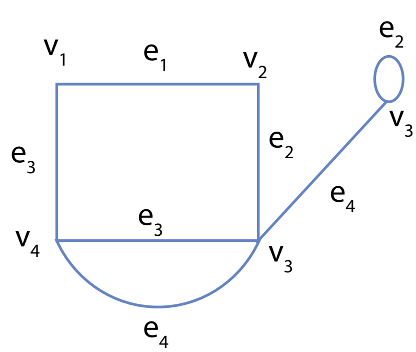


Fig (1.3)

d(v1) = 2

Notes

d(v2) = 2

d(v3) = 4

d(v4) = 3

d(v5) = 3

Note:

❖ It is necessary to count the edges' degrees twice in the self loop.

❖ The number of edges in the graph needs to equal the sum of the vertices' degrees twice over.

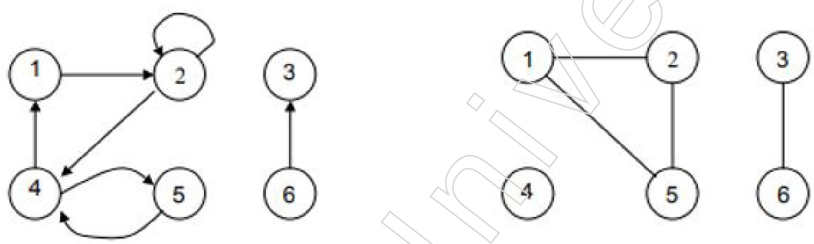
1.1.4 Network and Weighted Graphs

A network: what is it?

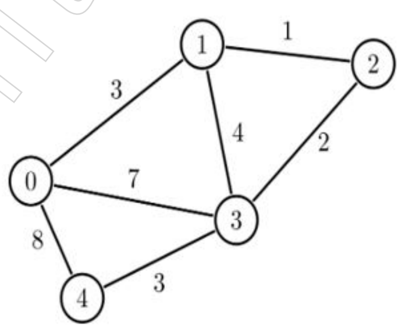
- Informally, a graph is a collection of nodes connected by a network of lines or arrows.
- Network = graph.

What is the theory of networks?

- A collection of methods for analysing graphs is provided by network theory.
- Complex systems network theory offers methods for dissecting the organisation of a network, which is a system of interacting agents.
- When utilising a graph-theoretic representation, network theory is applied to a system.



Weighted Graph: A graph that has assigned a numerical weight to each branch is called a weighted graph. Thus, a weighted graph is a particular kind of labelled graph where the labels are numbers, which are typically interpreted as positive. A weighted graph is a unique kind of graph where the edges are given weights that correspond to various relative measurement units, such as cost and distance. Given below is an example of a weighted graph:



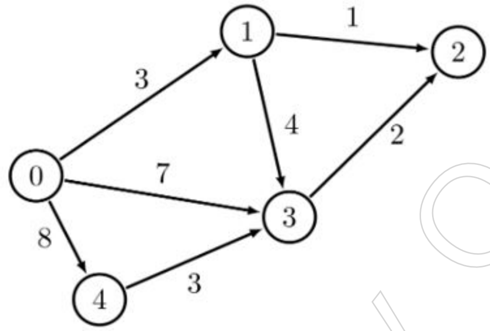
There are five nodes and seven edges (each with a weight) in the graph. For instance, the weight of the edge {0,3} is 7 and the weight of the edge {1,2} is 1.

Real-world items and node relationships are frequently modeled using weighted graphs. The aforementioned illustration can be viewed as a section of a Google map, with

Notes

roads serving as the boundaries and cities as the nodes. The time/distance between two cities determines the weight of each edge.

A weighted undirected graph with undirected edges is shown in the example above. Weighted graphs, however, can also be directed. Given below is an example of a directed weighted graph:



Certain actual processes can also be modeled using weighted directed graphs. As an illustration, node 0 in the preceding scenario can be thought of as a storage location from which we must move certain resources to the destination, let's say node 2. The remaining nodes can be thought of as intermediary locations. The directions in which the resources can be moved are indicated by each edge. The greatest quantity of resources that can be delivered through an edge is indicated by its weight.

Keep in mind that while all of the weights in the example above are positive, zero and negative weights are also permitted.

A representation of weighted graphs

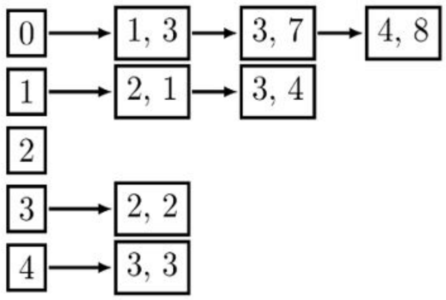
One can use either an adjacency list or an adjacency matrix to represent an unweighted graph. Both of these structures can also be used to represent a weighted graph.

An adjacency matrix for a weighted network with n nodes is a n×n matrix A, where A[i , j] is the edge's weight between two neighboring nodes, i and j. Should there be no boundary between i along with j and the matching value is set to ∞. Assuming that there are no self-loops in a graph, the diagonal members of the matrix are assigned to 0 0. The following is an adjacency matrix for the directed one mentioned above:

	0	1	2	3	4
0	0	3	∞	7	8
1	∞	0	1	4	∞
2	∞	∞	0	∞	∞
3	∞	∞	2	0	∞
4	∞	∞	∞	3	0

The nodes of the graph are represented by the first row and column of the matrix. The weights are represented by the remaining cells. For example, A[1,3]=4 since there is an edge from node 1 to node 3 with the weight 4. A[3,1]=∞ since there is no edge from node 3 to node 1.

The weighted and unweighted graphs' adjacent lists are comparable. The only distinction is that, in order to represent the weight of an edge, we must store an extra value in each list node. An adjacency list for the directed graph above looks like this:



A pair of values is present in every node in the list. The weight of the associated edge is represented by the second number, which is the label of a node in the graph. For instance, the edge {0,1} is represented by the first node in the first list that has weight 3. The second node in the same list corresponds to the edge {0,3} which has weight 7.

1.1.5 Walks, Paths and Circuits in Graphs

1. Walk: An alternating series of vertices, edges and incident elements that begins and finishes at the same vertex is known as a walk in a graph.

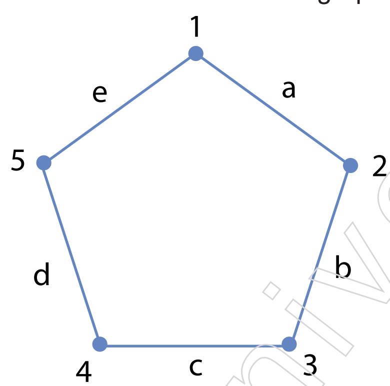


Fig (1.6)

In the above fig (1.6) 1a2b3c4d5e1 is a walk.

- ❖ A walk with no edges repeated is called a path.
- ❖ A path that doesn't repeat any vertices is called a walk. In the above Fig (1.6) 1a2b3c4d5e is a path.
- ❖ If the initial and terminal vertices of a path differ, the walk is said to be open.
- ❖ If the initial and terminal vertices of a walk are the same, the walk is considered closed.

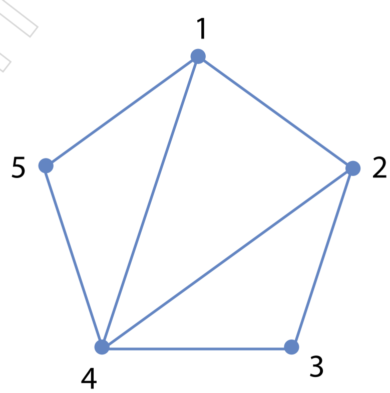


Fig (1.7)

Notes

- ❖ In the above figure (1.7) 1 → 2 → 3 → 4 → 1 → 5 is an open walk.
- ❖ In the above figure (1.6) 1a2b3c4d5e1 is a closed walk

2. Path: The term "path" in a graph refers to a sequence of unique vertices joined by a finite or infinite sequence of edges.

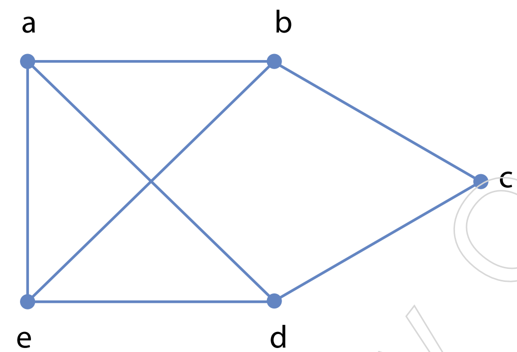


Fig (1.8)

The above figure (1.8) shows that a, b, c, d, e is a path.

3. Circuit: A graph is defined as a circuit that is closed and has no duplicate edges.

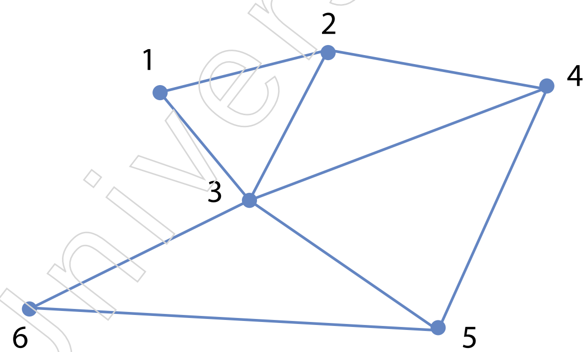


Fig (1.9)

In the above figure (1.9) 1 → 2 → 4 → 3 → 6 → 5 → 3 → 1 is a circuit.

- ❖ In a circuit, an edge cannot be repeated, but a vertex may.
- ❖ Neither the edge nor the vertex is repeated in a cycle.

1.1.6 Path and Cycle in Graphs

Path: It is a trail in which no vertex or edge is duplicated; that is, if we navigate a graph without encountering any repetitions of either. It is an open walk since the path acts as a trail.

A walk without a repeating vertex is another way to define a path. It is unnecessary to put this in the path definition because it clearly suggests that there won't ever be any duplicate edges.

- ❖ No repeating vertex
- ❖ Not repeating the edge

Notes

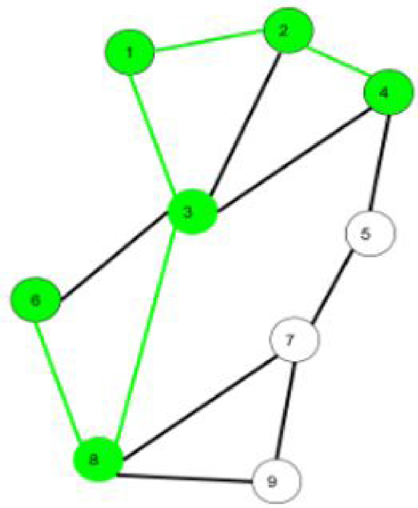


Figure: path

Here 6->8->3->1->2->4 is a Path

Cycle:

By traversing a graph, we may ensure that neither a vertex nor an edge are repeated, but rather that the starting and ending vertices remain the same. In other words, we can repeat the starting and ending vertices until we reach a cycle.

- ❖ No repeating vertex
- ❖ Not repeating the edge

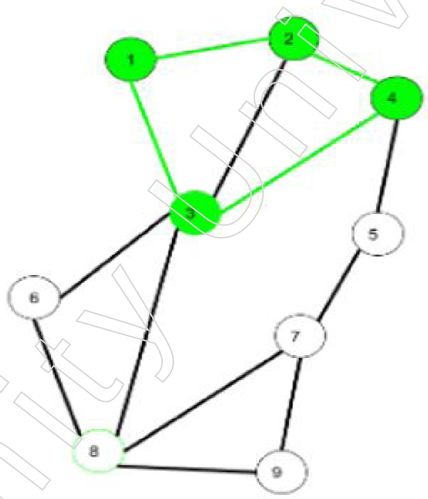


Figure: Cycle

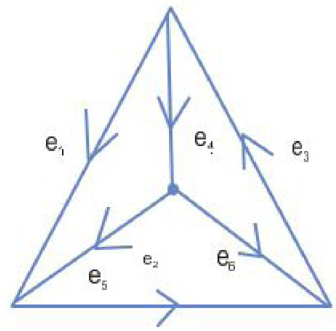
Here is a cycle: 1->2->4->3->1.

- ❖ Cycle is a closed path.
- ❖ Nothing about these can be repeated, not even the vertices or edges.

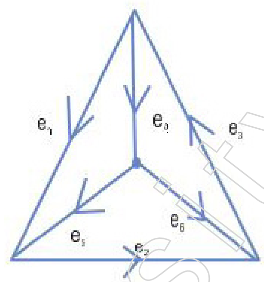
1.1.7 Connectedness in Graphs

1. **Strongly connected graph:** If there is at least one directed path connecting each vertex in the graph to every other vertex, the graph is considered strongly linked.

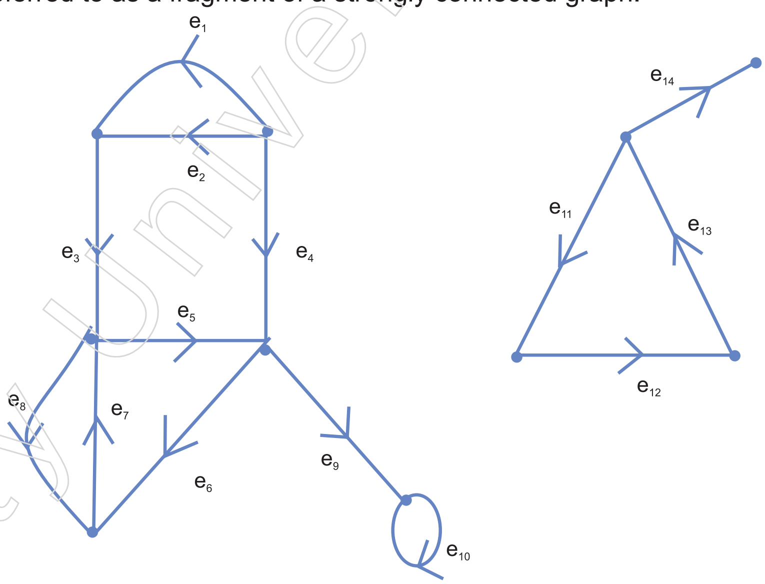
Notes



2. **Weakly connected graph**: If an undirected graph that corresponds to it is linked, then the graph is said to be weakly connected.



❖ When a maximally strongly linked subgraph exists within each component of G, it is referred to as a fragment of a strongly connected graph.

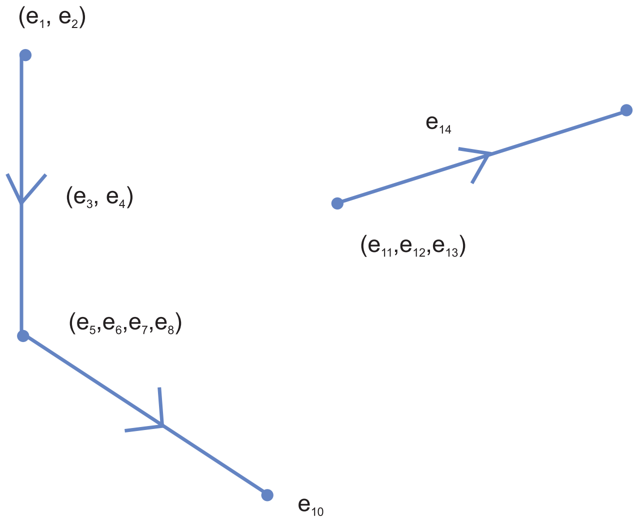


❖ G, the condensation of a digraph G is a digraph where all directed edges from one strongly connected component to another are replaced by a single directed edge and every strongly linked fragment is replaced by a vertex.

A strongly connected graph's condensation is just a vertex.

❖ There is no directed circuit in a digraph's condensation.

Notes



1.1.8 Handshaking Lemma

The total number of vertex degrees in any graph G(V,E) equals twice the total number of edges.

For example

$$\sum_{i=1}^{n} d(V_i) = 2e$$

Proof:

Let us examine the graph G, which has edges labeled "E" and vertices labeled "V." We can now provide a precise example to support this theorem. In light of this, we examine the graph in Fig. (1.3) above.

There are five vertices and seven edges in the graph.

$$d(v_1) = 2, d(v_2) = 2, d(v_3) = 4, d(v_4) = 3, d(v_5) = 3$$

$$d(v_1) + d(v_2) + d(v_3) + d(v_4) + d(v_5) = 2 + 2 + 4 + 3 + 3$$

=> 14= twice the number of edges

Therefore

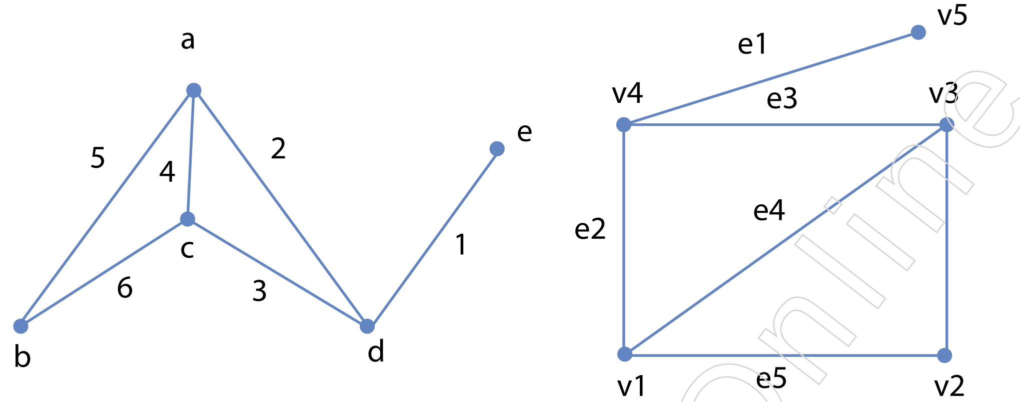
$$\sum_{i=1}^{n} d(V_i) = 2e$$

1.1.9 Isomorphism of Graphs

One way to define an isomorphism graph is as a graph where a single graph can exist in several forms. This implies that the number of edges, vertices and edge connectivity can all be the same in two distinct graphs. Isomorphism graphs are the name given to these kinds of graphs.

An example of an isomorphism between two graphs is when the incidence relationship between the vertices and edges of the graphs G and G' is retained and there is a one-to-one correspondence between them.

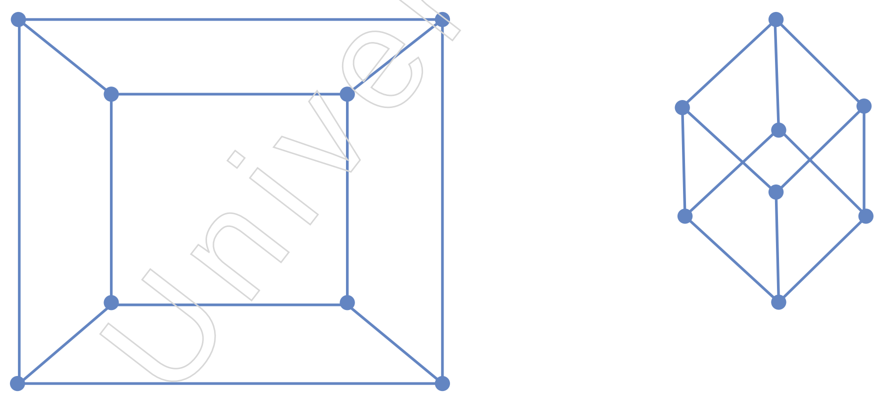
Notes



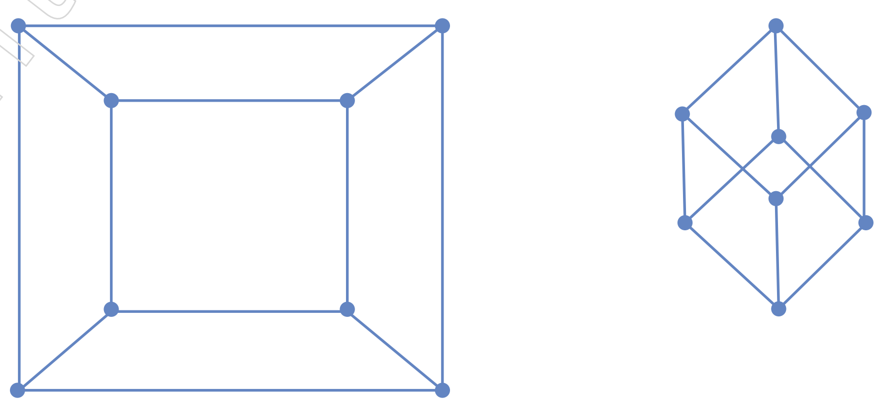
- The two graphs mentioned above are isomorphic.
- The vertices a, b, c, d and e in the graph above correspond to v_1,v_2,v_3,v_4 and v_5 respectively. The edges 1,2,3,4,5 and 6 correspond to e_1,e_2,e_3,e_4 and e_5 respectively. Therefore, G And G' are isomorphism.

Note:

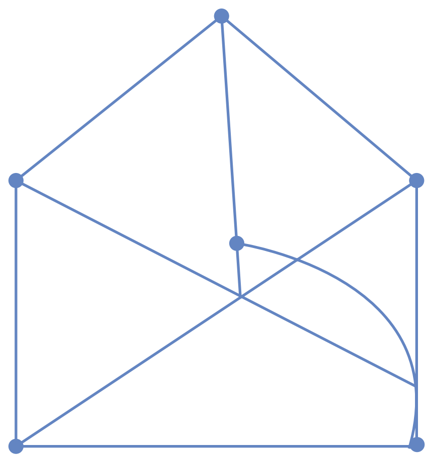
In the absence of vertex and edge labels, isomorphic graphs are identical graphs that may be depicted in a different way.



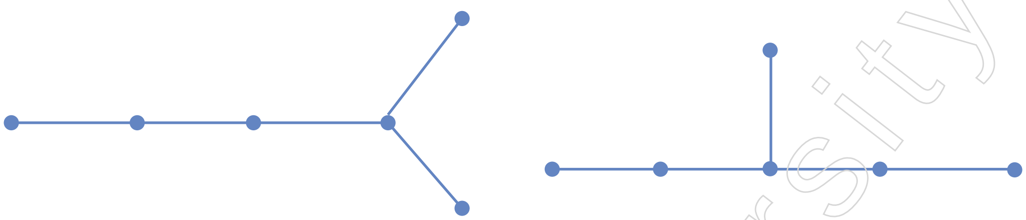
1. Both the number of vertices and edges in the isomorphic graphs must be equal.
2. The same number of vertices at a specific degree.



Notes



Isomorphic Graphs



Non-Isomorphic graphs

1.1.10 Subgraphs and Induced Graphs

Subgraph: A graph ‘g’ is said to be a subgraph of a graph ‘G’, if all the vertices and all the edges of ‘g’ are in ‘G’ and each edge of ‘g’ has the same end vertices in ‘g’ as in ‘G’.

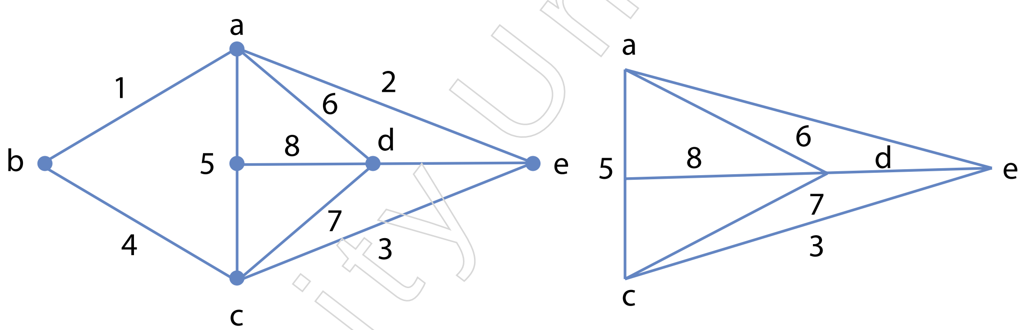


Fig. 12

A graph called “G” and a subgraph called “g” that contains every edge in the graph “G” are depicted in the above picture.

Note:

1. Every graph has a subgraph of its own.
2. A graph ‘G’ subgraph is a subgraph of ‘G’.
3. A graph “G” that has a single vertex is also a subgraph of “G.”
4. A single edge in “G” that has its end vertices together with it also forms a subgraph of “G”.

When ‘g_1’ and ‘g_2’ two or more subgraphs of “G,” are said to be edge disjoint of “G” if ‘g_1’ and ‘g_2’ do not share any edges.

Notes

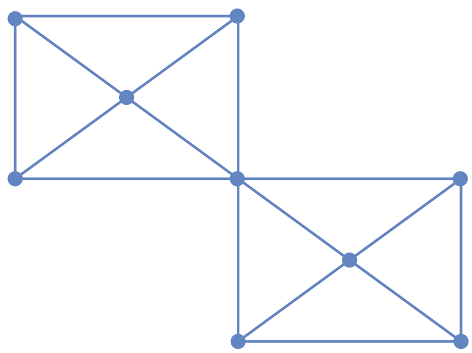


Fig (1.13a) [G]

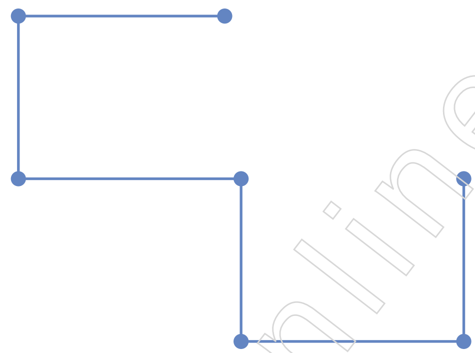


Fig (1.13b) [g_1]

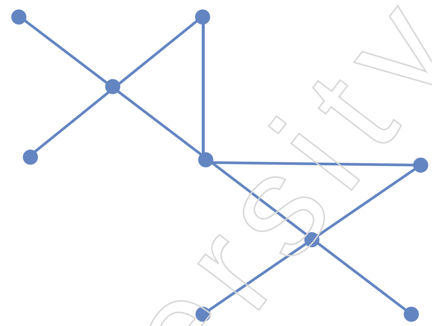
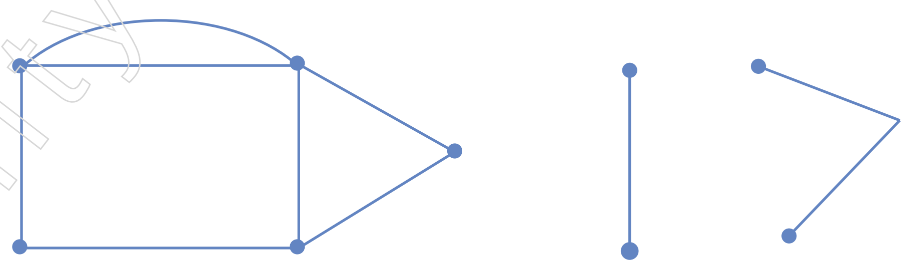


Fig (1.13c) [g_2]

Subgraphs with discontinuous edges are depicted in the above image.

1. Vertices may be shared by edge disjoint graphs, but edges are not shared by them.
2. Subgraphs are considered to be vertex disjoint if they share no vertices.

The below figure (1.14) represents the vertex disjoint subgraphs 'H_1' and 'H_2' do not have any vertices in common.

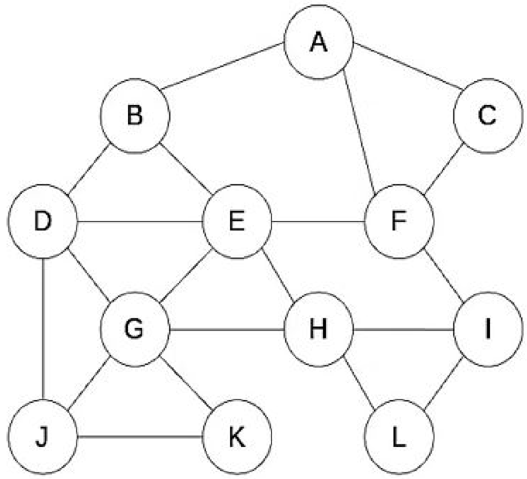


G H_1 H_2

Fig (1.14)

Induced Graph:

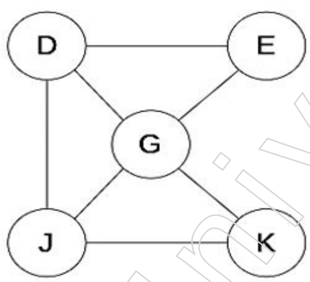
Definition: An induced subgraph is a graph that is created by taking a subset of the graph's vertices and all of the edges (from the original graph) that link the pairs of vertices in that subset together.



Notes

An induced subgraph is a special case of a subgraph. If S is a subset of G's nodes, then the subgraph of G induced by S is the graph that has S as its set of vertices and contain all the edges of G that have both endpoints in S. This definition covers both directed and undirected graphs. Also, it covers the weighted just as the unweighted ones.

Given G and S, the induced subgraph is unique. There's only one subgraph induced by {D,E,G,J,K} in the above graph:



The concepts of an induced subgraph and an ordinary subgraph are very similar. The difference is that an induced subgraph includes all the edges that have both endpoints in the inducing set S, whereas an ordinary subgraph may miss some.

So, an induced subgraph keeps both adjacency and non-adjacency of the inducing vertices, in contrast to an ordinary subgraph that preserves only non-adjacency.

1.2 Advanced Concepts in Graph Theory

The primary goal of Advanced Graph Theory is to solve problems by applying the key ideas of graph theory, together with a thorough understanding of its computer science applications. An in-depth grasp of graphs, as well as the basic ideas and models that underpin the theory, algorithms and methods of proof used in the discipline of graph theory, are provided by this course. The substantial influence of emerging applications of graph theory in the field of computer science. Graph theory contributes to the resolution of technology-driven, research-oriented problems.

1.2.1 Planar Graphs

If there is a geometric representation of a graph G that can be drawn on a plane without any of its edges intersecting, that graph is said to be planar. Nonplanar refers to a graph whose edges cross over when it is drawn on a plane.

Notes

Embedding is the process of creating a graph's geometric representation on any surface so that no edges cross. Thus, to state that G is a nonplanar graph. It is our responsibility to demonstrate that no geometric representation of G can fit inside a plane. In equivalent terms, a geometric graph G is said to be planar if an isomorphic graph to it exists and that graph is embedded in a plane. If not, G is not planar. A plane representation of a planar graph G is defined as an embedding of G on a plane.

Take the graph represented by, for example. Given how intriguing the edges e and f are, it is evident that the geometric representation is not embedded in a plane. Nevertheless, if we redrew an edge outside the quadrilateral while keeping the remaining edges the same, we would have embedded the new geometric graph in the plane, demonstrating the planar nature of the graph being represented. Another illustration would be the two isomorphic diagrams here, which are two distinct geometric renderings of the same graph. One of the diagrams represents a plane, while the other does not. Naturally, the graph is flat. Conversely, none of the three configurations can be drawn on a plane without edges intersecting. The nonplanar nature of the graph that these three distinct representations depict explains why.

A graph G can be either planar or nonplanar; it can be expressed either way, using its geometric representation or an abstract notation G = (V, E, Γ).

1.2.2 Spanning Graphs

Subgraphs are similar to graph subsets. A graph G can be represented as a "spanning subgraph" if it has all of the vertices of graph G but only some of the edges. Cycles and isolated nodes are possible components of a spanning subgraph.

Spanning trees arise when a spanning graph has all of its nodes connected without any cycles.

There are many different uses for subgraphs. In order to prevent packet loops in IP networks, spanning graphs—which in turn give rise to spanning trees—are used extensively in the field of bioinformatics. Subgraphs are also frequently used in the analysis of electrical circuits. The list of applications for spanning graphs is extensive.

Occasionally, a graph is provided, but only one method is required to connect the vertices.

For instance, a network that contains redundant connections.

Redundancy is not a concern when routing data; rather, it is essential to have a method of transferring the data to its intended destination.

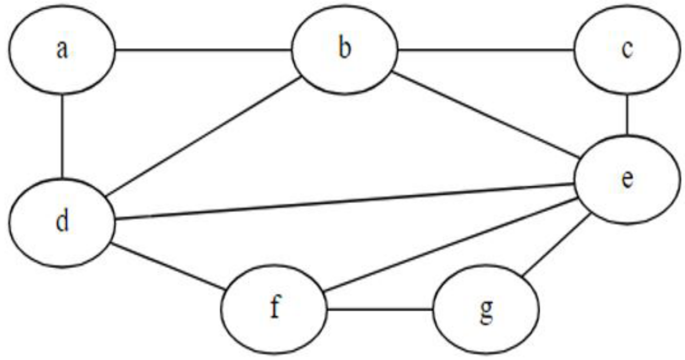
Therefore, the primary objective is to reduce the graph to a tree when examining routing decisions. The tree specifies the precise method by which the data will be transmitted.

In the event that a link is unsuccessful, we have the option of selecting an alternative tree.

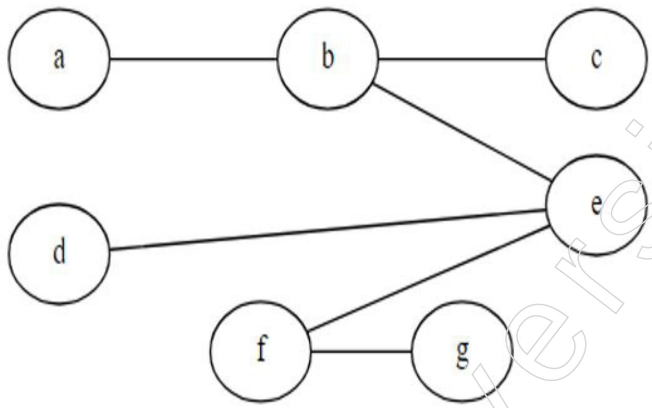
A spanning tree is a subgraph of a graph G = (V, E) that is a tree and connects all of the vertices.

For instance, consider the following graph:

Notes



The outcome is a spanning tree, which is achieved by removing edges until a tree is left.



It is evident that a spanning tree will have |V|−1 edges, as is the case with any other tree.

1.2.3 Reachability in Graphs

If a directed graph is the pair of objects G = (V, E), and each node v in V represents a reachable marking and each edge e in E represents a transition marking, then that graph is said to be reachable. This means that even for a finite petri net (directed bipartite network), the set of reachable markers can be infinite.

Using paths, we can extend edge relations to a relation on arbitrary pairs of nodes in the graph. To define concepts such as distance and reachability, we employ them.

- If and only if there is a path from u to v, we define v as reachable from u in graph G.

  It pertains to distance. Furthermore, we define the distance from u to v in a graph G as the length of the shortest path from u to v or ∞ if v is not reachable from u. This distance is denoted dG(u, v) or d(u, v) when the graph is clear from context.

  Note that the relation R on pairs of vertices is not transitive if we consider edges to be encoding this relation. For example, uRv if and only if (u, v) ∈ E. Therefore, uRv and vRw do not necessarily imply that uRw.

- On the other hand, it is true that if v is accessible from u and w is accessible from v, then w is accessible from v (by concatenating the paths). It is not difficult to demonstrate that reachability as a relation (the pairs that are reachable from each other) is the transitive closure of the standard relation induced by the edges. Specifically, it is the minimum transitive relation that contains the edge relation.

Notes

- Reachability is the transitive closure of the relation imposed by the edges of a graph. If we add an edge (u, v) to a graph G whenever u can reach v, this new graph is the edge minimal graph containing the original graph. In this graph, a (u, v) and (v, w) edge in the new graph imply a (u, w) edge in the new graph.
- The structure of the reachability relation is now a natural subject of reasoning. If we construct a graph in which an edge is added from u to v for each pair of vertices when and only if u is capable of reaching v.

For undirected graphs, it is straightforward to assert that this will generate a diverse array of complete graphs or disjoint cliques.

1.2.4 Union and Intersection of Graphs

1. **Definition: Union:** The **union** of two graphs $G_1=(v_1, e_1)$ and $G_2=(v_2, e_2)$ is another graph G_3 written as $G_3 = G_1 \bigcup G_2$ whose vertex set $V_3 = V_1 \bigcup V_2$ and the edges are $E_3 = E_1 \bigcup E_2$
2. **Definition: Intersection:** The intersection of two graphs, G_1 and,G_2 is a graph, $G_3 = G_1 \cap G_2$ consisting only of those vertices and edges that are in both G_1 and G_2 whose vertex set $V_3 = V_1 \cap V_2$ and the edges are $E_3 = E_1 \cap E_2$
3. **Definition:** The ring sum of two graphs, G_1 and,G_2 (written as $G_1 \oplus G_2$ is a graph) consisting of the vertex set v_1 union v_2 and edges are either in G_1 and G_2 but not in both.
4. Definition:

❖ If G_1 and G_2 are said to be **commutative** then

$$G_1 \bigcup G_2 = G_2 \bigcup G_1$$

$$G_1 \bigcap G_2 = G_2 \bigcap G_1$$

$$G_1 \oplus G_2 = G_2 \oplus G_1$$

❖ If G_1 and G_2 are **edge disjoint** then G_1 intersection G_2 is a null graph

$$G_1 \oplus G_2 = G_1 \bigcup G_2$$

❖ If G_1 and G_2 are **vertex disjoint** then G_1 intersection G_2 is empty

$$\Rightarrow G_1 \bigcap G_2 = \phi$$

❖ For any graph ,

$$G \bigcup G = G \bigcap G = G$$

$$G \oplus G \ = null \ \ graph$$

$$G \oplus g = G - g \ \ whenever \ \ g \subseteq G$$

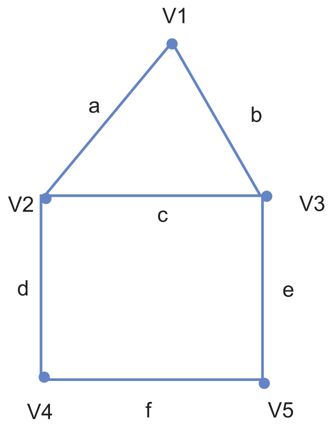
5. Definition: A graph G is said to have been decomposed into two subgraphs, if

$$g_1 \bigcup g_2 = G$$

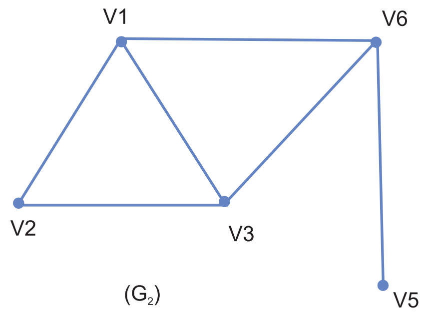
$$g_1 \bigcap g_2 = a \ \ null \ \ graph$$

The below diagram shows the Union, Intersection and Ring sum of the graphs.

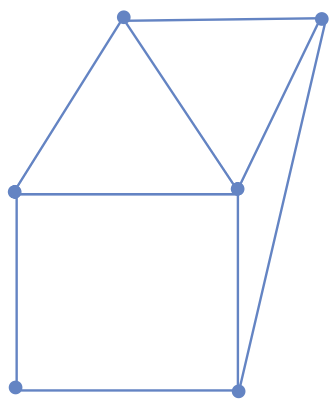
Notes



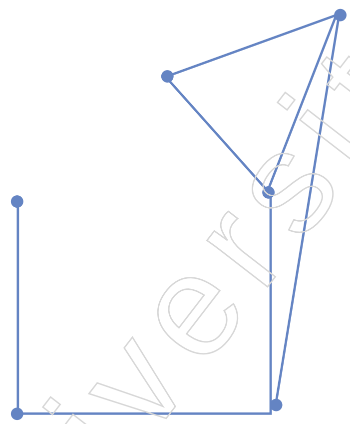
(G_1)



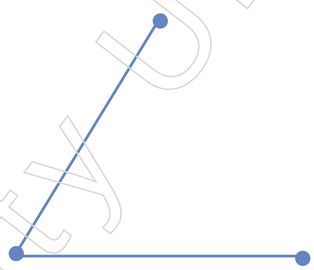
(G_2)



G_1 Union G_2



G_1 ringsum G_2



G_1 Intersection G_2

Summary

- In mathematics, a graph is a visual representation of any arranged data.
- To illustrate a variety of data types, the graphical representation employs bar graphs, frequency tables, line graphs, circle graphs, line plots, and other visual aids.
- Parallel edges are defined as two edges that share the same initial vertex and end vertex.
- If a vertex is incident to an edge, it is one of the two vertices that connect the edges.
- The Handshaking Lemma stipulates that the sum of the degrees of each vertex in a graph G (V, E) is equal to twice the number of edges. for example,

Notes

$$\sum_{i=1}^{n} d(V_i) = 2e$$

- A graph G is considered planar if there is a geometric representation of G that can be traced on a plane such that no two of its edges intersect.
- Spanning trees are generated when all vertices of a spanning graph are connected without any cycles.
- If there is a geometric representation of a graph G that can be drawn on a plane without any of its edges intersecting, that graph is said to be planar.
- If and only if there is a path from u to v, we define v as reachable from u in graph G.
- The **union** of two graphs $G_1=(v_1, e_1)$ and $G_2=(v_2, e_2)$ is another graph G_3 written as $G_3 = G_1 \bigcup G_2$ whose vertex set $V_3 = V_1 \bigcup V_2$ and the edges are $E_3 = E_1 \bigcup E_2$
- The intersection of two graphs, G_1 and,G_2 is a graph, $G_3 = G_1 \cap G_2$ consisting only of those vertices and edges that are in both G_1 and G_2 whose vertex set $V_3 = V_1 \cap V_2$ and the edges are $E_3 = E_1 \cap E_2$

Glossary

- A graph is composed of a collection of points, also referred to as vertices or nodes, that are connected by a collection of lines, also known as edges or arcs.
- Degree: The total number of connected edges of a node is denoted by its degree.
- Subgraph: A subgraph is a component of the graph.
- Trail: Plan a route that avoids passing through any boundaries more than once.
- Walk: A network that is connected in a specific order by a collection of edges that form a continuous path.
- Network theory Network theory offers a variety of techniques for graph analysis.
- Weighted Graph: A graph is referred to as a weighted graph if each branch has been assigned a numerical value.
- Isomorphism graph: A graph in which a singular graph can exist in multiple forms is referred to as an isomorphism graph.
- An induced subgraph is a graph that is generated by combining a subset of the graph's vertices with all of the edges (from the original graph) that connect the pairs of vertices in the subset.
- The **union** of two graphs $G_1=(v_1, e_1)$ and $G_2=(v_2, e_2)$ is another graph G_3 written as $G_3 = G_1 \bigcup G_2$ whose vertex set $V_3 = V_1 \bigcup V_2$ and the edges are $E_3 = E_1 \bigcup E_2$
- The intersection of two graphs, G_1 and,G_2 is a graph, $G_3 = G_1 \cap G_2$ consisting only of those vertices and edges that are in both G_1 and G_2 whose vertex set $V_3 = V_1 \cap V_2$ and the edges are $E_3 = E_1 \cap E_2$

Check Your Understanding

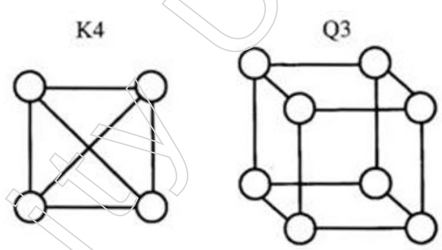
1. When a vertex is not next to every other vertex, it is referred as ____________ vertex.
   a) Pendant
   b) Isolated
   c) Incident
   d) Simple

Notes

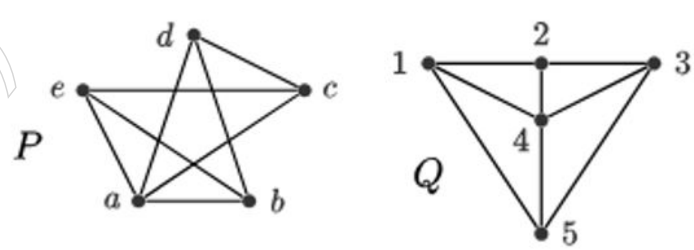
2. Graphs G_1 and G_2 have a total of n_1 and n_2 edges, respectively. Therefore, the number of edges in $G_1 \vee G_2$ is
   a) $n_1 + n_2$
   b) $n_1 + n_2 - 1$
   c) n_1
   d) n_2
3. A cycle on n vertices is isomorphic to its complement. The value of n is
   a) 4
   b) 3
   c) 5
   d) 6
4. If a graph is planar, then it is embeddable on a
   a) Sphere
   b) Square
   c) Circle
   d) Triangle
5. Let G = k_n where n ≥ 5. Then number of edges of any induced sub graph of G with 5 vertices
   a) 6
   b) 5
   c) 8
   d) 10

Exercise

1. Identify the Planar Graph:



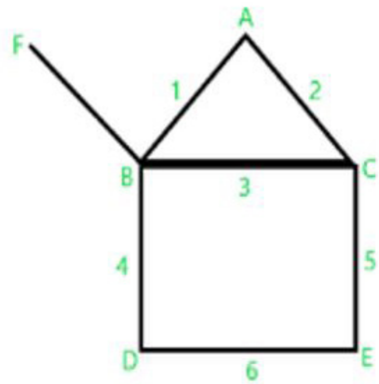
2. For each simple connected undirected graph with more than two vertices, what will be its degree?
3. For a simple graph, a path is always a trail. Justify the statement with the help of example.
4. Define Graph, simple graph, finite graph, Walk, Path and Circuit of a graph with example.
5. Prove that the following graphs P and Q are isomorphic.



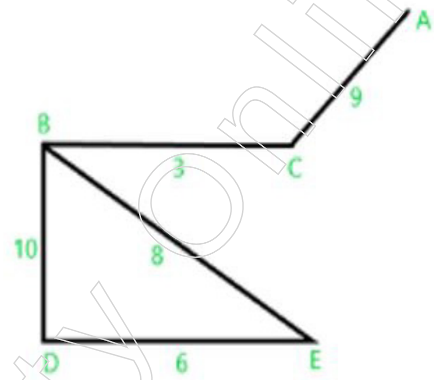
Notes

Learning Activities

1. Think of an eight-vertices, undirected random graph. Between two vertices, there is a 1/2 chance that an edge will exist. How many unordered cycles of length three are expected?
2. Find the union and intersection of following graphs:



G1(V1,E1)



G2(V2,E2)

Check Your Understanding (Answers)

1. a) 2. d) 3. b) 4. d)
5. c)

Further Readings and Bibliography

1. Dr D.S.C ," Graph Theory And Combinatorics All India", Prism Publications, 1 January 2012
2. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 2nd Edition, 2000.
3. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009
4. Harikishan, ShivrajPundir and Sandeep Kumar, Discrete Mathematics, Pragati Publication, 7th Edition, 2010.

Module -II: Graph Transversal Methods

Notes

Learning Objectives:

At the end of this module, you will be able to:

- Define Euler Graph and its Concept
- Solve the Shortest Path Problem Using Dijkstra's Algorithm
- Learn Hamiltonian Circuits and Graph
- Understand the Travelling Salesman Problem
- Define Bipartite Graphs
- Discuss Search Algorithms Like Fleury Algorithm
- Search Shortest Path Using Depth First Search (DFS) and Breadth First Search (BFS) Algorithm
- Define Topological Sorting Algorithm in Graph

Introduction

In the field of computer science, graph traversal, which is also known as graph search, is the process of visiting (as well as modifying and/or examining) each vertex in a graph. The traversals are classified according to the order in which the vertices are visited. A method of locating vertices in a graph is to employ graph traversal. To ascertain the order of node positioning, a graph traversal is implemented. You can search all of the nodes and edges without establishing a loop, as it also searches for edges without doing so.

An Euler graph is a graph in which there exists a closed trail that visits every edge exactly once. A graph that contains an Eulerian circuit (a circuit that uses each edge exactly once) is called an Eulerian graph.

The shortest path problem involves finding the shortest path or minimum distance between two vertices in a graph.

A Hamiltonian graph is a graph that contains a Hamiltonian cycle, a cycle that visits every vertex exactly once and returns to the starting vertex. The Traveling Salesman Problem is a classic optimization problem that seeks the shortest possible route that visits a set of vertices exactly once and returns to the origin vertex. A graph is bipartite if and only if it contains no odd-length cycles. Bipartite graphs are commonly used in modelling relationships and matching problems. Fleury's Algorithm is a simple algorithm for finding an Eulerian path or circuit in a graph. Depth-First Search is a graph traversal algorithm that explores as far down a branch as possible before backtracking. Breadth-First Search is a graph traversal algorithm that explores all vertices at the present depth level before moving on to vertices at the next depth level.

2.1 Euler Graph and Hamiltonian Graph

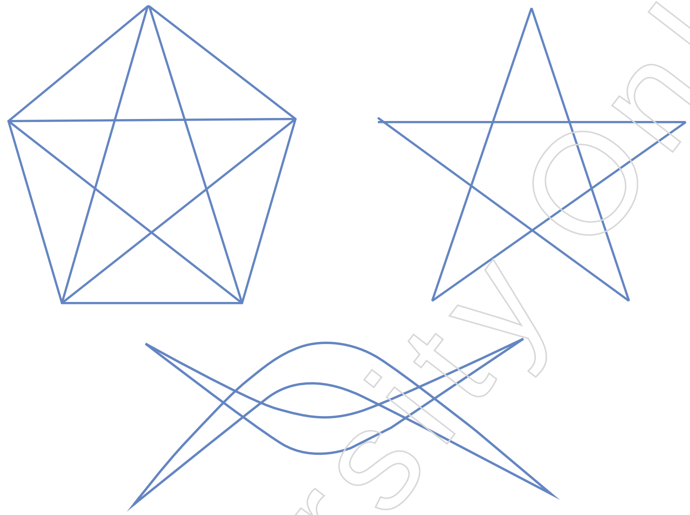
An Euler graph is a graph in which there exists a closed trail that visits every edge exactly once. A graph that contains an Eulerian circuit (a circuit that uses each edge exactly once) is called an Eulerian graph. A Hamiltonian graph is a graph that contains a Hamiltonian cycle, a cycle that visits every vertex exactly once and returns to the starting vertex. A circuit that contains every vertex in a graph G precisely once, with the exception of the beginning and last vertices, is called a Hamiltonian circuit. A walk from one vertex to

Notes

another in a graph G that travels through every vertex and every edge of G exactly once is called an Eulerian path.

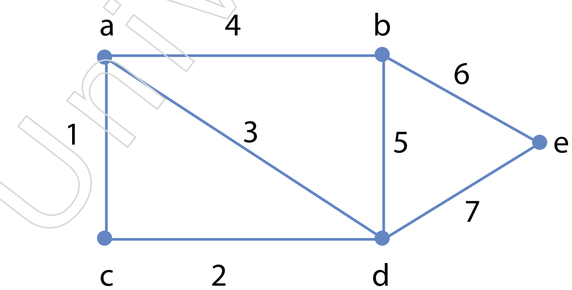
2.1.1 Euler Graph

A graph is referred to as an Euler graph if a closed walk across it includes every edge in the graph. This type of walk is known as an Euler line.



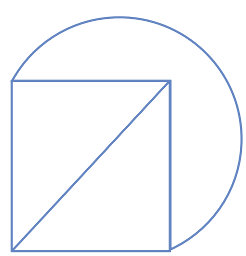
The Euler graph is shown in the above diagram.

Unicursal lines are open walks that include, trace, or cover every edge on a graph without drawing any edges back. It is claimed to be an open Euler line as well. Term used to describe a linked graph with a unicursal line



Theorem 1:

A certain connected graph If and only if every vertex in a graph G has an even degree, then that graph is a euler graph.



Proof:

Necessary condition:

Suppose that G is an euler graph

To Prove:

All vertices in G are of even degree:

G is a graph of eulers. An euler line, or closed walk, is present in it.

Tracing this closed walk allows us to know that, each time it reaches a vertex 'v,' it passes through two new incidents, one of which we entered and the other of which already existing.

This means that since we entered and existed at the same vertex at the start and finish of the walk, respectively, it is true not only of all intermediate vertices but also of the terminal vertex.

Therefore, G is a euler graph and each vertex's degree is even.

Sufficient condition:

Assume that a graph G has exactly one even degree vertex.

We now build a walk that traverses all of G's edges, beginning at any random vertex ('v'), making sure that no edge appears more than once. We keep tracing as far as we can.

We can leave every vertex because they are all of same degree, so we can enter the tracer and quit at any vertex "v."

When the tracing comes to a conclusion, we will finally reach "v" because "v" is also of even degree.

Case: (i)

If the closed walk 'h' that we have just traced contains every edge in G, then G is a euler graph.

Case: (ii)

In the event that 'h' does not contain every edge in 'G,' we subtract every edge from 'G' to create a subgraph 'h' of G made up of the remaining edges.

The degrees of the vertices in "h" are likewise even since all of the vertices in both G and "h" have even degrees.

Moreover, because G is linked, "h" must touch "h" at least once at vertex "a."

We can build a new walk in 'h' again, starting from 'a'. Considering that 'h' has vertices of even degrees. At vertex "a," this walk in "h" needs to come to an end. However, this walk in "h" can be joined with "h" to create a new walk that has more edges than "h" and begins and finishes at vertex "v." We can keep going through this process until we have a closed walk that goes around every edge in G.

G is therefore a euler graph.

Theorem 2:

In a connected graph 'G with exactly 2k odd vertices, there exists 'k' edge disjoint subgraph such that they together contain all edges of 'G' and that each is unicursal graph.

Proof:

Let the odd vertices of the given graph G can be named as $v_1,v_2\ldots v_k$; $w_1,w_2,,,,,w_k$

In any arbitrary order add 'k' edges to G between the vertex pairs (v_1,w_1), (v_2,w_2), (v_3,w_3), (v_k,w_k) to form a new graph G'.

Notes

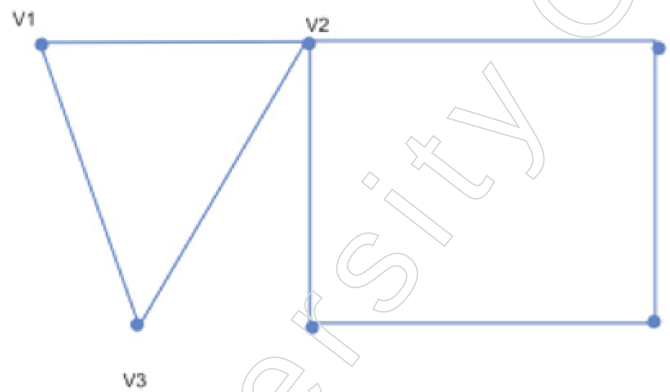
G' comprises all of its vertices, including the euler line, because each vertex has an even degree.

The euler line will now split into "k" walks, each of which is a unicursal line, if we remove the "k"-edges that we just inserted.

A single unicursal line will survive the initial removal. That will be divided into unicursal lines by the second removal and so on until there are 'k' edges between them. Each additional removal will divide a unicursal line into two unicursal lines.

Theorem 3:

Show that if and only if a linked graph G can be broken down into circuits, then it is a euler graph.



Proof:

Assume that graph G is a union of edge disjoint circuits, or that graph G may be broken down into circuits.

Every vertex in G has an even degree since each vertex in a circuit has a degree of two.

G is therefore a euler graph.

In contrast,

Consider a vertex v_1,. There are atleast two edges incident at v_1

Let one of these edges between v_1 and v_2

Since vertex v_2 is also of even degree, it must have atleast another edge say between v_2 and v_3

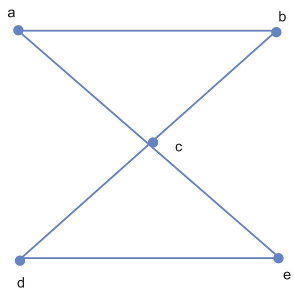
By moving forward in this manner, we finally reach a vertex that has already been crossed, creating a circuit. Let's take the circuit out of G. The remaining graph's vertices must all have an even degree. Remove a second circuit from the remaining graph in the same manner that you eliminated the circuit from G. This technique should be repeated until no edges remain.

Therefore a graph that is connected If and only if G is able to be divided into circuits, then it is a euler graph.

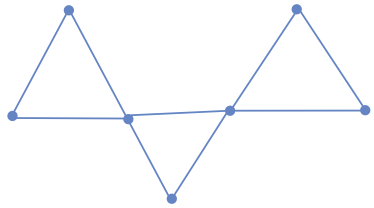
Definition:

An Euler line in a Euler graph is created by taking any closed walk from a vertex V until one reaches a vertex where they can choose any edges that haven't been visited yet. An arbitrarily traceable graph from the vertex is what is known as such a graph.

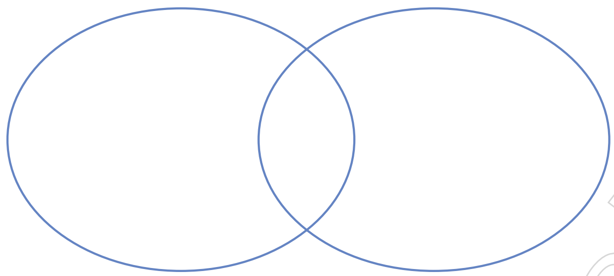
Notes



Arbitrarily traceable graph from G



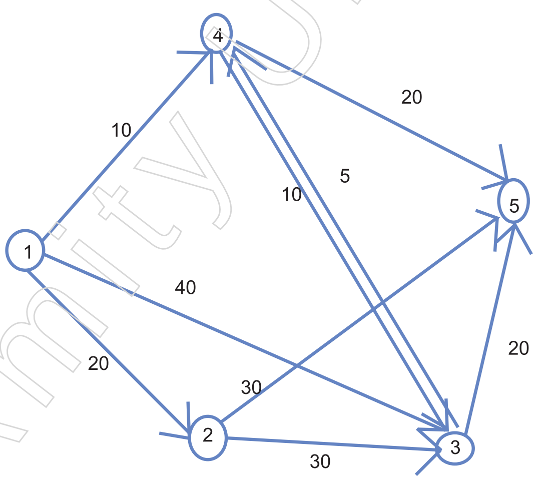
Euler graph not arbitrarily traceable



Arbitrarily traceable graph from all vertices

2.1.2 Shortest Path Problem

A weighted network (V,E,C) with vertex set V, edge set E and weight set C that specifies weight scij for edges (i,j) ∊E is presented. We also receive a node ∊V as a starting point. Finding the shortest path between any two nodes in the network is known as the "one-to-all shortest path problem."



Algorithms Solving the Problem

Dijkstra's algorithm:

Only non-negative cost problems are resolved by it. i.e. $c_i \geq 0$ for all (i,j) ϵE

Characteristics of the Dijkstra Algorithm:

Dijkstra's algorithm is a strong and versatile tool because a far wider range of issues than you might initially believe can be formulated as shortest path problems.

Notes

- Dijkstra's algorithm, for instance, is used to automatically determine directions between physical locations. This is shown in driving instructions on websites like Google Maps and Map Quest.
- Dijkstra's algorithm has been applied to networking and communications applications to solve the shortest path problem, or min-delay path problem.
- It is also utilised in the planning of plants and facilities, robotics and transportation to solve a range of shortest path issues.
- This kind of problem is solved using Dijkstra's method, which determines the shortest path between a particular node and every other node in the network. Node s is referred to as the starting node or initial node.
- The starting values for the distances between each node and the rest of the network are assigned by the Dijkstra algorithm.
- It works in stages, with the algorithm improving the distance values at each stage.
- At every stage, the shortest path between two nodes is ascertained.

Features:

- ❖ Each node is characterised by its state by the algorithm. Two characteristics make up a node's state:
  - (i) Distance value
  - (ii) Status indication
- ❖ A node's distance value is a scalar that indicates how far away it is from other nodes.
- ❖ The status label is an attribute that indicates if a node's distance value is equal to the shortest distance to another node or not.
- ❖ When the distance value of a node equals the smallest distance between nodes, the node's status label becomes Permanent.
- ❖ If not, the system maintains and incrementally modifies the nodes' states, designating a node's status label as Temporary. There is only one node identified as current at each phase.

Algorithm Steps

Step 1. Initial Stage

- Assign the zero-distance value to nodes and label it as Permanent. [The state of node s is (0,p).]
- Assign to every node a distance value of ∞ and Label them as Temporary. [The state of every other node is (∞,t).].
- Designate the nodes as the current node

Step 2.

- Distance Value Update and Current Node Designation Update, Let i be the index of the current node.
- Find the set J of nodes with temporary labels that can be reached from the current node i by a link (i, j). Update the distance values of these nodes.
- For each $j \in J$, the distance valued j of node j is updated as follows new
- d_j=min d_j {d_i+c_{ij}} where c_{ij} is the cost of link (i,j), as given in the network problem.

Notes

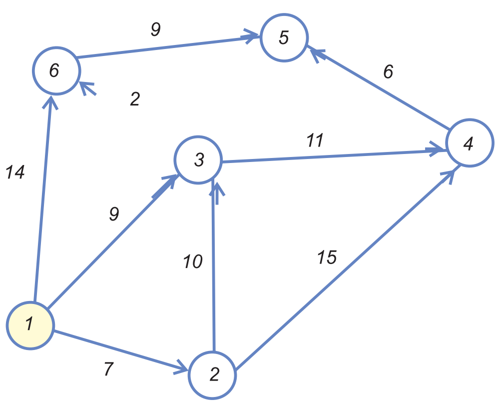
- Determine a node j that has the smallest distance valued among all nodes.
- j ∈ J, find j such that min dφ=dι*j∈J
- Permanently alter node J's label and declare it to be the active node.

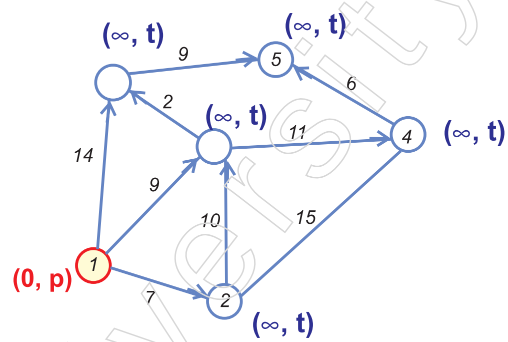
Step 3. Terminal Stage

- Stop; we're done if every node that may be accessed from another node has a permanent label.
- All temporary labels become permanent if we are unable to connect to any temporary labeled node from the present node.
- Otherwise, go to Step 2.

Example 1:

To find the shortest path from node 1 to all other nodes using Dijkstra's algorithm.

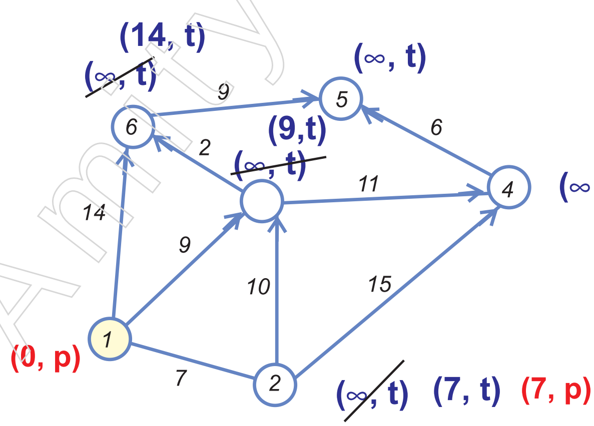




Step1: Initial Stage

- As the current status, Node 1 is assigned.
- Node 1 is in the state of (0,p).
- All other nodes possess state (∞,t).

Step 2:



❖ The present node 1 can connect to nodes 2, 3 and 6.

❖ Update these nodes' distance measurements d_2 = min{∞, 0 + 7} = 7

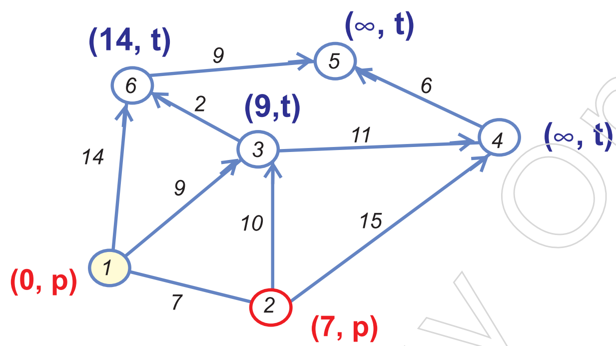
❖ d_3 = min{∞, 0 + 9} = 9 , d_6 = min{∞, 0 + 14} = 14. At this moment, node 2 has the smallest distance value out of nodes 2, 3 and 6.

Notes

- ❖ While the status of nodes 3 and 6 stays temporary, node 2's status label switches to permanent, indicating that its state is (7,p).
- ❖ Node 2 takes over as the active node.

Step: 3

Thus, the graph ends at step 2.



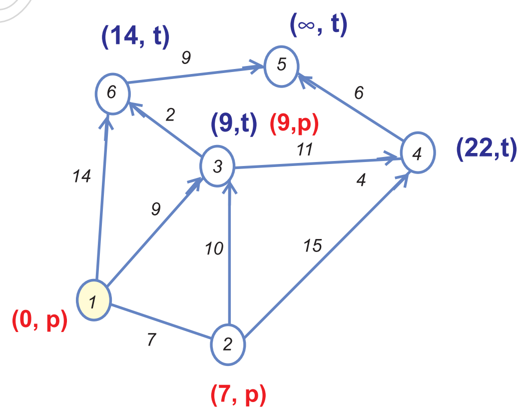
- ❖ We will repeat step 2 because we are not finished and not every node has been reached from node 1.
- ❖ Node 2 takes over as the active node.

Another Implementation of Step: 2

- The present node 2 can connect to nodes 3 and 4.
- Update these nodes' distance measurements.

  d_3= min{9,7+10}=9

  d_6= min{∞,7+15} = 22
- As of right now, node 3 has the least distance value between it and node 4.
- Node 3's status label becomes permanent, but Node 6's status stays temporary.
- Node 3 takes over as the active node.



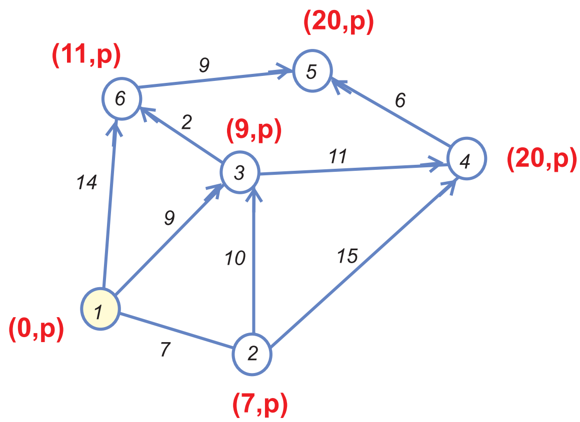
- Since Step 3 fails, we are not finished, thus we carry out Step 2 again.

Another Step: 2

- ❖ From the present node 6, node 5 is accessible.
- ❖ Revise the node 5 distance value.

d_5= min{∞,11+9} = 20

Notes

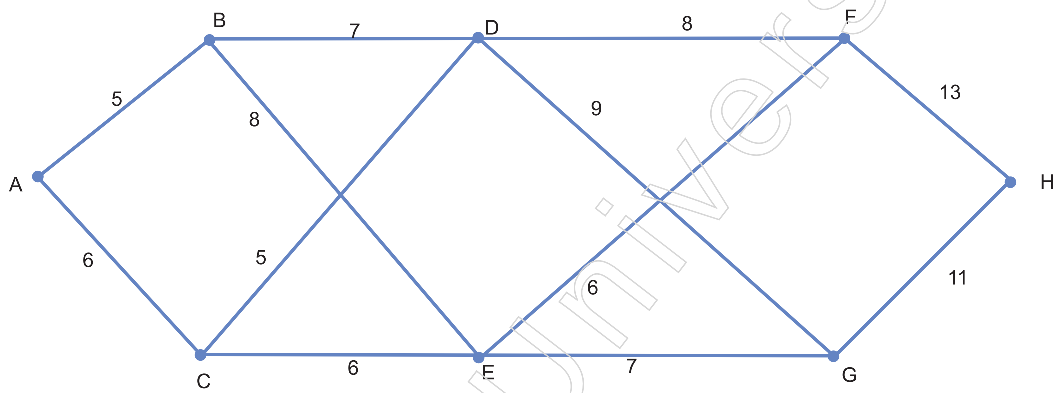


- Since node 5 is now the sole contender, its status is changed to permanent.
- Node 5 takes over as the active node.
- We are unable to access any further nodes from node 5.

Node 4 is consequently given a permanent label.

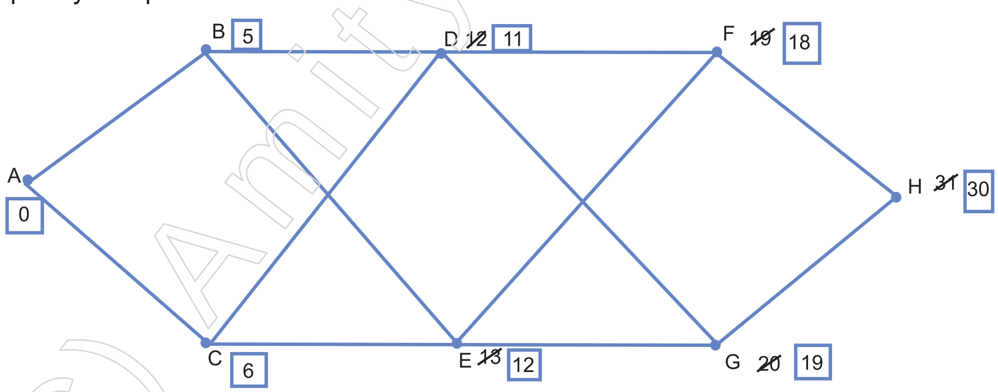
Example 2:

On the network below, find the shortest path between A and H.



Solution:

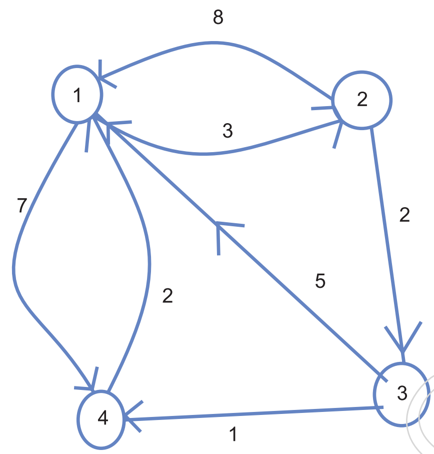
The completely labeled diagram below illustrates the values at each vertex, both temporary and permanent.



Example 3:

Utilising dynamic programming, find every pair of the shortest path

Notes



Solution:

Let us start

1→2,1→3,1→4

2→1,2→3,2→4

3→1,3→2,3→4

4→1,2→2,4→3

Now by solving dynamic programming, it can be solved to taking sequence of decisions. In each we have to take a decision.

Now we have to preparing a matrix using the problem

$$A^0 = \begin{matrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & \infty \\ 5 & \infty & 0 & 1 \\ 2 & \infty & \infty & 0 \end{matrix}$$

If there is no edge or an absence of an edge then it takes to be infinity

Step: 1

Let us consider the First row and first column it remains the same also the diagonals should be remains the same

$A^k [i,j\}=\min\{A^{k-1} [i,j],A^{k-1} [i,k]+A^{k-1} [k,j]\}$

(i) $A^0[2,3]= A^0[2,1]+ A^0[1,3]$

$2 \quad < \quad 8 \quad + \quad \infty$

(ii) $A^0[2,1]= A^0[2,1]+ A^0[1,4]$

$\infty \quad > \quad 8 \quad + \quad 7$

(iii) $A^0[2,1]= A^0[2,1]+ A^0[1,4]$

$\infty \quad > \quad 5 \quad + \quad 3$

In the same we have to prepare the values and form a matrix A^1

$$A^1 = \begin{matrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & 15 \\ 5 & 8 & 0 & 1 \\ 2 & 8 & \infty & 0 \end{matrix}$$

Notes

Step: 2

Here, too, we assume that the diagonals stay the same. We create matrix A^(2)by utilising the matrix A^(1)above, which requires that the second row and second column remain constant.

(i) $A^1[1,3] = A^1[1,2] + A^1[2,3]$

$2 > 3 + 2$

(ii) $A^1[1,4] = A^1[1,2] + A^1[2,4]$

$7 < 3 + 15$

In the same we have to prepare the values and form a matrix A^1

$$A^2 = \begin{matrix} 0 & 3 & 5 & 7 \\ 8 & 0 & 2 & 15 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{matrix}$$

Step: 3

In this case as well, we assume that the diagonals stay the same and create matrix A^(2)using the matrix A^1 above, which requires that the third row and third column remain constant.

(i) $A^2[1,2] = A^2[1,3] + A^2[3,2]$

$3 > 5 + 8$

In the same we have to prepare the values and form a matrix A^1

$$A^3 = \begin{matrix} 0 & 3 & 5 & 6 \\ 7 & 0 & 2 & 3 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{matrix}$$

Step: 4

Likewise, we had the matrix A^4 ready.

$$A^4 = \begin{matrix} 0 & 3 & 5 & 6 \\ 5 & 0 & 2 & 3 \\ 3 & 6 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{matrix}$$

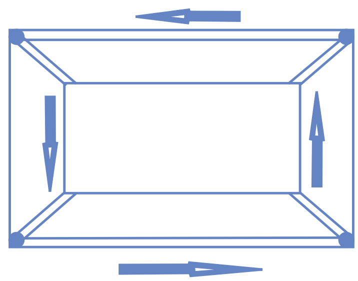
$A^k [i,j] = \min\{A^{k-1} [i,j], A^{k-1} [i,k] + A^{k-1} [k,j]\}$

2.1.3 Hamiltonian Graph

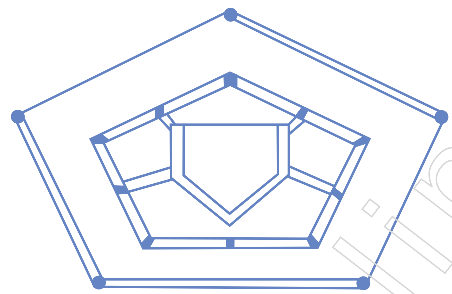
In a linked graph, a Hamiltonian circuit is a closed walk that passes over each vertex in G precisely once, with the exception of the starting and final vertices.

If a circuit in a linked graph G contains each vertex in G, it is said to be Hamiltonian. Consequently, a graph with n vertices and exactly n edges has a Hamiltonian circuit.

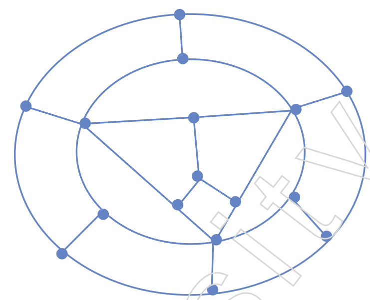
Notes



Hamiltonian circuit

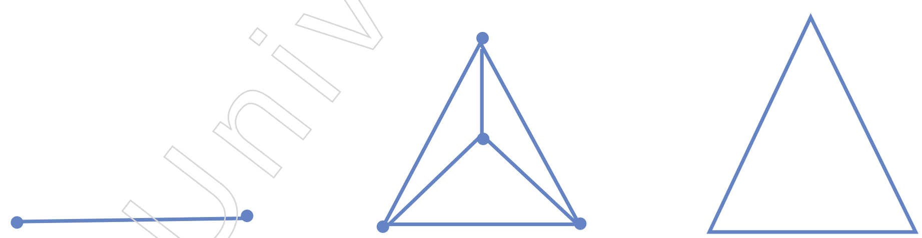


Graph of decahedron with Hamiltonian circuits



Graph without Hamiltonian circuits

A path remains in a Hamiltonian circuit if any one of its edges are removed. We refer to this path as a Hamiltonian path. A complete graph is a basic graph where there is an edge connecting each pair of vertices.



complete graphs

Not complete graph

Theorem:

In a complete graph with 'n' vertices there are n-1/2 edges disjoint Hamiltonian circuits if 'n' is an odd number greater than or equal to 3

Notes

Proof:

A complete graph G of ‘n’ vertices has n(n-1)/2 edges and a Hamiltonian circuit in G consist of ‘n’ edges.

Thus, when n is odd, the number of edges discontinuous Hamiltonian circuit can be represented as follows.

A Hamiltonian circuit is the subgraph (of the full graph of ‘n’ vertices) in the figure.

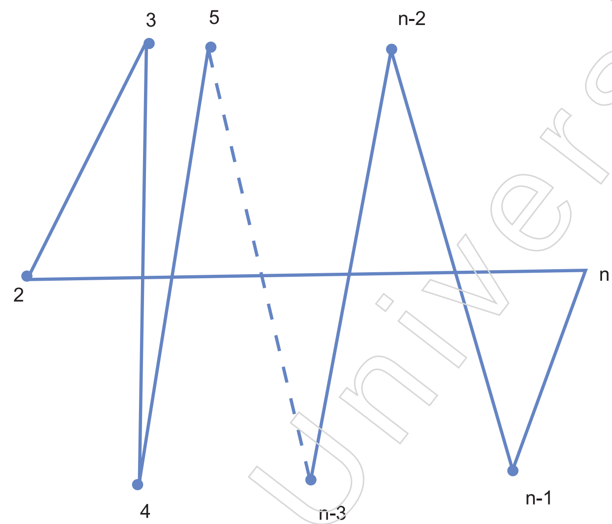
Rotate the polygonal pattern clockwise while maintaining the vertices on a circle by

$$\frac{360}{n-1}, 2.\frac{360}{n-1}, 3.\frac{360}{n-1}, \ldots\ldots \frac{n-3}{2}, \frac{360}{n-1} \text{ degrees}$$

Note that every rotation results in a Hamiltonian circuit that is unique from the others in terms of edges.

As a result, there are n-3/2 new Hamiltonian circuits that are edge disjoint from each other as well as from the circuit in the figure.

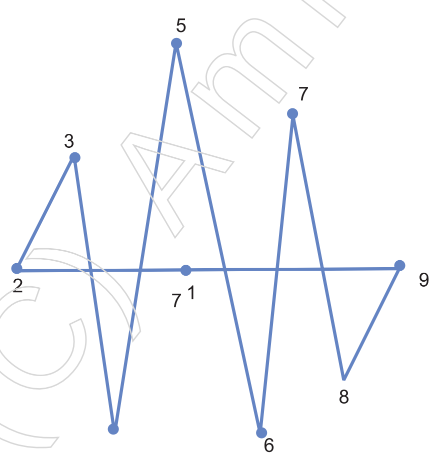
Thus, n-1/2 edge disjoint Hamiltonian circuits exist.



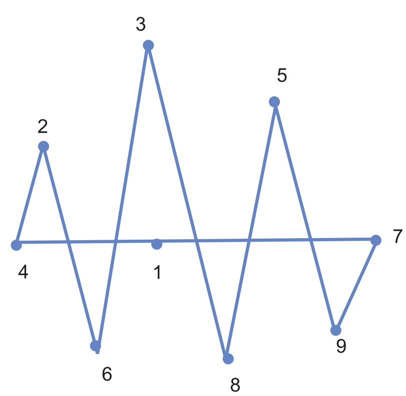
Problem: 1

Find the 4 edge disjoint Hamiltonian cycles in K_9

Solution

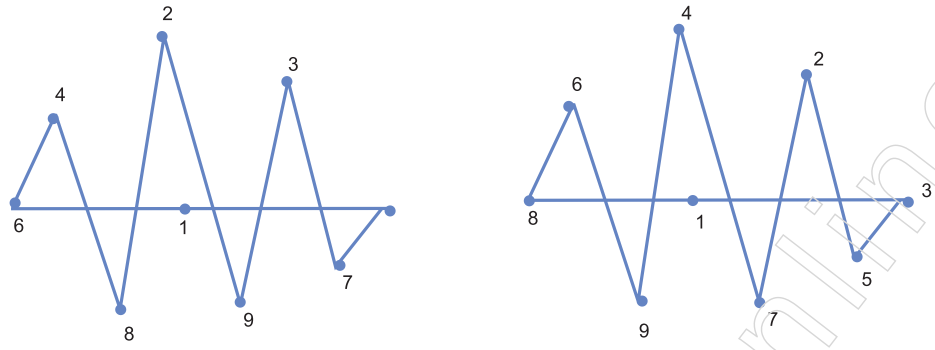


(i)



(ii)

Notes



(iii) (iv)

Edge disjoint hamiltonian cycles in K_9

Dirac's Theorem:

A vertex in G be at least sufficient (but by no means necessary condition) for a simple graph G to have a Hamiltonian circuit is that the degrees of every n/2, where 'n' is the number of vertices in G.

Proof:

Assume that G is non-hamiltonian circuit

Let G be the maximal non-hamiltonian with $n \geq 3$

$d(v_i) \geq \frac{n}{2}$, i=1 to n

We know that,

Any complete graph is Hamiltonian.

Therefore G cannot be complete.

There exist a pair of vertices u and v in G, which are not adjacent

Consider G'=G+uv by choice of G. G' becomes Hamiltonian

Given that G is not Hamiltonian. this means that the edge uv must be present in every Hamiltonian cycle of G. In other words, eliminating an edge uv from any Hamiltonian cycle of G yields an a(u–v) Hamiltonian path in G.

Let p=$v_{1,} v_{2,} v_{3,}$…….,v_n with v_1 = u as original and v_n=v as terminous

By definition,

$A = \{v_i \,/\, uv_i \in E(G)\}$

$B = \{v_i \,/\, v_i u \in E(G)\}$

Now, $uv \notin E(G)$

$v \notin A$ $u \notin B$ and $v \notin B$, $v \notin A \cup B$ and $|A \cup B| < n$

$\Rightarrow vk \in A \cap B$

$v_{1,} v_{2,} v_{3,}$…….,$v_{k,}$ $v_{n,} v_{k-1}$, v_1 is a Hamiltonian cycle of G

Which is a contradiction to the fact that G is non-hamiltonian

Therefore $A \cap B = \phi$

(i.e.) $|A \cap B| = 0$

Now the definition of A and B,

|A|=dG(u) and |B|=dG(V)

Therefore dG(u) +dG(V)=|A| + |B|

$$\leq |A \cup B| + |A \cap B|$$

$$< n+0$$

dG(u) +dG(V) < n

By the hypothesis,

$$\Rightarrow dg(u) \geq \frac{n}{2}$$

$$\Rightarrow dg(v) \geq \frac{n}{2}$$

$\Rightarrow$ dG(u) +dG(V) $\geq \frac{n}{2} + \frac{n}{2}$

$\Rightarrow$ dG(u) +dG(V) $\geq n$

Which is a contradiction

Therefore our assumption is wrong

Therefore G must be Hamiltonian.

2.1.4 Traveling Salesman Problem

A salesman and a group of cities are the subjects of the traveling salesman issues. The salesman is required to travel to every city, beginning in a specific city (such as his hometown) and ending in the same city. The traveling salesman's dilemma lies in having to cut down on the overall trip duration.

- A salesman needs to visit a certain number of cities, such as A, B, C and D.
- Every pair of cities has the distance, duration and cost between them listed.
- The salesperson must travel to each city exactly once before returning to his home city.
- The challenge is figuring out the shortest path in terms of money, time and distance.

Phase –I

- ❖ To get the best answer, TSP can first be solved as an Assignment problem (AP) using the Hungarian technique.
- ❖ After that, examine the TSP state.
- ❖ The AP solution will be the best option even for TSP if the criterion is met.

If not go to Phase –II

- ❖ I - the solution can be adjusted by inspection.
- ❖ II - form a single circuit.
- ❖ III - The iterative procedure – Branch and Bound method.

Example:

A traveling salesperson has made plans to arrive in each place once, depart from a specific city and then journey back to the beginning point. The table below shows the total cost of travel in rupees. Determine the least expensive path.

Notes

To city

From city	A	B	C	D
A	0	25	75	45
B	35	0	150	25
C	35	40	0	15
D	65	75	130	0

Solution:

Step 1: (Row Reduction Method)

First assign the ∞ for the diagonal values.

	A	B	C	D
A	∞	0	50	20
B	10	∞	125	0
C	20	25	∞	0
D	0	10	65	∞

Step 2: (Column Reduction Method)

	A	B	C	D
A	∞	0	0	20
B	10	∞	75	0
C	20	25	∞	0
D	0	10	15	∞

Notes

Step: 3

	A	B	C	D
A	∞	$\boxed{0}$	0	20
B	10	∞	75	$\boxed{0}$
C	20	25	∞	0
D	$\boxed{0}$	10	15	∞

Step: 4

	A	B	C	D
A	∞	0	0	30
B	10	∞	65	0
C	20	15	∞	0
D	0	0	5	∞

Step: 5

	A	B	C	D
A	∞	$\boxed{0}$	0	30
B	10	∞	65	$\boxed{0}$
C	20	15	∞	0
D	$\boxed{0}$	0	5	∞

Step: 6

	A	B	C	D
A	∞	0	$\boxed{0}$	40
B	$\boxed{0}$	∞	55	0
C	10	5	∞	$\boxed{0}$
D	0	$\boxed{0}$	5	∞

Notes

A→C→D→B→A

75+15+75+35

=Rs. 200

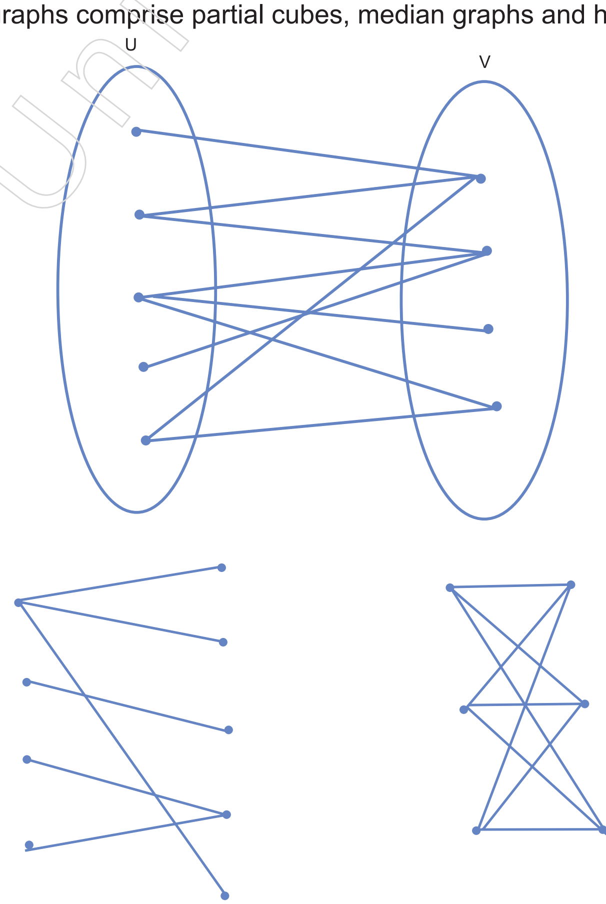
2.1.5 Bipartite Graphs

A graph that has its vertices split into two independent and distinct sets, U {\displaystyle U} U and V {\displaystyle V} V, so that each edge joins a vertex in U {\displaystyle U} U to a vertex V, is called a bipartite graph. Bigraph is the name of it. Comparatively, every edge can be represented as (u, v), linking a vertex from set U to a vertex from set V, or as (v, u), linking a vertex from set V to a vertex from set U.

A bipartite graph is said to be balanced if both sets have equal cardinality, which is defined as having an equal number of vertices in each set. Alternatively, if every vertex in set U has an edge to every vertex in set V and vice versa, the bipartite graph is referred to as complete bipartite graph.

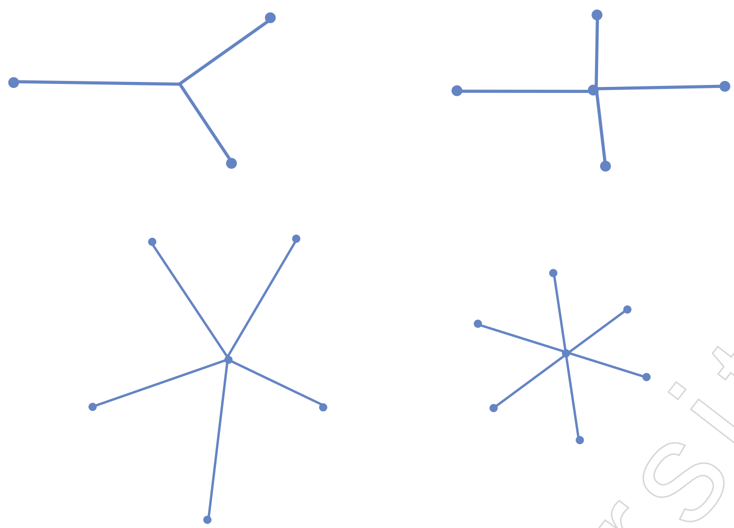
A few of the attributes include:

- Every tree has two parts.
- Cycle graphs are bipartite when the number of vertices is even.
- All planar graphs with even-length faces are bipartite.
- The complete bipartite graph on m and n vertices, denoted by K_(n,m) is the bipartite graph G = (U , V , E) {\displaystyle G=(U,V,E)}, where U and V are disjoint sets of size m and n, respectively and E connects every vertex in U with all vertices in V. it follows that K_(m,n) has mn edges.
- Bipartite graphs comprise partial cubes, median graphs and hypercube graphs.

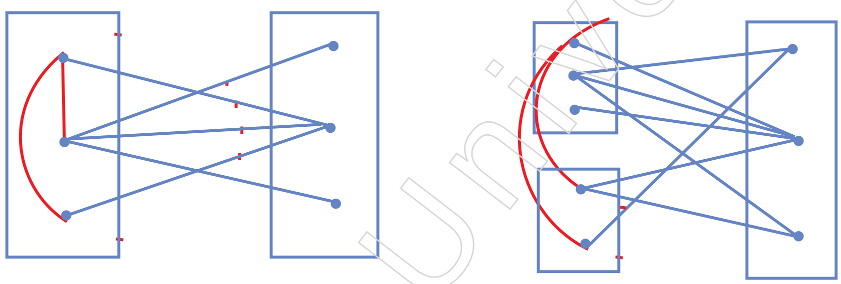


Bipartite graphs are what the graphs above are known as.

When a network's vertices can be divided into two subsets v_1 and v_2 such that no edge has both endpoints in the same subset, and every possible edge that could relate vertices in different subsets is part f the graph. The structure of bipartite graph is $(V_{1,}V_2,E)$ where $v_1 \in V_1$ and $v_2 \in V_2$

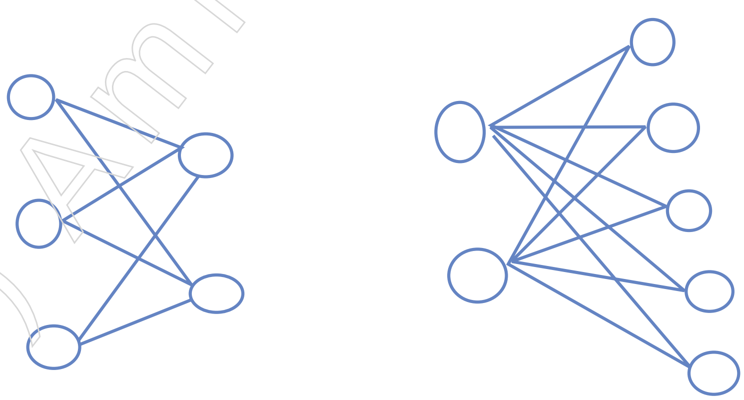


We refer to the graphs above as full bipartite graphs.



Non Bipartite graphs due to shared features

Instances of fully bipartite graphs include:



$K_{3,2}$ $K_{2,5}$

Notes

2.1.6 Fleury Algorithm

Named after the Swiss mathematician Leonhard Euler, Euler Circuits and Paths are intriguing ideas that offer a strong framework for delving into and resolving network and interconnected structure-related challenges. We will explore an algorithm named for the mathematician Fleury, whose work has had a major impact on this area.

A unique kind of graph known as an Eulerian graph has a path that passes through each edge precisely once. It visits every edge without repeating, beginning at one vertex (referred to as the "initial vertex") and ending at another (referred to as the "terminal vertex").

An Euler Circuit, on the other hand, is a graph's closed path. It starts and ends at the same vertex, just like an Euler Path, except it covers each edge exactly once. The initial and terminal vertices in this instance are the same.

The Fleury algorithm, which bears the name of the French mathematician and engineer Paul-Victor Fleury, is an effective method for locating Eulerian circuits and pathways inside graphs. A precise and trustworthy technique for figuring out whether a given graph has Eulerian pathways, circuits, or neither is Fleury's algorithm. This algorithm uncovers the Eulerian structures buried within the graph by systematically exploring it and recording the edges it visits. It does this by going through a sequence of phases.

Algorithm:

A methodical approach to locating Eulerian circuits and pathways in graphs is Fleury's algorithm. It provides an easy-to-follow, sequential method for revealing a graph's hidden structures. We begin with a given graph that we want to examine for the existence of Eulerian circuits or pathways before using Fleury's approach.

First, we confirm that there are two or zero odd vertices in the provided graph. If there are zero odd vertices, we start at any point. If there are two odd vertices, we begin at one of them and proceed through each edge one at a time. In situations where choosing between a bridge (cut edges) and something else is possible, we invariably choose the non-bridge option. We stop when we run out of edges.

The main steps involved are as follows:

1. Choose a Starting Vertex: To get started, we decide which vertex in the graph to start with. This will be the starting point.
2. Adhere to the Edges: We travel an unexplored edge from the selected vertex to a nearby vertex. Only the edges that haven't been explored yet should be followed.
3. Mark Went to the Edges: We designate an edge as "visited" after navigating it so that we don't go back.
4. Continue Until Stuck: We keep going, selecting only previously unexplored edges, until we run out of time to go forward without going back to an edge. This is when Fleury's algorithm's power to guide us comes into play.

Fleury's Flowchart

Notes

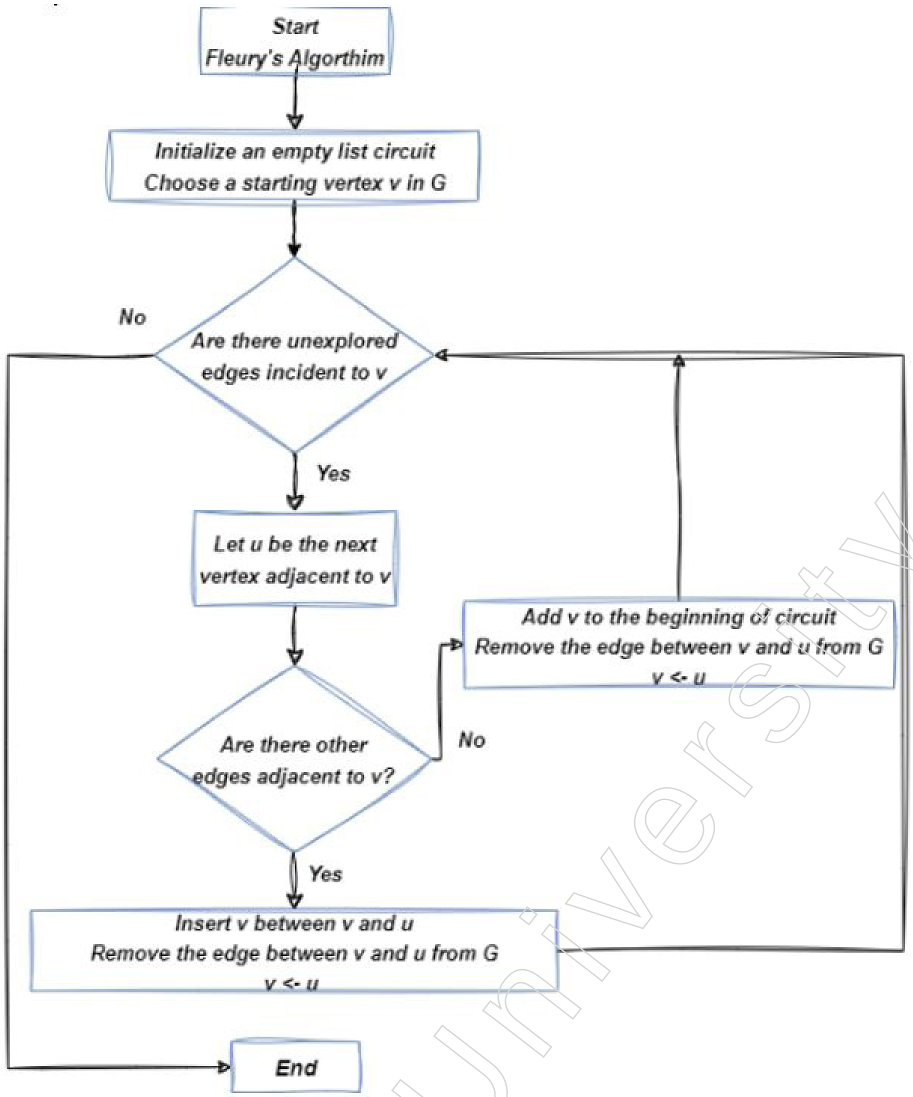


Figure : Fleury flowchart

Fleury's algorithm handles bridge edges in a way that is important. In a graph, bridge edges are edges that, if eliminated, would result in an increase in the number of connected components. A bridge edge can provide problems in the setting of Eulerian circuits and routes. When we come across a bridge edge during our research, according to Fleury's algorithm, we should only choose it if it is the only choice available. Additionally, the bridge edge should be stored as a "candidate" edge so that it can be used later in the algorithm if needed, ensuring that we can return to the initial vertex.

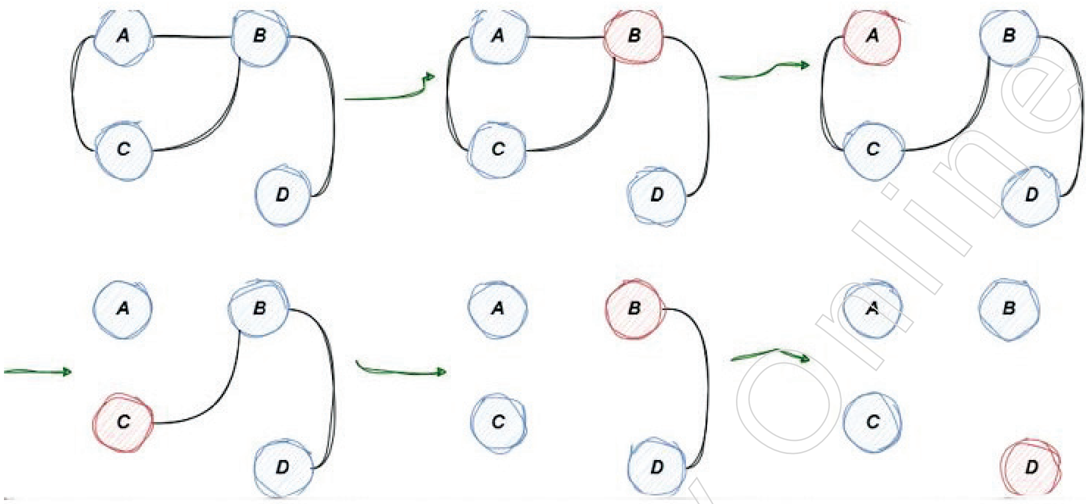
Until we've traveled down every edge, gone back to the initial vertex, or arrived at the intended terminal vertex, the algorithm keeps running.

There are checks in Fleury's algorithm to see if a graph is Eulerian. There isn't an Eulerian path or circuit if, at any stage of the procedure, we find that the graph has unexplored edges. There won't be a solution in these situations since we can't get to the intended terminal vertex or go back to the beginning vertex.

Example

To illustrate Fleury's algorithm, let's look at a simple graph. Our goal is to determine whether an Eulerian circuit exists given this graph's four vertices (A, B, C and D) and four edges.

Notes



Any of the two available odd-degree vertices, “B” and “D,” can be the starting point for a pathway.

Let’s start our walk at vertex ‘B’ now. Three edges radiate from vertex “B.” We don’t select the edge “B-D” because there is a bridge there and we are unable to go back to “D.” You can use either of the two remaining edges. Let’s say we go with “B–A.” When this edge is eliminated, we proceed to vertex “A.” After choosing and erasing the single edge from vertex “A,” we go on to vertex “C.” The current name of the Euler tour is “B–A A–C.”

Again, vertex B only has one edge, so we choose it, remove it and move on to vertex “D”. Euler tour is now “B–A A–C C–B B–D”.

Since there are no more edges to be located, this is where we end. The tour that ends is called “B-A A-C C-B B-D.”

2.1.7 Depth-First Search

A recursive technique called depth first search, also known as depth first traversal, searches every vertex in a graph or tree data structure. To traverse a graph is to visit every node in the graph.

Algorithm for Depth First Search

Every vertex in the graph is classified into one of two groups by a standard DFS implementation:

1. Visited
2. Not Visited

The algorithm’s goal is to prevent cycles and label every vertex as visited.

Here’s how the DFS algorithm functions:

- To begin, arrange any vertex in the graph on top of a stack.
- Place the item at the top of the stack on the visited list.
- Make a list of the nodes that are next to that vertex. Place the ones at the top of the stack that are not on the visited list.
- Until the stack is empty, keep repeating steps 2 and 3.

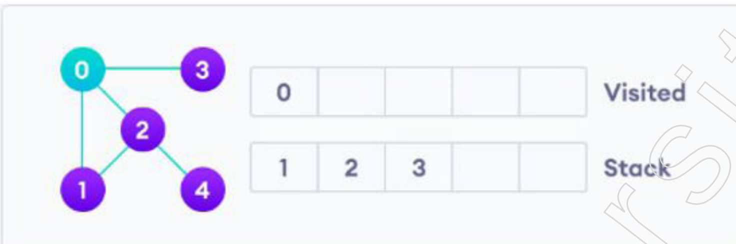
Notes

Example of Depth First Search

Using an example, let's examine how the Depth First Search algorithm functions. We make use of a five-vertex, undirected graph.

Undirected graph with 5 vertices

Beginning at vertex 0, the DFS algorithm adds all of its neighboring vertices to the stack and places it in the Visited list.

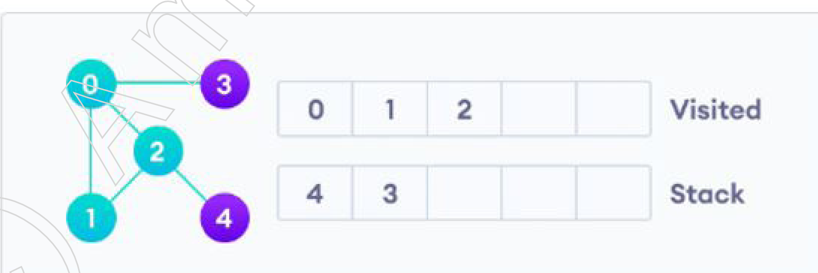


Visit the element and put it in the visited list

Next, we go to the nodes that are next to element 1, which is at the top of the stack. We visit 2 instead of 0 because it has already been visited.

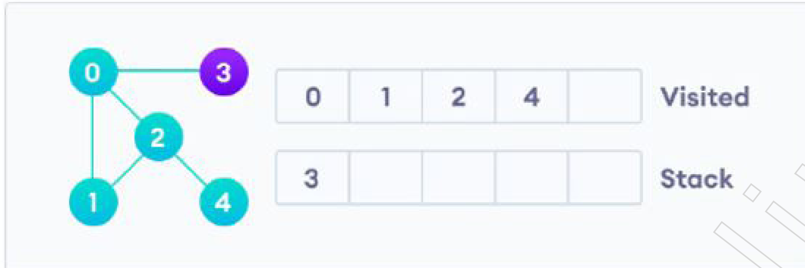
Visit the element at the top of stack

We put the unvisited adjacent vertex of vertex 2 to the top of the stack and visit it since it is located in vertex 4.



Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

Notes



Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

We have finished the Depth First Traversal of the graph when we visit the final element 3, which has no unvisited neighboring nodes.

After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the Depth First Traversal of the graph.

Implementing the DFS Algorithm

1. To locate the route.
2. Finding a graph's strongly connected components.
3. To determine whether the graph is bipartite.
4. In order to identify cycles in a graph.

2.1.8 Breadth-First Search

To traverse a graph is to visit every node in the graph. A recursive technique for looking through every vertex in a graph or tree data structure is called breadth first traversal, sometimes known as breadth first search.

BFS algorithm

Every vertex in the graph is classified into one of two categories by a standard BFS implementation:

1. Made a visit
2. Unvisited

The algorithm's goal is to prevent cycles and label every vertex as visited.

The algorithm functions as follows:

- To begin, place any vertex in the graph at the back of the queue.
- Move the item at the front of the queue to the visited list.

- Make a list of the nodes that are adjacent to that vertex. Place those not on the visited list at the back of the line.
- Repeat steps 2 and 3 until the queue is empty.

The network may consist of two distinct disconnected sections, therefore we can perform the BFS algorithm on each node to ensure that we cover every vertex.

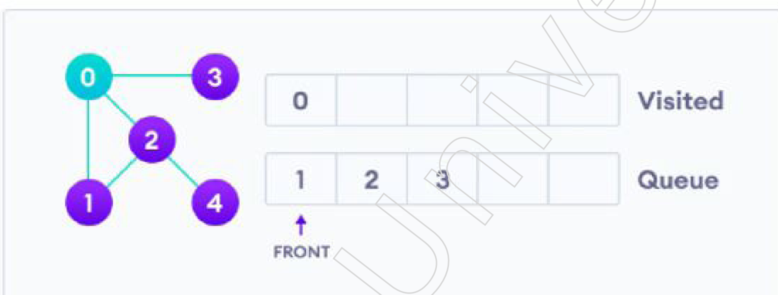
Notes

BFS illustration

Using an example, let's examine how the Breadth First Search algorithm functions. We make use of a five-vertex, undirected graph.

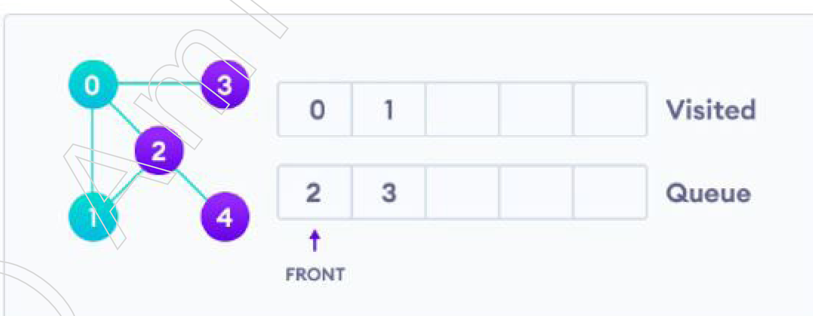
Undirected graph with 5 vertices

Beginning at vertex 0, the BFS algorithm adds all of its neighboring vertices to the stack and places it in the Visited list.



Visit start vertex and add its adjacent vertices to queue

We then visit the element (1), which is at the head of the queue and visit its neighboring nodes. We visit 2 instead of 0 because it has already been visited.



Visit the first neighbour of start node 0, which is 1

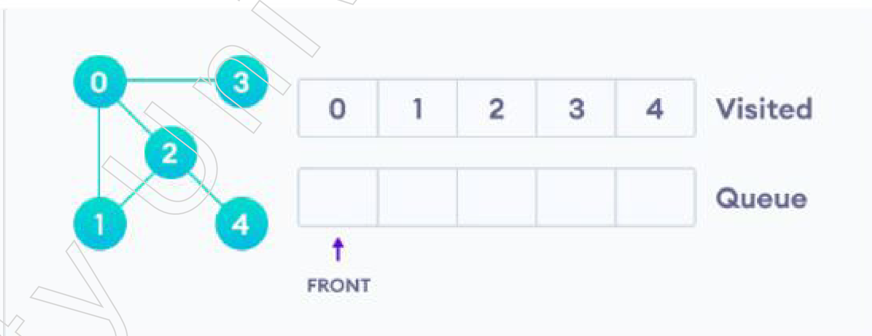
We move vertex 2 to the back of the queue and visit 3 at the front since vertex 2 has an unvisited adjacent vertex in 4.

Notes

Visit 2 which was added to queue earlier to add its neighbours

4 remains in the queue

Since the single node that is near to node 3, is 0 has already been visited, only 4 are left in the queue. We go there.



Visit last remaining item in the queue to check if it has unvisited neighbors

We have finished the graph's Breadth First Traversal because the queue is empty.

Algorithm Applications for BFS

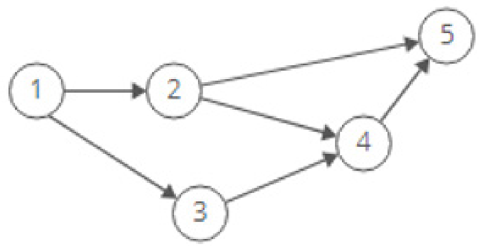
1. Creating an index through a search index.
2. For algorithms used in GPS navigation.
3. For path finding.
4. Using the Ford-Fulkerson algorithm, determine the network's maximum flow.
5. Finding cycles in an undirected graph.
6. In the minimum spanning tree.

2.1.9 Topological Sorting

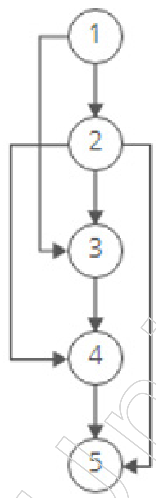
An array of nodes with each node appearing before all of the nodes it points to is produced by the topological sort algorithm given a directed graph.

TOPOLOGICAL ORDERING: The arrangement of nodes within an array.

As an illustration, consider this:

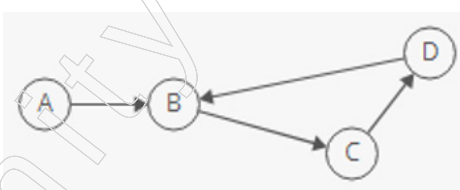


Node 1 comes before nodes 2 and 3 in the ordering since it points to them. Furthermore, nodes 2 and 3 show ahead of node 4 in the ordering since they both point to it.



Thus, the graph's topological ordering would be [1, 2, 3, 4, 5].

Check out this cycle-driven directed graph:



An impractical set of restrictions is produced by the cycle: B must come before and after D in the ordering.

Cyclic graphs typically don't have any legitimate topological orderings.

The Algorithm

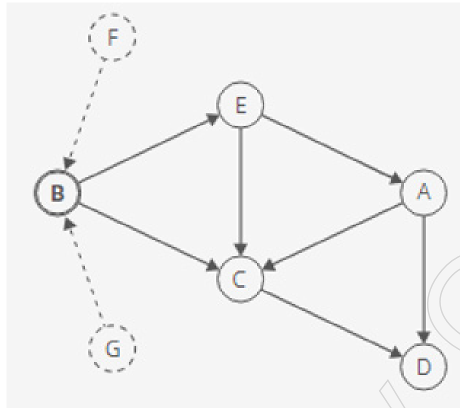
This is a typical design pattern for algorithms:

1. Determine how to obtain the initial item.
2. Take the first item out of the equation.
3. Repeat.

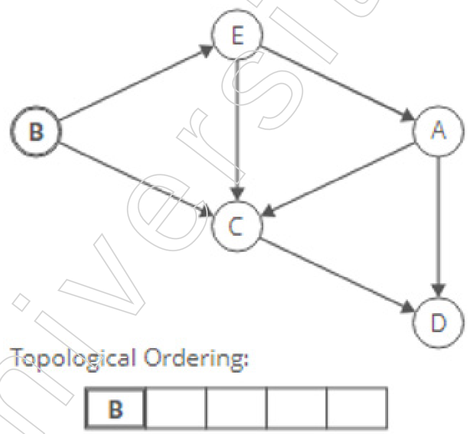
How can we generate this directed graph's topological ordering?

Notes

It is now time to examine the topological ordering's initial node. No incoming directed edges are allowed on that node; it needs to have an indegree of 0. Because the reason for this is that the nodes pointing to it would have to arrive first if it had coming directed edges.

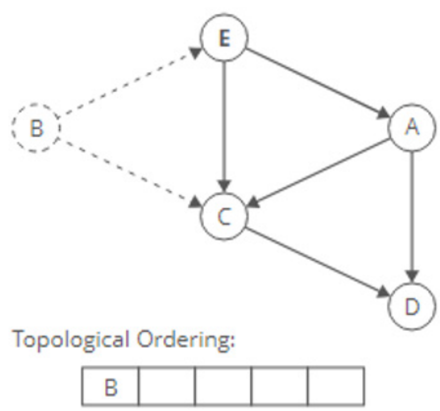


Consequently, we will add a node to the topological ordering that has an indegree of zero.



That takes care of our topological ordering's first node. And the one after that?

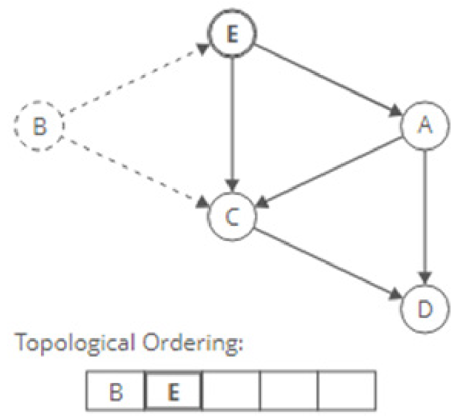
A node and all of its outgoing edges can be removed from the graph once it has been included in the topological ordering.



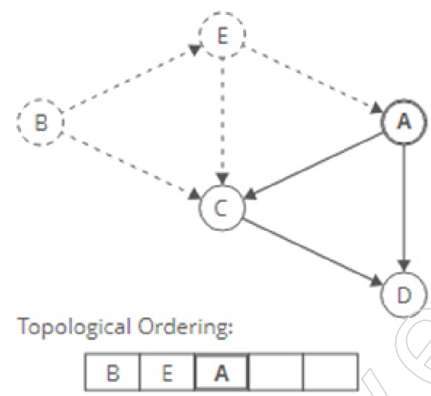
Next, we can use the same strategy we used before: find any node that has an indegree of zero and include it in the ordering.

This is how it appears on our graph. Taking a node with an indegree of zero, we will remove it from the network and add it to our topological ordering:

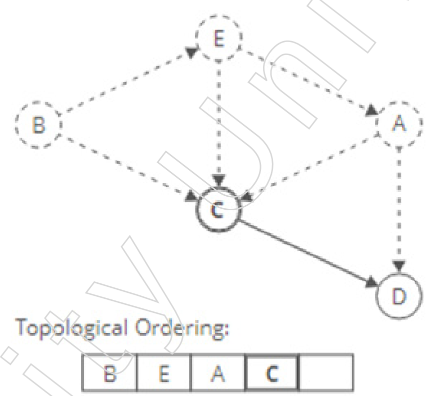
Notes



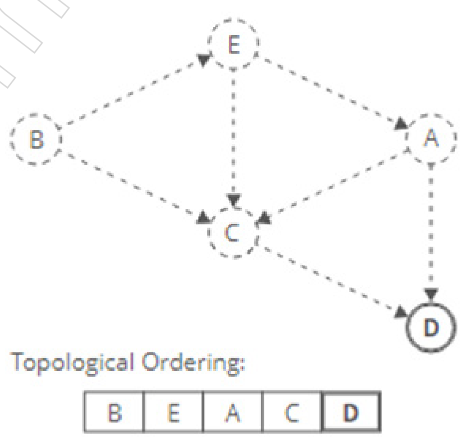
And repeat



Then repeat



Until we're



Notes

Out of nodes.

Implementation

We'll apply the above-described strategy:

1. Find a node that has no edges coming in.
2. Include that node in the hierarchy.
3. Take it out of the graph.
4. Repeat.

Until there are no more nodes with indegree zero, we will continue this cycle. Two things could cause this to occur:

1. Nodes are no longer present. They have all been added to the topological ordering and removed from the graph.
2. A few nodes remain, but they are all connected by incoming edges. This indicates that there is no topological ordering and that the graph contains a cycle.

Application

The most popular use of topological sorting is the ordering of process steps in which certain processes are interdependent.

For instance, when preparing a chocolate bundt cake,

1. Before placing the ingredients in the bundt pan, they must be combined.
2. Before pouring in the batter, the bundt pan needs to be floured and greased.
3. Before the cake bakes, the oven must be preheated.
4. The cake must be baked before it can be allowed to cool.
5. Before icing, the cake must cool.

Although we've used a lighthearted example here, the same reasoning holds well for any collection of dependent activities, such as assembling parts of a sizable software project, analysing data in a Map-Reduce job, or enabling hardware components during bootup. (For instance, before the BIOS attempts to load the bootloader from disk, the mother board needs to initialize the hard drive.)

Summary

- A Euler graph is a graph in which each connected vertex has an even degree. To put it differently, an Euler graph is a specific type of connected graph that is equipped with an Euler circuit.
- The one-to-all shortest path problem is the challenge of identifying the shortest path from nodes to all other nodes in the given network.
- The algorithm of Dijkstra is implemented to autonomously determine the route between physical locations, such as driving directions on websites like Google Maps or Map quest.
- The degree of every n/2 vertex in G is Dirac's Theorem, which is at least sufficient (but by no means necessary) for a simple graph G to have a Hamiltonian circuit. Note that n is the number of vertices in G.
- The travelling salesman issues pertain to a salesman and a collection of cities. Beginning in a specific location (such as his or her birthplace) and concluding in the

same city, the salesperson is obligated to travel to each city. The primary obstacle for the travelling salesman is to reduce the duration of the journey.

- The Fleury algorithm, named after the French mathematician and engineer Paul-Victor Fleury, is a highly effective approach for identifying Eulerian circuits and pathways within graphs.
- The topological sort algorithm generates an array of nodes in which each node is located before all of the nodes it refers to, given a directed graph.

Glossary

- Euler graph: A closed walk in a graph that comprises all of the edges of the graph is referred to as an Euler line, and the resulting graph is referred to as an Euler graph.
- A Hamiltonian circuit in a connected graph is a closed walk that traverses every vertex of G exactly once, with the exception of the starting and final vertex.
- A complete graph is a basic graph in which an edge exists between every pair of vertices.
- A bipartite graph is a graph whose vertices can be partitioned into two disjoint and independent sets, U and V, such that every edge connects a vertex in U to a vertex in V.
- Depth First Search (DFS) is an edge-based approach. It operates in two phases, utilising the Stack data structure: visited vertices are popped if there are no visited vertices, and they are placed onto the stack otherwise.
- The Breadth First Search (BFS) algorithm is employed to locate a node within a graph data structure that meets a specific set of criteria. It visits each node at the current depth level, starting with the graph's root, before proceeding to the nodes at the subsequent depth level.
- Topological Ordering: The arrangement of nodes within an array.

Check your Understanding

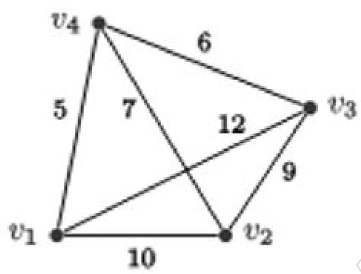
1. A graph is called a __________ if it is a connected acyclic graph.
   a) Cyclic graph
   b) Tree
   c) Regular graph
   d) Trivial graph
2. The graph in which, there is a closed trail which includes every edge of the graph is known as?
   a) Hamiltonian graph
   b) Euler graph
   c) Planar graph
   d) Directed graph
3. Let G be a simple undirected planar graph on 10 vertices with 15 edges. If G is a connected graph, then the number of bounding faces in any embedding of G on the plane is equal to
   a) 6
   b) 5
   c) 4
   d) 7

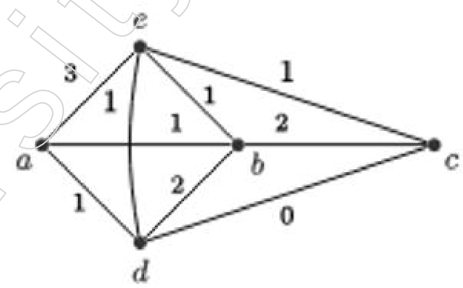
Notes

4. The Fleury algorithm is an effective method for locating ___________ and pathways inside graphs.
   a) Planar graph
   b) Eulerian circuits
   c) Bipartite graph
   d) Hamiltonian graph
5. Depth First Search, or DFS, is an edge-based method which uses ____________.
   a) Stack data structure
   b) Queue data structure
   c) both a and b.
   d) None of the above

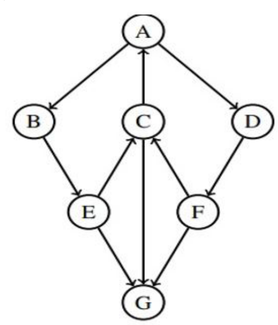
Exercises:

1. Find a Hamiltonian cycle of least cost in each of the following graphs:

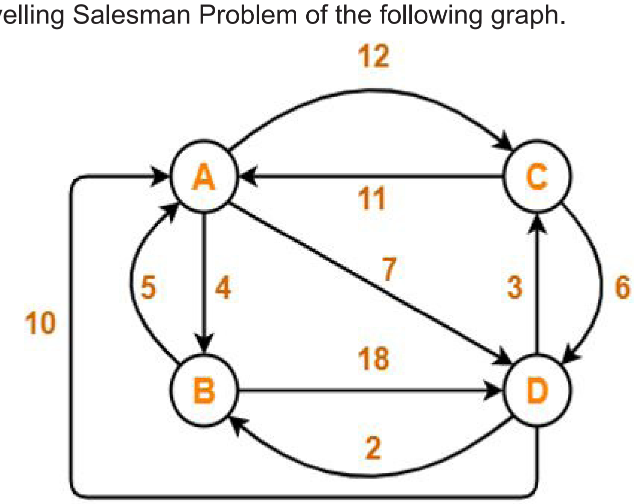




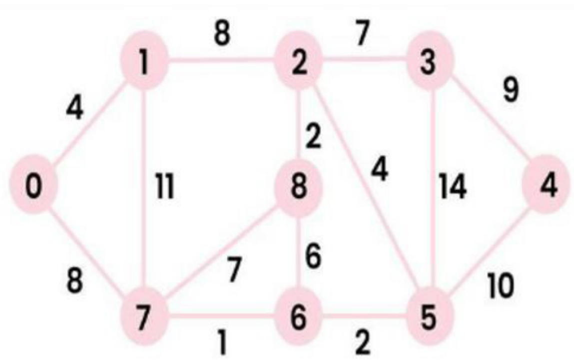
2. Find all pairs of natural numbers (m, n) for which Km, n is Eulerian.
3. Perform a Depth First Search on the following graph starting at A.



4. Solve the Travelling Salesman Problem of the following graph.



5. Find the Shortest Path from Source to all vertices using Dijkstra's Algorithm.



Learning Activities

1. Let G = (V, E) be a directed graph. Prove that G has a directed cycle if DF S(G) has a back edge.
2. For the topological sort algorithm, describe how would you find a source vertex in the graph and hence analyse the total runtime complexity of the algorithm.

Check Your Understanding (Answers)

1. c)	2. b)	3. a)	4. b)
5. a)			

Further Readings and Bibliography

1. Dr. D.S.C," Graph Theory and Combinatorics All India", Prism Publications, 1 January 2012.
2. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 2nd Edition, 2000.
3. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009.
4. Harikishan, ShivrajPundir and Sandeep Kumar, Discrete Mathematics, Pragati Publication, 7th Edition, 2010.

Module - III: Trees

Notes

Learning Objectives:

At the end of this module, you will be able to:

- Learn the basics of trees
- Define Binary Trees and its properties
- Describe Rooted trees and Path Length in Rooted trees
- Learn Spanning Trees and Spanning Trees of Weighted Graphs
- Define Fundamental Circuits in Graph
- Infer Cut Sets and Cut Vertices
- Analyse Fundamental Cut Set
- Learn the concept of Minimum Spanning Tree (MST)
- Identify Edge and Vertex Properties in MST
- Differentiate between Kruskal's Algorithm and Prim's Algorithm for Finding MST

3.1 Introduction to Trees

Trees are graphs that lack even a single cycle. Hierarchical structures are represented in a graphical format. Trees are the most fundamental of all graph classes. Despite their simplicity, they possess a complex framework. Trees provide a diverse array of practical applications, ranging from simple family trees to complex computer science data structure trees.

Binary trees are widely used in computer science for tasks such as sorting and searching. A rooted tree is a tree in which one vertex is designated as the root. In rooted trees, the distance from the root to any other node is known as the path length. Path length is significant in various algorithms, including those for finding the height or depth of the tree.

A spanning tree of a graph is a subgraph that includes all the vertices of the original graph and is a tree. Fundamental circuits (or cycles) in a graph are cycles formed by adding an edge to a spanning tree. A fundamental cut set in a graph is a set of edges associated with a spanning tree. Each fundamental cut set corresponds to the edges that, when removed, would separate a particular pair of vertices. A minimum spanning tree (MST) is a spanning tree of a weighted graph that has the smallest possible total edge weight.

Kruskal's algorithm is a greedy algorithm that finds an MST by sorting all the edges in non-decreasing order of their weights and adding them one by one to the growing spanning tree, provided they do not form a cycle.

Prim's algorithm is another greedy algorithm that finds an MST by starting from an arbitrary vertex and growing the MST one edge at a time by adding the smallest edge that connects a vertex in the growing MST to a vertex outside it.

3.1.1 Basic of Trees

A connected graph without any circuits is called a tree.

Theorem 1:

Prove that there is one and only one path between every pair of vertices in a tree T.

Proof:

T is connected, hence every pair of vertices in T must have at least one path connecting them.

Assume for the moment that there are two vertices between T's "a" and "b". There are two different routes.

It is impossible for a tree to have a circuit that is contained in the union of these two pathways. Every pair of vertices has a unique path as a result.

Theorem 2:

Prove that in a graph G there is one and only one path between every pair of vertices G is a tree.

Proof:

In a graph G, there is one and only one path between every pair of vertices.

$\Rightarrow$ G is connected

Suppose, G has a circuit.

$\Rightarrow$ There is atleast one pair of vertices a, b such that, there are two distinct paths between a and b.

G cannot have a circuit since there is only one path connecting each pair of vertices in G.

G is therefore a tree.

Theorem 3:

Prove that a tree with 'n' vertices has n-1 edges

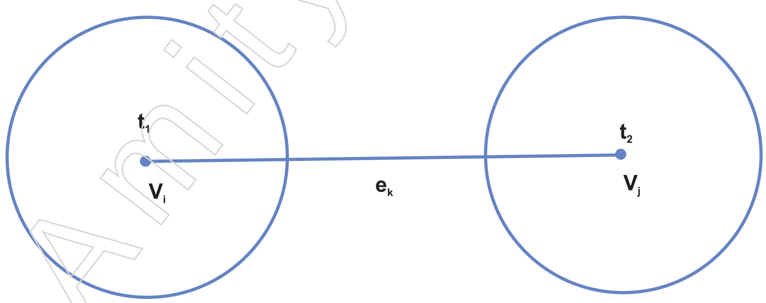
Proof:

The Theorem will be proved by induction on the number of vertices.

It is easy to see that the theorem is true for n = 1,2,3.

Assume that the theorem for all trees with fewer than n vertices.

Let us now consider a tree T with n vertices.



T

In T let ek be an edge with end vertices V_i and V_j .

According to theorem, there is no other path between V_i and V_j except e_k.

Deletion of e_k from T will disconnect the graph as shown in figure.

Notes

Further more $T\text{-}e_k$ consist of exactly two components, and since there were no circuits in T to begin with, each of these component is a tree.

Let the number of vertices of tree t_1 and t_2 be n_1 and n_2 respectively such that

$n_1+n_2=n$

Also $n_1<n$ and $n_2<n$

By induction hypothesis, the number of edges in t_1 and t_2 are n_1-1 and n_2-1 respectively.

Thus $T\text{-}e_k$ consist of

$n_1-1+ n_2-1= n_1 + n_2-2$

$\Rightarrow$ n-2 edges

Therefore T has exactly n-1 edges.

Theorem 4:

Prove that a connected graph with ‘n’ vertices and ‘n-1’ edges is a tree.

Proof:

Let G be a connected graph with ‘n’ vertices and ‘n-1’ edges

We show that G contains no circuits

Assume to the contrary that G contain circuit. Remove an edge from a circuit so that, the resulting graph is again connected.

Continue this process of removing one edge from one circuit at a time till the resulting graph ‘H’ is a tree.

As H has ‘n’ vertices so number of edges in H is n-1. Now the number of edges in G is greater than the number of edges in H.

So, n-1 > n-1, which is not possible

Hence G has no circuits and therefore is a tree.

Definition:

A graph G is said to be minimally connected if removal of anyone edge from it

disconnects the graph. Clearly a minimally connected graph has no circuits.

Theorem 5:

Prove that a graph is a tree if it is minimally connected.

Proof:

Let the graph G be minimally connected.

Then G has no circuits and therefore is a tree.

Conversely,

Let G be a tree.

Then G contains no circuits and deletion of any edge from G disconnect the graph.

Hence G is minimally connected.

Theorem:

Prove that a Graph G with n vertices, n-1 edges no circuits is connected.

Proof:

Suppose that there exist a circuitless graph G with `n' vertices and (n-1) edges which is disconnected.

In that case, **G** will consist of two or more circuitless component. Without loss of generality, Let G consist of two components g_1 and g_2.

Add an edge **e** between a vertex v_1 in g_1 and v_2 in g_2.

In G, there was no path between v_1 and v_2, so adding e did not result in the creation of a circuit.

This $G \cup e$ is an impossible circuitless connected graph with n vertices and n edges, or a tree.

G is hence connected.

Note:

- G is both circuitless and linked (or)
- G has (n-1) edges and is connected (or)
- G has (n-1) edges and is circuitless (or)
- Every pair of vertices in G has exactly one path connecting them (or)
- The graph G has low connectivity.

Definition: If the degree of a vertex in a tree is one, it is said to be pendent.

Theorem:

Show that there are always at least two pendent vertices in any tree with two or more vertices.

Proof:

Suppose that the tree T has `n' vertices.

Case: (i)

n = 2

T is of the form

Hence we have two pendent vertices.

Case: (ii)

n>2

Let T have n+1 edges.

As we are aware,

Each edge adds two degrees. In the graph, there are 2(n-1) degrees.

n vertices have a degree of 2(n-1). There cannot be a single degree zero vertex in a tree.

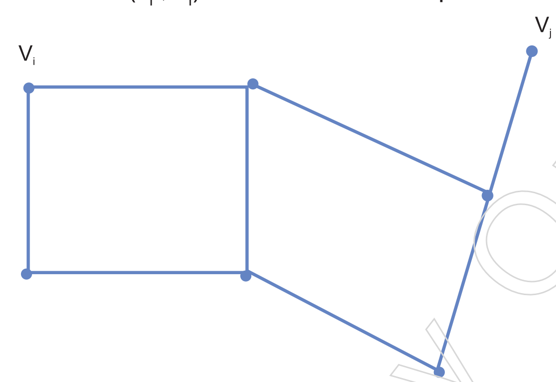
Consequently, there are at least two degree-one vertices.

Notes

Consequently, a tree must have at least two pendent vertices.

Definition:

The length of the shortest path between two vertices v_i and v_j in a connected graph G is represented by the distance d (v_i, v_j) between those two points.



d(V_i,V_j)=3

Definition:

Consider x to be a nonempty set. It is argued that the function "t" is the metric of a "x." If it meets the requirements listed below:

(i) Non-negativity f(x,y)≥0 and f(x,y)=0 iff x=y.

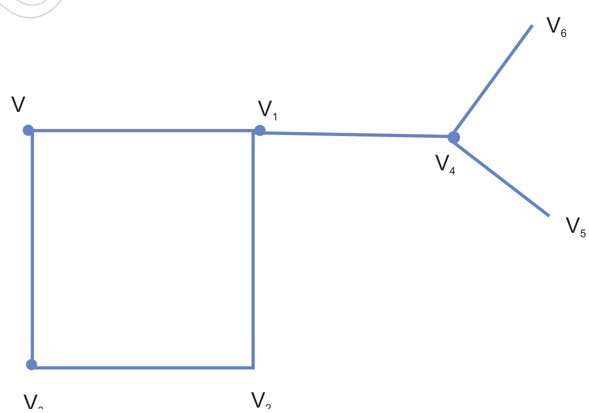
(ii) Symmetry : f(x,y) = f(y,x)

(iii) Triangle inequality: f(x,y) ≤ f(x,z) if (z,y) for some z.

Definition:

The distance between a vector V and the vertex in graph G that is furthest from V is known as the vector's eccentricity, or E(v).

(i.e,) E(v) = max d (v,v_i)
$V_i \in G$



d(v,v_1) =1, d(v,v_3) =1, d(v,v_2) =2 , d(v,v_4) =2 , d(v,v_5) =3

Therefore E(v) =3

Definition:

In graph G, the vertex with the least eccentricity is referred to as the center of G.

E(d) =3 E(b) =1

E(c) =2 E(a) =2

Therefore ‘b’ is the center.

Definition:

We call two centers on a graph G that have two centers, bicenters.

Theorem:

Show that all trees have one or two centers.

Proof:

Case: (i)

T has one vertex.

The vertex itself its center.

Case: (ii)

T has two vertices.

T is of the form.

Therefore ‘T’ has two centers.

Case: (iii)

T has more than two vertices by theorem,

There are atleast two pendent vertices.

Let v be any vertex in T.

The maximum distance, max d (v, v_i) from given vertex v to any other vertex v_i occurs only when v_i is a pendent vertex.

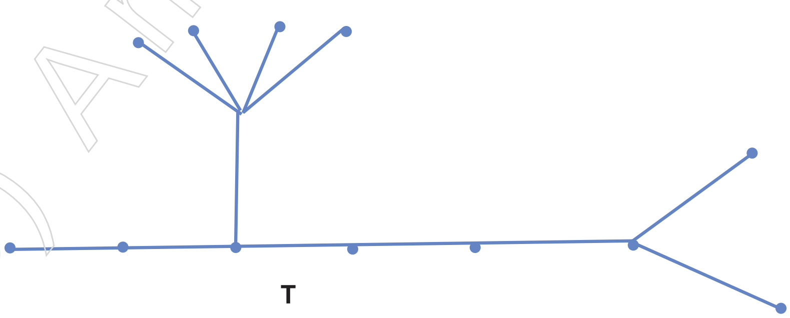
Delete all the pendent vertices from T.

The remaining graph T' is still a tree.

T' is also true with eccentricity of each vertex in one less than of T.

The center of T' are the center of T, only by deleting all the pendent vertices in T'.

We get another tree T” with the same center.



T

Notes

Continuing in this manner until either a vertex or an edge remains.

The tree has a single center if there is just one vertex. The tree has two centers, or the edge's end vertices, if there is only one edge.

Thus, there are one or two centers on the tree.

Remark:

If a tree T has two centers, the two centers must be adjacent.

Definition:

An eccentricity of a center in a tree defined as the radius of tree.

Definition:

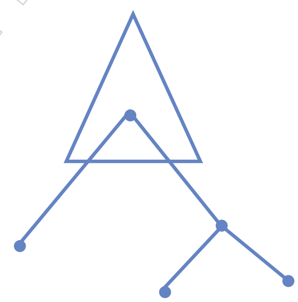
The diameter of a tree T is defined as the length of longest path in tree.

3.1.2 Binary Trees

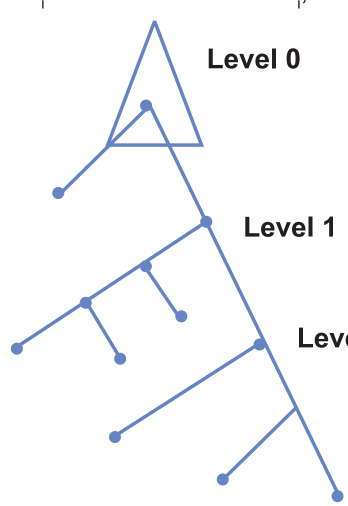
A binary tree is characterised as having precisely one degree two vertex with the rest vertices being either tree one or tree three.

A binary tree always has an odd number of vertices.

An internal vertex is any non-pendent vertex in a tree.



In a binary tree a vertex v_i is said to be at **level** l_i, if v_i at a distance of l_i from the root.



Notes

The maximum level l_{max} of any vertex in binary tree is called the height of a tree.

Note:

Minimum possible height of an n-vertex binary tree is min

$l_{max} = \lceil \log_2(n+1)-1 \rceil$ where $\lceil n \rceil$ denotes the smallest integer ≥ n.

Max l_{max} =n-1 /2.

Every pendent vertex (v_j) of a binary tree has a positive real number w_j associated with it, which is the definition of a weighted path length.

Given $w_1, w_2, \ldots, w_n$ the problem is to construct a binary tree that minimize.

$\sum w_j l_j$ where l_j is the level of pendent vertex v_j, and the sum is taken over all pendent vertices.

3.1.3 Rooted Trees and Path Length in Rooted Trees

Rooted Trees

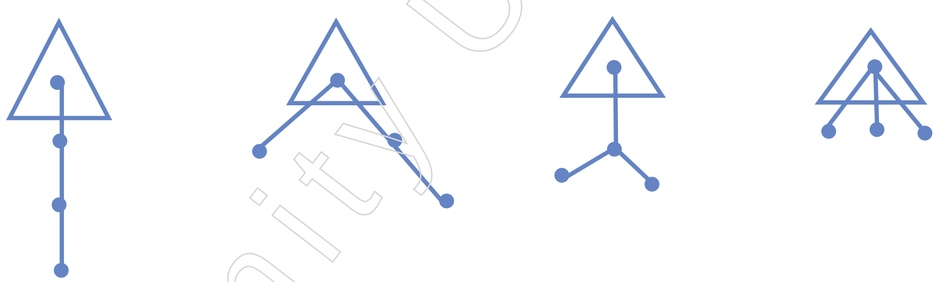
A tree in which one vertex called the root is distinguished from all the others is called a rooted tree.

A directed tree is referred to as a rooted tree if it contains a single node or vertex called the root, whose incoming degrees are zero and all other vertices have incoming degrees of one.

Note:

1. A rooted tree is an empty tree since it lacks nodes.
2. A rooted tree is a single node that has no offspring.

Non-rooted trees are called free tree.

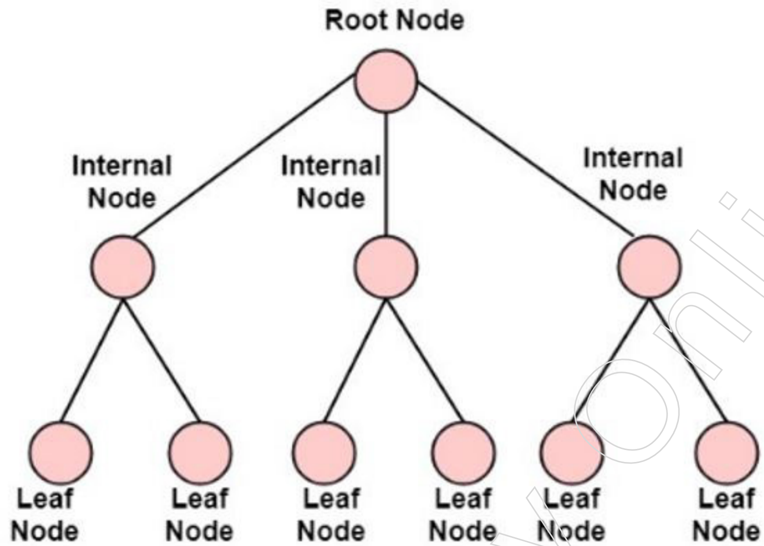


Path length of a rooted tree

The total length of all the paths from a tree's root to all of its nodes is known as the path length. The average path length from the root to a node is obtained by dividing the total number of nodes in the tree by the length of the path.

The length of the path from the root to the element's node determines how many comparisons are required to identify an element in a search tree, hence the average path length of a tree indicates how many comparisons are typically made during an insertion, deletion, or member operation on the tree.

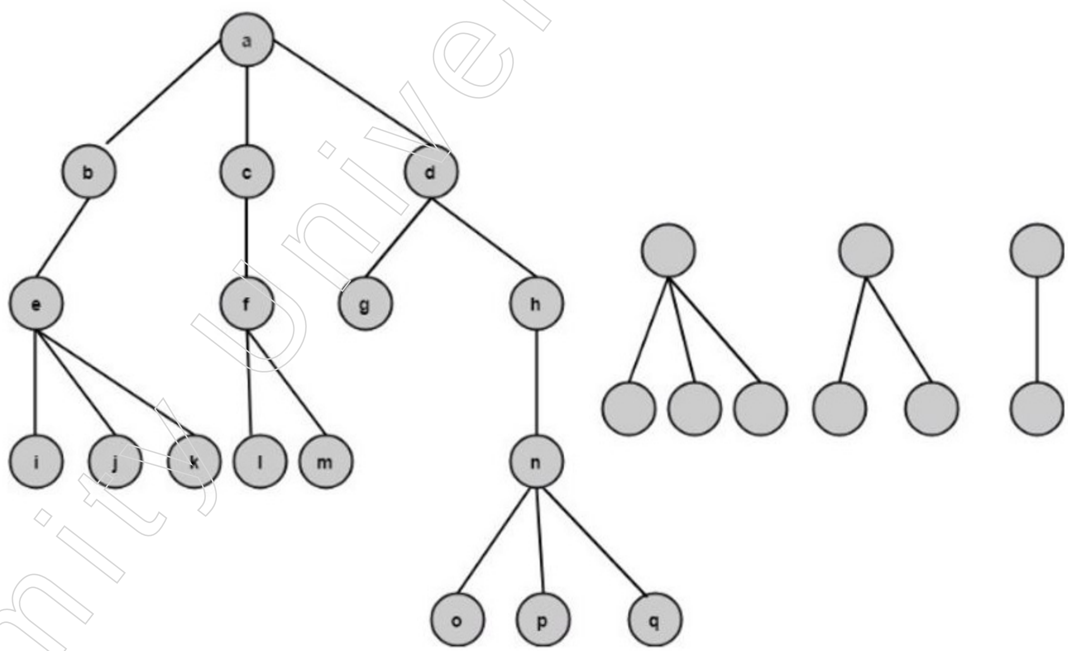
Notes



Path length of a Vertex: In a rooted tree, the number of edges in the path from the root to the vertex is the vertex's path length.

Example:

Determine the paths of the nodes b, f, l and q as indicated in fig:



Answer: Node B has a path length of one.

Node f's path length is two.

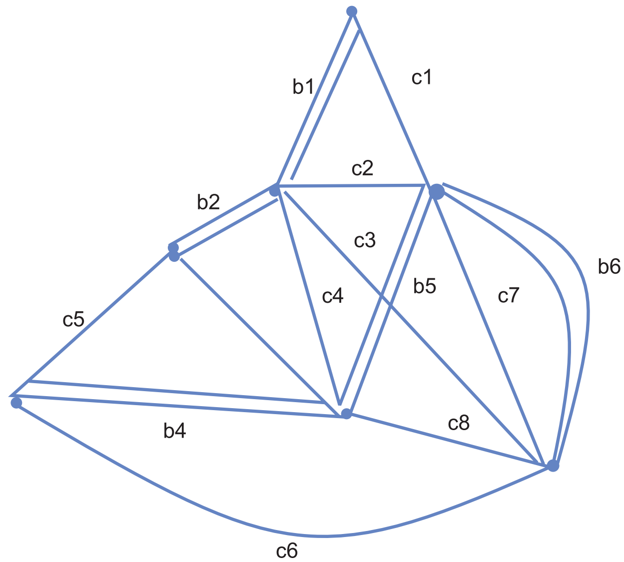
Node L's path length is three

The node Q has a path length of four.

3.1.4 Spanning Trees and Spanning Trees of a Weighted Graph

A tree t that is a subgraph of a linked graph G and contains every vertex in G is referred to as a spanning tree of G. The term "skeleton" or "scaffolding" of G can also apply to spanning trees.

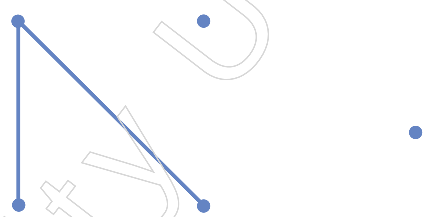
A forest is an area with trees in it.



The complement of the complete graph k_n is the empty graph with n vertices.

G

G'



G-G'

Theorem:

Every connected graph has atleast one spanning tree.

Proof:

Let G be a connected graph.

Case: (i)

G has no circuits.

Since G is connected. G is a tree

Therefore, G itself is a spanning tree.

Notes

Notes

Case: (ii)

G has circuits.

If an edge is removed from a circuit in G, a linked graph is still left behind.

Repeat the process until you reach the edge of the final circuit in the graph that contains all of G's vertices if there are still circuits in the remaining graph.

It is therefore a spanning tree.

Branch: A spreading tree's edge T is known as a branch of T.

Chord: A chord is an edge of G that is not a part of a certain spanning tree T.

A chord can also be referred to as a connection or a tie.

Note:

The chord called the (also known as the tie set or cotree) is a group of chords that make up the subgraph T.

Theorem:

With respect to any of its spanning trees, a connected graph of n vertices and e edges has n-1 branches and e-n+1 chords.

Proof:

Let G be a graph with `n' vertices and e edges, suppose that T is any spanning tree in G.

$\Rightarrow$T is a tree with n vertices.

$\Rightarrow$T has n-1 edges.

The number of branches in T is n-1.

Therefore, the remaining e-(n-1) are chords of T.

There are e-n+1 chords of T

Definition: Let G be a graph with 'k' -components 'n' vertices, 'e' edges.

Rank(r)=n-k

Nullity(μ)=e-n+k

Note:

Rank of G = number of branches in any spanning tree of G.

Nullity of G = number of chords in G.

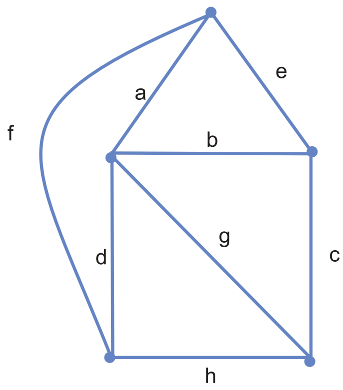
Rank + Nullity = number of edges in G.

The cyclomatic number or first Betti number of a graph is another term for its nullity.

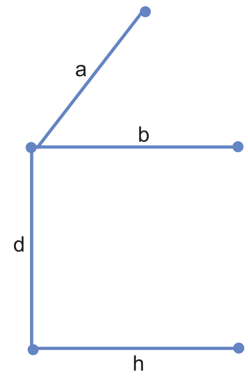
Definition:

The distance between two spanning tree T_i and T_j of a graph G is defined G as the no of edges of G present in one tree but not in other.

The distance may written as $d(T_i,T_j)$

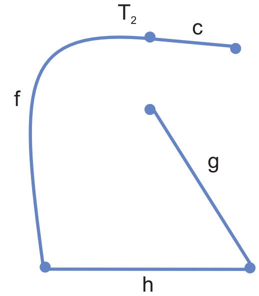


G



T_1

T_2



$d(T_1,T_2)=3$

Remark:

A metric is the distance of a graph's spanning tree from itself.

The distance between the spanning tree of a graph is a metric.

ie.,) i) $d(T_i,T_j) \geq 0$ $d(T_i,T_j) = 0$ iff $T_i = T_j$

ii) $d(T_i, T_j) = d(T_i,T_j)$

iii) $d(T_i,T_j) \leq d(T_i,T_e) + d(T_k,T_j)$

Remark:

Max $d(T_i,T_j)$ =1/2 max $N(T_i \oplus T_j)$

$\leq r$, the rank of G

Max $d(T_i,T_j) \leq \mu$

Max $d(T_i,T_j) \leq \min(\mu,r)$

Definition:

For a spanning tree T_0 of a graph G, let max $d(T_i,T_j)$denote the maximal distance between T_0 and any other spanning tree of tree. Then T_0 is called a **central tree** of G if max $d(T_o,T_i) \leq d(T,T_j)$ for every tree T of G.

3.1.5 Fundamental Circuits

Circuits:

Theorem:

Prove that the connected graph G's subgraph C is a circuit if and only if

i) Each of G's cospanning trees has at least one connection in C and

ii) If D is a subgraph of C and D=C, then a cospanning tree exists, with none of its links being in D.

Notes

Proof:

First, let's look at the scenario in which C is a circuit.

Then, (i) is true since C contains at least one link from each cospanning tree. D is clearly a forest rather than a circuit if it is a genuine subgraph of C.

Next, we can add to D to make it a spanning tree of G, meaning that D does not include any links in T and that some spanning tree T of G includes D.

Consequently, (ii) is accurate.

The situation in which both (i) and (ii) are true is now under consideration.

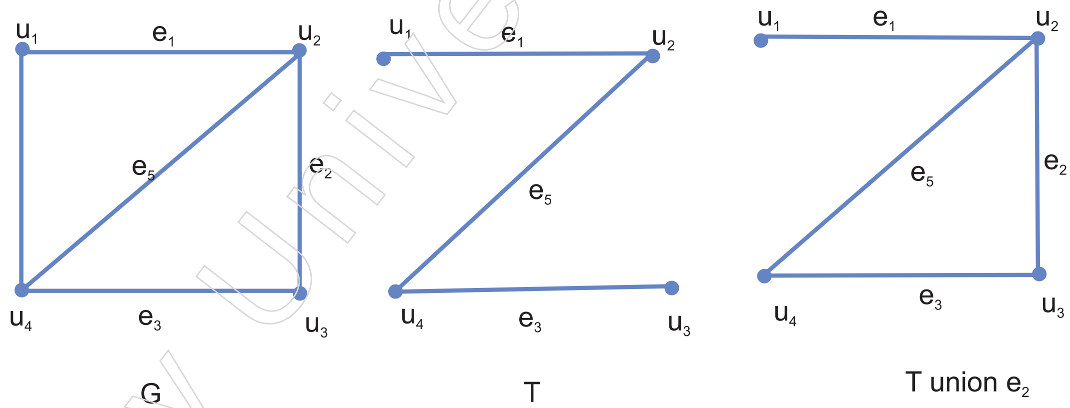
Given that C is a forest otherwise and that we can supplement it to make it a spanning tree of G, then there must be at least one circuit in there.

In C, we take a circuit C′. As a result of C′ being a circuit with a link from each cospanning tree, (ii) being true, C′=C is false.

Therefore, C=C′ is a circuit.

Fundamental Circuits:

Any spanning tree T in a linked graph G is allowed. A fundamental circuit is one that is established when a chord is added to a spanning tree; it can be created by adding any one chord to T.



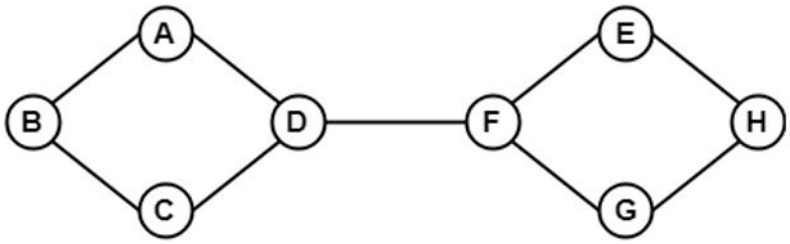
3.1.6 Cut Sets and Cut Vertices

Connectivity: A foundational idea in graph theory is connectivity. It indicates the connectedness or disconnectedness of a graph. It is impossible to move from one vertex to another in a graph without connection.

- If a path connects each pair of vertices in a graph, the graph is said to be connected. There must be a path to travel from each vertex to every other vertex. This is referred to as a graph's connectedness.
- If there are several disconnected vertices and edges in a graph, it is considered disconnected.
- Network applications, transportation network routing, network tolerance and other areas require an understanding of graph connection theories.

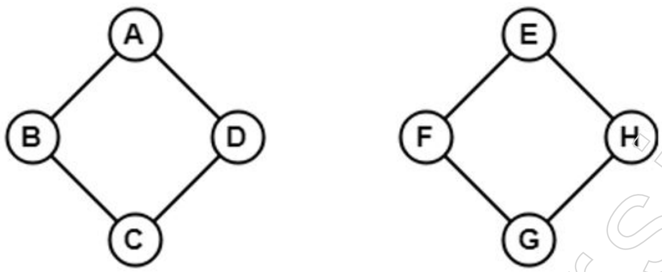
Example:

Notes



It is possible to go from one vertex to another in the example above. Here, we can follow the line B -> A -> D -> F -> E -> H to journey from vertex B to H. It is therefore a connected graph.

Example:



Since there is no direct or indirect path connecting vertex B and H in the example above, it is not possible to cross from one to the other. It is therefore a disconnected graph.

Let's examine a few fundamental ideas in connectivity.

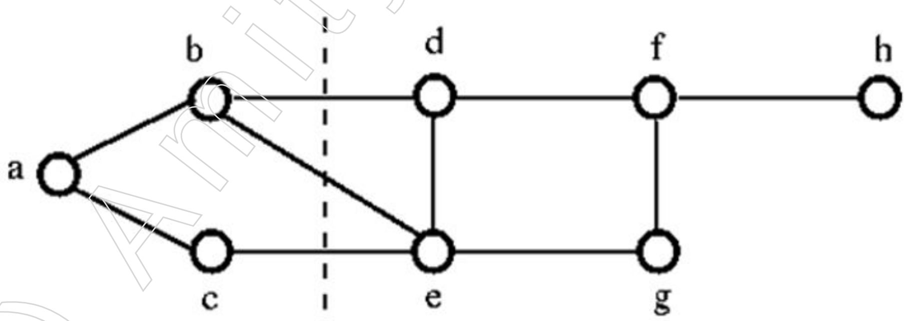
Cut sets:

A set S of edges in a connected graph G that have the following characteristics is called a cut set:

G is disconnected when all of S's edges are eliminated.

G remains connected even if some (but not all) of S's edges are removed.

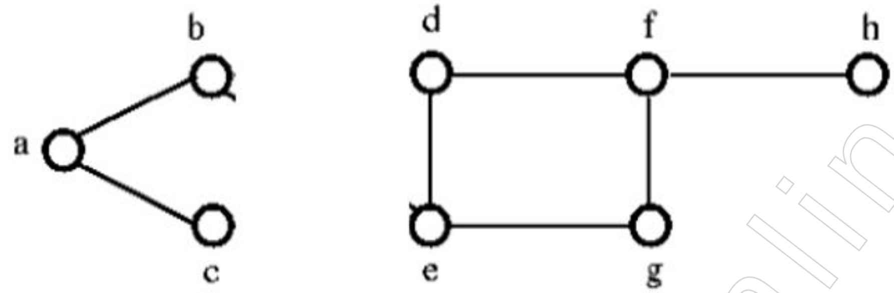
Example



We must eliminate the three edges in the graph G above in order to disconnect it. For example, bd, be and ce. It cannot be disconnected with only two of its three edges removed. {bd, be, ce} is a cut set as a result.

Following the removal of the chopped set, the graph above would like this:

Notes



Theorem:

The edge set F of the connected graph G is a cut set of G if and only if

(i) F includes at least one branch from every spanning tree of G and

(ii) $H \subset F$, then there is a spanning tree none of whose branches is in H.

Proof:

Let us first consider the case where F is a cut set.

Then, (i) is true.

If $H \subset F$ then $G-H$ is connected and has a spanning tree T. This T is also a spanning tree of G.

Hence, (ii) is true.

Let us next consider the case where both (i) and (ii) are true.

Then $G-F$ is disconnected. If $H \subset F$ there is a spanning tree T none of whose branches is in H.

Thus, T is a subgraph of $G - H$ and $G - H$ is connected.

Hence, F is a cut set.

Theorem:

Prove that a circuit and a cut set of a connected graph have an even number of common edges.

Proof:

We choose a circuit C and a cut set F of the connected graph G.

$G-F$ has two components $G_1= (V_1, E_1)$ and $G_2= (V_2, E_2)$

If C is a subgraph of G_1 or G_2=, then the theorem is obvious because they have no common edges.

Let us assume that C and F have common edges. We traverse around a circuit by starting at some vertex v of G_1.

Since we come back to v, there has to be an even number of edges of the cut V_1, V_2 in C.

Cut vertices:

A cut-vertex is a single vertex that, when removed, causes a graph to become disconnected.

Consider a connected graph, G. If G-v, or Remove v from G, yields a disconnected graph, then a vertex v of G is referred to as a cut vertex of G.

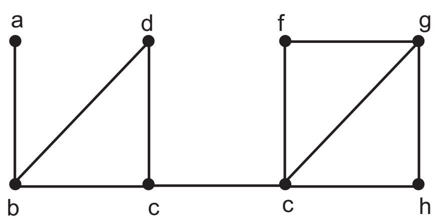
A graph will split into two or more graphs when a vertex is removed. We refer to this vertex as a cut vertex.

Note: Let G be a graph with n vertices.

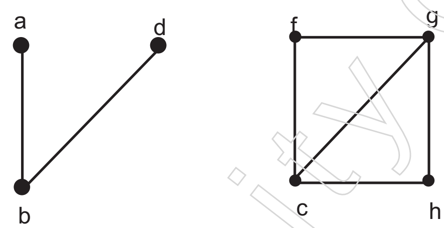
- There can be a maximum of (n-2) cut vertices in a connected graph G.
- If a cut vertex is removed, a graph could become unconnected.
- A graph may have at least one more component when a vertex is removed.
- A cut vertex is any non-pendant vertex in a tree.

Example1:

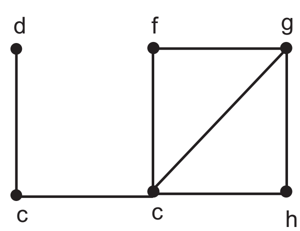
Original graph:



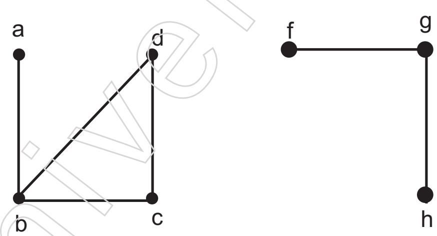
Vertex c is a cut vertex:



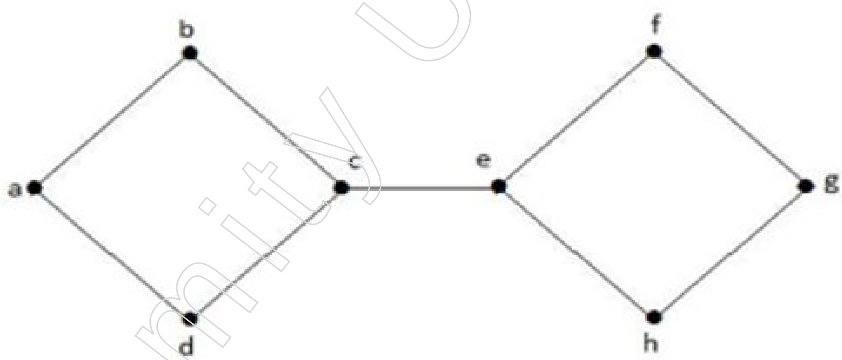
Vertex b is a cut vertex:

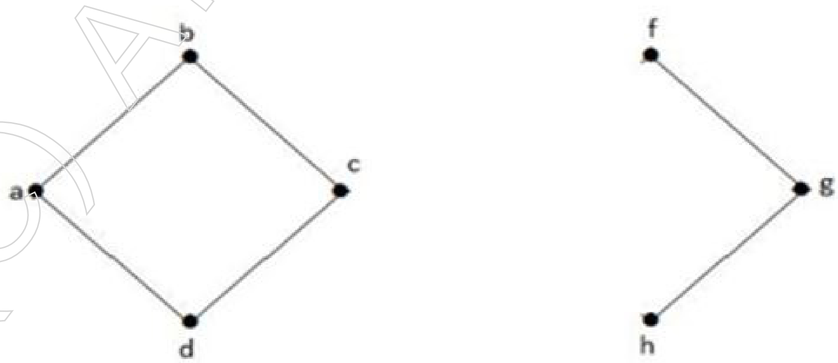


Vertex e is a cut vertex:



Example 2





Notes

Vertex 'e' in the graph above is a cut-vertex. The above graph will become a disconnected graph at the removal of vertex 'e'.

3.1.7 Fundamental Cut Set

Theorem:

The branches of T whose corresponding fundamental cut set includes c precisely create a fundamental circuit corresponding to links of the cospanning tree T∗ of a connected graph.

Proof:

There exists a fundamental circuit C that corresponds to link c of T∗. The other edges $b_{1, \ldots}, b_k$, of C are branches of T.

We denote l_i as the fundamental cut set that corresponds to branch b_i. Then, b_i is the only branch of T which is in both C and l_i.

On the other hand, c is the only link of T∗ in C. We know that the common edges of C and l_i are b_i and c, equivalently c is an edge of l_i.

Then, we show that there is no c in the fundamental cut sets $l_{(k+1)}, \ldots, l_{(n-1)}$ that correspond to the branches $b_{(k+1), \ldots}, b_{(n-1)}$ of T .

For instance, if c were in $l_{(k+1)}$, then the fundamental cut set $l_{(k+1)}$ and the circuit C would have exactly one common edge.

So c is only in the fundamental cut sets $l_{i, \ldots}, l_k$

Theorem:

In a connected graph, the fundamental cut set corresponding to branch b of the spanning tree T is made up of all the links in T∗ whose corresponding fundamental circuit contains b.

Proof:

Suppose that I is a fundamental cut set corresponding to T's branch b. Other edges $c_{1,\ldots}, c_k$ of I are links of T ∗.

Let c_i denote the fundamental circuit that corresponds to c_i. Then, c_i is the only link of T∗ in both I and c_i .

On the other hand, b is the only branch of T, the common edges of I and c_i are b and c_i, in other words, b is an edge of c_i .

Additionally, we demonstrate that the basic circuits $c_{(k+1), \ldots}$ $c_{(m-n+1)}$ corresponding to the links $c_{(k+1), \ldots}$, $c_{(m-n+1)}$ do not include b.

For example, if b were in $c_{(k+1)}$, then the fundamental circuit $c_{(k+1)}$ and the cut set I would have exactly one common edge. Hence, the branch b is only in fundamental circuits $c_{1, \ldots}$, c_k.

3.2 Minimum Spanning Tree

The total weight of all the edges in the tree equals the cost of the spanning tree. One may find numerous spanning trees. The minimum spanning tree is the spanning tree that has the lowest cost out of all the spanning trees. Multiple minimum spanning trees may also exist.

Notes

3.2.1 Minimum Spanning Tree (MST): Concept

A subgraph that is a tree that joins all of the vertices of a connected and undirected graph is called a spanning tree. Multiple spanning trees are possible for a single graph. A tree with V vertices and V-1 edges is said to be a spanning tree. In a spanning tree, every node can be accessed from every other node.

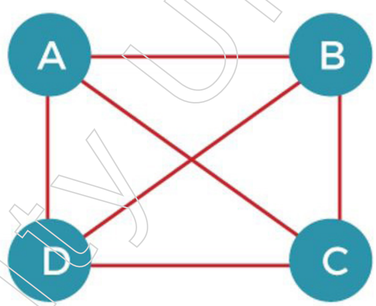
A Minimum Spanning Tree (MST) or minimum weight spanning tree for a weighted, linked, undirected graph is a spanning tree whose weight is less than or equal to the weight of every other viable spanning tree. The total weight assigned to all of a spanning tree's edges determines the tree's weight. In other words, MST is the spanning tree with the least weight among all the spanning trees in the given graph.

The minimum spanning tree's general properties are:

- The spanning tree gets disconnected if we remove any of its edges. As a result, none of the spanning tree's edges can be eliminated.
- A loop is created in the spanning tree if we add an edge. As such, we are unable to increase the spanning tree's edge count.
- If every edge in a graph has a different weight, then there can only be one minimum spanning tree. There may be more than one minimum spanning tree if the edge weight is ambiguous.
- An undirected graph can have n^{n-2} spanning trees in total.
- There is at least one spanning tree present in every linked and undirected graph.
- There isn't a spanning tree in the unconnected graph.
- To create a spanning tree, we can eliminate the maximum (e-n+1) edges from a complete graph.

Let's look at an example to better grasp the final characteristic.

Examine the entire graph, which is provided below:



The number of spanning trees that can be made from the above complete graph equals to $n^{n-2} = 4^{4-2} = 16$.

Therefore, 16 spanning trees can be created from the above graph.

The maximum number of edges that can be removed to construct a spanning tree equals to e-n+1 = 6 - 4 + 1 = 3.

3.2.2 Edge and Vertex Properties in MST

A spanning tree with a weight less than or equal to the weight of every other feasible spanning tree is known as a Minimum Spanning Tree (MST) or minimum weight spanning tree for a weighted, linked, undirected graph. The total weight assigned to all of a spanning tree's edges determines the tree's weight. In other words, MST is the spanning tree with the least weight among all the spanning trees in the given graph.

Notes

Minimum Spanning Tree requirements:

- It can't create a cycle, meaning no edge can be traversed more than once.
- No alternative spanning tree with a lower weight may exist.

Properties of MST:

1. Possible Multiplicity:

   G(V, E) is a graph if and only if each spanning tree of graph G has (V – 1) edges, where V and E are the graph's vertice and edge counts, respectively. That is, the spanning tree does not contain the edges (E – V + 1). A minimum spanning tree with the same weight may exist in multiple instances. A graph's minimal spanning trees are all those whose edge weights are equal.

2. Cut Property:

   For every cut C of the graph, an edge is said to be a member of all the MSTs of the graph if its weight in the cut-set of C is strictly less than the weights of every other edge in the cut-set of C.

3. Cycle Property:

   Any edge E in cycle C of the graph cannot be a member of an MST if its weight exceeds the sum of the weights of all other edges in cycle C. Due to its highest weight among all the edges in cycle ABD, edge BD in the aforementioned figure cannot be found in any minimal spanning tree.

4. Uniqueness:

   There can only be one, that is, a unique minimum spanning tree, if every edge has a different weight.

5. Subgraph with Minimum Cost

   The smallest spanning tree for any conceivable combination of spanning trees needs to have the lowest weight feasible. Nevertheless, there can be more spanning trees that have the same weight as the least spanning tree; they could all be regarded as minimum spanning trees.

Minimum Cost Edge: An edge is included in any MST if it is a graph's unique minimum cost edge. For instance, the edge AB (with the least weight) in the preceding diagram is always included in MST.

Since every spanning tree is minimally acyclic, adding a new edge will cause the tree to become cyclic.

Minimum Spanning Tree (MST) algorithms include the following:

1. The Prim's Algorithm
2. The Kruskal's Algorithm

3.2.3 Kruskal's Algorithm for Finding MST

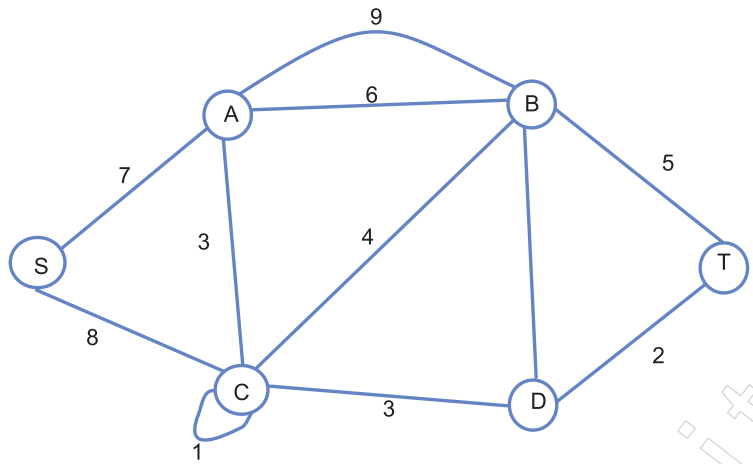
The minimal spanning tree for a linked weighted graph can be found using Kruskal's Algorithm. The primary goal of the algorithm is to identify the subset of edges that allow us to navigate across each vertex in the graph. It adopts a rapacious strategy that looks for the best option at every turn.

In order to process the edges, Kruskal's algorithm takes the weight values of each edge in order, from least to greatest. It then adds V-1 edges to the cycle, terminating after each edge that does not create a cycle. The MST, a single tree, eventually emerges from the forest of trees formed by its boundaries.

Notes

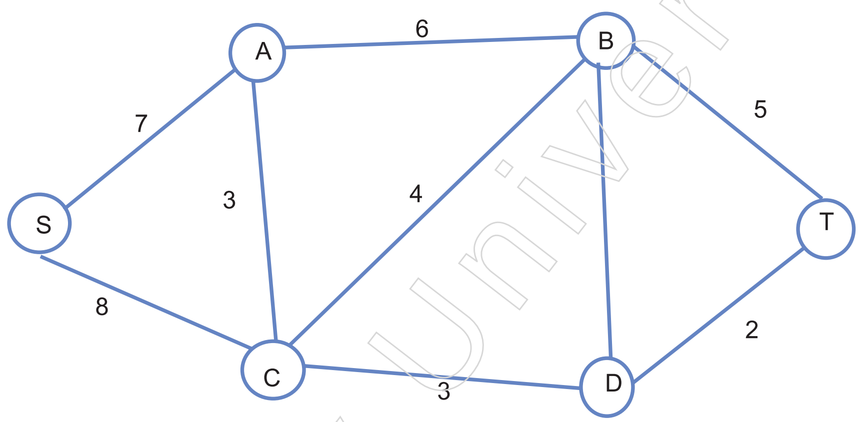
Example:

Using the Kruskal technique, determine the minimal spanning tree.



Step 1:

Eliminate all parallel edges and loops.



Step: 2

Put all of the edges in ascending weight order.

(i) B →D=2

(ii) D →T=2

(iii) A→C=3

(iv) C→D=3

(v) C→B=4

(vi) B→T=5

(vii) A→B=6

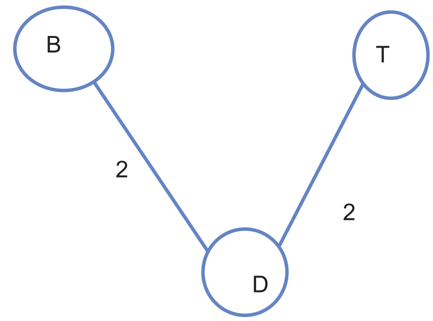
(viii) S→A=7

(ix) S→C=8

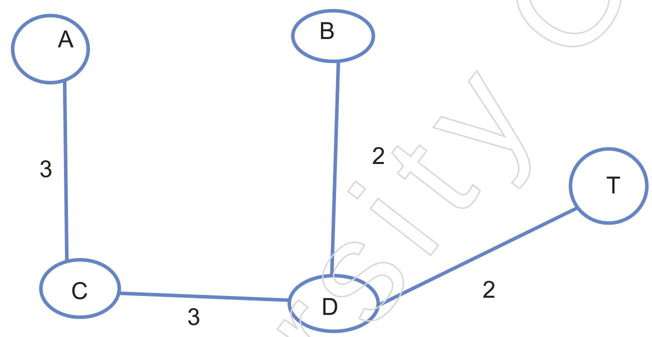
Step: 3

Include the edge with the least weight.

Notes



A→C or C→D are the edges since 3 is the next least expensive.

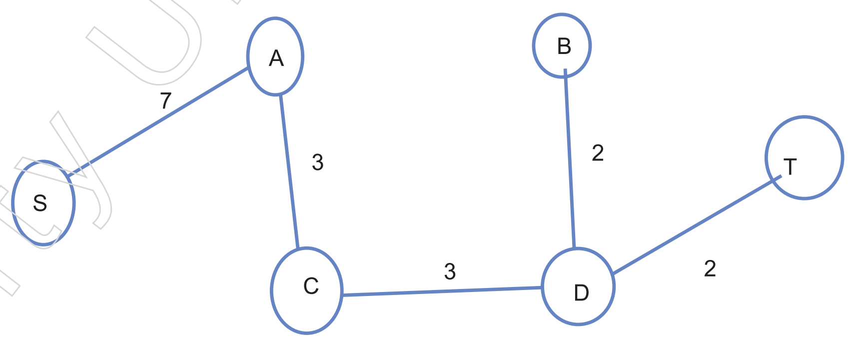


The next lowest cost is 4, but taking 4 closes the circuit, so we reject it to prevent a circuit.

The next lowest cost is hence 5, at which point it also closes the circuit. So throw it away.

The next lowest cost is six, at which point it also closes the circuit. So throw it away.

7 and 8 are the next lowest costs.



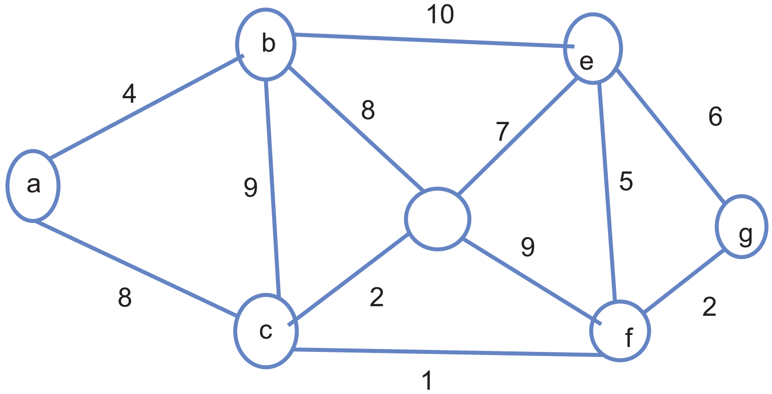
The minimal spanning tree is thus finished.

3.2.4. Prim's Algorithm for Finding MST

At each step, Prim's algorithm adds a new edge to a single expanding tree. Assume that any vertex is a single-vertex tree at first. Then, add V-1 edges to it by always adding the minimum-weight edge (a crossing edge for the cut defined by tree vertices) that joins a vertex on the tree to a vertex that is not yet on the tree.

Example:

Prim's algorithm can be used to find the minimum spanning tree.



Solution:

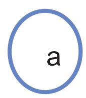
Step: 1

Consider the vertex "a."

S={a}

Step: 1

Let us take the vertex 'a'



S={a}

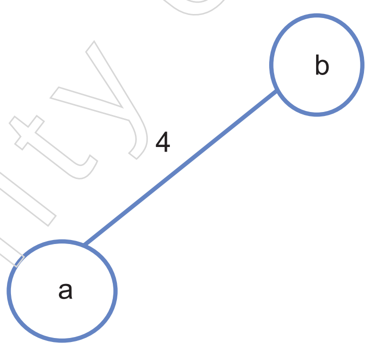
The remaining vertices={b,c,d,e,f,g}

Starting vertex A={ϕ}

The lighted edges={a,b}

Step: 2

Let S={a,b}



S={a,b}

The remaining vertices={c,d,e,f,g}

Starting vertex A={{a,b}}

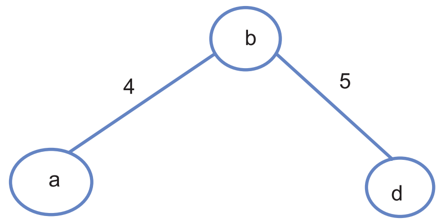
The lighted edges are b→d or a→c and both have the weight 8.

But we have to visit the weight from an adjacent to the unvisited vertex

So we have to select the vertex 'b'

Notes

Step: 3



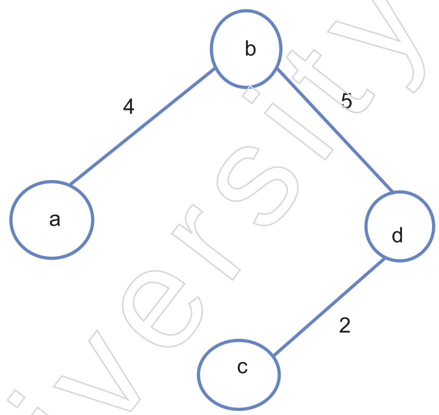
S={a,b,d}

The remaining vertices={c,e,f,g}

Starting vertex A={{a,b},{b,d}}

The lighted edges are d→c

Step: 4



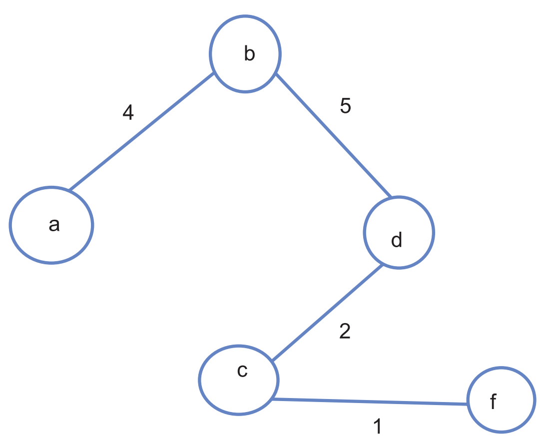
S={a,b,c,d}

The remaining vertices={e,f,g}

Starting vertex A={{a,b},{b,d},{d,c}}

The lighted edges are c→f

Step: 5



S={a,b,c,d,f}

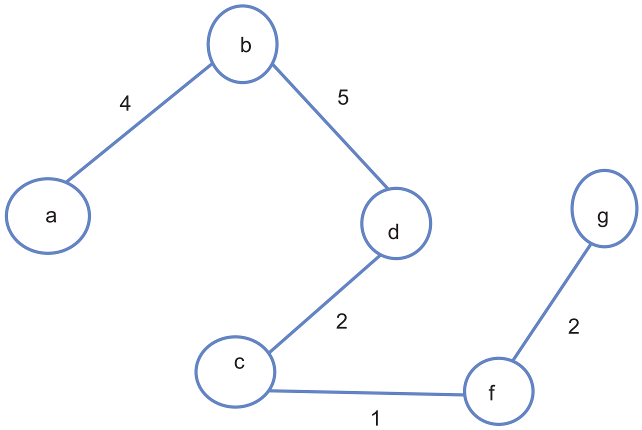
The remaining vertices={e,g}

Starting vertex A={{a,b},{b,d},{d,c},{c,f}}

The lighted edges are f→g

Notes

Step: 6



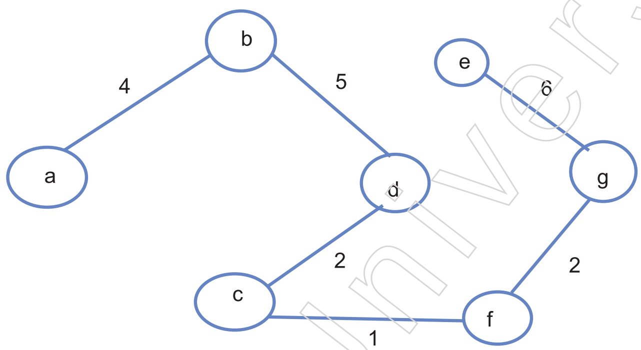
S={a,b,c,d,f,g}

The remaining vertices={e}

Starting vertex A={{a,b},{b,d},{d,c},{c,f},{f,g}}

The lighted edges are g→e

Step: 7



S={a,b,c,d,e,f,g}

The remaining vertices={ ϕ }

Starting vertex A={{a,b},{b,d},{d,c},{c,f},{f,g},{g,e}}

The lighted edges are g→e

We have now finished the minimum spanning tree.

Summary

- Each pair of vertices in a tree must have at least one path between them, as it is a connected graph without any circuits.
- A connected graph with 'n' vertices and 'n-1' edges is recognised as a tree.
- The extent of the shortest path between two vertices v_i and v_j in a connected graph G is indicated by the distance d (v_i, v_j).
- The eccentricity E(v) of a vector V in a graph G in the distance from V to the vertex farthest from V in G, i.e., E(v) = max d (v, v_i)
- If a directed tree contains a single node or vertex, the root, whose incoming degrees are zero, and all other vertices have incoming degrees of one, it is referred to as a rooted tree.
- The distance between two spanning trees T_i and T_j of a graph G is defined as the

Notes

number of edges of G that are present in one tree but not in the other.

- A fundamental circuit is a circuit that is constructed by adding a chord to a spanning tree. This circuit is created by adding any one chord to T.
- A cut set is a set S of edges in a connected graph G that satisfy the following properties:
  - ❖ When every edge in S is removed, G is disconnected.
  - ❖ Even if some (but not all) of S's edges are eliminated, G is still connected.
- The fundamental circuit, which corresponds to the links of the cospanning tree T* of a connected graph, is precisely the set of branches of T whose corresponding fundamental cut set includes c.
- The cost of the spanning tree is equivalent to the sum of the weights of all the edges in the tree. Among all the spanning trees, the minimum spanning tree is identified as the one with the lowest cost.
- The Minimum Spanning Tree for a specific graph can be determined by employing Prim's method. Prim's approach enables the reduction of the total edge weights by identifying the subset of edges that contain each vertex in the graph.

Glossary

- Pendant: In a tree, the vertex is said to be pendent if its degree is one.
- Eccentricity: A vertex with minimum eccentricity in graph G is called a centre of G.
- Binary tree: It is defined as a tree in which there is exactly one vertex of degree two, and each of the remaining vertices is of tree one or three.
- Rooted tree: A tree in which one vertex called the root is distinguished from all the others is called a rooted tree.
- Path length: The total length of all the paths from a tree's root to all of its nodes is known as the path length.
- Subgraph: A tree t is said to be a spanning tree of a connected graph G if T is a subgraph of G and T contains all vertices of G
- Branch: An edge in a spanning tree T is called a branch of T.
- Chord: An edge of G that is not in a given spanning tree T is called a chord.
- A cut-vertex is a single vertex that, when removed, causes a graph to become disconnected.
- Spanning tree: The subgraph of an undirected connected graph is called a spanning tree.
- Kruskal's Algorithm: A linked weighted graph's smallest spanning tree can be found using Kruskal's algorithm. The algorithm's primary goal is to identify the subset of edges that allow us to navigate across each vertex in the graph. It takes a rapacious strategy that looks for the best answer at every turn.

Check Your Understanding

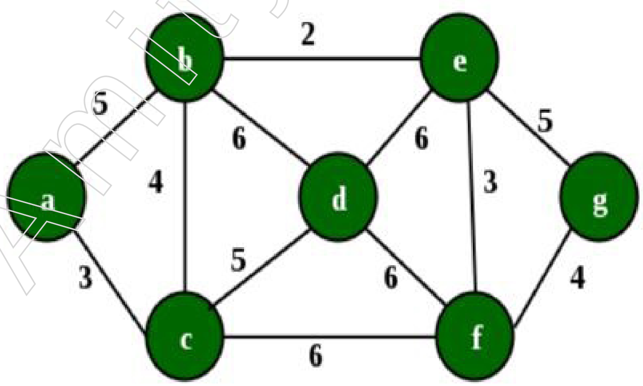
1. In the case of a spanning tree of a graph G, which of the following statements is incorrect?
   a) The tree spans G
   b) It is a subgraph of the G
   c) It contains all the G's vertices
   d) It may be cyclic or acyclic

Notes

2. __________ can be used to solve the Traveling Salesman Problem.
   a) Bellman-Ford algorithm
   b) DFS traversal
   c) Spanning tree
   d) Minimum spanning tree
3. What determines whether a graph vertex is even or odd, according to ____?
   a) Is either odd or even. Its degree might be odd or even
   b) The total number of edges in a graph
   c) A graph's total vertex count can be even or odd
   d) None of the above
4. The binary tree's Breadth First Search traversal is used in __________.
   a) Networks of peers to peers
   b) Cloud-based software
   c) Euler route
   d) Weighted graphs
5. What is the total number of edges of a k− regular graph with n vertices?
   (a) k_n
   (b) kn/2
   (c) k + n
   (d) kn^2

Exercise

1. Prove that G is a tree by observing that there is a single path between each pair of vertices in a graph G.
2. Demonstrate that a graph G is connected if it contains n vertices, n-1 edges, and no circuits.
3. Consider the following graph, using Kruskal's algorithm calculate minimum spanning tree.

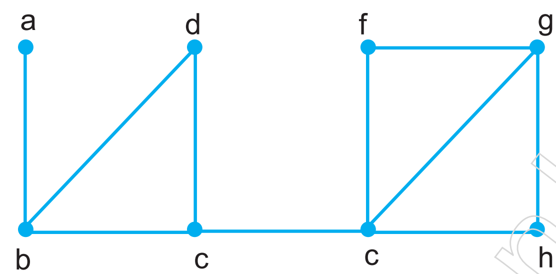


4. Point out the difference between Prim's and Kruskal's algorithm for MST.
5. Prove that in a certain connected graph If and only if every vertex in a graph G has an even degree, then that graph is a Euler graph.

Notes

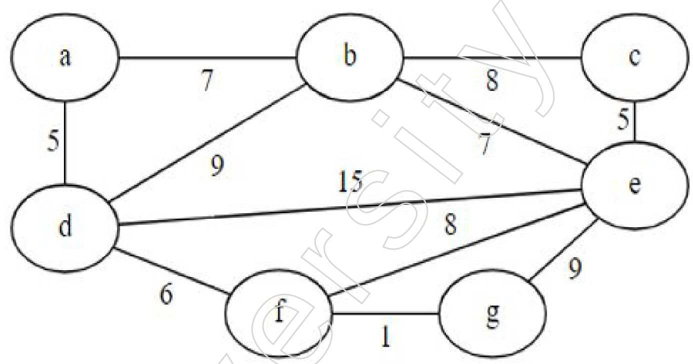
Learning Activities

1. Consider the following graph G:



Let the number of cut vertices is x and the number of cut edges is y. What is the value of x^y ?

2. Calculate the minimum spanning tree of given weighted graph.



Check Your Understanding (Answers)

1. d) 2. d) 3. b) 4. a)
5. c)

Further Readings and Bibliography

1. Dr D.S.C.," Graph Theory and Combinatorics All India", Prism Publications, 1 January 2012.
2. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 2nd Edition, 2000.
3. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009.
4. Harikishan, Shivraj Pundir and Sandeep Kumar, Discrete Mathematics, Pragati Publication, 7th Edition, 2010.

Module -IV: Directed Graph and Planar Graphs

Learning Objectives

At the end of this module, you will be able to:

- Identify the types of Graphs: Directed or Planar Graphs
- Learn about Directed Trees
- Summarise the Network Flows with the Maximum-Flow and Minimum-Cut
- Discuss the Matrix Representation of a Graph
- Define the Planar Graphs: Combinatorial and Geometric Duals
- Infer the features of Kuratowski's Graphs
- Understand the basic criteria on Detection of Planarity
- Define the Thickness and Crossing properties of Graph

Introduction

A directed graph is a type of graph that contains directed edges and is frequently referred to as a digraph. In formal terms, an edge is an unordered pair {v, w} if v and w are vertices, and a directed edge, also known as an arc, is an ordered pair (v, w) or (w, v). Typically, an arrow is used to represent this.

In the realm of graph theory, a planar graph is a graph that can be either embedded in the plane or marked on the plane with the edges only overlapping at either end. In other words, it can be drawn in a manner that ensures no edge intersects another edge. Drawings of this nature are referred to as plane graphs, or planar embeddings of graphs. Plane graphs are linear graphs that have mappings from nodes to points on a plane and from edges to plane curves on that plane. The extreme points of each curve are the points that are transmitted from its end nodes, and all curves are disconnected except for their extremes.

Network flow problems involve determining the optimal way to send flow through a network from a source to a sink while respecting capacity constraints on the edges. The Maximum-Flow and Minimum-Cut Theorem is a fundamental result in network flow theory. It states that in a flow network, the maximum amount of flow that can be sent from the source to the sink is equal to the total weight of the edges in the minimum cut, which is the smallest set of edges that, if removed, would disconnect the source from the sink.

Kuratowski's theorem provides a characterisation of planar graphs. According to this theorem, a graph is planar if and only if it does not contain a subgraph that is a subdivision of either of two specific non-planar graphs: K_5 (the complete graph on five vertices) or K_33 (the complete bipartite graph on two sets of three vertices). These two graphs are known as Kuratowski's graphs. If a graph contains a subgraph homoeomorphic to K_5 or K_33, it is non-planar.

Detecting whether a graph is planar involves determining if it can be drawn on a plane without any edges crossing. The thickness of a graph is the minimum number of planar subgraphs into which the graph can be decomposed. It is a measure of how "non-planar" a graph is. Crossing number is another measure, defined as the minimum number of edge crossings required to draw the graph in the plane. These concepts help in understanding the complexity of embedding graphs in two-dimensional surfaces and have practical implications in areas like VLSI design and network visualisation.

Notes

4.1 Directed Graphs

A directed graph is a graph with directed edges, commonly referred to as digraphs. Typically, an arrow is used to represent this; in formal terms, an edge is an unordered pair {v,w} if v and w are vertices and a directed edge, also known as an arc, is an ordered pair (v,w) or (w,v). Drawn as an arrow from v to w is the arc (v,w). A graph that has both arcs (v,w) and (w,v) is not considered to have “multiple edges” because the arcs are separate. It is possible for there to be more than one arc; specifically, an arc (v,w) may appear more than once in the multiset of arcs. As before, if a digraph has no loops or many arcs, it is called simple.

4.1.1 Directed Graphs and Connectedness

When a graph G is said to be a **directed graph or digraph** it consists of vertex set and the edge set $V=\{v_1, v_2, \ldots\}$ and $E=\{e_1, e_2, \ldots\}$ such that every pair of vertices representing any edge by an ordered or a direct pair (v_1, v_2) where v_1 is the tail and v_2 is the head of the edge

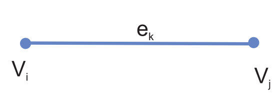


Fig 3.1a

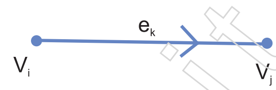


Fig 3.1b

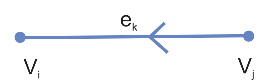
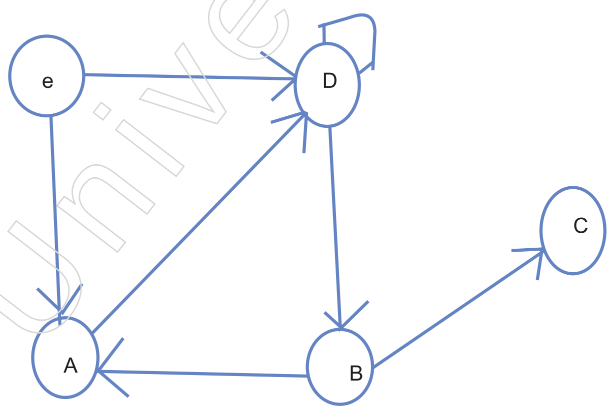


Fig 3.1c

Fig a, b and c depicts the directed graph.

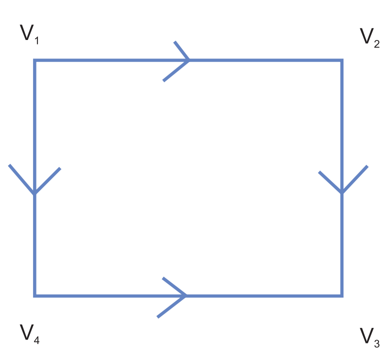


Directed graph G

Properties:

(i) Number of edges incident out of vertex v_i, $d^+(v_i)$ is said to be **out degree** or out valene

(ii) Number of edges incident into of vertex v_i, $d^-(v_i)$ is said to be **in degree** or in valene

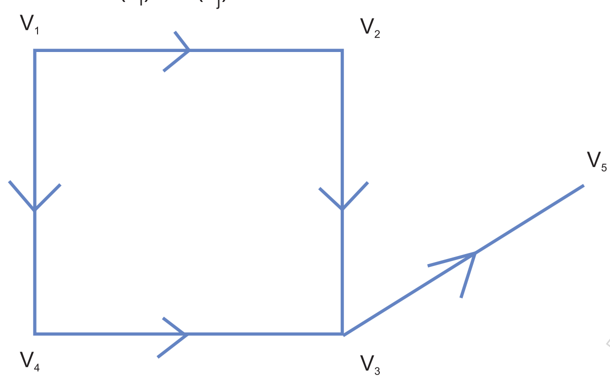


Notes

$d^+(v_1) = 2$	$d^-(v_1) = 0$
$d^+(v_2) = 1$	$d^-(v_2) = 1$
$d^+(v_3) = 0$	$d^-(v_3) = 2$
$d^+(v_4) = 1$	$d^-(v_4) = 1$

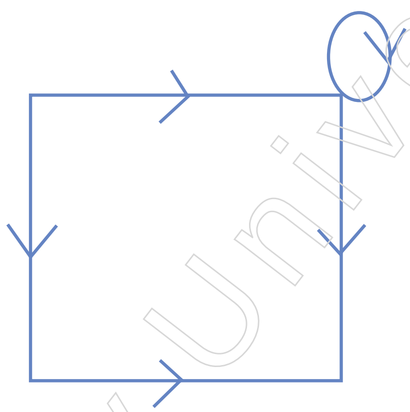
(iii) For an isolated vertex in degree = out degree = 0

(iv) In Pendent vertex $d^+(v_i) + d^-(v_j) = 1$

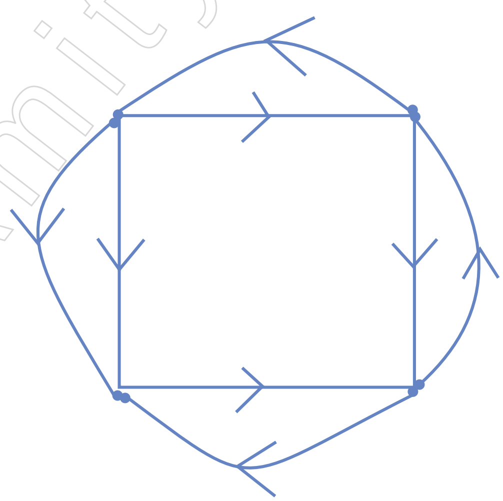


(v) In a simple digraph there are no self loops and no parallel edges

(vi) In a Asymmetric digraph that have atmost one directed edge between the pair of vertices and that are allowed to have self loops

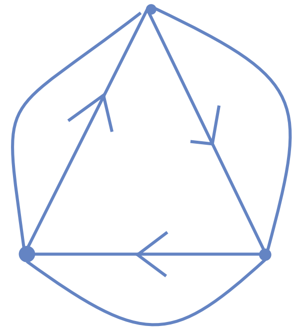


(vii) In a symmetric digraph if there is a vertex from ‘a’ to ‘b’ and the same for vertex ‘b’ to vertex ‘a’, it satisfies for all vertex ‘ab’

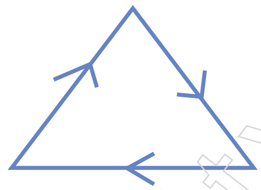


(viii) In complete symmetric digraph is a simple digraph in which there is exactly one edge directed from each vertex to every other vertex

Notes



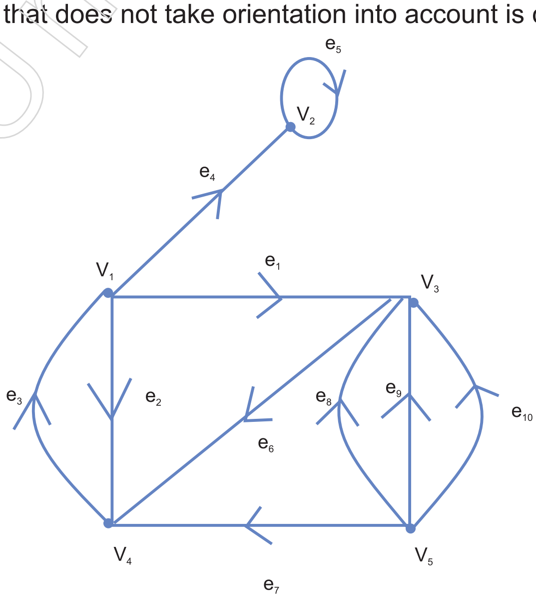
(ix) In a complete asymmetric digraph is also a simple graph there is an exactly one edge between a pair of vertices. It is also called as a tournament.



(x) A digraph is balanced for all v_i indegree = outdegree .

Note:

1. An alternating sequence of vertices and edges, beginning with v_i and ending with v_j such that each edge is oriented form the vertex preceding it to the vertex following it (no edge retracting)
2. In a directed graph, a walk in the matching undirected graph is equivalent to a **semi-walk**.
3. An edge in a **directed circuit** is orientated from the vertex before it to the vertex after it. It is a closed circuit that begins and ends on vi.
4. A closed stroll that does not take orientation into account is called a **semi-circuit**.



In the above figure,

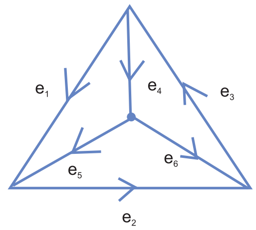
- Directed walk: $v_1\ e_2\ v_4\ e_3\ v_1\ e_4\ v_2$
- Semi walk: $v_1\ e_2\ v_4\ e_7\ v_5$
- Directed circuit: $v_1\ e_1\ v_4\ e_6\ v_4\ e_3\ v_1$
- Semi circuit: $v_1\ e_2\ v_4\ e_7\ v_5\ e_9\ v_3\ e_1\ v_1$

Notes

Types of Connectedness:

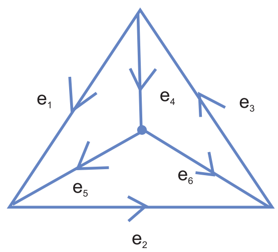
1. Strongly connected graph:

If there is at least one directed path connecting each vertex in the graph to every other vertex, the graph is considered strongly linked.



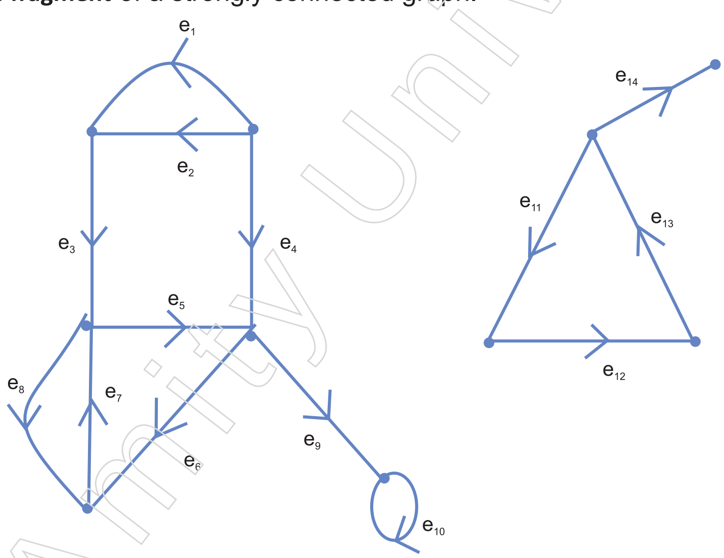
2. Weakly connected graph:

If an undirected graph that corresponds to it is linked, then the graph is said to be weakly connected.



Definition:

When a maximally strongly linked subgraph exists within each component of G, it is referred to as a **fragment** of a strongly connected graph.



Definition:

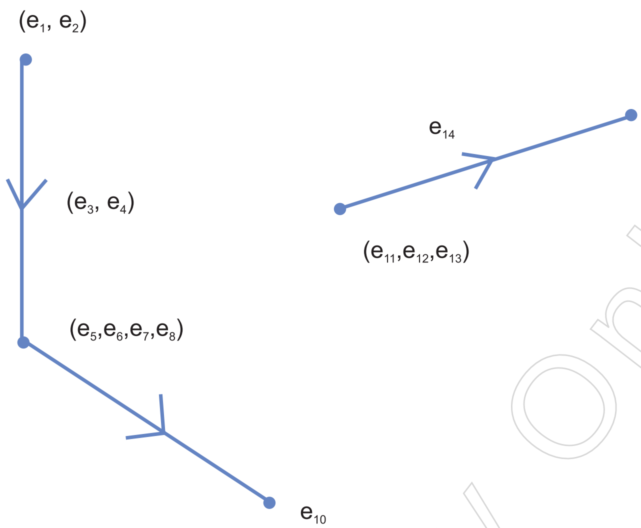
A digraph's condensation G G is a digraph where all directed edges connecting strongly connected components to one another are replaced by a single directed edge and each strongly connected fragment is replaced by a vertex.

Remark:

A vertex is all that remains of a strongly linked graph after condensation.

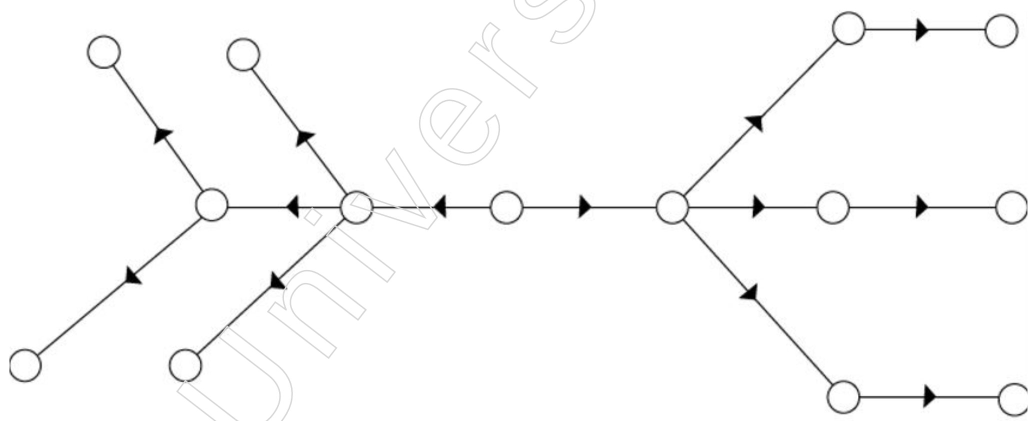
A digraph's condensation lacks a directed circuit.

Notes



4.1.2 Directed Trees

An acyclic directed graph is called a directed tree. As seen in fig., it has one node with indegree 1 and every other node has indegree 1.



Directed Trees

https://static.javatpoint.com/tutorial/dms/images/directed-trees.png

A node with an outdegree of 0 is referred to as an exterior, terminal, or leaf node. Nodes with an outdegree of one or more are referred to as internal nodes.

4.1.3 Network Flows

Definition:

A directed network D, with each arc (i, j) assigned a positive real number c_{ij} called the capacity of the arc, and two distinguished vertices called a sink [t] and a source (s), a second set of non-negative real numbers, x_{ij}, assigned to the arcs is called an (s, t) -feasible flow if $0 <= x_{ij} <= c_{ij}$ for all arcs, and $\sum_{j=1}^{n} x_{ij} \sum_{j=1}^{n} = x_{ji}$ for all vertices i, except s and t, if there is no arc (i, j) then $x_{ij} = 0$.

Definition:

A cut is a division of the directed network D's vertices into two sets, one including s (called S) and the other containing t (called T). A cut can be found in the set of arcs that begin in S and end in T and it is typically associated with this set. The total of the capacities on these arcs is the cut's capacity. A minimal cut is one that has the smallest possible capacity.

Notes

Remark:

- The value of any feasible flow is less than or equal to the capacity of any cut.
- If the value of some feasible flow equals the capacity of some cut, then the flow is a maximum flow and the cut is a minimum cut.
- For any arc (i, j) of the directed network, the slack, $s_{ij} = c_{ij} - x_{ij}$.

Algorithm:

- Finding a path from s to t such that every arc on the path has positive slack is a straightforward approach for locating a larger flow.
- For each x_{ij} of the path, we can add the least slack on these arcs to create a larger flow. Sadly, achieving a maximum flow is not usually the result of this process.
- Take a look at the directed network's underlying graph now. Find the orientation of every edge in a chain that runs from s to t in this graph.
- The direction of these arcs in the original network should then be compared to this one. The term "forward" refers to an arc when its orientation is the same; "backward" refers to an arc when its orientation is different.
- A link from s to t in the underlying graph that provides positive flow on each backward arc and positive slack on each forward arc is known as a flow enhancing chain in a directed network.
- When dealing with a flow augmenting chain, the largest possible flow is produced by adding the smallest possible value to the flow on all forward arcs and subtracting it from the flow on all backward arcs. This process takes place for both positive slacks on forward arcs and positive flows on backward arcs.

Note:

If and only if the network has no flow boosting chains, a viable flow is maximum.

The capacity of a minimal cut represents the value of a directed network's maximum flow.

Labeling Algorithm:

- A portion of the directed network's arcs will be regarded as belonging to sets F and B.
- All arcs with positive slack are in group F, while all arcs with positive flow are in group B. These sets are not disjoint, as you can see.
- As it moves forward, the algorithm labels certain vertices while scanning the others.
- The process comes to an end after it has scanned every labeled vertex. An augmenting chain can be found if t has been labeled; if not, one cannot be found.
- The vertex s is first assigned the label A, but it is an unscanned vertex; we will refer to it as i.
- Scan i, meaning
  1. for all arcs (i,j), if (i,j) is in F and j is unlabeled, then label j with (i+).
  2. for all arcs (j,i), if (j,i) is in B and j is unlabeled, then label j with (i-).
- Continue until either all labeled vertices have been scanned or labeled.
- In the event that t is labeled, the augmenting chain is found by beginning at t and

Notes

working your way backward via the labels to s.

- The +'s and -'s show whether the arc is a forward or backward arc.

Flows and Matchings in Bipartite Graphs

- We can create a directed network in a bipartite graph (X, Y) in the following way:
- Include a vertex s and connect it to every other vertex in X using arcs. Give a capacity of 1 to each of these arcs.
- Join every edge in the bipartite graph that goes from X to Y.
- All these arcs should have an infinite capacity.
- Add a vertex t at the end, then connect it to t using arcs that connect each vertex in Y. Put this directed network N and take note that each of these arcs has a capacity of 1.
- You will see that any feasible flow of N where every flow value is an integer (which means it must be either 0 or 1) corresponds to a matching of the original graph (the matching is made up of edges that match arcs that have positive flow between X and Y) and that any matching of the graph results in a feasible flow.
- The number of edges in the matching determines the flow's value.

Direct relation between flows and matchings:

A directed network's coverage is equivalent to one on the graph.

Assume that A and B are subsets of X and Y, respective. This set, K = A U B, will be discussed. An obvious cut in the network can be seen if we set S = {s} U (X-A) U B and T = {t} U (Y-B) U A.

Theorem:

'K' is a covering of the graph if and only if (S,T) is a cut of the network of finite capacity.

Proof:

Let 'K' be a covering of the graph.

Suppose (S,T) is a cut of finite capacity.

Then no arcs from X to Y are in this cut, in particular there are none from X-A to Y-B in the cut.

So, all arcs from A go to Y-B and all arcs from X-A go to B.

Therefore, every edge of the graph has one vertex in A or one vertex in B and so, K is a covering.

The arcs in the cut that start at s all go to A and those that end at t all start in B, each having capacity 1, so the capacity of the cut is |A U B|.

Now, suppose that K is a covering of the graph. Let A be the intersection of K with X and B the intersection of K with Y.

Then Form S and T as before.

Consider an arc from a vertex in S to one in T. If this arc starts at s, it must go to A.

If it starts in X-A, it must go to B since K is a covering, so such an arc does not go to T.

If it starts in B, then it must go to t and so, has capacity 1.

Thus, all arcs from S to T have finite capacity

So the cut (S, T) has finite capacity.

Finally, for a bipartite graph, the size of a maximum matching equals the size of a minimum cover.

Notes

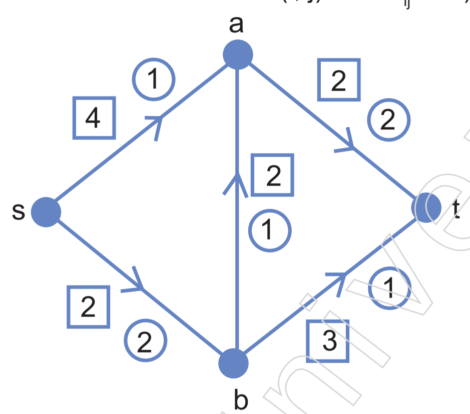
4.1.4 The Maximum-Flow and Minimum-Cut Theorem

A directed graph D with two distinct vertices, referred to as a sink (t) and a source (s) and a positive real integer c_{ij}, denoting the arc's capacity, is the representation of a capacitated single source-single sink network. There is at least one out-degree for the source and one in-degree for the sink. Although it is not necessary according to the definition, in nearly all of the examples we take into consideration, the source will have an in-degree of 0 and the sink an out-degree of 0. These will be referred to as **(s,t)-networks.**

Given an (s,t)-network, a second set of non-negative real numbers, x_{ij}, assigned to the arcs is called an **(s,t) - feasible flow** if

1. $0 \leq x_{ij} \leq c_{ij}$ for all arcs, and
2. $\sum_j x_{ij} = \sum_j x_{ji}$ for all vertices i, except s and t

(Where it is understood that if there is no arc (i, j) then $x_{ij} = 0$.).



At s and t, the conservation law (the second condition above) does not hold and we define

$v_t = \sum_j x_{jt} - \sum_j x_{tj}$ and $v_s = \sum_j x_{sj} - \sum_j x_{js}$.

Now, if we sum over all the vertices i, we get:

$\sum_i \left(\sum_j x_{ij} - \sum_j x_{ji} \right) = v_s - v_t$.

But the left hand side of this equation is 0, so $v_s = v_t$ and we call this common value v, and define it to be the **value** of the flow.

For the above figure, the value of that flow is 3.

Definition:

A cut is a division of the vertices of the (s,t)-network D into two sets, S and T, respectively, one containing s and t. The set of arcs that begin in S and terminate in T can also be used to identify a cut and this set is typically used to do so. The total of the capacities on these arcs is the capacity of a cut. A cut that has the smallest capacity is called a **minimum cut.**

Remark:

- The value of any feasible flow is less than or equal to the capacity of any cut.

Notes

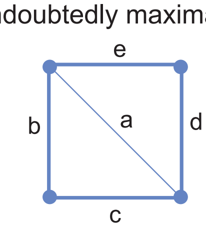
- If the value of some feasible flow equals the capacity of some cut, then the flow is a maximum flow and the cut is a minimum cut.
- For any arc (i,j) of the (s,t)-network, its **slack** is defined as, $s_{ij} = c_{ij} - x_{ij}$.
- A simple algorithm for finding a larger flow is to find a path from s to t such that all the arcs on the path have positive slack.
- We can add the smallest slack on these arcs to each xij of the path to obtain a larger flow. Unfortunately, this procedure does not always lead to a maximum flow.
- Consider the underlying graph of the directed network. Given a path from s to t in this graph, orient the all the edges of the path from s to t.
- Now, compare this orientation with the orientation that these arcs have in the original network. If the orientation is the same for an arc, it will be called a forward arc, if the orientation is opposite, the arc is called a backward arc.
- A **flow augmenting path** in the (s,t)-network, is a path from s to t in the underlying graph, so that each forward arc has positive slack and each backward arc has a positive flow.
- Given a flow augmenting path, a larger flow is obtained by taking the smallest value of the positive slacks on forward arcs and the positive flows on backward arcs and adding this value to the flow on all forward arcs and subtracting it from the flow on all backward arcs of the path.

Labeling Algorithm for Finding Flow Augmenting Paths:

- Some of the arcs of the (s,t)-network will be considered as members of two sets, F and B.
- F consists of all arcs that have positive slack and B consists of all arcs that have a positive flow. Notice that these sets are not disjoint. A forward arc in F or a backward arc in B is called a usable arc.

Matchings, Transversals and Vertex Covers:

If there isn't an endpoint shared by any two edges in set M of a graph G, then set M of edges is referred to be a matching of G. A maximal matching is one that is not a part of any larger matching, whereas a maximum matching of graph G is one that has the most edges. Although a maximum matching is undoubtedly maximal, this is not always the case.



Only the last two matchings in the graph above are maximal matchings; the other matchings are all maximal matchings: {a}, {b,d} and {c,e}.

Job Assignment Problem revisited:

The following is how we can create a (s,t)-network given a bipartite graph (X, Y):

- Include a vertex s and connect it to every other vertex in X using arcs. X to Y is the orientation of every edge in the bipartite graph.
- Add a vertex t at the end, then connect it to t using arcs that connect each vertex in Y. Assign a capacity of 1 to each arc leaving s or entering t; however, the arcs connecting X and Y will have an infinite capacity. Give this network the name N.
- Keep in mind that any feasible flow of N where every flow value is an integer (the

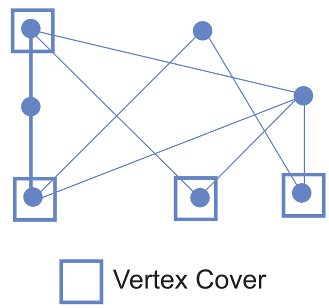
Notes

matching must be either 0 or 1) corresponds to a matching of the original graph (the matching is made up of the edges that match the arcs between X and Y that have positive flow) and any matching of the graph results in a feasible flow.

- The number of matching edges determines the flow's value.

Definition:

Assume that G is a graph and that C is a subset of its vertices. If every edge in graph G is incident with at least one vertex in set C, then set C is a vertex cover of graph G. A vertex cover with the fewest vertices is called a minimum vertex cover.



Theorem:

K is a vertex covering of the graph if and only if (S,T) is a cut of the network of finite capacity.

Proof:

Suppose (S,T) is a cut of finite capacity.

Then, this cut does not have any arcs from X to Y; specifically, it does not contain any arcs from X-A to Y-B.

Thus, every arc from A to Y-B and every arc from X-A to B. K is a covering since each edge of the graph has one vertex in either A or B as a result.

The cut's capacity is |A∪ B| since all of the arcs that begin at s go to A and all of the arcs that end at t start in B. Each arc has capacity 1.

Assume that K is a graph vertex covering. Let A represent the point where K meets X and let B represent the point where K meets Y.

As before, form S and T. Examine an arc from a vertex in S to a vertex in T. This arc has capacity 1 since it must travel to A if it begins at s. An arc of this type does not go to T if it begins in X-A since it must proceed to B because K is a vertex covering.

It has capacity 1 as it must travel to t if it begins at B. The cut (S, T) has a finite capacity because every arc from S to T has a finite capacity.

Theorem:

Let G be a bipartite graph. Then the size of a maximum matching equals the size of a minimum vertex cover.

Proof:

Let G be a bipartite graph.

Examine the bipartite graph's corresponding network.

There won't be any changes to flows if the infinite capacity are changed to 2.

Notes

Since all capacity are integers, there is a maximum flow and its values will also be integers.

The number of edges in a maximum matching, denoted as n, represents the value of this flow.

The minimal capacity of any (S, T) cut is denoted by 'n'.

Since the capacity of the trivial cut, S = {s}, is finite, the minimum must also be finite. A covering with |A∪ B| = n corresponds to any cut with the smallest capacity. However, a matching with n edges and a cover with n vertices now exist.

So the cover must be a minimum cover.

Theorem of Max-Flow Min-Cut

According to the max-flow min-cut theorem, the smallest sum of a cut is precisely equal to the maximum flow along any network from a given source to a given sink. It is possible to verify this theorem with the **Ford-Fulkerson algorithm**. This algorithm determines a network's or graph's maximum flow.

Let us formally define the max-flow min-cut theorem. Let be a flow on G and let G=(X,A) be a flow network. The capacity of the minimal cut CT dividing S and P is equal to the amount of the maximum flow from the source S to the sink P:

Max (Ø) = Min (C(CT))

The residual network, augmented path and cut are the three key ideas on which the Ford-Fulkerson algorithm is built.

$G_f(V, E_f)$ is a definition of a residual network, where $E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$.

The formula to calculate residual capacity is $c_f(u, v) = C(u,v) - f(u,v)$.

Within the residual network G_f, an augmenting path p is a straightforward path that connects source node S to sink node P.

This approach also makes use of a Netflow graph to display the flow and capacity for each edge in the graph. The algorithm begins with a feasible flow across the graph, which is then incrementally improved.

In the event that this flow reaches its maximum, both the minimal cut and the flow function satisfying this value can be found. Its goal is to draw attention to an improved path that corresponds with the flow if it is not at maximum.

The flow value between the source and sink nodes is first set to 0 by the method. We identify an augmented path and raise the flow value at each iteration. When no more enhanced pathways can be found, the algorithm will end and the flow value will be returned.

4.1.5 Matrix Representation of a Graph

Introduction

A computer can easily and effectively see a graph with the use of a matrix. It is simple to manipulate matrices mechanically. The study of graph structural properties can benefit from the use of numerous well-known matrix algebraic results.

Definition:

Let G be a graph with 'n' vertices, 'e' edges and no self loops. Define an n by 'e' matrix

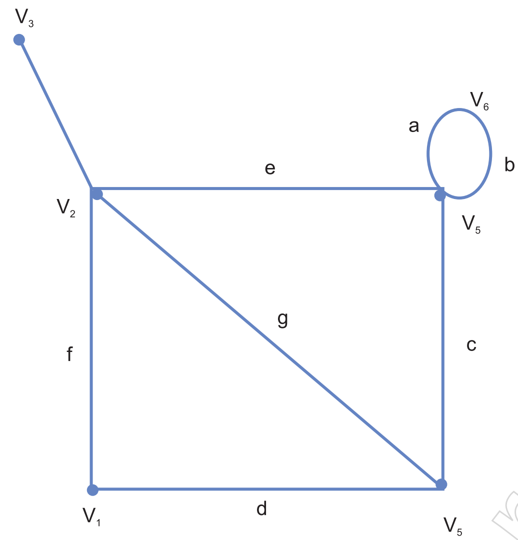
Notes

A=[a_{ij}] whose ‘n’ rows correspond to the ‘n’ vertices and the ‘e’ columns correspond to the ‘e’ edges .

The matrix element

a_{ij} = 1, if the ‘j’th edge e_j is incident on the ‘i’th vertex v_i and

a_{ij} = 0, otherwise



$$\begin{array}{cc} & \textbf{Edges} \\ \text{Vertices} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Graph and incidence matrix

- In the above graph, such a matrix is called the vertex –edge incidence matrix or simply matrix.
- The incidence matrix contains only two elements 0 and 1.
- Such a matrix is called binary matrix or a (0,1) matrix.

Observations:

- Each edge is incident on exactly two vertices, each column of ‘A’ has exactly two 1’s.
- The number of 1’s in each row is equals the degree of the corresponding vertex.
- A row with all 0’s therefore represents an isolated vertex.
- Parallel edges in a graph produce identical columns in its incidence matrix.

Definition:

Each row in an incidence matrix A(G) may be regarded as a vector over in the vector space of a graph G

Notes

$$A(G) = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \cdot \\ \cdot \\ A_n \end{bmatrix}$$

Definition:

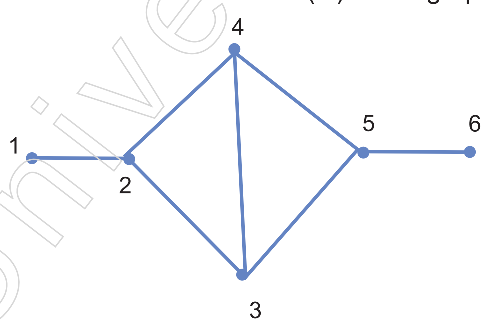
Let ‘g’ be a subgraph of a graph ‘G’ and let A(g) and A(G) be the incidence matrices of ‘g’ and ‘G’ respectively. A(g) is a submatrix of A(G). There is a non-to-one correspondence between each ‘n’ by ‘k’ submatrix of A(G) and a subgraph of G with ‘k’ edges of vertices in G.

Theorem: (Matrix Tree Theorem)

Given a loopless graph G with vertex set $v_1, v_2, ...v_n$, let a_{ij} be the number of edges having v_i and v_j as their end vertices. Let Q^* be the matrix with entry *(i, j)* being $-a_{ij}$ where $i \neq j$ and $d(v_i)$ when $i = j$. If Q^* is the matrix obtained by deleting any one row, say s^{th} row and any one column, say t^{th} column of Q, then $\tau(G) = (-1)^{s+t}$ det Q^*. $[i,e,\ \tau(G)$ is a confactor A_{st} of $G]$.

Example:

Now we use the Matrix Tree Theorem to fine τ (G) of the graph G given in Figure



$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{pmatrix}$$

matrix Q

We omit the second row and the second column to find the confactor A_{22}

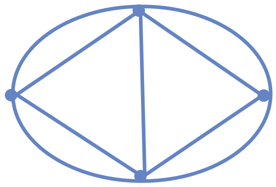
$$A_{22} = (-1)^{2+2} \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & -1 & -1 & 0 \\ 0 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{vmatrix}$$

$$1\begin{vmatrix} 3 & -1 & -1 & 0 \\ -1 & 3 & -1 & 0 \\ -1 & -1 & 3 & -1 \end{vmatrix} + 0 + 0 + 0 + 0 = 8$$

Notes

Problem 1

Find $\tau(G)$ for the graph G given to figure



Solution:

The matrix Q is

$$\begin{pmatrix} 4 & -2 & -2 & 0 \\ -2 & 5 & -1 & -2 \\ -2 & -1 & 5 & -2 \\ 0 & -2 & -2 & 4 \end{pmatrix}$$

$\tau(G)$ = the cofactor A_{11} = $(-1)^2 \begin{vmatrix} 5 & -1 & -2 \\ -1 & 5 & -2 \\ -2 & -2 & 4 \end{vmatrix}$

= 5(20-4)+1(-4-4)-2(2+10)

= 80-8-24

= 48

4.2 Planar Graphs

The question of planarity is of great significance from a theoretical point of view. In addition, planarity and other related concepts are useful in any practical situations. For instance, in the design of a printed-circuit board, the electrical engineer must know if he can make the required connections without an extra layer of insulation. The solution to the puzzle if three utilities, requires the knowledge of whether or not the corresponding graph can be drawn in a plane.

4.2.1 Planar Graphs: Combinational and Geometric Duals

A graph exists as an abstract object, devoid of any geometric connotation of its ability of being drawn in three-dimensional Euclidean space. For example, an abstract graph can be defined as

A graph exists as an abstract object, devoid of any geometric connotation of its ability of being drawn in three-dimensional Euclidean space. For example, an abstract graph G_1 can be defined as

G_1 = (V, E, ψ),

Where the set V consists of the five objects named a, b, c, d and e, that is,

V={a,b,c,d,e}

Notes

And the set E consists of seven objects (none of which is in set) named 1,2,3,4,5,6 and 7, that is,

E={1,2,3,4,5,6,7},

And the relationship between the two sets is defined by the mapping ψ, which consists of

$$\psi = \begin{cases} 1 \longrightarrow (a,c) \\ 2 \longrightarrow (c,d) \\ 3 \longrightarrow (a,d) \\ 4 \longrightarrow (a,b) \\ 5 \longrightarrow (b,d) \\ 6 \longrightarrow (d,e) \\ 7 \longrightarrow (b,e) \end{cases}$$

Here, he symbol 1 → (a,c) says that object 1 from set E is mapped onto the (unordered) pair (a,c) of objects from set V.

Now it so happens that this combinatorial abstract object G_1 can also be represented by means of a geometric figure. Moreover, it is also true that any graph can be represented by means of such a configuration in three-dimensional Euclidean space.

Definition:

A graph G is said to be **planar** if there exists some geometric representation of G which can be drawn on a plane such that no two of its edges intersect. A graph that cannot be drawn on a plane without a crossover between its edges is called nonplanar.

A drawing of a geometric representation of a graph on any surface such that no edges intersect is called embedding. Thus, to declare that a graph G is nonplanar. We have to show that of all possible geometric representation of G none can be embedded in a plane. Equivalently, a geometric graph G is planar if there exists a graph isomorphic to G that is embedded in a plane. Otherwise, G is nonplanar. An embedding of a planar graph G on a plane is called a plane representation of G.

For instance, consider the graph represented by. The geometric representation clearly is not embedded in a plane, because the edges e and f are interesting. But if we redraw edge outside the quadrilateral, leaving the other edges unchanged, we have embedded the new geometric graph in the plane, thus showing that the graph which is being represented is planar. As another example, the two isomorphic diagrams in are different geometric representations of one and the same graph. One of the diagrams is a plane representation; the other one is not. The graph, of course, is planar. On the other hand, you will not be able to draw any of the three configurations in on a plane without edges intersecting. The reason is that the graph which these three different diagrams in represent is nonplanar.

A graph G [which may be given by an abstract notation G = (V, E, Γ) or by one of its geometric representation] is planar or nonplanar.

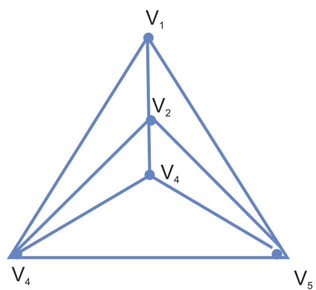
Different Representations of a Planar Graph

It may have appeared that one's ability to draw a planar graph in a plane depended on his ability to draw many crooked lines through devious routes. This is not the case. The following important and somewhat surprising result, due to Fary, tells us that there is no need to bend edges in drawing a planar graph to avoid edge intersections. Any simple planar graph can be embedded in plane such that every edge is drawn as a straight-line segment.

Notes

Definition:

A plane representation of a graph divides the plane into regions also called windows, faces, or meshes) as shown in Figure A region is characterised by set of



Straight line representation of the graph

Theorem:

A planar graph may be embedded in a plane such that any specified region (i.e., specified by the edges forming it) can be made the infinite region.

Proof:

In terms of the regions on the sphere, we see that there is no real difference between the infinite region and the finite regions on the plane.

Therefore, when we talk of the regions in the plane representation of a graph, we include the infinite region.

Also, since there is no essential difference between an embedding of a planar graph on a plane or on a sphere (a plane may be regarded as the surface of a sphere or infinitely large radius), the term "plane representation" of a graph is often used to include spherical as well as planar embedding.

Theorem:

Prove that a connected planar graph with n vertices and e edges has e-n+2 regions.

Proof

It will suffice to prove the theorem for a simple graph, because adding a self-loop or a parallel edge simply adds one region to the graph and simultaneously increases the value of e by one.

We can also remove all edges that do not form boundaries of any region. Addition (or removal) of any such edge increases (or decreases) e by one and increases (or decreases) n by one, keeping the quality e-n unaltered.

Since any simple planar graph can have a plane representation such that each edge is a straight line, any planar graph can be drawn such that each region is a polygon (a polygon net). Let the polygonal net representing the given graph consist of f regions or faces, and let k_p be the number if p-sided regions. Since each edge is on the boundary of exactly two regions,

$$3.k_3 + 4.k_4 + 5.k_5 + \ldots + r.k_r = 2.e.$$

Notes

Where k is the number of polygons, with maximum edges.

Also,

$$k_3 + k_4 + k_5 + \ldots + k_r = f.$$

The sum of all angles subtended at each vertex in the polygonal net is

$$2\pi n.$$

Recalling that the sum of all interior angles of a p-sided polygon is π (p-2), and the sum of the exterior angles is π (p+2), let us compute the expression as the grand sum of all interior andlesf-1 finite regions plus the sum of the exterior angles of the polygon defining the infinite region. This sum is

$$\pi(3-2).k_3 + \pi(4-2).k_4 + \ldots + \pi(r-2).k_r + 4\pi$$

$$= \pi(2e-2f) + 4\pi.$$

Equating we get

$$2\pi(e-f) + 4\pi = 2\pi n.$$

or $$e-f+2 = n.$$

Therefore, the number of regions is

$$f = e-n+2.$$

Theorem:

In any simple, connected planar graph with f regions, n vertices, and e edges (e>2), the following inequalities must hold:

1. $e \geq \frac{3}{2} f$ 2. $e \leq 3n - 6$

Proof:

Since each region is bounded by at least three edges and each edge belongs to exactly two regions,

$$2e \geq 3f,$$

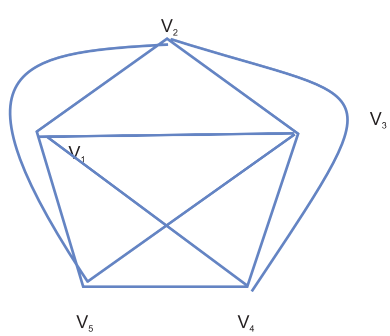
or $$e \geq 3/2 f.$$

Substituting for f from Euler's formula is inequality

$$e \geq 3/2\,(e-n+2)$$

or $$e \leq 3n-6.$$

It is often useful in finding out if graph is nonplanar. For, example, in the case of K_3, the complete graph of five vertices.



(e)

$$n = 5, \quad e = 10, \quad 3n - 6 = 9 < e.$$

Thus the graph violates and hence it is not planar.

Incidentally, this is an alternative and independent proof of the non-planarity of Kuratowski's first graph.

Every simple planar graph must satisfy the above equation the mere satisfaction of this inequality does not guarantee the planarity of a graph.

For example, Kuratowski's second graph, $K_{3.3}$, satisfies because

$$e = 9,$$

$$3n- 6 = 3.6-6 = 12.$$

Yet the graph is non-planar.

To prove the non-planarity of Kuratowski's second graph, we make use of the additional fact that no region in the graph can be bounded with fewer than four edges.

Hence, if this graph were planar, we would have

$$2e \geq 4f,$$

And substituting for f from Euler's formula,

$$2e \geq 4(e-n+2),$$

or $$2.9 \geq 4(9-6+2),$$

or $$18 \geq 20, \quad \text{a contradiction.}$$

Hence the graph cannot be planar.

Plane Representation and Connectivity:

In a disconnected graph the embedding of each component can be considered independently. Therefore, it is clear that a disconnected graph is planar if and only if its components is planar. Similarly, in a separable (or 1-connected) graph the embedding of each block (i.e., maximal non-separable subgraph) can be considered independently. Hence a separable graph is planar if and only if each of its blocks is planar. Therefore, in questions of embedding or planarity, one need consider only nonseparable graphs.

We must define the meaning of unique embedding can be made to coincide by suitably rotating one sphere with respect to the other and possibly distorting regions (without letting a vertex cross an edge). If of all possible embeddings on a sphere no two are distinct, the graph is said to have a unique embedding on a sphere (or a unique plane representations).

For example, consider two embeddings of the same graph.

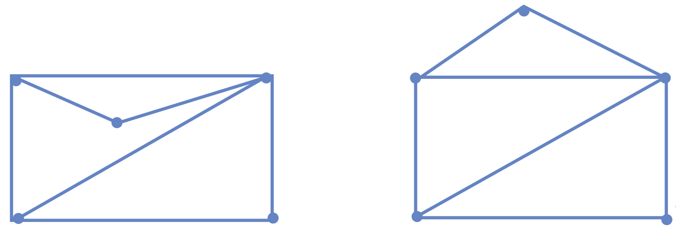
The embedding (b) has a region bounded with five edges, but embedding (a) has no region with five edges. Thus, rotating the two spheres on which (a) and (b) are embedded will not make them coincide. Hence the two embeddings are distinct and the graph has no unique plane representation.

On the other hand, the embeddings when considered on a sphere, can be made to coincide. (Remember that edges can be bent and in a spherical embedding there is no infinite region), when a graph is uniquely embeddable in a sphere.

Remark:

The spherical embedding of every planar 3-connected graph is unique.

Notes



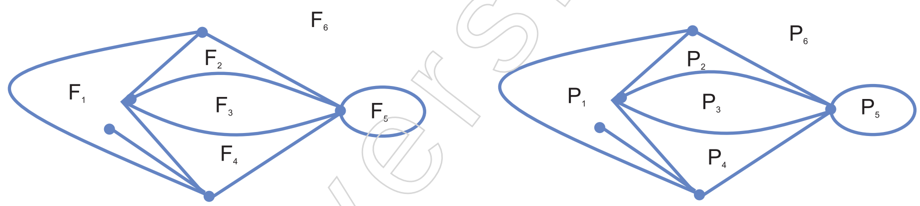
(a) (b)

Two distinct plane representation of the same graph

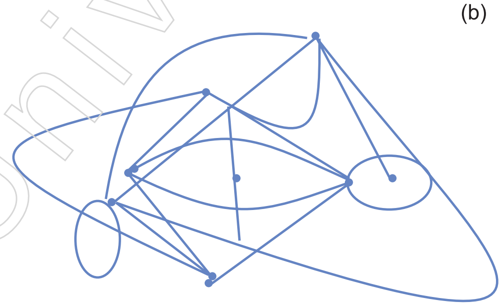
Geometric Dual

Consider the plane representation of a graph in figure with six regions or faces F_1, F_2, F3, F4, F_5, and F_6.

Let us place six points $p_1, p_2, \ldots, p_6$. One in each of the regions, as shown in figure. Next let us join these six points according to the following procedure:



(a) (b)



Construction of a dual graph

If two regions F_i and F_j are adjacent (i.e., have a common edge), draw a line joining points p_i and p_j that intersects the common edge between F_i and F_j exactly once.

If there is more than one edge common between F_i and F_j, draw one line between points p_i and p_j for each of the common edges.

For an edge e lying entirely in one region, say F_k, draw a self-loop at point p_k intersecting e exactly once.

By this procedure we obtain a new graph G* consisting of six vertices, p_1, p_2, . . . , p_6 and of edges joining these vertices. Such a graph G* is called a dual (or strictly speaking, a geometric dual) of G.

Clearly, there is a one -to-one correspondence between the edges of graph G and its dual G* - one edge of G* intersecting one edge of G.

Observations that can be made about the relationship between a planar graph G and its dual G* are

Notes

1. An edge forming a self-loop in G yields a pendant edge in G*.
2. A pendant edge in G yields a self-loop in G*.
3. Edges that are in series in G produce parallel edges in G*.
4. Parallel edges in G produce edges in series in G*.
5. The result of the general observation that the number of edges constituting the boundary of a region F_i in G is equal to the degree of the corresponding vertex p_i in G*, and vice versa.
6. Graph G* is also embedded in the plane and is therefore planar.
7. Considering the process of drawing a dual G* from G, it is evident that G is a dual of G* [see figure]. Therefore, instead of calling G* a dual of G, we usually say that G and G* are dual graphs.
8. If n, e, f, r, and μ denote as usual the numbers of vertices, edges, regions, rank, and nullity of a connected planar graph G, and if n*, e*, f*, r*, and μ^* are the corresponding numbers in dual graph G*, then

$$n^* = f,$$

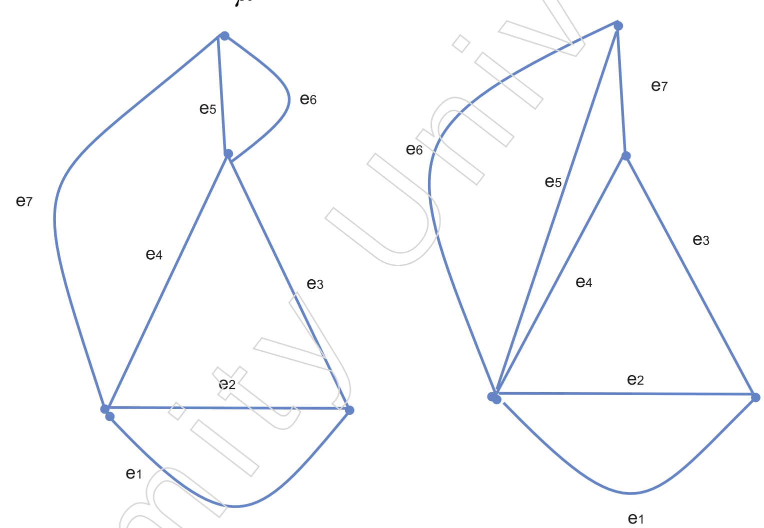
$$e^* = e,$$

$$f^* = n.$$

Using the above relationship, one can immediately get

$$r^* = \mu,$$

$$\mu^* = r.$$



Dual of graphs

Unique the above Dual Graphs:

In an incident on a pendant vertex is called a pendant edge.

Other words, are all duals of a given graph isomorphic. From the methods of constructing a dual, it is reasonable to except that a planar graph G will have a unique dual if and only of it has a unique plane representation or unique embedding on a sphere.

For instance, in figure the same graph (isomorphic) had two distinct embeddings, (a) and (b). Consequently, the duals of these isomorphic graphs are non-isomorphic, as shown in figure.

Notes

The graphs in figure, however, are 2-isomorphic. This stated without proof, is a generalisation of this example.

Theorem:

All duals of a planar graph G are 2-isomorphic; and every graph 2-isomorphic; to a dual of G is also a dual of G.

Proof:

It is quite appropriate to refer to a dual as the dual of a planar graph.

Since a 3-connected planar graph has a unique embedding on a sphere, its dual must also be unique. In other words, all duals of a 3-connected graph are isomorphic.

Combinatorial Dual:

We have defined and discussed duality of planar graphs in a purely geometric sense. The following provides us with an equivalent definition of duality independent of geometric notions.

Theorem:

A necessary and sufficient condition for two planar graphs and to be duals of each other is as follows: There is a one-to-one correspondence between the edges in and the edges in such that a set of edges in forms a circuit if and only if the corresponding set in forms a cut-set.

Proof:

Let us consider a plane representation of a planar graph G. Let us also draw (geometrically) a dual G* of G. Then consider an arbitrary circuit Γ in G. Clearly, Γ will form some closed simple curve in the plane representation of G dividing the plane into two areas. (Jordan Curve Theorem).

Thus the vertices of G* are partitioned into two nonempty, mutually exclusive subsets one inside Γ and the other outside.

In other words, the set f edges Γ * in G* corresponding to the set Γ in G is a cutset in G*. (No proper subset of Γ * will be a cut-set in G*).

Likewise it is apparent that corresponding to a cut-set S in G such that S is a circuit. This proves the necessity portion

To prove the sufficiency, let G be a planar graph and let G’ be a graph for which there is a one-to-one correspondence between the cut-sets of G and circuits of G’ and vice versa.

Assume that G* is G’s dual graph. The circuits of G’ and cut-sets of G, as well as the cut-sets of G and circuits of G*, are equivalent in a one-to-one relationship.

Consequently, the circuits of G’ and G* have a one-to-one correspondence, suggesting that G’ and G* are 2-isomorphic and that G’ must be a dual of G.

Dual of a Subgraph:

Let G be a planar graph and let G* be its dual. Let a be an edge in G and let a* be the matching edge in G*. Let’s say that we remove edge a from G and then look for G-a’s dual. Removing edge a would combine two territories that were bordered by it into one. Hence, by removing the matching edge a* and then fusing the two end vertices of a* in G*-a*, the dual (G - a)* can be obtained from G*. However, a* creates a self-loop if edge an is not on the boundary. G*-a* is the same as (G-a)* in that scenario. Because of this, if a graph G has a dual G*, then any subgraph of G can have its dual derived by repeatedly using this process.

Notes

Dual of a Homeomorphic Graph:

G* is the dual of G, a planar graph. Assume that an is an edge in G and that a* is its corresponding edge in G*. Let us assume that we add a vertex of degree two to edge a, so that a becomes two edges in series and so we generate an extra vertex in G. To G*, it will merely add an edge parallel to a*. Similarly, one of the matching parallel edges in G* will be eliminated when two edges in series are merged in reverse. If a graph G has a dual, then any graph that is homeomorphic to G can be constructed from G* using the method described above.

So far we have been studying duality for planar graphs only. This was forced upon us because the very definition of duality depended on the graph being embedded in a plane. However, now that provides us with an equivalent abstract definition of duality (namely, the correspondence between circuits and cut-sets), which does not depend on a plane representation of a graph, we will see if the concept of duality can extended to nonplanar graphs also. In other words, given a non-planar graph G, can we find another graph G' with one-to-one correspondence between their edges such that every circuit in G corresponds to a unique cut-set in G' and vice versa. The answer to this question is no, as shown in the following important theorem, due to Whitney.

Theorem:

Prove that a graph has a dual if and only is it is planar.

Proof:

It simply remains to prove the "only if" portion. In other words, all we need to do is demonstrate that a nonplanar graph lacks a dual. Consider a nonplanar graph, G. Kuratowski's theorem thus states that G contains a graph homeomorphic to either K_5 or K_{33}. As we have already seen, a graph G can only have a dual if there is a dual for each of its subgraphs, g and graphs that are homeomorphic to g.

This if we can shoe that neither K_5 or K_{33} has a dual, we have proved the theorem.

This we shall prove by contradiction as follows:

(a) Assume that K_{33}. possesses a dual D. Note that circuits in D correspond to cut-sets in K_{33}. and vice versa. Given that K_{33}. lacks a cut-set with two edges, D also lacks a circuit with two edges. In other words, D doesn't have any parallel edge content. D has no cut-set with fewer than four edges since all of the circuits in K_{33}. are four or six edges long. As a result, each vertex in D has a degree of at least four. D must have at least five vertices, each of degree four or more, since it lacks any parallel edges and each vertex has a degree of at least four. D must therefore have a minimum of (5 x 4)/2 = 10 edges. This is a contradiction, because K_{33} has nine edges and so must its dual. Thus K_{33} cannot have a dual. Likewise

(b) Assume that there is a dual H in graph K5. Keep in mind that K5 has

(1) ten edges,

(2) the absence of two parallel edges,

(3) a two-edged cut-set without cuts and

(4) Cut-sets consisting of four or six edges.

Notes

Thus, graph H needs to have

(1) ten edges,

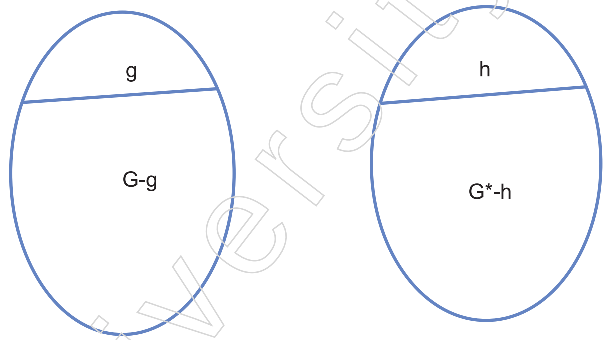
(2) No vertex with a degree lower than 3.

(3) the absence of two parallel edges and

(4) There are only four and six circuits total. Now, graph H has two parallel edges of length three and a hexagon, or a circuit of length six. There must be at least seven vertices in H since H has ten edges and both of these are prohibited. Every one of these vertices has a degree of at least three. H therefore has a minimum of 11 edges. which contradicts itself.

An additional comparable combinatorial definition of duality is provided, as well.

If there exists a one-to-one correspondence between the edges of two planar graphs, G and G*, such that if g is any subgraph of G and h is the corresponding subgraph of G*, then Rank of (G*-h) = rank of G* - nullity of g, then two planar graphs are said to be combinatorial duals (or duals of each other).



Rank of (G*-h)=Rank of G*-Nullity of g

Combinatorial duals

It takes a lot of work to demonstrate the equivalence of geometric and combinatorial duals. We just refer to the geometric and combinatorial duals as the dual, rather than the geometric or combinatorial duals, because they are the same.

Self- Dual Graphs:

A planar graph G is referred to as self-dual if it is isomorphic to its own dual. The four-vertex complete graph is a self-dual graph, as may be demonstrated with ease. Self-dual graphs present several intriguing traits as well as open challenges.

4.2.2 Kuratowski's Graphs

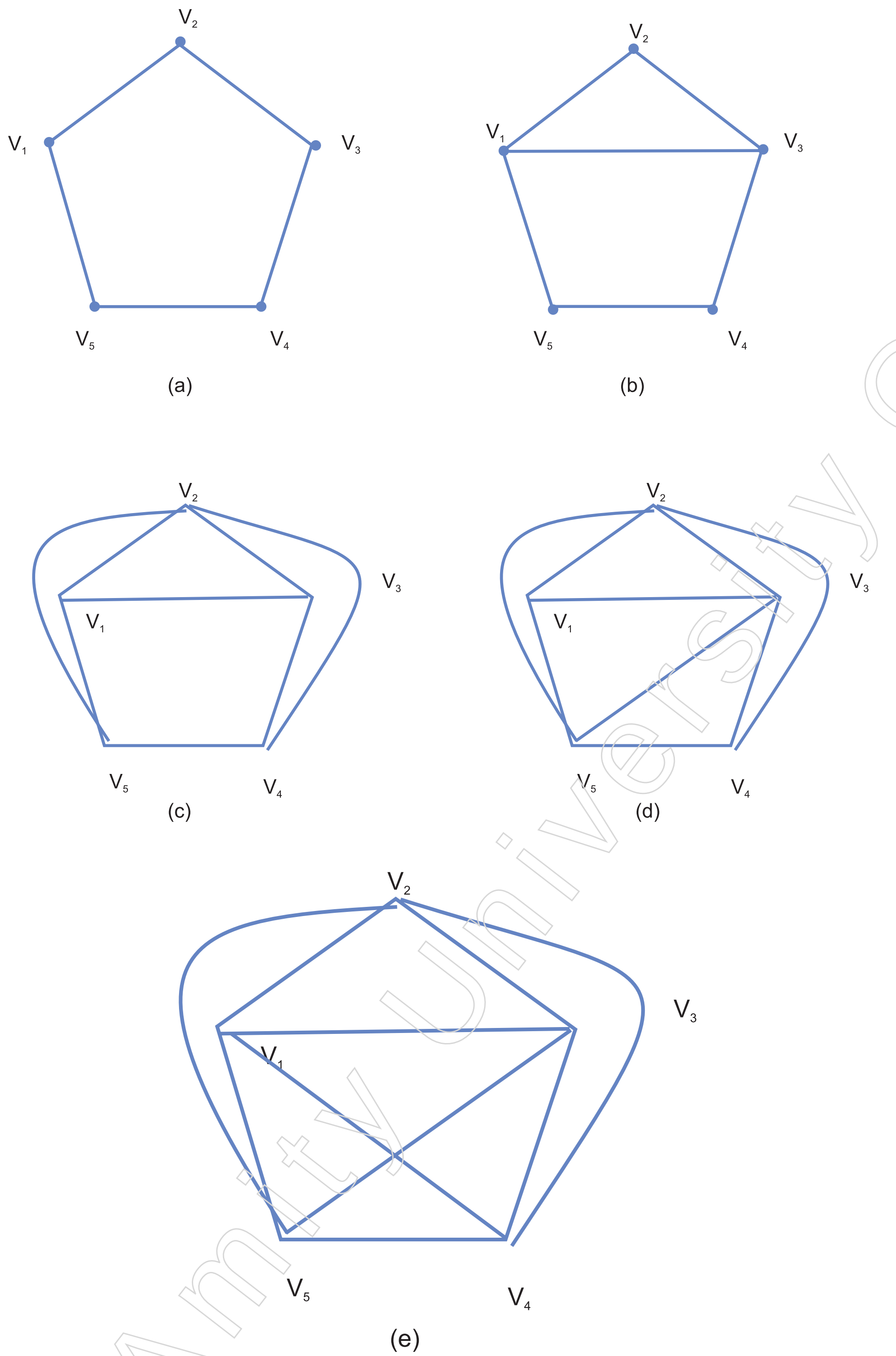
Theorem:

Prove that the complete graph of five vertices is nonplanar. Proof Let the five vertices in the complete graph be named v_1, v_2, v_3, v_4 and v_5

A complete graph, as you may recall, is a simple graph in which every vertex is joined to every other vertex by means of an edge.

This being the case, the "meeting" of edges at a vertex is not considered an intersection.

Notes



The five vertices of complete graph

A circuit going from v_1 to v_2 to v_3 to v_4 to v_5 to v_1—that is, a pentagon. As seen in above fig (a). According to the Jordan Curve Theorem, this pentagon must split the paper's plane into two sections: an inside region and an outside region.

An edge that connects vertex v_1 to v_3 must be drawn and it can be drawn within or outside the pentagon as long as it doesn't cross any of the five previously drawn edges.

Assume we decide to create a line inside the pentagon that runs from v_1 to v_3. (The argument remains the same if we opt for the outside.)

Notes

One edge needs to be drawn from v2 to v4 and another from v2 to v5. We draw these two edges outside the pentagon since it is not possible to draw one of them inside the pentagon without also drawing over this other edge. Without going over the boundary between v2 and v4, it is impossible to draw the edge between v3 and v5 outside of the pentagon.

Therefore, v3 and v5 have to be connected with an edge inside the pentagon.

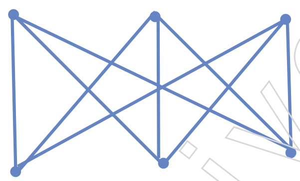
The border between v1 and v4 still needs to be drawn.

This edge requires a crossover in order to be positioned inside or outside the pentagon. As a result, a plane cannot contain a graph.

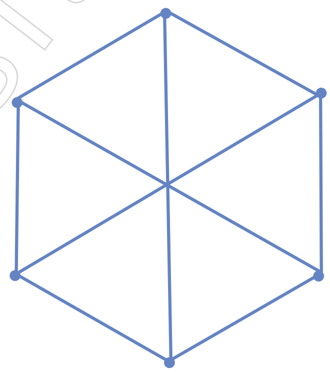
The first of Kuratowski's two graphs is a complete graph with five vertices.

Kuratowski's second graph, which has nine edges and six vertices, is a regular connected graph. It can be easily observed that the two graphs are isomorphic by examining their shared geometric representations.

It is possible to demonstrate that Kuratowski's second graph is likewise nonplanar by using visual geometric reasons that are comparable to those used in the proof.



(a)



(b)

Kuratowski's second graph

Remark:

- Kuratowski's second graph is also nonplanar.
- Both are regular graphs.
- Both are nonplanar.
- Removal of one edge or a vertex makes each a planar graph.
- Kuratowski's first graph is the nonplanar graph with the smallest number of vertices and Kuratowski's second graph is the nonplanar graph with the smallest number of edges.
- Thus, both are the simplest nonplanar graphs.

4.2.3 Detection of Planarity

Set of Basic Circuits:

A set C of circuits in a graph is said to be a complete set of basic circuits if

(i) every circuit in the graph can be expressed as a ring sum of some or all circuits in C and

(ii) no circuit in C can be expressed as a ring sum of others in C.

The significance of complete sets of basic circuits will be clearer in Chapter 6, in relation to the vector space of a graph. It may, however, be mentioned here that whereas a set of fundamental circuits with respect to a spanning tree always constitutes a complete set of basic circuits, the converse does not hold for all graphs.

An entire set of fundamental circuits has an extra feature in a planar graph.

The set of circuits forming the interior regions of a planar, linked graph G in a plane representation represents the entire set of fundamental circuits. The ring total of the circuits defining the regions contained in Γ can be used to express Γ in any given circuit in G. Note that each edge occurs in a maximum of two of these fundamental circuits. Therefore, we can identify the entire collection of fundamental circuits for each planar graph G such that no edge appears in more than two of these basic circuits. Planar graphs are characterised in another well-known way by this result and its opposite.

Theorem:

If and only if there is a full set of basic circuits (all μ of them, where μ is G's nullity) with no edge appearing in more than two of them, then a graph G is said to be planar.

Proof:

Two issues plague three of these timeless characterisations. To begin with, they are very challenging to apply to a huge graph. Secondly, they do not provide a plane representation of the graph if it is planar.

These shortcomings have led to the recent discovery of a number of map-construction techniques, where the planarity test itself is predicated on a number of other techniques, some of which are quite comparable, that have been applied to digital computers. The majority of these techniques begin by reducing the provided graph to one or more straightforward, non-separable graphs where $e \leq 3n - 6$ and every vertex has degree three or higher. The building algorithm is then used in a way that determines which of the two succeeds in being stated on such algorithms.

All algorithms require a lot of work and effort; some are better than others. It is far from over to find a straightforward, elegant and useful characterisation of a planar graph.

4.2.4 Thickness and Crossing

A crossing on a graph is created when two edges of the graph cross each other. You may find the overall number for that graph design by adding up all of these crossings. As a result, reducing the number of crossings by changing the vertices' locations is one of the primary issues.

Usually, this is done to make graphs easier to read or to address a practical issue like circuit board design or the construction of railroad or road crossings. The crossing number is the smallest number of crossings that can be found in any graph drawing.

There is no algorithm that can determine the minimal number of crossings of every graph in deterministic polynomial time, making computing the crossing number an NP-hard task. We are aware of the rectilinear crossing number for complete rectilinear graphs, which is the number of vertices up to 27.

Unless for graphs where we already know the crossing number, it would be at least as difficult to have software minimize or draw the graph with the minimum crossing because this task is NP-hard. A triangular convex hull is the best option in all other scenarios. By manually creating full graphs with few vertices, you can find out why.

Notes

Local Crossing Number:

The local crossing number is one area that has gotten less attention despite the fact that there has been a lot of research on the crossing number. The greatest number of crossings on a single edge is known as the local crossing number in a graph drawing. The local crossing number for a given graph is its minimal local crossing in any drawing of that graph.

Summary

- A directed graph is a graph with directed edges, commonly referred to as digraphs.
- A graph is said to be weakly connected if its corresponding undirected graph is connected.
- A graph is said to be strongly connected fragment, within each component of G, the maximal strongly connected subgraph is called as fragments.
- The condensation of strongly connected graph is simply a vertex.
- A partition of the vertices of the directed network D into two sets, one containing s, called S and the other containing t, called T, is called a cut.
- A directed network D, with each arc (i, j) assigned a positive real number c_ij called the capacity of the arc and two distinguished vertices called a sink [t] and a source (s).
- A graph with vertices that can be split into two disjoint sets so that every edge links a vertex in one set to a vertex in the other is called a bipartite graph. In the disjoint sets, there are no edges connecting vertices.
- Kirchhoff's law states that the total flow entering and exiting a node, also known as a vertex, should equal one another.
- A matrix is a convenient and useful way of representing a graph to a computer. Matrices lend themselves easily to mechanical manipulations.
- A graph exists as an abstract object, devoid of any geometric connotation of its ability of being drawn in three-dimensional Euclidean space.
- A plane representation of a graph divides the plane into regions also called windows, faces, or meshes.
- A graph exists as an abstract object, devoid of any geometric connotation of its ability of being drawn in three-dimensional Euclidean space.

Glossary

- Out degree: Number of edges incident out of vertex v_i, d^+ (vi) is said to be out degree.
- In degree: Number of edges incident into of vertex v^i, d^- (vi) is said to be in degree.
- A directed circuit is a closed circuit starting from vi and ending on vi such that each edge is oriented from the vertex preceding it to the vertex following it.
- Strongly connected graph: A graph is said to be strongly connected if there is at least one directed path from one vertex to every other vertex.
- Directed tree: An acyclic directed graph is called a directed tree.
- Maximum flow: The greatest amount of flow that the network or graph would permit to go from a source node to a sink node is known as a maximum flow.
- Bipartite graph: When a set of graph vertices is divided into two separate sets, no neighboring graph vertices in either set remains, the resulting sets are called bipartite graphs, or bi-graphs. A k-partite graph with k=2 as a specific example is called a bipartite graph.

Notes

- Kuratowski's Theorem: Assume that G is a graph. If G has a subgraph that is a subdivision of either K3,3 or K5, then G is nonplanar if and only if it does.
- A graph G is said to be planar if there exists some geometric representation of G which can be drawn on a plane such that no two of its edges intersect. A graph that cannot be drawn on a plane without a crossover between its edges is called nonplanar.

Check Your Understanding

1. Which statement or statements below is/are TRUE?

   P: A cycle is a walk whose end vertices are identical.

   Q: A path with identical end vertices is called a cycle.

   a) P only
   b) Q only
   c) Both P and Q
   d) Neither P and Q

2. How many edges can a bipartite graph with 12 vertices have at most?

   a) 36
   b) 38
   c) 24
   d) 32

3. __________ stipulates that the overall flow into and out of a vertex, sometimes referred to as a node, must match one another.

   a) Fleury Algorithm
   b) Prims Algorithm
   c) Kruskal Algorithm
   d) Kirchhoff's law

4. When a graph has a pair of _____________, it's known as a crossing on the graph.

   a) Edges that cross
   b) Vertices that overlap
   c) Edges and vertex both
   d) None of the above

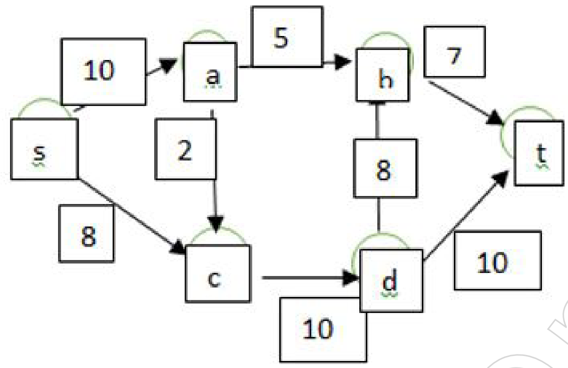
5. G is a simple undirected graph. Some vertices of G are of odd degree. Add a node v to G and make it adjacent to each odd-degree vertex of G. The resultant graph is sure to be

   a) Regular
   b) Complete
   c) Hamiltonian
   d) Euler

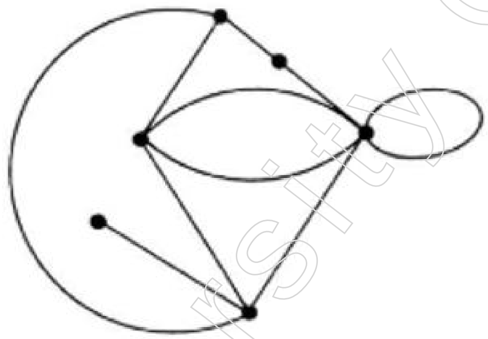
Exercise

1. Define directed graphs and their connectedness with the help of suitable example.
2. Find the maximum flow from the following graph.

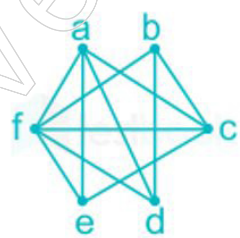
Notes



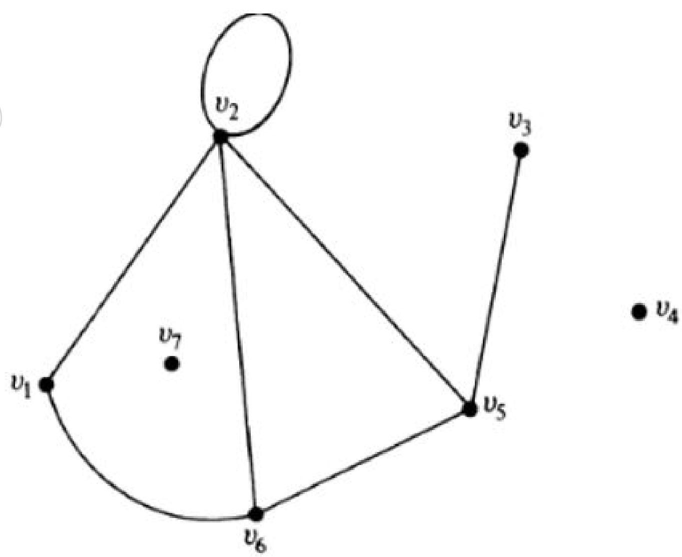
3. Find the geometric dual of the following planar graph.



4. Verify whether the graph is planar or not?



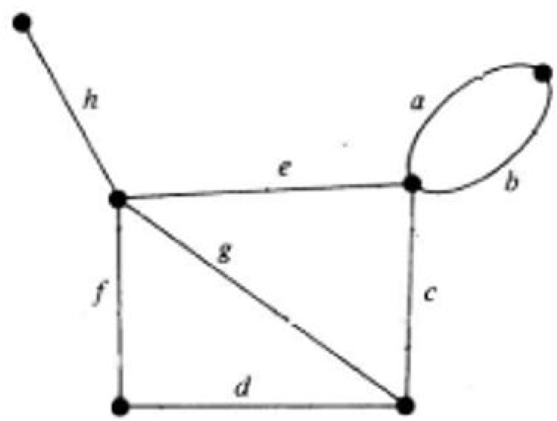
5. In the following graph identify isolated vertices, series edges and pendant vertex.



and prove that the number of vertices of odd degree in a graph is always even.

Learning Activities

1. Prove that a connected planar graph with n vertices and e edges has e-n+2 regions.
2. Find all fundamental circuits and cut sets of the following graph and show that the ring sum of any two cut sets in this graph is either a third cut set or an edge disjoint union of cut sets.



Notes

Check Your Understanding (Answers)

1. a) 2. c) 3. d) 4. a)
5. d)

Further Readings and Bibliography

1. Dr. D.S.C ," Graph Theory And Combinatorics All India", Prism Publications, 1 January 2012.
2. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 2nd Edition, 2000.
3. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009.
4. Harikishan, ShivrajPundir and Sandeep Kumar, Discrete Mathematics, Pragati Publication, 7th Edition, 2010.

Module - V: Combinatorics

Notes

Learning Objectives

At the end of this module, you will be able to:

- Introduce Combinatorics in Graph
- Describe Partition sets and Counting Functions
- Identify the number of Partitions into Odd or Unequal Sets
- Define Necklace and Euler's Functions
- Analyse set of Symmetries
- Enumerate Odd and Even cases
- Learn Combinatorial Designs
- Describe Ramsey Theory
- Define Inclusion-Exclusion Principle
- Infer Recurrence Relations
- Understand Polya's Enumeration Theorem

Introduction

The characteristics of mathematical relations are delineated by the concept of "combinatorics." Mathematicians employ this term to denote the more extensive subset of discrete mathematics. It is frequently employed in the field of computer science to estimate and analyse algorithms, as well as to create formulas.

It is a branch of mathematics that concentrates on the specific characteristics of finite structures, as well as counting as both a tool and an objective in and of itself.

Ramsey theory is a branch of mathematics that studies conditions under which order must appear. It asserts that in sufficiently large structures, certain properties will always emerge. The inclusion-exclusion principle is a fundamental counting technique used in combinatorics to calculate the size of the union of multiple sets. It corrects the overcounting that occurs when simply adding the sizes of the sets by subtracting the sizes of their intersections and adding back the size of their pairwise intersections, and so on.

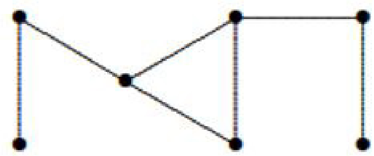
Recurrence relations are equations that define sequences recursively: each term of the sequence is defined as a function of the preceding terms. They are widely used in computer science, mathematics, and economics for modelling problems.

Polya's Enumeration Theorem is a combinatorial method used to count the number of distinct objects under group actions, particularly symmetries. It generalises the concept of counting distinct objects that are invariant under a set of symmetries.

5.1 Introduction to Combinatorics

It's common to sum up combinatorics as being about counting and in fact, counting is a big component of combinatorics. But as the name implies, it goes beyond this; it's really about mixing things. Counting problems come up, such as "How many ways can these elements be combined?" However, there are other queries, such whether a specific combination is feasible or which combination is, in some way, the "best." All of these will be seen, but numbering is especially important.

Different kinds of networks, or more accurately, graphs, which are models of networks, are the subject of graph theory. Instead of the graphs of analytical geometry, they are what are commonly referred to as "points connected by lines," such as:



5.1.1 Partitions

A partition of a positive integer, also known as an integer partition, is a method to express n as the sum of positive integers in number theory and combinatorics. A pair of sums with only their summand order as a difference are regarded as belonging to the same partition. (The sum becomes a composition if order matters.) For instance, there are five unique ways to divide 4:

4
3 + 1
2 + 2
2 + 1 + 1
1 + 1 + 1 + 1

The composition that varies with order While the two different compositions, 1 + 2 + 1 and 1 + 1 + 2, indicate the same partition, 2 + 1 + 1, 1 + 3 is the same as 3 + 1.

A part is another name for a summand in a partition. The partition function p(n) yields the number of partitions of n. Consequently, p(4) = 5. λ is a partition of n, as indicated by the notation λn.

Young diagrams and Ferrers diagrams are two ways to graphically represent partitions.

They can be found in many areas of mathematics and science, such as the study of symmetric groups and polynomials, as well as group representation theory in general.

Example:

The seven partitions of 5 are:

- 5
- 4 + 1
- 3 + 2
- 3 + 1 + 1
- 2 + 2 + 1
- 2 + 1 + 1 + 1
- 1 + 1 + 1 + 1 + 1

Partitions are not represented as an expression with plus signs in certain sources, but rather as a series of summands. The partition 2 + 2 + 1, for instance, could instead be expressed as the tuple (2, 2, 1) or in the even more compact form (2^2, 1), where the superscript denotes the number of times a term appears.

Partition of a set:

A partition of a set X is a collection of non-empty subsets of X such that X is a disjoint union of the subsets and each element x in X is in exactly one of these subsets.

Notes

Comparatively, if and only if each of the subsequent circumstances is true, a family of sets P is a partition of X.

- The empty set (that is, $\varnothing \notin P$) is absent from the family P.
- X is the result of adding all of the sets in P. X is said to be covered by the sets in P.
- When two unique sets in P cross each other, it remains empty. Pairwise disjoint elements are those in P.

The sets in P are called the blocks, parts or cells of the partition. The rank of P is $|X| - |P|$, if X is finite.

Examples:

- The empty set has exactly one partition, namely for any non-empty set X, P = {X} is a partition of X, called the trivial partition.
- Particularly, every singleton set {x} has exactly one partition, namely { {x} }.
- For any non-empty proper subsetA of a set U, the set A together with its complement form a partition of U.
- The set { 1, 2, 3 } has these five partitions (one partition per item):
  - o { {1}, {2}, {3} }, sometimes written 1|2|3.
  - o { {1, 2}, {3} }, or 12|3.
  - o { {1, 3}, {2} }, or 13|2.
  - o { {1}, {2, 3} }, or 1|23.
  - o { {1, 2, 3} }, or 123 (in contexts where there will be no confusion with the number).
- The following are not partitions of { 1, 2, 3 }:
  - o { {}, {1, 3}, {2} } is not a partition (of any set) because one of its elements is the empty set.
  - o { {1, 2}, {2, 3} } is not a partition (of any set) because the element 2 is contained in more than one block.
  - o { {1}, {2} } is not a partition of {1, 2, 3} because none of its blocks contains 3; however, it is a partition of {1, 2}.

Partitions of n:

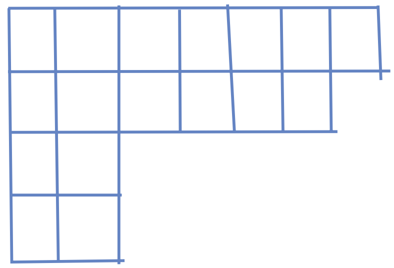
An unordered combination of positive integers, known as the parts, that add up to n is called a partition of n. Repetition is permitted.

Stated otherwise, a partition can be defined as a multiset of positive integers and if the multiset's integer sum equals n, then the partition is of size n. The components of a partition are typically written in descending order.

Ferrers Diagram and Conjugate Partition:

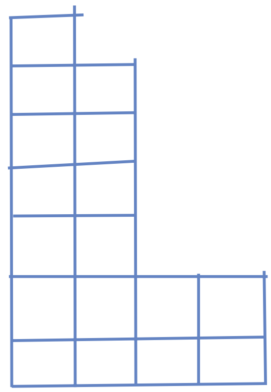
The Ferrers diagram, also called Young diagram, of a partition $\lambda \vdash n$ is a rectangular array of n boxes, or cells, with one row of length j for each part j of λ. For example, the diagram of (7,5,2,2) is

Notes



The conjugate of a partition λ +n is the partition of n whose diagram you get by reacting the diagram of λ about the diagonal so that rows become columns and columns become rows. We use the notation for the conjugate of λ.

In our example above, with λ =(7,5,2,2), the diagram of



5.1.2 Counting Functions

Combinatorics can be traced back more than 3000 years to India and China. For many centuries, it primarily comprised the solving of problems relating to the permutations and combinations of objects. The use of the word "combinatorial" can be traced back to Leibniz in his dissertation on the art of combinatorial in 1666. Over the centuries, combinatorics evolved in recreational pastimes.

These include the Konigsberg bridges problem, the four- colour map problem, the Tower of Hanoi, the birthday paradox and Fibonacci's 'rabbits' problem. In the modern era, the subject has developed both in depth and variety and has cemented its position as an integral part of modern mathematics.

Undoubtedly part of the reason for this importance has arisen from the growth of computer science and the increasing use of algorithmic methods for solving real-world practical problems. These have led to combinatorial applications in a wide range of subject areas, both within and outside mathematics, including network analysis, coding theory and probability.

As per the Fundamental Counting Principle, if there are m · n potential outcomes for one event and n possible outcomes for another, then the total possible outcomes for the two events combined are m · n. The amount of ways an object can be selected from a total of objects (order does not matter) is called a combination.

Addition and multiplication rules:

1) What is the total number of possible crossword puzzles?

   Let's say we have to choose four balls from a bag of twenty, numbered 1 through twenty.

Notes

2) How frequently do two balls that were chosen have consecutive numbers? The word alphabet can be rearranged in how many different ways?

We note several points regarding the aforementioned issues. Unlike many mathematical issues, a priori rarely uses abstract or specialised jargon. Even though they seem simple at first, several of these issues will be quite difficult to resolve. Furthermore, we observe that these challenges usually consist of choosing, organising and counting different kinds of things, even though they seem diverse and unconnected. We'll deal with the counting problem. It goes without saying that we want to be able to count without really counting.

Put differently, is it possible to determine the quantity of objects possessing a particular attribute without listing each one individually? This frequently requires profound mathematical understanding. Now, we will provide two widely used methods that are excellent for counting without really counting. The following examples are a great way to become motivated about these strategies.

Example:

Let the cars in New Delhi have license plates containing 2 alphabets followed by two numbers. What is the total number of license plates possible? Solution: Here, we observe that there are 26 choices for the first alphabet and another 26 choices for the second alphabet.

After this, there are two choices for each of the two numbers in the license plate.

Hence, we have a maximum of 26× 26 × 10 × 10 = 67600 license plates.

Example:

Let the cars in New Delhi have license plates containing 2 alphabets followed by two numbers with the added condition that \in the license plates that start with a vowel the sum of numbers should always be even". What is the total number of license plates possible?

Solution:

Here, we need to consider two cases.

Case 1:

The license plate doesn't start with a vowel. Then using the previous example, the number of license plates equals

21× 26× 10×10 = 54600

Case 2:

The license plate starts with a vowel. Then the number of license plates equals

5 × 26 × (5× 5+5 × 5) = 6500.

Hence, we have a maximum of 54600 + 6500 = 61100 license plates.

Generalisation of the first example leads to what is referred to as the rule of product and that of the second leads to the rule of addition. To understand these rules, we explain the involved ideas.

Suppose we have a task to complete and that the task has some parts. Assume that each of the parts can be completed on their own and completion of one part does not result in the completion of any other part. We say the parts are compulsory to mean that each of the parts must be completed to complete the task. We say the parts are alternative to mean that exactly one of the parts must be completed to complete the task. With this setting we state the two basic rules of combinatorics.

Basic counting rule:

Let $n, m_1, \ldots, m_n \in N$.

Multiplication/Product rule:

If a task consists of n compulsory parts and the i-th part can be completed in m_i ways, i = 1, 2,…..n, then the task can be completed in $m_1, m_2, \ldots\ldots, m_n$ ways.

Addition rule:

If a task consists of n alternative parts and the i-th part can be completed in mi ways, i = 1; : : : ; n, then the task can be completed in

$m_1 + m_2 + \ldots.. + m_n$ ways.

Let us consider the following examples.

1. How many three-digit natural numbers can be formed using digits

0,1,….., 9?

Identify the number of parts in the task and the type of the parts (compulsory or alternative).

Solution:

The task of forming a three digit number can be viewed as lining three boxes kept in a horizontal row. It has three compulsory parts.

Part 1:

Choose a digit for the left most place.

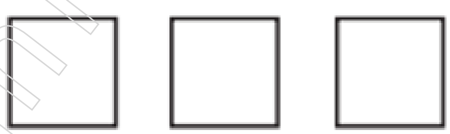
Part 2:

Choose a digit for the middle place.

Part 3:

Choose a digit for the rightmost place.

Apply multiplication rule.



$9 \times 10 \times 10 = 900$

Prime Counting Function:

The prime counting function is the function $\pi(x)$ giving the number of primes less than or equal to a given number 'x'. For example, there are no primes ≤ 1, so $\pi(1) = 0$. There is a single prime (2) ≤ 2, so $\pi(2) = 1$. There are two primes (2 and 3) ≤ 3, so $\pi(3) = 2$. And so on.

The notation $\pi(n)$ for the prime counting function is slightly unfortunate because it has nothing whatsoever to do with the constant $\pi = 3.1415 \ldots$. This notation was introduced by

Notes

number theorist Edmund Landau in 1909 and has now become standard. In the words of Derbyshire (2004, p. 38), “I am sorry about this; it is not my fault. You’ll just have to put up with it.”

Letting p_n denote the ‘n’th prime, p_n is a right inverse of $\pi(n)$ since

$$\pi(Pn) = n \tag{1}$$

for all positive integers. Also,

$$p_{\pi(n)} = n \tag{2}$$

iff ‘n’is a prime number.

The first few values of $\pi(n)$ for $n = 1$, 2, ... are 0, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, 6, ... The Wolfram Language command giving the prime counting function for a number $x$is PrimePi[x], which works up to a maximum value of $x \approx 8 \times 10^{13}$.

The notation $\pi_{a,b}(x)$ is used to denote the modular prime counting function, i.e., the number of primesof the form $a\,k + b$ less than or equal to x.

One of the most fundamental and important results in number theory is the asymptotic form of $\pi(n)$ as $n$becomes large. This is given by the prime number theorem, which states that

$$\pi(n) \sim \mathrm{Li}(n), \tag{3}$$

where $\mathrm{Li}(x)$ is the logarithmic integral and $\sim$is asymptotic notation. This relation was first postulated by Gauss in 1792.

Prime Counting Function:

The prime counting function can be expressed by Legendre’s formula, Lehmer’s formula, Mapes’ method, or Meissel’s formula. A brief history of attempts to calculate $\pi(n)$ is given by Berndt (1994). The following table is taken from Riesel (1994), where $O(x)$ is asymptotic notation.

An approximate formula due to illustrated above, is given by

$$\pi(n) \approx \frac{n}{h_n} \tag{4}$$

where h_n is related to the harmonic number H_n by $h_n = H_n - 3/2$. This formula is within ≈ 2 of the actual value for $50 \le n \le 1000$. The values for which $\pi(n) - n/h_n > 0$ are 1, 109, 113, 114, 199, 200, 201, ... This quantity is positive for all $n \ge 1429$.

An upper bound for $\pi(n)$ is given by

$$\pi(n) < \frac{1.25506\ n}{1n_n} \tag{5}$$

for $n > 1$, and a lower bound by

$$\frac{n}{1n_n} < \pi(n) \tag{6}$$

for $n \ge 17$.

It gives the formula

$$\pi(n) = -1 + \sum_{j=3}^{n} \left[(j-2)! - j \left[\frac{(j-2)!}{j} \right] \right] \tag{7}$$

for $n > 3$, where $\lfloor x \rfloor$is the floor function.

A modified version of the prime counting function is given by

$$\pi_0(p) \equiv \begin{cases} \pi(p) & \text{for } p \text{ composite} \\ \pi(p)-\frac{1}{2} & \text{for } p \text{ rime} \end{cases} \quad (8)$$

$$\pi_0(p) \equiv \sum_{n=1}^{\infty} \frac{\mu(x) f(x^{1/n})}{n} \quad (9)$$

where $\mu(n)$ is the Möbius function and $f(x)$ is the Riemann prime counting function.

It also shows that

$$\frac{d\pi(x)}{dx} \sim \frac{1}{x1nx} \sum_{n=1}^{\infty} \frac{\mu(n)}{n} x1/n \quad (10)$$

where $\mu(n)$is the Möbius function.

The smallest $x$such that $x \geq n\,\pi(x)$ for $n = 2$, 3, ... are 2, 27, 96, 330, 1008, ..., and the corresponding $\pi(x)$ are 1, 9, 24, 66, 168, 437, ... The number of solutions of $x = n\,\pi(x)$ for $n = 2$, 3, ... are 4, 3, 3, 6, 7, 6, ...

It shows that for sufficiently large x,

$$\pi^2(x) < \frac{ex}{1nx} \pi\left(\frac{x}{e}\right) \quad (11)$$

This holds for $x = 6$, 9, 10, 12, 14, 15, 16, 18, ... (OEIS A091886). The largest known prime for which the inequality fails is 38358837677. The related inequality

$$[1i(x)]^2 < \frac{ex}{1nx} 1i\left(\frac{x}{e}\right) \quad (12)$$

where

$$li(x) \int_0^x \frac{dt}{\ln t} \quad (13)$$

is true for $x \geq 2418$ and no smaller number.

Riemann Counting function:

Riemann defined the function $f(x)$ by

$$f(x) \equiv \sum_{\substack{p^v sx \\ p \text{ prime}}} \frac{1}{v} \quad (1)$$

$$= \sum_{n=1}^{[1gx]} \frac{\pi\left(x^{1/n}\right)}{n} \quad (2)$$

$$= \pi(x) + \frac{1}{2}\pi\left(x^{1/2}\right) + \frac{1}{3}\pi\left(x^{1/3}\right) + \ldots \quad (3)$$

Amazingly, the prime counting function $\pi(x)$ is related to $f(x)$ by the Möbius transform

$$\pi(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} f(x^{1/n}) \quad (4)$$

Notes

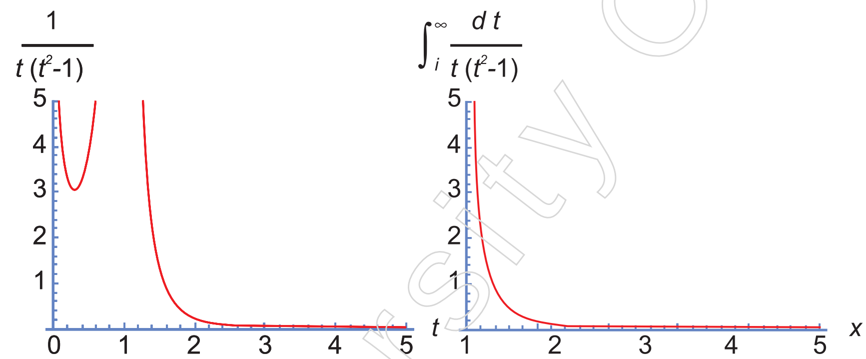
where $\mu(n)$ is the Möbius function. More amazingly still, $f(x)$ is connected with the Riemann zeta function $\zeta(s)$ by

$$\frac{\ln[\zeta(s)]}{s} = \int_0^{\infty} f(x) x^{-s-1} dx \tag{5}$$

$f(x)$ is also given by

$$f(x) = \lim_{t\to\infty} \frac{1}{2\pi i} \int_{2-iT}^{2+iT} \frac{x^s}{x} \ln\zeta(s) ds \tag{6}$$

where $\zeta(z)$ is the Riemann zeta function, and (5) and (6) form a Mellin transform pair.



Riemann (1859) proposed that

$$f(x) = \mathrm{li}(x) - \sum_{\rho} \mathrm{li}(x^{\rho}) - \ln 2 + \int_x^{\infty} \frac{dt}{t(t^2-1)\ln t^5} \tag{7}$$

where $\mathrm{li}(x)$ is the logarithmic integral and the sum is over all nontrivial zeros ρ of the Riemann zeta function $\zeta(z)$.

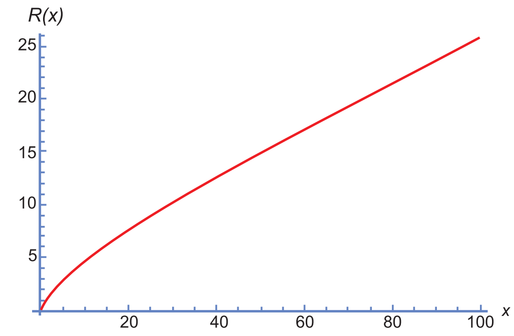
Actually, since the sum of roots is only conditionally convergent, it must be summed in order of increasing $\mathrm{I}[\rho]$ even when pairing terms ρ with their “twins” $1-\rho$, so

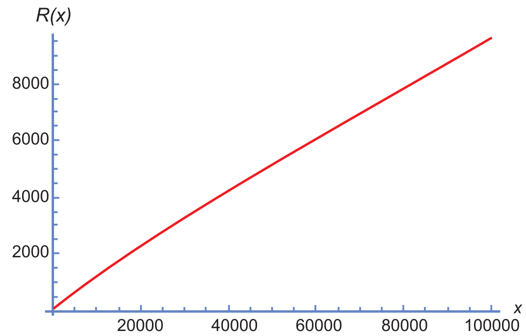
$$\sum_{\rho} \mathrm{li}(x^{\rho}) = \sum_{\mathrm{I}(\rho)>0} \left[Li(x^{\rho}) + Li(x^{1-\rho}) \right] \tag{8}$$

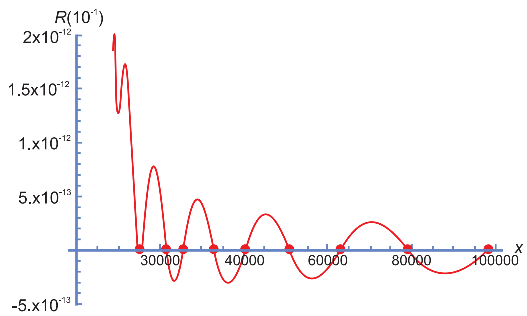
This formula was subsequently proved. The integral on the right-hand side converges only for $x > 1$, but since there are no primes less than 2, the only values of interest are for $x \geq 2$. Since it is monotonic decreasing, the maximum therefore occurs at $x = 2$, which has value

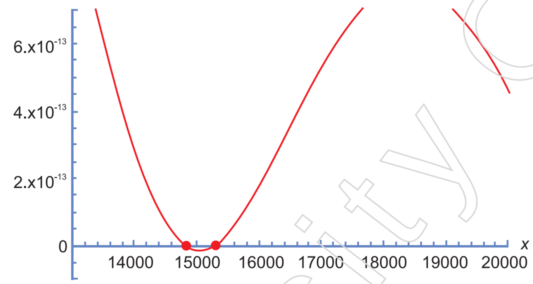
$$\int_2^{\infty} \frac{dt}{t \ln t (t^2-1)} = 0.14001010114328692668... \tag{9}$$

Notes





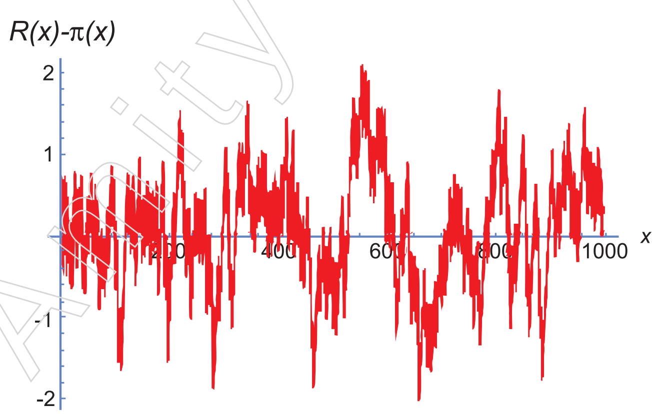




Riemann also considered the function

$$R(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} \mathrm{l}i(x^{1/n}) \qquad (10)$$

Sometimes also denoted $\mathrm{Ri}(x)$ obtained by replacing $f(x^{1/n})$ in the Riemann function with the logarithmic integral $\mathrm{l}i(x^{1/n})$, where ζ (z) is the Riemann zeta function and μ(n) is the Möbius function which illustrate the fact that *R*(x) has a series of zeros near the origin. These occur at 10^{-x} for x=14827.7, 15300.7, 21381.5, 25461.7, 32711.9, 40219.6, 50689.8, 62979.8, 78890.2, 98357.8, ..., corresponding to x = 1.829 X 10^{-14828}, 2.040 X 10^{-15301}, 3.289 X 10^{-21382}, 2.001 X 10^{-25462}, 1.374 X 10^{-32712}, 2.378 X 10^{-40220}, 1.420 X 10^{-50690}, 1.619 X 10^{-62980}, 6.835 X 10^{-78891}, 1.588 X 10^{-98358}, ...

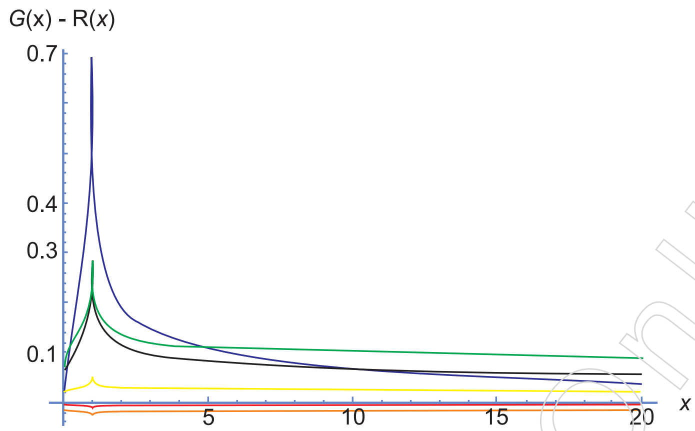


The quantity $R(x) - \pi(x)$ is plotted above.

This function is implemented in the Wolfram Language as Riemann R[x].

Ramanujan independently derived the formula for $R(n)$, but non rigorously. The following table compares $\pi(10^n)$ and $R(10^n)$ for small n. Riemann conjectured that $R(n) = \pi(n)$, but this was disproved by Littlewood in 1914.

Notes



The Riemann prime counting function is identical to the Gram series

$$G(x) = 1 + \sum_{k=1}^{\infty} \frac{(1nx)^k}{kk!\zeta(k+1)} \quad (11)$$

where $\zeta(z)$ is the Riemann zeta function, but the Gram series is much more tractable for numeric computations. For example, the plots above show the difference $G(x) - R(x)$ where $R(x)$ is computed.

5.1.3 Number of Partitions into Odd or Unequal Parts

The number of partitions with odd parts, q(n), is the same as the number of partitions with distinct parts for every positive number. Leonhard Euler demonstrated this outcome in 1748 and Glaisher's theorem was subsequently developed to generalise it.

Among the 22 partitions of the number 8, there are 6 that contain only odd parts:

- 7 + 1
- 5 + 3
- 5 + 1 + 1 + 1
- 3 + 3 + 1 + 1
- 3 + 1 + 1 + 1 + 1 + 1
- 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1

Alternatively, we could count partitions in which no number occurs more than once. Such a partition is called a partition with distinct parts. If we count the partitions of 8 with distinct parts, we also obtain 6:

- 8
- 7 + 1
- 6 + 2
- 5 + 3
- 5 + 2 + 1
- 4 + 3 + 1

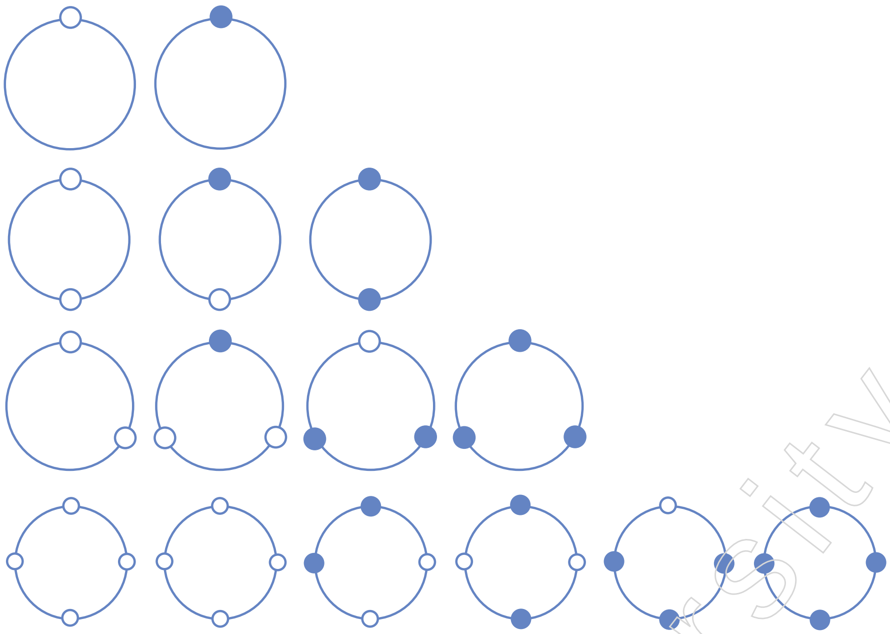
This is a general property. For each positive number, the number of partitions with odd parts equals the number of partitions with distinct parts, denoted by q(n).

For every type of restricted partition there is a corresponding function for the number of partitions satisfying the given restriction. An important example is q(n).

5.1.4 Necklaces

Notes

Reconstructing necklaces (cyclic configurations of binary values) from incomplete information is the subject of the necklace problem in recreational mathematics.



In the technical combinatorial sense, an a-ary necklace of length n is a string of 'n' characters, each of a possible types. Rotation is ignored, in the sense that $b_1\, b_2 \ldots b_n$ is equivalent to $b_k\, b_{k+1} \cdots b_n\, b_1\, b_2 \cdots b_{k-1}$ for any k.

In fixed necklaces, reversal of strings is respected, so they represent circular collections of beads in which the necklace may not be picked up out of the plane (i.e., opposite orientations are not considered equivalent). The number of fixed necklaces of length n composed of a types of beads $N(n, a)$ is given by

$$N(n,a) = \frac{1}{n}\sum_{i=1}^{v(n)} \phi(d_i)a\, n/d_i \qquad (1)$$

where d_i are the divisors of n with $d_1 \equiv 1, d_2, \ldots, d_v(n) \equiv n$, $v(n)$ is the number of divisors of n, and $\phi(x)$ is the totient function.

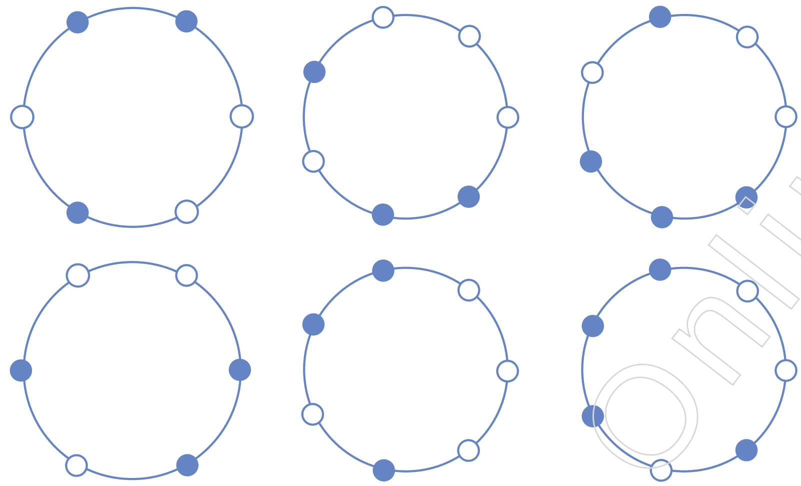
For free necklaces, opposite orientations (mirror images) are regarded as equivalent, so the necklace can be picked up out of the plane and flipped over. The number $N'(n, a)$ of such necklaces composed of n beads, each of a possible colors, is given by

$$N'(n,a) = \frac{1}{2n}\begin{cases} \sum_{i=1}^{v(n)} \phi(d_i)a^{n/d_i} + na^{(n+1)/2} & \text{for } n \text{ odd} \\ \sum_{i=1}^{v(n)} \phi(d_i)a^{n/d_i} + \frac{1}{2}n(1+a)a^{n/2} & \text{for } n \text{ even} \end{cases} \qquad (2)$$

For $a = 2$ and $n = p$ an odd prime, this simplifies to

$$N'(\rho,2) = \frac{2^{\rho-1}-1}{\rho} + 2^{(\rho-1)/2} + 1 \qquad (3)$$

Notes



A table of the first few numbers of necklaces for $a = 2$ and $a = 3$ follows. Note that $N(n, 2)$ is larger than $N'(n, 2)$ for $n \geq 6$. For $n = 6$, the necklace 110100 is inequivalent to its mirror image 001011, accounting for the difference of 1 between $N(6, 2)$ and $N'(6, 2)$.

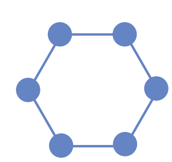
Similarly, the two necklaces 0010110 and 0101110 are inequivalent to their reversals, accounting for the difference of 2 between $N(7, 2)$ and $N'(7, 2)$.

n	*N (n, 2)*	*N' (n, 2)*	*N' (n, 3)*
Sloane	**A000031**	**A000029**	**A027671**
1	2	2	3
2	3	3	6
3	4	4	10
4	6	6	21
5	8	8	39
6	14	13	92
7	60	46	1219
8	108	78	3210
9	188	126	8418
10	352	224	22913
11	632	380	62415
12	1182	687	173088
13	2192	1224	481598

Let's look at the problem of determining how many different ways there are to arrange n persons in a circle so that no one has the same two neighbors more than once. There are 21 such combinations for 8 individuals.

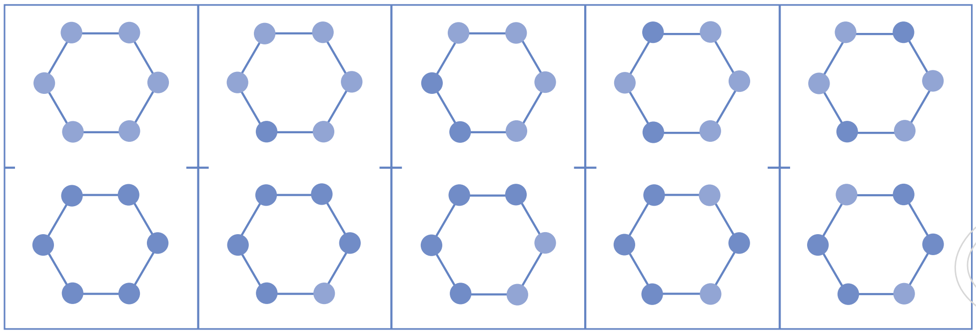
Example:

In his store, a jeweler offers six-beaded necklaces for sale. How many different kinds of necklaces must he make given the two different colours of beads in order to have every possible combination of colours?



Notes

It is possible to colour an immovable beaded necklace in 64 different ways, as we have two colour options for each of the six beads. But if we are allowed to turn and mirror it, there are just 13 different kinds. Additionally, there are just 8 different configurations if the colours are interchangeable—that is, a totally light necklace is equal to a completely dark necklace.



The set of necklaces with 0, 1, or 2 beads of one colour.

One could speculate that it has to do with symmetry. Twelve symmetries make up the hexagon: three reactions through opposed vertices, three reactions through opposite sides and rotation by 0, 60, 120, 180, 240, or 300 degrees. The number of symmetries, the number of unique variants and the number of possible red necklaces do not appear to be related, yet there undoubtedly is. Here, we solve the six-beaded necklace problem and similar ones using group theory and combinatorics.

The study of symmetry is one of the most significant and elegant subjects that unites several fields of contemporary mathematics. Many of us have an innate understanding of symmetry and we frequently consider some forms or patterns to be more or less symmetrical than others. In a way, a square is “more symmetric” than a rectangle, which is “more symmetric” than any random four-sided form. Is it possible for us to clarify these concepts?

Group theory investigates general approaches to investigating symmetry in a variety of different settings. It is the mathematical study of symmetry. numerous of the numerous subjects we have previously studied are connected by group theory, such as sets, cardinality, number theory, isomorphism and modular arithmetic.

5.1.5 Euler’s Function

By counting the numbers smaller than a given number, n, that are co-prime (that is, have no positive factor in common with n other than 1), the Euler’s Totient Function calculates. Thus, there are four integers that are co-prime with eight, namely 1, 3, 5 and 7.

Zn, it is often useful to inquire as to if something similar holds true for Un. Here, we examine Un within the framework of the preceding section. We present a new number, the Euler phi function, denoted υ(n) for positive integers n, to facilitate the analysis.

Definition:

The number of non-negative integers smaller than n that are comparatively prime to n is denoted by υ(n). Stated differently, $\phi(n)$ represents the number of items in Un if n>1 and $\phi(1)=1$.

Example:

$\phi(2)=1$, $\phi(4)=2$, $\phi(12)=4$ and $\phi(15)=8$

Notes

Example:

Since 0 is not a prime number and 1, 2,..., p−1 are all relatively prime to p, if p is a prime, then $\phi(p)=p-1$.

It turns out that $\phi(n)$ has an astonishingly simple form for every number n; that is, $\phi(n)$ can be found using a straightforward formula. For primes, we've already shown how easy it is. We shall progressively approach any n, as is common with many results in number theory, by first examining powers of a single prime.

Euler's φ function:

$\varphi(n)$ is the number of integers $m \in [1,n]$ with m coprime to n.

Or, it is the order of the unit group of the ring **Z**/n**Z.**

Euler:

If a is coprime to n, then

$$a^{\varphi(n)} \equiv 1 (\mathrm{mod}\, n)$$

Euler's theorem is the basis of the RSA Cryptosystem:

If integers *E,D* satisfy $ED \equiv 1 \pmod{\varphi(n)}$, then

$$a^{ED} \equiv a \pmod{n},$$

For every integer a coprime to n, (In fact, this holds for all integers a if n is square free, such as the product of two different large primes.)

Encrypt message "a" : $b \equiv a^E \pmod{n}$,

Decrypt: $a = b^D \pmod{n}$.

To encrypt, one should know *E* ,n. To decrypt, *D* as well.

Note that it is easy to compute a^E (mod n) given a, *E*, n and similarly it is easy to compute b^D(mod n).

Further, given $\varphi(n)$, it is easy to come up with pairs *E,D* with

$ED \equiv 1 \pmod{\varphi(n)}$. Indeed, keep choosing random numbers *D* until one is found that is coprime to $\varphi(n)$, and then use Euclid's algorithm to find *E.*

As a public-key system, *E*, n are released to the public, but *D* is kept secret. Then anyone can send you encrypted messages that only you can read.

The security of the RSA Cryptosystem is connected to our ability to compute $\varphi(n)$:

For *p* prime, $\varphi(p) = p-1$; more generally,

$\varphi(p) = p^k - p^{k-1} = p^k(1-1/p)$.

By the **Chinese Remainder Theorem,**

$$\varphi(n) = n \prod_{\substack{p\,prime \\ p/n}} (1-1/p).$$

So, knowing the prime factorization of n, one can compute $\varphi(n)$ rapidly—in deterministic polynomial time.

What about the converse? Given n and $\varphi(n)$ can one compute the prime factorization of n in deterministic polynomial time?

Yes, if $n=pq$, where *p* ,*q* are different primes, since

φ (n) = (*p*-1) (*q*-1), so that n-1-φ (n) = *p* + *q*.

In general, the Extended Riemann Hypothesis implies that there is a deterministic, polynomial time algorithm to compute the prime factorization of n, given n and φ (n).

A random polynomial time algorithm to get a nontrivial factorization:

We may assume φ (n) < n-1 and that n and φ (n) are coprime.

Write φ (n) = 2^s m, where m is odd.

Choose a at random from [1,n-1]. We may assume a and n are coprime.

Note that,

$a^{\phi(n)} - 1 \equiv (a^m - 1)(a^m + 1)(a^{2m} + 1)...(a^{2s-1} + 1)$

Theorem:

If *p*is a prime and *a* is a positive integer, then

$\phi(p^a)=p^a-p^a-1$

Proof:

The number of non-negative integers less than $n=p^a$ that are relatively prime to *n*. As in many cases, it turns out to be easier to calculate the number that are *not* relatively prime to *n*, and subtract from the total. List the non-negative integers less than *pa*: 0, 1, 2, …, p^{a-1}; there are p^a of them.

The numbers that have a common factor with p^a (namely, the ones that are not relatively prime to *n*) are the multiples of *p*: 0, *p*, 2*p*, …, that is, every *p* th number. There are thus $p^{a/p=pa-1}$ numbers in this list,

so $\phi(p^a)=p^a-p^a-1$

Example:

$\phi(32)=32-16=16$,

$\phi(125)=125-25=100$

Example:

Since

U20={[1],[3],[7],[9],[11],[13],[17],[19]},

U4={[1],[3]},

U5={[1],[2],[3],[4]}.

both U20 and U4×U5 have 8 elements.

In fact, the correspondence discussed in the Chinese Remainder Theorem between Z20 and Z4×Z5 is also a 1-1 correspondence between U20 and U4×U5:

[1] ↔ ([1],[1])

[3] ↔ ([3],[3])

[7] ↔ ([3],[2])

[9] ↔ ([1],[4])

[11] ↔ ([3],[1])

Notes

$[13] \leftrightarrow ([1],[3])$

$[17] \leftrightarrow ([1],[2])$

$[19] \leftrightarrow ([3],[4])$

Theorem:

If a and b are relatively prime and $n=ab$, then $\phi(n)=\phi(a)\phi(b)$

Proof.

We want to prove that $|Un|=|Ua|\cdot|Ub|$

As indicated in the example, we will actually prove more, by exhibiting a one to one correspondence between the elements of Un and Ua×Ub.

We already have a one to one correspondence between the elements of Zn and Za×Zb. Again as indicated by the example, we just have to prove that this same correspondence works for Un and Ua×Ub.

That is, we already know how to associate any $[x]$ with a pair $([x],[x])$;

we just need to know that $[x]\in Un$ if and only if $([x],[x])\in Ua\times Ub$.

$\Rightarrow [x]$ is in Un if and only if $(x,n)=1$ if and only if $(x,a)=1$ and $(x,b)=1$ if and only if $([x],[x])\in Ua\times Ub$

Euler's function on average

The chance that two random integers are both divide by the prime p is $1/p^2$.

Should be

$$\alpha := \prod_{pprime}(1-1/p^2).$$

We have

$$1-1/p^2 = \left(\frac{p^2}{p^2-1}\right)^{-1} = \left(1+\frac{1}{p^2}+\frac{1}{p^2}+...\right)^{-1}$$

So that

$$\alpha = \left(\sum_{n=1}^{\infty}\frac{1}{n^2}\right)^{-1} = \frac{1}{\zeta(2)} = \frac{6}{\pi^2}.$$

Thus the probability that two random integers are coprime is $\frac{6}{\pi^2}$.

Another interpretation: choose a lattice point (m, n) at random, the probability that it is "visible" is $\frac{6}{\pi^2}$.

Using these thoughts it is easy to see that

$$\sum_{n\le x}\varphi(n) \approx \frac{3}{\pi^2}x^2$$

As x $\longrightarrow \propto$.

Extreme values of φ (n) are also fairly easy:

$$\limsup \frac{\varphi(\mathrm{n})}{\mathrm{n}} = 1,$$

$$\liminf \frac{\varphi(\mathrm{n})}{\mathrm{n}} = 0,$$

in fact,

$$\liminf \frac{\varphi(n)}{n/\log\log n} = e^{-\gamma}$$

Notes

Clearly $\frac{\varphi(\mathrm{n})}{\mathrm{n}}$ jumps around a bit: if n has only large prime factors, the ratio is close to 1, but if n has many small prime factors, it is close to 0.

One can ask for a "distribution function". That is, let u be a real variable in [0,1] and consider.

$$\{\mathrm{n}: \frac{\varphi(\mathrm{n})}{\mathrm{n}} /\mathrm{n} \leq \mathrm{u}\}.$$

Euler's Totient Function:

The role of the totient: The number of positive integers ≤ that are relatively prime to (i.e., do not share any factor with) n is defined as Ø(n), commonly known as Euler's totient function, where 1 is counted as being relatively prime to all numbers.

The totient function Ø(n) can be easily defined as the number of totatives of n, as a totative is a number that is less than or equal to and relatively prime to a given integer. For instance, Ø(24)=8 because there are eight totatives of 24 (1, 5, 7, 11, 13, 17, 19 and 23).

The Euler Phi[n] provides an implementation of the totient function.

The cototient of n is represented by the number n-Ø(n), which indicates how many positive integers ≤n share at least one prime factor in common with n.

For all values of n less than 3, Ø(n) is always even. Although the Euler Phi[0] equals 0 for compatibility with its Factor Integer[0] command, by convention, Ø(0)=1. Ø(n) for n = 1, 2,... has the following initial values: 1, 1, 2, 2, 4, 2, 6, 4, 6, 4, 10,... Plotting Ø(n) above for tiny n yields the totient function, which is given by the Möbius transform of 1, 2, 3, 4,....

Regarding a prime P,

Ø (p) is equal to p – 1.

Given that any integers smaller than P are comparatively prime to P. If m=P^a is a power of a prime, then the numbers that have a common factor with m are the multiples of p: p, 2 p, ...,(P^{a-1}) p . There are $P^{(a-1)}$ of these multiples, so the number of factors relatively prime to P^a is

$$\phi(p^{\alpha}) \quad = \quad p^{\alpha} - p^{\alpha-1} \qquad (2)$$

$$= \quad p \quad (p \quad 1) \qquad (3)$$

$$= \quad p^{\alpha}\left(1 - \frac{1}{p}\right) \qquad (4)$$

Now let q be some other prime dividing m . The integers divisible by q are ,q ,2 q, ...,(m/q) q . But these duplicate , p q, 2 p q, ...,(m/(p q)) p q.

So the number of terms that must be subtracted from φ p to obtain φ p q is

$$\phi_p(m) \quad = \quad m - \frac{m}{p} \qquad (5)$$

Notes

$$= \quad m\left(1-\frac{1}{p}\right) \quad (6)$$

Now let q be some other prime dividing m. The integers divisible by q are q, $2q$, ..., $(m/q)q$. But these duplicate pq, $2pq$, ..., $(m/(pq))pq$.

So the number of terms that must be subtracted from ϕ_p to obtain ϕ_{pq} is

$$\Delta\phi_q(m) = \frac{m}{q}-\frac{m}{pq} \quad (7)$$

$$= \frac{m}{q}\left[1-\frac{1}{p}\right] \quad (8)$$

and

$$\phi_{pq}(m) = \phi_p(m)-\Delta\phi_q(m) \quad (9)$$

$$= m\left(1-\frac{1}{p}\right)-\frac{m}{q}\left(1-\frac{1}{p}\right) \quad (10)$$

$$= m\left(1-\frac{1}{p}\right)\left(1-\frac{1}{q}\right) \quad (11)$$

By induction, the general case is then

$$\phi(n) = n\prod_{p|n}\left(1-\frac{1}{p}\right) \quad (12)$$

$$= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)....\left(1-\frac{1}{p_r}\right) \quad (13)$$

where the product runs over all primes p dividing n. An interesting identity relating $\phi(n^k)$ to $\phi(n)$ is given by

$$\phi(n^k) = n^{k-1}\phi(n) \quad (14)$$

Another identity relates the divisors d of n to n via

$$\sum_{d/n}\phi(d) = n \quad (15)$$

The totient function is connected to the Möbius function $\mu(n)$ through the sum

$$\sum_{d} d\,\mu\left(\frac{n}{d}\right) = \phi(n) \quad (16)$$

where the sum is over the divisors of n, which can be proven by induction on n and the fact that $\mu(n)$ and $\phi(n)$ are multiplicative.

The totient function has the Dirichlet generating function

$$\sum_{n=1}^{\infty}\frac{\phi(n)}{n^s} = \frac{\zeta(s-1)}{\zeta(s)} \quad (17)$$

Notes

The totient function satisfies the inequality

$$\phi(n) \geq \sqrt{n} \tag{18}$$

for all n except $n = 2$ and $n = 6$. Therefore, the only values of n for which $\phi(n) = 2$ are $n = 3$, 4, and 6. In addition, for composite n,

$$\phi(n) \leq n - \sqrt{n} \tag{19}$$

also satisfies

$$\liminf_{n\to\infty} \phi(n)\frac{\ln\ln n}{n} = e^{-y} \tag{20}$$

where γ is the Euler-Mascheroni constant. The values of n for which $\phi(n) < e^{-\gamma} n/(\ln\ln n)$ are given by 3, 4, 5, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 22, ..

The divisor function satisfies the congruence

$$n\sigma(n) \quad = \quad 2(\mathrm{mod}\,\phi(n)) \tag{21}$$

$$= \quad \begin{cases} 0(\mathrm{mod}\,\phi(n)) if\, \phi(n)=2 \\ 2(\mathrm{mod}\,\phi(n)) otherwise \end{cases} \tag{22}$$

for all primes $p \geq 5$ and no composite with the exception of 4, 6, and 22, where $\sigma(n)$ is the divisor function. No composite solution is currently known to

$$n - 1 \equiv 0(\mathrm{mod}\ \phi(n)) \tag{23}$$

A corollary of the Zsigmondy theorem leads to the following congruence,

$$\phi(a^n + b^n) \equiv 0(\mathrm{mod}\, n) \tag{24}$$

The first few 'n' for which

$$\phi(n) = \phi(n+1) \tag{25}$$

are given by 1, 3, 15, 104, 164, 194, 255, 495, 584, 975, ... which have common values $\phi(n) = 1$, 2, 8, 48, 80, 96, 128, 240, 288, 480, ...

The only $n < 10^{10}$ for which

$$\phi(n) = \phi(n+1) = \phi(n+2) \tag{26}$$

is n=5186, giving

$$\phi(5186) = \phi(5187) = \phi(5188) - 2^5 3^4 \tag{27}$$

Values of $\phi(n)$ shared among n that are close together include

$$\phi(25930) \quad = \quad \phi(25935) = \phi(25940) = \phi(25942) \tag{28}$$

$$= \quad 2^7 3^4 \tag{29}$$

$$\phi(404471) \quad = \quad \phi(404473) = \phi(404477) \tag{30}$$

$$= \quad 2^8 3^2 5^2 7 \tag{31}$$

then for every positive integer m, there are primes p and q such that

$$\phi(p) + \phi(q) = 2m \quad = \tag{32}$$

$$\phi(\sigma(n)) = n \tag{33}$$

where $\sigma(n)$ is the divisor function.

Notes

5.1.6 Set of Symmetries

Many people understand symmetry intuitively. Some of the shapes in the figure below seem to be more symmetrical than others. But even with our general symmetry intuitions, it might not be obvious how to make this assertion more accurate. Is it appropriate to talk about "how much" symmetry a shape has?

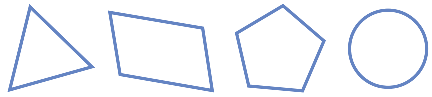
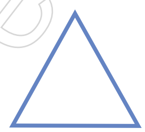


Figure: some symmetry polygons

Is there a way to clarify the claims that the circle is "more symmetric" than any regular polygon and that the regular pentagon is "more symmetric" than the equilateral triangle? This segment will examine symmetry and its emergence in diverse settings that we are acquainted with, particularly in the geometry of regular polygons (2D) and regular polyhedra (3D), like the Platonic solids. In many different branches of modern mathematics, as well as in chemistry, physics and even economics, the study of symmetry is a common issue.

If a graph remains unchanged when it is reflected over a line, then the graph is symmetric with regard to that line. The graph's axis of symmetry is denoted by this line. If a point is always on the graph and is symmetric with respect to the x-axis, then the graph is symmetric.

In order to better investigate the concept of symmetry, let's start by looking at one specific example, the equilateral triangle shown below.



Rotation Symmetries

You can spin an equilateral triangle by 120, 240, or 360 degrees without significantly altering it. You wouldn't notice anything different if you were to close your eyes and a friend rotated the triangle by one of those angles when you opened them. On the other hand, you would see that the triangle's bottom edge is no longer completely horizontal if your friend rotated it by 31 or 87.

Rotational symmetries are also present in many other shapes that are not regular polygons. For example, all the shapes shown in Figure 39 contain symmetries related to rotation. For the first example, the rotation options are limited to 180, 360, or 0.

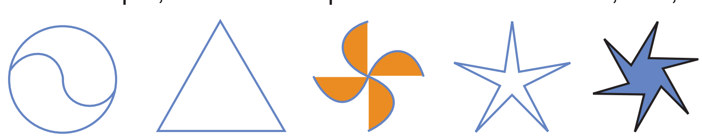


Figure: Several shapes with rotational symmetries.

It is possible to rotate the third form in any integer multiple of 90. Any whole number multiple of 72 can be used to rotate the fourth form. Any whole number multiple of 60 can be used to rotate the fifth form.

Rotational symmetry of order n, more broadly speaking, is the property that allows a form to be rotated by any multiple of 360 /n without losing its visual integrity. Other forms with rotational symmetries of any order are conceivably possible to develop. A form is said to have only the trivial (order n = 1) symmetry if the only rotations that retain its shape are multiples of 360.

Notes

Mirror Reflection Symmetries:

Mirror reflection symmetry is another kind of symmetry that can be seen in two-dimensional geometric forms. More precisely, we can reflect some shapes through a line drawn through them without altering their appearance. We refer to this as a mirror reflection symmetry.

Upon closer inspection of the equilateral triangle (refer to Figure 40), we see that there are three separate mirror lines that allow us to reflect the form without affecting its appearance.

The entire shape would appear differently if we were to reflect the triangle through any other line.

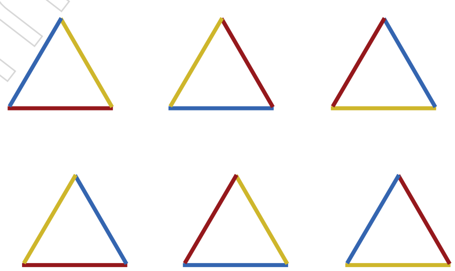
Counting Symmetries:

Counting an object's symmetries is one technique to measure the "amount" of symmetry present in it. One way to count symmetries of an object would be to count both its rotational and mirror reflection symmetries. It can be difficult to count a shape's symmetries, though. It is not immediately obvious which symmetries are appropriate to count and which ones, if any, are not.

Consider rotating the equilateral triangle by 120, 240 and 360 to get an idea of why we might not count some symmetries. Since the numbers by which we are rotating the triangle are obviously different, we could be tempted to count them individually. However, take note that the triangle can also be rotated by 480, 600 and 720 degrees. Do those qualify as distinct symmetries for us? In case we decide to count them, what would prevent us from counting an endless quantity of rotational symmetries for a triangle, or any other form for that matter?

Colouring or naming the form in some other way is one technique to reduce the number of symmetries we count. As seen in Figure, we can, for instance, colour each edge of the equilateral triangle. Then, symmetries can be represented as colour shifts that maintain the uncoloured shape fixed. Triangles in either row can be obtained from other triangles in that row by rotation, while triangles in the opposite row can be obtained from triangles in that row by reflection.

Carefully counting symmetries is made possible by using this colouring. Should two distinct methods of modifying the form yield same colouring, then the two symmetries ought to be considered identical. To count as having the same symmetry, we can rotate the equilateral triangle by 120 or 480, for example and achieve the same colouring. Equilateral triangle with coloured edges is what is produced when the triangle is rotated by 0 and 360 degrees, as well.



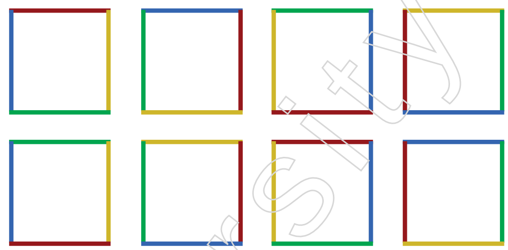
Any triangle in either row can be rotated to obtain another triangle in the row and triangles in the other row can be reflected to obtain triangles from triangles in that row. We count those in the same way because they have the same colouration. As a result, we prefer 120 to 480, despite their equivalency rotations, or "doing nothing" to 360 rotations,

Notes

despite their equivalency. To avoid confusion, we use a number between 0 and 360 (not including 360 itself) to denote the angle of a revolution. The rotations (0, 120 and 240) and three reflections—one for each of the mirror planes that travel through a corner and the triangle's center—leave us with six symmetries of the triangle. The way these symmetries look can be used to visualize how they transform the coloured triangle.

Symmetries of The Square:

Because a square has more symmetries than a triangle, it is "more symmetric" in a certain sense. A square with coloured edges arranged in various ways is depicted in the figure below. Once more, you may have observed that squares in the same row can be obtained from each other by rotations, while squares in different rows can only be obtained from each other by reflections. Upon closer examination, it becomes evident that there are no more rotations or mirror reflection symmetries; hence, these images embody all eight square symmetries.



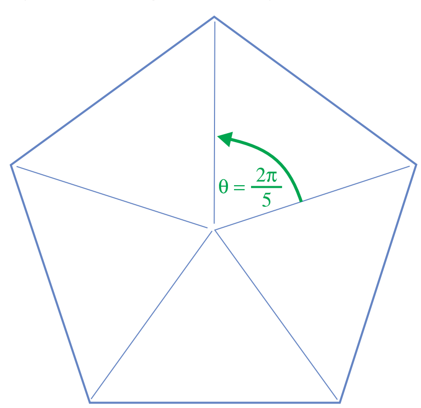
Squares

Other kinds of symmetries occur in infinite systems and in higher dimensions, even though the focus of this section is on rotational and mirror re-flection symmetries of single objects in two dimensions.

Rotational Symmetry:

A spatial configuration F is said to have rotational symmetry, in an abstract sense, if it continues to be invariant under the group C=C(F). The group of rotations of F is indicated here as C(F) and it is thought of as a subgroup of the automorphism group Γ(F) of all automorphisms that preserve F. The example of planar figures in Cartesian space provides a concept of rotational symmetry that is easier to understand.

If there is a point such that a geometric object in R^d, with 'd' being arbitrary, appears exactly the same when rotated a specific number of degrees (or radians) about that point, then that object is said to have rotational symmetry. By counting the different ways an object may be rotated to look like itself, one can further refine this idea. This count is known as the degree or order of the symmetry. A plane figure that exhibits rotational symmetry of degree is the same whether rotated by 360/n degrees or by 2π/n radians.



Notes

Because it produces the same result when rotated about its center point by a=2πn/5 radians, for n = 0, 1, 2, 3, 4 and other values, the regular pentagon in the preceding picture possesses rotational symmetry of order 5. Any regular planar n-gon possesses rotational symmetry of order n; this is just one illustration of a more general fact. For the regular planar n-gon, the set of all these symmetries is represented by the group C^n. It is isomorphic to the cyclic group Z/nZ of integers modulo n and a proper subgroup of the dihedral group D^n of all the symmetries of the figure, both rotational and otherwise.

According to the definition above, an object with rotational symmetry of degree 1 can only have symmetry about a point when it is rotated by 360/1=360 degrees. It is obvious that only symmetric objects, or those whose rotational symmetry group is trivial, satisfy this requirement. Consequently, the most basic rotational symmetry that can exist is of order 2, which can be found, for example, in planar parallelograms.

According to some literature, rotational symmetry of order n is defined as the set of outcomes obtained from rotating a figure about a line ı as opposed to a point (Weyl 1982). Specifically, according to these sources, a figure is said to possess rotational symmetry of order n if it maintains its original shape following a 2π/n radian revolution around t, also referred to as the rotational axis. It is nevertheless possible to interpret the 2π/5 -radian clockwise rotation of the pentagon about its center point in the above figure as a -radian clockwise rotation about the segment/line that is defined by the center point and the top right vertex. These two viewpoints produce the same result, though.

A -radian rotation along any line I does not change the geometry of the d -sphere S^d in the (d+1) -dimensional Cartesian space R^(d+1)since it possesses perfect rotational symmetry. Due in part to this historical circumstance, several prehistoric societies believed that the circle and/or the sphere were divine (Weyl 1982).

Due to its widespread occurrence in many naturally occurring phenomena, such as snowflakes, crystals and flowers, rotational symmetry is not only a widely understood mathematical concept but also a widely applied one.

Knot Symmetry:

A symmetry of a knot K is a homeomorphism of $\mathbb{R}^3$ which maps K onto itself. More succinctly, a knot symmetry is a homeomorphism of the pair of spaces $(\mathbb{R}^3, K)$. Hoste et al. (1998) consider four types of symmetry based on whether the symmetry preserves or reverses orienting of $\mathbb{R}^3$ and K,

1. preserves $\mathbb{R}^3$, preserves K (identity operation),
2. preserves $\mathbb{R}^3$, reverses K,
3. reverses $\mathbb{R}^3$, preserves K,
4. reverses $\mathbb{R}^3$, reverses K.

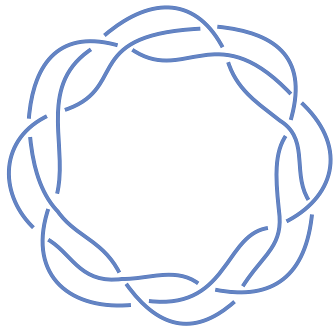
This then gives the five possible classes of symmetry summarized in the table below.

class	symmetries	knot symmetries
c	1	chiral, noninvertible
$+$	1, 3	$+$ amphichiral, noninvertible
$-$	1, 4	$-$ amphichiral, noninvertible
i	1, 2	chiral, invertible
a	1, 2, 3, 4	$+$ and $-$ amphichiral, invertible

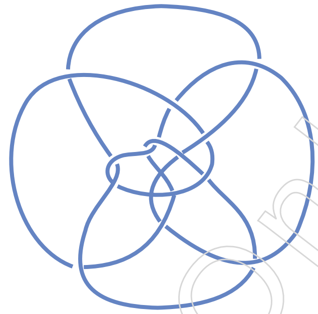
Hyperbolic knots require a finite symmetry group that might be either dihedral or cyclic (Kodama and Sakuma 1992, Hoste et al. 1998). For knots that are not hyperbolic,

Notes

the classification is a little more intricate. Notably, according to Hoste et al. (1998), every knot that has eight crossings or fewer is either amphichiral or invertible. In any alternating diagram of the link, any symmetry of a prime alternating link must be evident up to flypes.



D_{16}



D_9

According to Sloane (A052411 and A052412 for Z_1), there are a total of -crossing knots associated with cyclic symmetry groups Z_k and dihedral symmetry groups D_k (A052415 through A052422). These numbers are provided in the tables below (Hoste et al. 1998). Only a single knot each with symmetry groups Z_3,D_{14} and D_{16} (upper left) exists among knots with 16 or fewer crossings. Only the hyperbolic knot (above right) and the satellite knot (below left) have symmetry group D_9. Furthermore, there are 2–4 and 10 satellite knots with 14–15 and 16 crossings, respectively, that are members of the dihedral group D_∞ .

n	Z_1	Z_2	Z_3	Z_4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	2	0	0	0
8	24	3	0	0
9	173	14	0	0
10	1047	57	0	0
11	6709	210	0	0
12	37177	712	0	2
13	224311	2268	1	0
14	1301492	7011	0	11

n	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{14}	D_{16}
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0
5	0	4	0	2	0	0	0	0	0	0	0	0
6	4	12	0	3	0	0	0	1	0	0	0	0
7	13	23	3	4	0	3	0	0	0	0	0	0
8	66	62	1	5	0	1	0	0	0	1	0	0

9	217	134	2	11	0	0	0	0	0	0	0	0
10	728	309	6	18	0	8	1	2	0	0	0	0
11	2391	647	1	21	2	3	1	2	0	0	0	0
12	7575	1463	4	31	2	2	0	0	0	0	1	0
15	23517	3065	50	53	3	12	0	2	1	4	0	0
16	73263	6791	15	89	0	10	1	8	1	1	0	1

Notes

5.1.7 Enumeration in the Odd and Even Cases

A list of all such objects must be expressly produced in order to enumerate a collection of objects that satisfy a given set of properties. The enumeration problem is the one in which finding or counting every possible solution is required

A generating function

$$F(x) = \sum_n a_n x^n$$

Is said to enumerate

Even Codes:

When k is an integer, an even number is an integer of the form n=2k. Thus, the even numbers are...-4,-2, 0, 2, 4, 6, 8, 10,....The congruence n≡0(mod 2) holds for even n because the even integers are integrally divisible by two. While n≡0(mod 4) is an even number for which it is called a doubly even number, an even number n for which n≡2(mod 4) likewise holds is referred to as a singly even number. A non-even integer is referred to as an odd number.

The oddness of a number is called its parity, so an odd number has parity 1, while an even number has parity 0.

The generating function of the even numbers is

$$\frac{2x}{(x-1)^2} = 2x + 4x^2 + 6x^3 + 8x^4 + ...$$

It is always possible to determine that the product of an even number and an odd number is even by writing

$$(2k)(2/+1) = 2\,[k(2/+1)],$$

This is even since it is divisible by 2.

Parity:

An integer's evenness or oddness is known as its parity. As a result, it can be said that whereas 7 and 12 have different parities (since 7 is odd and 12 is even), 6 and 14 have the same parity because they are both even.

The sum s_2 (n) of the bits in binary representation, or the digit count N_1(n) computed modulo 2, defines an alternative form of parity for an integer n. As a result, the number 10= 1010_2, for instance, has parity 2 (mod 2), or 0, because its binary form contains two 1s. Therefore, the parities of the first few integers (beginning with 0) are 0, 1, 1, 0, 1, 0, 1, 1, 0, 0,..., as summed up in following table.

Notes

N	binary	parity	N	binary	parity
1	1	1	11	1011	1
2	10	1	12	1100	0
3	11	0	13	1101	1
4	100	1	14	1110	1
5	101	0	15	1111	0
6	110	0	16	10000	1
7	111	1	17	10001	0
8	1000	1	18	10010	0
9	1001	0	19	10011	1
10	1010	0	20	10100	0

A generating function for parity is given by

$$\frac{1}{2}\left(\frac{1}{1-x}-\prod_{k=0}^{\infty}1-x^{2^k}\right)=x+x^2+x^4+x^7+.... \qquad (1)$$

The constant generated by interpreting the sequence of parity digits as a binary fraction $0.011010011\ ..._2$is called the Thue-Morse constant.

The parity function obeys the sum identity

$$\sum_{k=0}^{2^{n+1}}(-1)^{P(k)}(k+r)^n=0 \qquad (2)$$

for any n. For example, for $n=2$and $r=0$,

1 – 4 – 9 + 16 – 25 + 36 + 49 – 64 = 0 (3)

Odd Codes:

An odd number is an integerof the form$n=2\,k+1$, where $k$is an integer. The odd numbers are therefore ..., -3, -1, 1, 3, 5, 7, ... which are also the gnomonic numbers. Integers which are not odd are called even.

Odd numbers leave a remainder of 1 when divided by two, i.e., the congruence $n\equiv 1\ (\mathrm{mod}\ 2)$holds for odd n. The oddness of a number is called its parity, so an odd number has parity 1, while an even number has parity 0.

The generating function for the odd numbers is

$$\frac{x(1+x)}{(x-1)^2}=x+3x^2+5x^3+7x^4+....$$

The product of an even number and an odd number is always even, as can be seen by writing

(2 k) (2 / + 1) = 2 [k (2 / + 1)],

which is divisible by 2 and hence is even.

Parity Check Matrix:

A parity check matrix H of a linear code C of length n and dimension k over a field F is an ×(n-k)matrix whose rows yield the orthogonal complement of C; that is, if an element w of F^nis a code word of C, then wH=0. The null space of the generator matrix G is produced by the rows of H.

Notes

Generator Matrix:

Given a linear code C, a generator matrix $\mathsf{G}$of $C$is a matrix whose rows generate all the elements of C, i.e., if $\mathsf{G} = (g_1\ g_2\ \ldots\ g_k)^{\mathrm{T}}$, then every code word w of C can be represented as

$$w = c_1 g_1 + c_2 g_2 + \ldots + c_k g_k = cG$$

in a unique way, where $\mathbf{c} = (c_1\ c_2\ \ldots\ c_k)$.

An example of a generator matrix is the Golay code, which consists of all 2^{12}possible binary sums of the 11 rows.

Golay Code:

Perfect linear error correction is achieved using the Golay code. The Golay code comes in binary and ternary forms, which are basically different from one another.

The binary version $G_{23}$is a $(23, 12, 7)$ binary linear code consisting of $2^{12} = 4096$code words of length 23 and minimum distance 7. The ternary version is a $(11, 6, 5)$ternary linear code, consisting of $3^6 = 729$ code words of length 11 with minimum distance 5.

A parity check matrix for the binary Golay code is given by the matrix $\mathsf{H} = (\mathsf{M}\ \mathsf{I}_{11})$, where $\mathsf{I}_{11}$is the 11×11 identity matrix and M is the 11×12 matrix.

$$\mathsf{M} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

By adding a parity check bit to each code word in G_{23}, the extended Golay code G_{24}, which is a nearly perfect $[24, 12, 8]$binary linear code, is obtained. The automorphism group of $G_{24}$is the Mathieu groupM_{24}.

A second M_{24}generator is the adjacency matrix for the icosahedron, with $\mathsf{J}_{12} - \mathsf{I}_{12}$ appended, where J_{12} is a unit matrix and I_{12} is an identity matrix.

An additional generator The 24-bit 0 word (000...000) is used by M_{24} to start a list and the first 24-bit word that differs by eight or more from every other word in the list is appended to it repeatedly.

Conway and Sloane enumerate numerous other techniques.

Even though Golay's original paper was only a half-page long, it has since been shown to have significant ties to a wide range of fields, including particle physics, group theory, graph theory, number theory, combinatorics and game theory.

Hamming Code:

A binary Hamming code $H_r$of length $n = 2^r - 1$(with $r \geq 2$) is a linear code with parity-

Notes

check matrix H whose columns consist of all nonzero binary vectors of length r, each used once. H_r is an $(n = 2^r - 1, k = 2^r - 1 - r, d = 3)$ code. Hamming codes are perfect single error-correcting codes.

5.1.8 Combinatorial Designs

The statistical theory of experiments, geometry and recreational mathematics are the foundational fields of combinatorial design theory, which emerged in the 18th and 19th centuries. In the latter part of the 20th century, design theory quickly expanded to become its own subfield of combinatorics. It has extensive applications in numerous other fields and close relationships with algebra, geometry, number theory and graph theory. Even non-mathematicians can understand most of the problems, but the solutions typically call for the use of both cutting-edge tools and methods from other branches of mathematics in addition to creative problem-solving strategies. The most basic issues are still unresolved.

Balanced Incomplete Block Designs

A pair (V, B) where V is a finite set and B is a collection of nonempty subsets of V is called a design (also known as a combinatorial design, block design, or other similar terms). While subsets in B are referred to as blocks, elements in V are referred to as points.

Balanced incomplete block designs are among the most significant design classes.

Definition:

A pair (V, B) with |V | = v and B being a collection of b blocks, each with cardinality k, such that every element of V is contained in exactly r blocks and every 2-element subset of V is contained in exactly λ blocks, is defined as a balanced incomplete block design (BIBD). The parameters of the BIBD consist of the numbers v, b, r, k and λ.

Given that both $r = \frac{\lambda(v-1)}{k-1}$; and $b = \frac{\lambda v(v-1)}{k(k-1)}$ must be integers, the following arithmetic prerequisites are evidently necessary for the existence of a BIBD(v, b, r, k, λ):

(1) $\lambda(v - 1) \equiv 0 \pmod{k - 1}$,

(2) $\lambda v(v - 1) \equiv 0 \pmod{k(k - 1)}$.

Parameter sets that satisfy (1) and (2) are called admissible.

The five parameters: v, b, r, k, λ are not independent; three of them: v, k and λ uniquely determine the remaining two as

$r = r = \frac{\lambda(v-1)}{k-1}$ and $b = r = \frac{vr}{k}$ k. Hence we often write (v, k, λ) - design (or (v, k, λ) − BIBD) to denote a BIBD (v, b, r, k, λ).

Example 1.

A (7, 3, 1) − BIBD (the "Fano plane"):

V = {0, 1, . . . , 6},

B = {{0, 1, 2}, {0, 3, 4}, {0, 5, 6}, {1, 3, 5}, {1, 4, 6}, {2, 3, 6}, {2, 4, 5}}.

Example 2.

A (11, 5, 2) − BIBD:

V = {0, 1, . . . , 10},

B = {{0, 1, 2, 6, 9}, {0, 1, 5, 8, 10}, {0, 2, 3, 4, 8}, {0, 3, 5, 6, 7}, {0, 4, 7, 9, 10}, {1, 2, 3, 7, 10}, {1, 3, 4, 5, 9}, {1, 4, 6, 7, 8}, {2, 4, 5, 6, 10}, {2, 5, 7, 8, 9}, {3, 6, 8, 9, 10}}.

The necessary conditions are also sufficient for the existence of a (v,k,1)-BIBD with

small k:

when $k = 2$, $v \geq 2$,

when $k = 3$, $v \equiv 1, 3 \pmod 6$,

when $k = 4$, $v \equiv 1, 4 \pmod{12}$,

when $k = 5$, $v \equiv 1, 5 \pmod{20}$.

If $v \equiv 1, 6 \pmod{15}$ and v 6 = 16, 21, 36, 46; 51, 61, 81, 166, 226, 231, 256, 261, 286, 316, 321, 346, 351, 376, 406, 411, 436, 441, 471, 501, 561, 591, 616, 646, 651, 676, 771, 796, 801.

In this case, a (v, 6, 1) − BIBD exists if $v \equiv 1, 6 \pmod{15}$. It has been established that a (v, 6, 1) − BIBD does not exist in the cases of orders v = 16, 21, 36 and 46; nevertheless, the existence issue for the remaining 29 orders remains unresolved.

An incidence matrix is a useful alternative to a list of a BIBD's blocks when representing it. The $v \times b$ matrix A = (a_ij) is the incidence matrix of a (v, k, λ) − BIBD (V, B), where $V = \{ x_i : 1 \leq i \leq v\}$ and $B = \{B_j : 1 \leq j \leq b\}$. In this case, $a_{ij} = 0$ otherwise and $a_{ij} = 1$ when $x_i \in B_j$.

Lemma 1. Given an incidence matrix A of a (v, k, λ) − BIBD, $AA^T = (r - \lambda)I + \lambda J$, where I is the identity matrix of $v \times v$ and J is the matrix of $v \times v$ all ones.

Theorem 2 (Fisher's inequality).

If a (v, k, λ) − BIBD exists with $2 \leq k < v$, then $b \geq v$.

This result, for instance, guarantees that a (21, 6, 1) − BIBD cannot exist, since b =14 < 21 = v, even though the above arithmetic necessary conditions are satisfied.

When r = k and v = b, a BIBD is said to be symmetric. It is owed to Bruck, Ryser and Chowla to establish the most basic prerequisite for the existence of symmetric designs.

Theorem 3 (Bruck-Ryser-Chowla). Let v, k and λ be integers satisfying $\lambda(v-1) = k(k-1)$ and for which there exists a symmetric (v, k, λ) − BIBD.

(1) If v is even, then $n = k - \lambda$ is a square.

(2) If v is odd, then the equation $z^2 = nx^2 +(-1)^{v-1/2} \lambda y^2$ has a solution in integers x, y, z not all zero.

The dual of D is a design D* = (B, V), where B corresponds to a set of elements and V to a set of blocks, such that $B \in B$ is an element contained in $v \in V$ if and only if v is contained in B in D. Thus, if M is an incidence matrix of D, then M^T is an incidence matrix of D^*.

Remark:

If and only if a BIBD is symmetric, then its dual is also a BIBD.

Furthermore, although a symmetric design and its dual share the same parameters, they are not always isomorphic.

Even when combined, the prerequisites listed above are insufficient to guarantee the presence of a symmetric (111, 111, 11, 11, 1) − BIBD, for example.

Notes

The Bruck-Ryser-Chowla theorem and Fisher's inequality can be easily verified by checking the set of parameters, however an extensive computer search and a thorough structural study have shown that no such design exists. For an infinite number of parameter combinations, the fundamental open problem of BIBD's general existence is still unsolved.

Definition 2:

If a bijection α exists, then two designs, (V_1, B_1) and (V_2, B_2), are isomorphic. $V_1 \rightarrow V_2$, where $B_2 = \{\alpha(x_i): x_i \in B_1\}$ and for any $B_1 \in B_1$, there exists $B_2 \in B_2$.

An isomorphism from a design to itself is called an automorphism. The whole automorphism group is the collection of all a design's automorphisms. Any subgroup of a design's complete automorphism group is its automorphism group. Specifically, a (v, k, λ) − BIBD is cyclic if and only if its automorphism group admits a cyclic group of order v.

In certain cases, specifying an automorphism group makes it considerably easier to develop a design. After that, choosing a set of foundation blocks that represent each block orbit under the designated automorphism group is sufficient. The group acts on these basic blocks to obtain the additional blocks.

Assume that G is an order v group. If every non-zero element of G has precisely λ representations as a difference d − d^' of elements d and d^' from D, then a k-element subset D of G is a (v, k, λ)- difference set.

Theorem 4.

A set $D = \{d_1, d_{2, \ldots}, d_k\}$ of k residues modulo v is a (v, k, λ)-difference set if and only if the sets $B_i = \{d_i + i, d_2 + i, \ldots, d_k + i\} \bmod v$, $i = 0, 1, \ldots, v - 1$ form blocks of a cyclic (v, k, λ) − BIBD.

Example 3.

A (13, 4, 1)-difference set in the group Z_{13} is {0, 1, 3, 9}.

Therefore, the base block of a cyclic (13, 4, 1) − BIBD is {0, 1, 3, 9}.

It is possible to expand the definition of a difference set to include more sets. Assume that G is an order v group. If every non-zero element of G appears exactly λ times as a difference $d_i^p - d_j^p$, then a collection $D = \{D_1, D_2, \ldots D_s\}$ of k-element subsets of G, where $D_i = \{d_1^i, d_2^i, \ldots, d_k^i\}$, i = 1, 2,... s, forms a (v, k, λ)-difference family.

Theorem 5.

If a set $D = \{D_1, D_2, \ldots D_s\}$ is a (v, k, λ)-difference family over the cyclic group G, then $Orb_G(D_1) \cup Orb_G(D_2) \cup \ldots \cup Orb_G(D_s)$ is the collection of blocks of a cyclic (v, k, λ) − BIBD.

Example 4.

{{0, 2, 10, 15, 19, 20}, {0, 3, 7, 9, 10, 16}} is a (21, 6, 3)-difference family in the group Z_{21}.

Let G be a group of order v−1. A collection $D = \{D_1, D_2, \ldots D_s\}$ of k-element subsets of $G \cup \{\infty\}$, is a 1-rotational (v, k, λ)-difference family if every element of $G \backslash \{0\} \cup \{\infty, -\infty\}$ occurs exactly λ times as a difference $d_i^p - d_j^p$.

Theorem 6.

If a set $D = \{D_1, D_2, \ldots D_s\}$ is a 1-rotational (v, k, λ)-difference family over the group G, thenOrb$_G$ $(D_1) \cup Orb_G(D_2) \cup \ldots \cup Orb_G(D_s)$ is the collection of blocks of a (v, k, λ) − BIBD admitting an automorphism group fixing one point and acting sharply transitively on the other points.

Notes

Example 5:

This is a 1-rotational (12, 3, 2)-difference family: {{0, 1, 3}, {0, 1, 5}, {0, 2, 5}, {0, 4, ∞}}.

Bose expanded on the idea of a difference family to create the strategy known as the method of pure and mixed differences. Assume that G is an additive abelian group and that T is a collection of t elements. A set V = G × T is considered. There are two types of differences that can arise from a pair of elements (x, i) ≠ (y, j) of V:

(1) if i = j then ± (x − y) is a pure difference of class i

(2) if i ≠ j then ± (x − y) is a mixed difference of class i j.

A pure difference of any class may equal to any nonzero element of G while a mixed difference may equal to any element of G.

Let us assume that there is a set of k-element sets D = = {D_1, D_2,... D_s , such that for every i in T, there are exactly λ instances of any nonzero element of G as a pure difference of class i and for every i in T, there are exactly λ instances of any element of G as a mixed difference of class i j, for all i. The sets in D thus constitute a basis for a (v, k, λ) – BIBD.

(V, B), where B = {D_i ± g : g ∈ G, i = 1, 2, . . . s}.

Example 6. Let G = Z_5 and T = {1, 2}.

D = {{0_1, 2_1, 3_1, 3_2}, {0_1, 2_2, 3_2, 4_2}, {0_1,1_1,0_2,2_2}} is a basis for a (10, 4, 2) − BIBD.

Example 7. Let G = Z_3 and T = {1, 2, 3}.

D = {{0_1, 1_1,0_2}, {0_2, 1_2, 0_3}, {0_1,0_3,1_2}, {0_1,1_2,2_3} is a basis for a (9, 3, 1) − BIBD.

One more fixed point can be added to the construction shown above.

Example 8.

Let V = (Z_7 × {1, 2}) ∪ {∞}.

D = {{0_1, 1_1,3_1}, {0_1,0_2,1_2}, {0_1,2_2,4_2}, {0_1,3_2,6_2}, {0_1, 4_2, ∞}} is a basis for a (15, 3, 1)− BIBD.

A complement of a design (V, B) is a design), $V,\overline{B}$ where B = {V \ B : B ∈ B}. Thus a complement of a BIBD (v, b, r, k, λ) is a BIBD (v, b, b − r, v − k, b − 2r + λ). A supplement of a BIBD (v, b, r, k, λ) is a BIBD obtained by taking all k-subsets which are not in B as blocks; in this way we get a BIBD

$$\left(v, \binom{v}{k} - b, \binom{v-1}{k-1} - r, k, \binom{v-2}{k-2} - \lambda\right).$$

A design (V', B') is a subdesign of (V, B) if V'⊂ V and B^' ⊂ B.

Given a design D = (V, B), a block intersection graph G(D) is a graph with the vertex set B and the edge set {{B_i, B_j} : B_i ∩ B_j≠ ∅}. In particular, for a (v, k, 1) − BIBD, G(D) is strongly regular.

5.1.9 Ramsey Theory

The idea that "complete disorder is impossible" is the metastatement of Ramsey theory. Put another way, no matter how complex a large system is, there will always be a smaller component that has a unique structure of some kind. Its arguably the oldest statement of its kind.

Notes

Proposition 1: Out of every six individuals, there are either three that are friends with each other or three who are not friends with each other.

This is a fairly basic graph theoretic statement rather than a sociological assertion: in any graph with six vertices, there is a triangle or three vertices without any connections connecting them.

Proof.

Assume a graph G = (V, E) with |V | = 6. Resolve a vertex v in V. We look at two situations.

- Let x, y and z be the three neighbors of v if v has a degree of at least three. We are done if any two of {x, y, z} are buddies because they join v to make a triangle. If not, We are also done and there are no two friends among {x, y, z}.
- There are at least three more vertices, x, y and z, that are not neighbors of v if the degree of v is at most 2. The argument in this instance supports the earlier one. {x, y, z} must be mutual friends for us to proceed; otherwise, we're done. Alternatively, no two of {v, x, y} are friends if there are two among {x, y, z} who are not friends, such as x and y.

In a broader sense, we take into account the following setup. We investigate whether there is a complete subgraph (a clique) of a given size such that all of its edges have the same colour. Specifically, we colour the edges of K_n (a complete graph on n vertices) with a given number of colours. We'll show that for a big enough n, this is always the case. It should be noted that the friendship question correlates to colouring K_6 with the colours "friendly" and "unfriendly." In a same manner, we begin with an arbitrary graph and seek to locate an independent set, sometimes known as a complement of a clique.

The definition of Ramsey numbers follows from this.

Definition 1. A set of t vertices such that every pair among them is an edge is called a clique of size t.

A collection of s vertices without an edge connecting them is called an independent set of size s.

According to Ramsey's theorem, there exists an independent set of size s or a clique of size t for any sufficiently big graph. A Ramsey number is the fewest number of vertices needed to accomplish this.

Definition 2: An independent set of size s or a clique of size t can be found in any graph on n vertices if the smallest number n is known as the Ramsey number R(s, t).

The smallest number n such that each colouring of the edges of K_n with k colours contains a clique of size s_i in colour i, for some i, is known as the Ramsey number $R_k(s_1, s_2, \ldots, s_k)$.

Take note that the finite nature of Ramsey numbers is not obvious from the outset! In fact, it's possible that for every given choice of s, t, there isn't a finite number that satisfies the requirements of R(s, t). This is not the case, as demonstrated by the following theorem, which also provides an explicit bound on R(s, t).

Theorem 1: (Theorem Ramsey).

There exists $R(s, t) < \infty$ such that any graph on R(s, t) vertices contains either a clique of size t or an independent set of size s for any $s, t \geq 1$. More specifically,

$$R(s,t) \leq \binom{s+t-2}{s-1}.$$

Notes

We note that this bound is more stringent than the one proposed by Ramsey.

Proof. Our findings demonstrate that R(s, t) ≤ R(s − 1, t) + R(s, t − 1). To visualize this, consider any graph G with n vertices and let n = R(s − 1, t) + R(s, t − 1). Resolve a vertex v ∈ V. We examine two scenarios:

- There are at least R(s, t−1) edges incident with v. Then we perform induction on the neighbors of v, which implies that either they contain an independent set of size s, or a clique of size t − 1. In the second scenario, we can add v to the clique to make it longer, so G contains either a clique of size t or an independent set of size s.
- R(s − 1, t) non-neighbors of v exist at least. Once induction is applied on v's non neighbors, we obtain a clique of size t or an independent set of size s − 1. We are done since, once more, v can be added to the independent set to make it larger.

Given that it follows by induction that these Ramsey numbers are finite. Moreover, we get an explicit bound. First, holds for the base cases where s = 1 or t = 1 since every graph contains a clique or an independent set of size 1. The inductive step is as follows:

The finite nature of these Ramsey numbers can be deduced from the fact that R(s, t) ≤ R(s−1, t)+R(s, t−1), In addition, we obtain a precise limit. Since every graph contains a clique or an independent set of size 1, the first, R (s, t) ≤ ((s+t-2)/(s-1)) $\left(\frac{s+t-2}{s-1}\right)$, of these two properties holds for the base cases where s = 1 or t = 1. Listed below is the inductive step:

$$R(s,t) \leq R(s-1,t) + R(s,t-1) \leq \binom{s+t-3}{s-2} + \binom{s+t-3}{s-1} = \binom{s+t-2}{s-1}$$

by a binomial coefficient standard identity.

A similar statement holds for a greater number of hues.

Theorem 2. For any $s_1, s_2, \ldots, s_k \geq 1$, there is $R_k(s_1, s_2, \ldots, s_k) < \infty$ such that for any k-colouring of the edges of k_n, $n \geq R_k(s_1, s_2, \ldots, s_k)$ there is a clique of size s_i in some colour i.

Here, we only outline the proof. For the sake of simplicity, let's suppose that k is even. We demonstrate that

$$R_k(s_1, s_2, \ldots, s_k) \leq R_{k/2}(R(s_1, s_2), R(s_3, s_4), \ldots, R(s_{k-1}, s_k)).$$

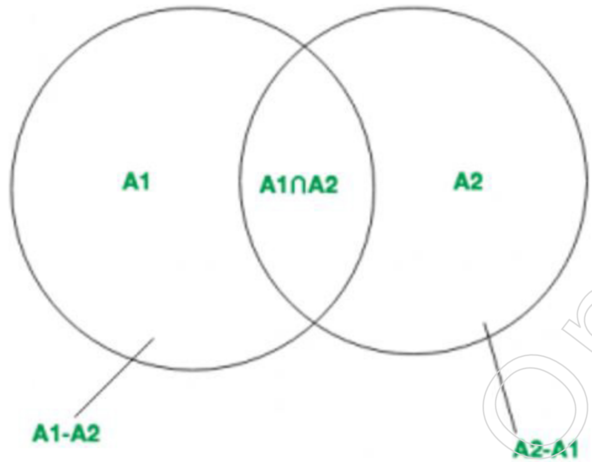
To prove this, let n = $R_{k/2}$ $(R(s_1, s_2), R(s_3, s_4), \ldots, R(s_{k-1}\ s_k))$ and consider any k-colouring of the edges of K_n. We pair up the colours: {1, 2}, {3, 4}, {5, 6}, etc. By the definition of n, there exists a subset S of $R(,s_{2i-1}), s_{2i})$ vertices such that all edges on S use only colours 2_{i-1} and 2i. By applying Ramsey's theorem once again to S, there is either a clique of size s_{2i-1} in colour 2_{i-1}, or a clique of size s_{2i} in colour 2_i.

5.1.10 Inclusion-Exclusion Principle

It is a counting technique used in combinatorics to determine the cardinality of the union set. In line with the fundamental inclusion-exclusion principle:

$(A_1\text{-}A_2)$, $(A_2\text{-}A_1)$ and $(A_1 \cap A_2)$ are disjoint sets for two finite sets, A_1 and A_2, which are subsets of the Universal set.

Notes



Hence it can be said that,

$$|(A_1 - A_2) \bigcup (A_2 - A_1) \bigcup (A_1 \bigcap A_2)| = |A_1| - |A_1 \bigcap A_2| + |A_2| - |A_1 \bigcap A_2| + |A_1 \bigcap A_2|$$

$$|A_1 \bigcup A_2| = |A_1| + |A_2| - |A_1 \bigcap A_2|$$

Similarly for 3 finite sets

A_1, A_2 and A_3

$$|A_1 \bigcup A_2 \bigcup A_3| = |A_1| + |A_2| + |A_3| - |A_1 \bigcap A_2| - |A_2 \bigcap A_3| - |A_1 \bigcap A_3| + |A_1 \bigcap A_2 \bigcap A_3|$$

Principle:

The inclusion-exclusion principle states that, for any number of finite sets A_1, A_2,A_3,…. A_i , the union of the sets is equal to the sum of the sizes of all single sets, the sum of the intersections of all two sets, the sum of the intersections of all three sets, the sum of the intersections of all four sets and $+(-1)^{i+1}$ Total of all intersections in the i-set.

Broadly speaking, one could say that,

$$|A_1 \bigcup A_2 \bigcup A_3 .. \bigcup A_i| = \sum_{1 \le k \le i} |A_k| + (-1) \sum_{1 \le k_1 < k_2 \le i} |A_{k_1} \bigcap A_{k_2}| + (-1)^2 \sum_{1 \le k_1 < k_2 < k_3 \le i} |A_{k_1} \bigcap A_{k_2} \bigcap A_{k_3}| .. + (-1)^{i+1} \sum_{1 \le k_1 < k_2 < k_3 < .. k_i \le i} |A_{k_1} \bigcap A_{k_2} \bigcap A_{k_3} .. \bigcap A_{k_i}|$$

Properties:

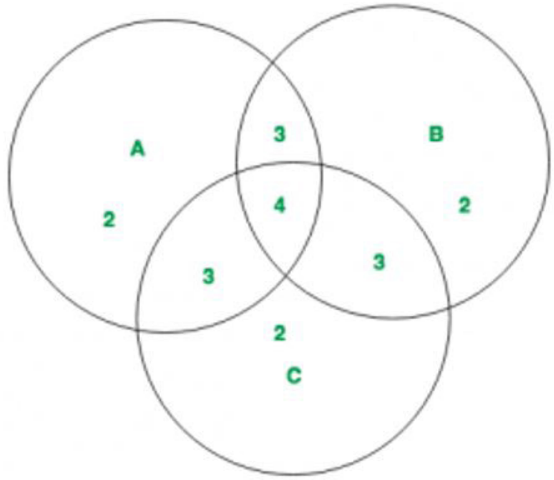
❖ Determines how many elements there are overall that meet at least one of various requirements.

❖ It avoids the issue of counting twice.

Example 1:

Three finite sets, A, B and C, are displayed in the diagram along with the appropriate values. Calculate

$$|A \bigcup B \bigcup C|$$



Notes

Solution:

The graphic indicates that the values of the appropriate regions are –

$$|A| = 2, |B| = 2, |C| = 2, |A \bigcap B| = 3, |B \bigcap C| = 3,$$

$$|A \bigcap C| = 3, |A \bigcap B \bigcap C| = 4$$

Using the principle of inclusion-exclusion,

$$|A_1 \bigcup A_2 \bigcup A_3| = |A_1| + |A_2| + |A_3| - |A_1 \bigcap A_2| - |A_2 \bigcap A_3| - |A_1 \bigcap A_3| + |A_1 \bigcap A_2 \bigcap A_3|$$

$$|A_1 \bigcup A_2 \bigcup A_3| = 2 + 2 + 2 - 3 - 3 - 3 + 4 = 1$$

Applications:

1. Derangements:

   To count the number of n object derangements, or permutations, in which an object never returns to its initial position (similar to the Hat-check issue).

   We can examine the number derangements in the following scenarios, for instance:

   The total number of derangements for i = 1 is 0.

   There is just one total derangement for i = 2. This is 21.

   There are two total derangements for i = 3. These are 231

   And

   312.

Approach:

Counting the elements in the union of several sets is possible because to the combinatorial counting technique known as the Inclusion-Exclusion Principle. According to the principle, the total size of the sets, less the size of their intersection, plus the size of the intersection of their pairwise intersections and so on, equals the size of the union of two or more sets.

The following describes how to apply the Inclusion-Exclusion Principle step-by-step in C++:

1. Specify which sets have to be mixed.

Notes

2. Determine how big each set is.
3. Determine how big each intersection between two sets is.
4. Determine how big each intersection between three sets is.
5. Up to the last intersection, keep calculating the sizes of each intersection of sets of four, five and so on.
6. Add up all of the set sizes.
7. The size of each pairwise intersection is subtracted.
8. Include every three-way intersection's size.
9. Up until the last crossing, keep adding and deleting intersections of progressively larger sizes.
10. Give back the total count.

5.1.11 Recurrence Relations

When an integer is written as the sum of positive integers, P(n), commonly written P(n), indicates how many possible ways there are to write it without regard to the order of the addends.

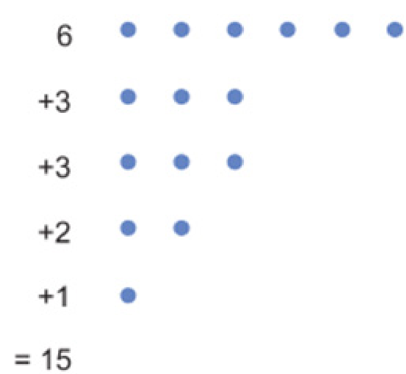
Partitions are often arranged in descending order of size by convention. Because four can be written, for instance

$$\begin{aligned} 4 &= 4 && (1)\\ &= 3+1 && (2)\\ &= 2+2 && (3)\\ &= 2+1+1 && (4)\\ &= 1+1+1+1, && (5) \end{aligned}$$

Consequently, P (4) = 5. P (n), which is implemented as Partitions P[n], is sometimes referred to as the number of unconstrained partitions at times.

For n = 1, 2,..., the values of P (n) are 1, 2, 3, 5, 7, 11, 15, 22, 30.

The indices 2, 3, 4, 5, 6, 13, 36, 77, 132 and so on correspond to the first few prime values of P (n), which are 2, 3, 5, 7, 11, 101, 17977, 10619863,...With 35219 decimal digits, the greatest known n that gives a likely prime is 1000007396; with 16569 decimal digits, the largest known n that gives a verified prime is 221444161.



When a number n is deliberately partitioned, the simplest version is known as the "natural representation," which is just the sequence of numbers in the representation (for example, (2, 1, 1) for the number 4 = 2+1+1). Alternatively, each number's number of occurrences alongside that number is shown by the multiplicity representation (e.g., (2, 1), (1, 2) for 4=2.1+1.2). An illustration of a partition is shown in the Ferrers diagram. The Ferrers diagram for the partition 6+3+3+2+1=15, for instance, is shown in the diagram above.

Euler gave a generating function for P (n) using the q-series

$$(q)_\infty \quad \equiv \quad \prod_{m=1}^{\infty}(1-q^m) \tag{6}$$

$$= \quad \sum_{-\infty}^{\infty}(-1)^n q^{n(3n+1)/2} \tag{7}$$

$$= \quad 1-q-q^2+q^5+q^7-q^{12}-q^{15}+q^{22}+q^{26}+\ldots. \tag{8}$$

Here, the exponents are generalised pentagonal numbers 0, 1, 2, 5, 7, 12, 15, 22, 26, 35, ... and the sign of the kth term (counting 0 as the 0th term) is (-1)(-1)$^{[(k+1)/2]}$ (with [x] the floor function). Then the partition numbers p(n) are given by the generating function p(n)

$$\frac{1}{(q)_\infty} \quad = \quad \sum_{n=0}^{\infty} P(n)q^n \tag{9}$$

$$= \quad 1+q+2q^2+3q^3+5q^4+\ldots. \tag{10}$$

When a number n is divided into m parts, the number of partitions into parts that are the largest is equal to m and when a number is divided into at most m parts, it is equal to the number of partitions into parts that do not exceed m. It is instantly evident from this that each of these outcomes can be interpreted by reading a Ferrers diagram column-wise or row-wise.

For example, if a_n for all n , then the Euler transform b_n is the number of partitions of n into integer parts.

Euler invented a generating function which gives rise to a recurrence equation in p(n),

$$P(n)=\sum_{k=1}^{n}(-1)^{k+1}[P(n-\frac{1}{2}k(3k-1))+P(n-\frac{1}{2}k(3+1))] \tag{11}$$

Other recurrence equations include

$$P(2n+1)=P(n)+\sum_{k=1}^{\infty}[P(n-4k^2-3k)+P(n-4k^2+3k)]- \sum_{k=1}^{\infty}(-1)^k[P(2n+1-3k^2+k)+P(2n+1-3k^2-K)] \tag{12}$$

and

$$P(n)=\frac{1}{n}\sum_{k=0}^{n-1}\sigma_1(n-k)P(k) \tag{13}$$

where $\sigma_1(n)$ is the divisor function as well as the identity

$$\sum_{k=[-(\sqrt{24n+1}+1)/6]}^{[(\sqrt{24n+1}-1)/6]}(-1)^k P(n-\frac{1}{2}k(3k+1))=0 \tag{14}$$

Where $\lfloor x \rfloor$ is the floor function and $\lfloor x \rfloor$ is the ceiling function.

A recurrence relation involving the partition function Q is given by

Notes

$$P(n)=\sum_{k=0}^{[n/2]} Q(n-2k)P(k) \qquad (15)$$

$$\sum_{n=0}^{\infty} P(5n+1)q^n=\prod_{n=1}^{\infty}\frac{(1-q^{5n})}{(1-q^{5n-4})(1-q^{5n-1})} \pmod 5 \qquad (16)$$

$$\sum_{n=0}^{\infty} P(5n+1)q^n=2\prod_{n=1}^{\infty}\frac{(1-q^{5n})}{(1-q^{5n-3})(1-q^{5n-2})} \pmod 5 \qquad (17)$$

$$\sum_{n=0}^{\infty} P(5n+3)q^n=3\prod_{n=1}^{\infty}\frac{(1-q^{5n-4})(1-q^{5n-1})(1-q^{5n})}{(1-q^{5n-3})^2(1-q^{5n-2})^2} \pmod 5 \qquad (18)$$

$$P(n)-P(n-1)-P(n-2)+P(n-5)+P(n-7)-P(n-12)-P(n-15)+...=0 \qquad (20)$$

where the sum is over generalised pentagonal numbers £n and the sign of the k th term is $(-1)(-1)^{([(k+1)/2])}$, as above.

Ramanujan stated without proof the remarkable identities.

$$\sum_{k=0}^{\infty} P(5k+4)q^k=5\frac{(q^5)_\infty^5}{(q)_\infty^6} \qquad (21)$$

and

$$\sum_{k=0}^{\infty} P(7k+5)q^k=7\frac{(q^7)_\infty^3}{(q)_\infty^4}+49q\frac{(q^7)_\infty^7}{(q)_\infty^8} \qquad (22)$$

To obtain the asymptotic solution

$$P(n)\sim\frac{1}{4n\sqrt{3}}e^{\pi\sqrt{2n/3}} \qquad (23)$$

$$P(n)=\frac{1}{\pi\sqrt{2}}\sum_{k=1}^{\infty}A_k(n)\sqrt{k}\,\frac{d}{dn}\left[\frac{\sinh\left(\frac{\pi}{k}\sqrt{n-\frac{1}{24}}\right)}{\sqrt{n-\frac{1}{24}}}\right] \qquad (24)$$

where

$$A_k(n)=\sum_{h=1}^{k}\delta GCD(h,k),1\exp\left[\pi i\sum_{j=1}^{k-1}\frac{j}{k}\left(\frac{hj}{k}-\left[\frac{hj}{k}\right]-\frac{1}{2}\right)-\frac{2\pi ihn}{k}\right] \qquad (25)$$

δ is the Kronecker delta and $\lfloor x \rfloor$ is the floor function. The remainder after N terms is

$$R(N)<CN^{-1/2}+D\sqrt{\frac{N}{n}}\sinh\left(\frac{K\sqrt{n}}{N}\right) \qquad (26)$$

where C and D are fixed constants.

To found an algebraic formula for the partition function P(n) as a finite sum of algebraic numbers as follows. Define the weight-2 meromorphic modular form F(z) by

$$F(z)\ \frac{}{2}\ \frac{E_2(z)-2E_2(2z)-3E_2(3z)+6E_2(6z)}{\eta^2(z)\eta^2(2z)\eta^2(3z)\eta^3(6z)} \qquad (27)$$

were $q=e^{2\pi iz}$, E_2 (q) is an Eisenstein series and η(q) is a Dedekind eta function.

Notes

Now define

$$R(z)=-\left(\frac{1}{2\pi i}\frac{d}{dz}+\frac{1}{2\pi y}\right)F(z) \quad (28)$$

where z = x+ iy . Additionally let Q^n be any set of representatives of the equivalence classes of the integral binary quadratic form Q(x,y)=ax_2+bxy+cy_2 such that 6|a with a>0 and b≡1(mod12) and for each Q (x,y), let a^Q be the so-called CM point in the upper half-plane, for which Q (a^Q,1)=0 .

Then

$$P(n)=\frac{Tr(n)}{24n-1} \quad (29)$$

where the trace is defined as

$$Tr(n)=\sum_{Q\in Q_n} R(\alpha_Q) \quad (30)$$

Let $f_o(x)$ be the generating function for the number of partitions $P_o(n)$ of n containing odd numbers only and $f^o(x)$ be the generating function for the number of partitions $P^D(n)$ of n without duplication, then

$$f_O(x) = f_D(x) \quad (31)$$

$$= \prod_{k=1,3,\dots}^{\infty}\sum_{l=0}^{\infty}x^{lk} \quad (32)$$

$$= \frac{1}{\prod_{k=1,3,\dots}^{\infty}1-x^k} \quad (33)$$

$$= \prod_{k=1}^{\infty}\left(1+x^k\right) \quad (34)$$

$$= \frac{1}{2}(-q;x)_\infty \quad (35)$$

$$= 1+x+x^2+2x^3+2x^4+3x^5+\dots.. \quad (36)$$

The identity

$$\prod_{k=1}^{\infty}(1+z^k)=\prod_{k=1}^{\infty}(1-z^{2k-1})^{-1} \quad (37)$$

is known as the Euler identity.

The difference between the number of partitions into an odd number of unequal parts and the number of partitions into an even number of unequal parts is represented by the generating function, which is given by

$$\prod_{k=1}^{\infty}(1-z^k)=1-z-z^2+z^5+z^7-z^{12}-z^{15}+\dots \quad (38)$$

where

$$ck = \begin{cases} (-1)^n & \text{for } k \text{ of the form} \frac{1}{2}n\ (3n\pm 1) \\ 0 & \text{otherwise} \end{cases} \quad (40)$$

Let $P_E(n)$ be the number of partitions of even numbers only and let $P_{EO}(n)$ ($P_{DO}(n)$) be

Notes

the quantity of divisions where each component is distinct and all even (odd). Then the generating function of $P_{DO}(n)$ is given by

$$f_{DO}(x) = \prod_{k=1,3,..}^{\infty} 1+x^k \tag{41}$$

$$= (-x;x^2)_\infty \tag{42}$$

The number of partitions that have no even part repeated is equal to the number that have no part that occurs more than three times and the number that have no part that is divisible by four and they all share the same generating functions.

$$P_3(n) = \prod_{k=1}^{\infty} \frac{1+x^{2k}}{1-x^{2k-1}} \tag{43}$$

$$= \prod_{k=1}^{\infty} (1+x^k+x^{2k}+x^{3k}) \tag{44}$$

$$= \prod_{k=1}^{\infty} \frac{1-x^{4k}}{1-x^k} \tag{45}$$

$$= \frac{(x^4)_\infty}{(x)_\infty} \tag{46}$$

In general, the generating function for the number of partitions in which no part occurs more than d times is

$$P_d(n) = \prod_{k=1}^{\infty} \sum_{i=0}^{d} x^{ik} \tag{47}$$

$$= \prod_{k=1}^{\infty} \frac{1-x^{(d+1)k}}{1-x^k} \tag{48}$$

The generating function for the number of partitions in which every part occurs 2, 3, or 5 times is

$$P_{2,3,5}(n) = \prod_{k=1}^{\infty} (1+x^{2k}+x^{3k}+x^{5k}) \tag{49}$$

$$= \prod_{k=1}^{\infty} (1+x^{2k})(1+x^{3k}) \tag{50}$$

$$= \prod_{k=1}^{\infty} \frac{1-x^{4k}}{1-x^{2k}} \frac{1-x^{6k}}{1-x^{3k}} \tag{51}$$

$$= \frac{(x^4)_\infty (x^6)_\infty}{(x^2)_\infty (x^3)_\infty} \tag{52}$$

The first few values are 0, 1, 1, 1, 1, 3, 1, 3, 4, 4, 4, 8, 5, 9, 11, 11, 12, 20, 15, 23, ... The number of partitions in which no part occurs exactly once is

Notes

$$P_1(n) = \prod_{k=1}^{\infty}(1+x^{2k}+x^{3k}+\ldots) \quad (53)$$

$$= \prod_{k=1}^{\infty}\frac{1-x^k+x^{2k}}{1-x^k} \quad (54)$$

$$= \prod_{k=1}^{\infty}\frac{1+x^{3k}}{1-x^{2k}} \quad (55)$$

$$= \prod_{k=1}^{\infty}\frac{1-x^{6k}}{(1-x^{2k})(1-x^{3k})} \quad (56)$$

Remark

1. One can get the number of partitions of n that have no even part that is repeated, the number of partitions that have no part that occurs more than three times and the number of partitions that have no part that is divisible by four.
2. The quantity of n partitions where no portion appears more than d times is equal to the quantity of n partitions where no term is a multiple of d+1.
3. There are the same number of partitions where each part appears twice, three times, or five times as there are where each part is congruent mod 12 to two, three, six, nine, or ten times.
4. The number of partitions where a part does not exist exactly once is equal to the number of partitions where a part does not congruence with 1 or 5 mod 6.
5. The difference between the total number of partitions with odd and even parts and the number of partitions with all even and distinct parts is equal. The inequality is satisfied by P(n).

$$P(n) \le \frac{1}{2}[P(n+1)+P(n-1)] \quad (58)$$

P(n,k) represents how many ways there are to write n as the sum of exactly k terms or, in other words, how many partitions there are into sections where the largest is exactly k. (Note that the partition function q is obtained if “largest is exactly k” is changed to “n° element greater than k” and “exactly k” is changed to “k or fewer.”) Since {3,1,1} and {2,2,1} are the partitions of 5 of length 3 and {3,2} and {3,1,1} are the partitions of 5 with maximum element 3, P(5,3)=2, for instance.

One can list all of the P(n,k) such partitions in the Integer Partitions.

[n, k].

P(n,k) can be computed from the recurrence relation

P(n,k)=P(n-1,k-1)+P(n-k,k) (59)

with P(n,k)= 0 for k > = n , P(n,n)= 1 and P n(,0) = 0 . The triangle of P(k,n) is given by

1

1 1

1 1 1

1 2 1 1 (60)

Notes

1 2 2 1 1

1 3 3 2 1 1

P (n, k) is the number of partitions of n having the biggest component.

It is possible to precisely solve the recurrence relation to obtain

$$P(n,1) = 1 \tag{61}$$

$$P(n,2) = \frac{1}{4}\left[2n-1+(-1)^n\right] \tag{62}$$

$$P(n,3) = \frac{1}{72}\left[6n^2-7-9(-1)^n+16\cos\left(\frac{2}{3}\pi n\right)\right] \tag{63}$$

For n less than k, P (n , k) = 0. It is also possible to explicitly provide the functions P(n,k) in the basic forms for the first few values of k.

$$P(n,2) = \left[\frac{1}{2}n\right] \tag{65}$$

$$P(n,3) = \left[\frac{1}{12}n^2\right] \tag{66}$$

Where $\lfloor x \rfloor$ is the floor function and $\lfloor x \rceil$ is the nearest integer function defines

$$t_k(n) = n+\frac{1}{4}k(k-3) \tag{67}$$

and then yields

$$P(n,2) = \left[\frac{1}{2}t_2(n)\right] \tag{68}$$

$$P(n,3) = \left[\frac{1}{12}t_3^2(n)\right] \tag{69}$$

$$P(n,4) = \begin{cases} \left[\frac{1}{144}t_4^3(n)-\frac{1}{48}t_4(n)\right] \\ \left[\frac{1}{144}t_4^3(n)-\frac{1}{12}t_4(n)\right] \end{cases} \tag{70}$$

Then obtained the exact asymptotic formula

$$P(n) = \sum_{k<\alpha\sqrt{n}} P_k(n)+O(n^{-1/4}) \tag{71}$$

where α is a constant. However, the sum diverges

$$\sum_{k=1}^{\infty} P_k(n)$$

(72)

5.1.12 Polya's Enumeration Theorem

Notes

The number of things that are invariant under a particular set of symmetries can be counted using Polya's enumeration theorem, a mathematical theorem. Put otherwise, it is a technique for calculating how many unique configurations or structures—as long as the objects' symmetries are taken into consideration—can be made out of a given set of objects. Here is a quick synopsis of the theorem along with some examples to show how to use it:

The Polya Enumeration Theorem

Let G be a collection of symmetries that are applied to a set X of n items. Let c(g) represent the number of length k cycles in the permutation g in G. The number of unique configurations of X under G is therefore determined by:

N = 1/|G| * Σ(g∈G) [a^(c(g)) * b^(k-c(g))]

where k is the total number of objects in each arrangement and a, b are parameters that rely on the kinds of objects being organised (e.g., colours, forms, etc.).

Example 1: Counting Necklace

Let's say we wish to arrange n various coloured beads into a circular necklace. Considering the circle's symmetries, how many different kinds of necklaces are possible?

The following is how we can apply Polya's theorem to address this issue. Let G be the cyclic group of order n that symbolizes the circle's rotations. It follows that c(g) = gcd(n,k), where k is the number of objects that g fixed. Given that necklaces are under consideration, k = n for all g in G since each necklace contains n objects.

Consequently, the following gives the number of unique necklaces:

N = 1/n * [a^n + 2a^(n/2) + 2a^(n/3) + b^(n/2) + b^(n)]

The number of necklaces that are invariant under all rotations is counted in the first term; the number of necklaces that are invariant under rotations by half the circle is counted in the second term; the number of necklaces that are invariant under rotations by one-third of the circle is counted in the third term and so on.

Example 2: Keeping track of bracelets

For example, let's say we wish to create a circular bracelet with a clasp using n different coloured beads. Taking the bracelet's symmetries into consideration, how many different bracelets can be made?

To overcome this issue as well, we can apply Polya's theorem. Let G be the dihedral group of order 2n that has the circle's rotations and reflections represented by it. Next, if g is a rotation, then c(g) = 1 and if g is a reflection, then c(g) = 2.

Thus, the following formula yields the number of unique bracelets:

N = 1/2n * [a^n + a(n/2) + b(n/2) + b(n)]

where the number of bracelets that remain unchanged under all rotations and reflections is counted in the first term; the number of bracelets that remain unchanged under rotations by half a circle is counted in the second term; the number of bracelets that remain unchanged under reflections across the vertical axis is counted in the third term; and the number of bracelets that remain unchanged under reflections across the horizontal axis is counted in the fourth term.

Notes

Polya's enumeration theorem can be used to count the number of unique arrangements of symmetries in things, as these examples show. Numerous further applications of the theorem can be found in group theory, combinatorics and other branches of mathematics.

In order to express Polya's Enumeration Theorem, we add some additional apparatus:

Definition 1: Type: Allow p to be an X permutation. The set {b1,..., bn}, where bi is the number of cycles of length i in the cycle decomposition of p, is thus the type of p.

Definition 2: Index polynomial cycle. For the group action φ, the cycle index polynomial Zφ is defined as 2.

$$Z_\phi(x_1, ..., x_n) = \frac{1}{|G|} \sum_{g \in G} \prod_{i=1}^{n} x_i^{b_i(g)}.$$

where bi(g) represents the i-th element of the assumed permutation type such that pg ∈ Sym(X).

Definition 3: Function equivalency. If two functions f ∈ Y^Xare in the same orbit of φ 0 (i.e., g ∈ G such that f2 = gf1), then they are said to be comparable under the action of G (f1 ~G f2).

Function equivalency is proven to be an equivalence relation by Proposition, which means that Y and X will have equivalency classes under function equivalency.

Definition 4: Setting up. A configuration is a class of equivalence of the equivalence relation ~G on Y_X

On the other hand, we have that every configuration C is merely an orbit around φ 0. And the set of configurations C is just Y^X/G under φ^' 0.

We can give elements in Y weights so that our functions are weighted differently.

Definition 5: Weight. Let w : Y → R be a weight assignment to each element in Y . Then the weight of a function f ∈Y_X is defined as

$$W(f) = \prod_{x \in X} w(f(x)).$$

As a result, every function in a configuration c has a common weight that we refer to as the configuration's weight (W(c)).

Summary

- Combinatorics is widely utilised in computer science to estimate and analyze algorithms, as well as to develop formulas that focuses on counting—both as a tool and a goal in and of itself—as well as specific characteristics of finite structures.
- In number theory and combinatorics, a partition of a positive integer, also called an integer partition, is a way of writing n as a sum of positive integers.
- According to the Fundamental Counting Principle, there are m · n total possible outcomes for the two events taken together if there are m possible outcomes for one event and n possible outcomes for a second event. The amount of options to select items from a total of objects is called a combination (order does not important).
- Basic counting rules are Addition, multiplication/Product rules , Prime Counting Functions, Reimann Counting Functions.

Notes

- The number of partitions with odd parts, q(n), equals the number of partitions with distinct parts for every positive number. Leonhard Euler demonstrated this result in 1748 and Glaisher's theorem was later developed to generalize it.
- Kinds of Symmetry are Rotational Symmetry, Square Symmetry, Mirror reflection symmetries, Counting symmetries, knot Symmetry.

Glossary

- Partition of a set: A partition of a set X is a set of non-empty subsets of X such that every element x in X is in exactly one of these subsets (i.e., X is a disjoint union of the subsets).
- Part: A summand in a partition is also called a part.
- Partition of n: The Ferrers diagram, also called Young diagram, of a partition λ +n is a rectangular array of n boxes, or cells, with one row of length j for each part j of λ.
- Necklaces: Reconstructing necklaces (cyclic configurations of binary values) from incomplete information is the subject of the necklace problem in recreational mathematics.
- Euler's Function: The Euler's Totient Function counts the values smaller than a given integer, n, that are co-prime with n, that is, have no shared positive factor other than 1. As a result, 4 numbers—1, 3, 5 and 7—are co-prime with 8 and smaller than it.
- Set of Symmetry: If reflecting a graph over a line does not modify the graph, then the graph is symmetric with respect to that line. This line is referred to as the graph's axis of symmetry. If every point on a graph is also on the graph, then the graph is symmetric with respect to the x-axis.
- Enumeration in the Odd and Even Cases: It is common knowledge that in a digraph, the total number of odd out-vertices and odd in-vertices is always even.
- Parity: The parity of an integer is its attribute of being even or odd.
- Ramsey Theory: Put another way, no matter how complex a large system is, there will always be a smaller component that has a unique structure of some kind. Its arguably the oldest statement of its kind.

Check Your Understanding

1. The rank of P is |X| − |P|, if X is __________.
   a) finite.
   b) infinite
   c) negative
   d) positive
2. A partition is a __________of positive integers, and it is a partition of n of the sum of the integer.
   a) even partition
   b) odd partition
   c) both a and b
   d) multiset
3. If a plane graph has k components, then n-e+f is equal to_________.
   a) k-1
   b) k

Notes

c) k+1

d) none of the above

4. If K_5 were planar graph, then e is less than or equal to_________.

a) 3n-6

b) 2n-6

c) n-6

d) All of the above

5. A 3-regular graph with four or more vertices is__________.

a) an euler graph

b) not an euler graph

c) a wheel

d) a cycle

Exercise

1. Explain even and odd codes. Highlight the relation between them.
2. Explain Euler's function with the help of example.
3. Define Ramsey Theory.
4. Explain the Inclusion-Exclusion Principle.
5. Explain Polya's Enumeration Theorem.

Learning Activities

1. Explain recurrence relations with the help of example.
2. In a store, a jeweler offers six-beaded necklaces for sale. How many different kinds of necklaces must he make given the two different colours of beads in order to have every possible combination of colours?

Check Your Understanding (Answers)

1. a) 2. d) 3. c) 4. a)

5. b)

Further Readings and Bibliography

1. Dr D.S.C," Graph Theory and Combinatorics All India", Prism Publications, 1 January 2012.
2. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 2nd Edition, 2000.
3. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009.
4. Harikishan, ShivrajPundir and Sandeep Kumar, Discrete Mathematics, Pragati Publication, 7th Edition, 2010.