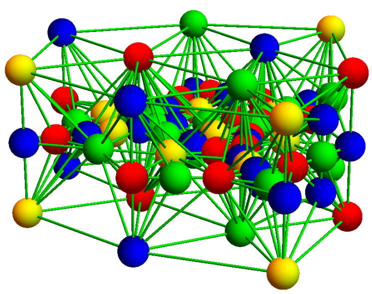
UNIVERSITY of the WESTERN CAPE

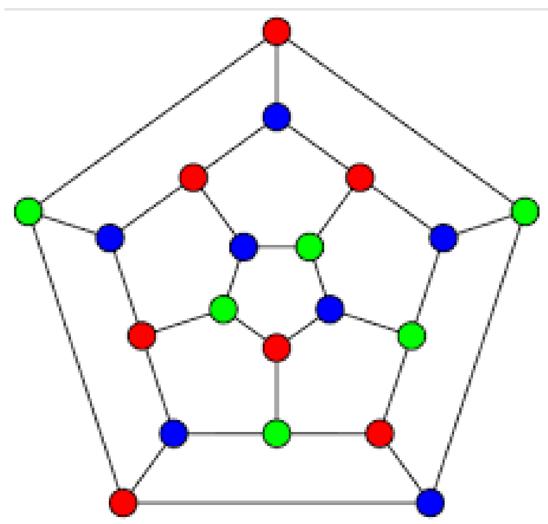
Department of Mathematics and Applied Mathematics

GRAPH COLORING: HISTORY, RESULTS, AND OPEN PROBLEMS

Cherubin Makembele
44 76 723

Supervisor: **Dr. Elham Mehdi-Nezhad**





Honors Project - Academic Year 2024

Contents

Abstract

This project explores the concept of graph coloring, a central topic in graph theory, by introducing essential graph-theoretical concepts that lay the foundation for understanding coloring problems. The study begins with the basic definitions and properties of graphs, such as order, size, and vertex degree, and progresses to more advanced topics like connectivity, isomorphisms, and the importance of subgraphs. These foundational ideas provide a framework for understanding the various types of graph coloring: vertex coloring, edge coloring, and total coloring. The paper examines key results, including the chromatic number, minimum coloring, perfect matchings, as well as important theorems and algorithms that guide the determination of optimal colorings. The project highlights the relevance of these concepts in solving practical problems, such as scheduling, resource allocation, map coloring, and optimizing path designs (desire lines). Through this structured approach, the paper demonstrates how understanding the properties of graphs is crucial to effectively applying graph coloring in both theoretical and real-world contexts.

Introduction

Graph coloring is a significant branch of graph theory focused on assigning different colors or numbers to the vertices (nodes), edges of a graph, or both. Among the various types of coloring, *vertex coloring* is the most studied, where a "proper coloring" of a graph ensures that no adjacent vertices or edges share the same color.

The origins of graph theory can be traced back to **Leonhard Euler**, who laid the groundwork for the study of graphs through his exploration of the *Seven Bridges of Königsberg*. Building on these foundations, mathematicians such as **Vadim Vizing**, **Robert Brooks**, and **Mehdi Behzad** made significant contributions to graph coloring, advancing the field enough to find applications in solving real-world problems.

Before delving into the details of graph coloring, this project addresses the essential basics of graph theory, particularly **simple graphs**, which form the core subject of this research.

This project begins with an introduction to simple graphs, covering their fundamental properties and providing a brief overview of the background and history of graph theory. It then explores the definitions, properties, and theorems of graphs, establishing the theoretical framework for this study. Next, the focus shifts to the definitions and concepts of graph coloring, including proper coloring and its variations. Finally, the project discusses the applications of graph coloring in various real-world scenarios.

By building a solid foundation in simple graphs, this project aims to offer a comprehensive understanding of graph coloring, its theoretical groundwork, and its practical implications.

1 Background and History: The Story of Graph Theory and the Rise of Graph Coloring

The Origin of Graph Theory

Graph theory emerged from a simple problem in the 18th century in the city of Königsberg (now Kaliningrad, Russia), which was a part of the Prussian Kingdom at the time. The city was divided by the Pregel River into four land masses, connected by seven bridges. The problem was as follows: Could one walk through the city and cross each bridge exactly once without retracing any steps? This was known as the Seven Bridges of Königsberg problem.

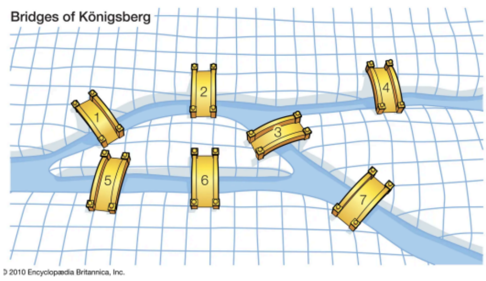


Figure 1: Bridges of Königsberg

Leonhard Euler and the Birth of Graph Theory (1736)

In 1736, the Swiss mathematician Leonhard Euler solved the problem, but in a way that was entirely novel. He realized that the problem could be translated into an abstract form, where the land masses were represented as points (vertices) and the bridges as lines (edges) connecting those points. Instead of focusing on the geography of Königsberg, Euler abstracted the problem into a graph.

Euler showed that it was **impossible** to cross each bridge exactly once by demonstrating that no solution existed based on the structure of the graph (the number of vertices with an odd degree). The solution to this problem involves Eulerian Trails. This marked the beginning of graph theory as a distinct branch of mathematics, **focused on the study of vertices and edges**. Euler's work was groundbreaking because it established the foundation of topology and graph theory, which would evolve into complex areas of study over the following centuries[4].

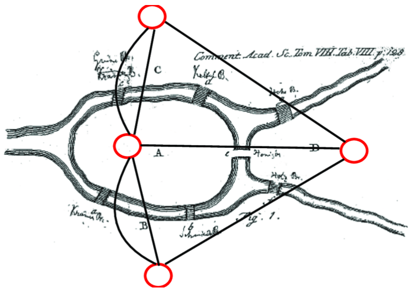


Figure 2: Bridges of Königsberg as a graph

The Rise of Graph Coloring

As graph theory expanded, a specific problem, graph coloring, emerged as a significant focus. The concept of graph coloring involves assigning "colors" (or labels) to the vertices or edges of a graph such that no two adjacent vertices (or edges) share the same color. This problem initially arose from the need to schedule or partition tasks in various real-world scenarios.

Early Developments in Graph Coloring

The first formal problem of graph coloring was posed by **Francis Guthrie in 1852**. He wondered whether it was possible to color the regions of a map using only four colors in such a way that no two adjacent regions (sharing a border) would have the same color. This was the famous Four Color Theorem, which was conjectured by **Guthrie and proved in 1976 by Kenneth Appel and Wolfgang Haken using a computer-assisted proof** [8].

The Four Color Theorem was important because it represented one of the earliest examples where graph theory was applied to a practical, real-world problem. **The theorem itself became a central point of research in graph theory**, and the methods developed for its proof laid the groundwork for further exploration of graph coloring problems [8].

2 Graphs: Definitions and Theorems

A graph G is defined as an ordered pair (V, E), where V is a set of vertices (nodes) and E is an edge set. Note that since V and E are sets, they are unordered. See Figure below.

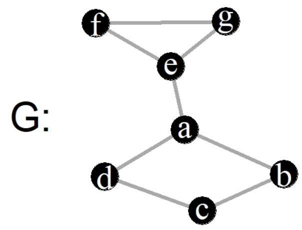


Figure 3: A graph

In the graph G above, the set of vertices is $V(G) = \{a, b, c, d, e, f, g\}$ and the set of edges is $E(G) = \{\texttt{ab,ad,ae,bc,cd,ef,eg,fg}\}$.

Definition 2.1 (Order of a graph). *The order of a graph is the number of vertices it contains. If a graph G has n vertices, its order is $n = |V(G)|$. In Figure 3, the order of G is $n = |V(G)| = 7$.*

Definition 2.2 (Size of a graph). *The size of a graph is the number of edges it contains. For a graph G with m edges, its size is $m = |E(G)|$. In Figure 3, the size of G is $m = |E(G)| = 8$.*

Definition 2.3 (Degree of a vertex). *The degree of a vertex $deg(v)$, $v \in G$ is the number of edges incident with v. The maximum degree of a G, denoted $\Delta(G)$ is defined as the set*

$$\Delta(G) = max\{deg(v)|v \in V\}$$

Similarly, the minimum degree of a G, denoted $\Delta(G)$ is defined as the set

$$\delta(G) = min\{deg(v)|v \in V\}$$

[6]

In Figure 3, $deg(a) = 3$, $deg(f) = 2$ and the minimal and maximal degrees are $\Delta(G) = 3, \delta(G) = 2$, respectively.

Theorem 2.1 (Graph theory's first Theorem). *In a graph G, the sum of degrees of the vertices is equal to twice the number of edges*

Proof. Let $S = \sum_{v \in V(G)} \deg(v)$. Notice that in counting S, we count the edges exactly twice, Hence $\sum_{v \in V(G)} \deg(v) = 2|E|$ $\square$

Types of graphs

There are many types of graph, each suited for modeling different kinds of relationship and structure and allowing resolution through optimization. These graphs help solve problems in areas like math, computer science, and data analysis, allowing us to represent anything from simple connections to complex networks. Here is a list of a few graphs:

1. **Empty Graphs :**

   An empty graph E_n, in general, is a graph with an empty edge set E, i.e. $G(V, E) = \{V, \{\}\}$. According to the vertex set constitution, an empty graph can be called:

   (a) **Null graph**: when the vertex set is also empty. Empty graphs are denoted E_n, where n is the number of vertices.

   (b) **Trivial graph**: when the vertex set consists of a single vertex E_1[22].

2. **Multi-Graphs:**

   Graphs that allow multiple edges (parallel edges) between the same pair of vertices.

   (a) **In undirected multi-graphs**, the edges `ab` and `ba` are the same, but multiple occurrences of `ab` (or `ba`) are allowed [22].

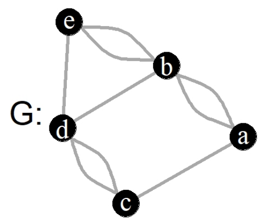


Figure 4: A multigraph (undirected)

   (b) **In directed multi-graphs**, the edges `ab` and `ba` are different edges, and multiple occurrences of each are allowed. e.g. digraphs [22].

3. **Directed Graphs (Digraphs):**

   Graphs where edges have a direction going from one vertex to another. Undirected graphs have edges with no direction [22].

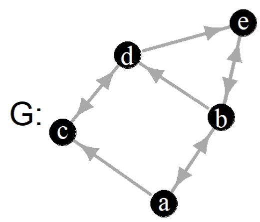


Figure 5: Directed graph

Simple Graphs

A simple graph, also called a **strict graph**, is an undirected graph that does not contain any loops or multiple edges. This means that simple graphs are:

- Directionless edges
- No multiple edges between the same pair of vertices
- No loops (edges from and to the same vertex) [22]

Here is a non exhaustive list of simple graphs:

1. **Connected Graphs**

   A graph that has a path P_n between every pair of vertices is said to be **"connected"**. In a graph, each maximal connected piece of a graph is called a **connected component**. For example, Figure 6 has two connected components, whereas Figure 7 only has one connected component.[6]

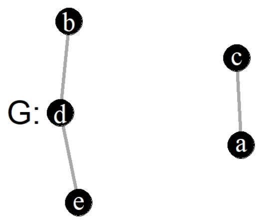


Figure 6: A simple graph

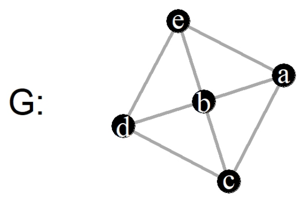


Figure 7: A connected graph

2. **Disconnected Graphs:**

   Disconnected graphs are not connected, meaning that there exists at least one pair of vertices with no path P_n between them. A disconnected graph G has n $\in \mathbb{N}$ connected components $G_1, G_2, \ldots, G_n$, eg: $\texttt{ae} \notin E$.

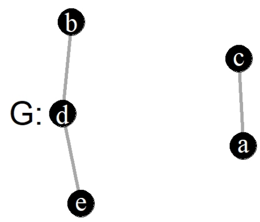


Figure 8: Disconnected graph

3. **Complete Graphs:**

   If a graph G has an edge between every pair of distinct vertices, it is said to be **complete** and denoted K_n. The number of edges in a complete graph (size of the graph) is given by the combination formula $\binom{n}{2}$, which represents the number of edges that link each pair of vertices from a set of n vertices[22]. This is calculated as follows:

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$$

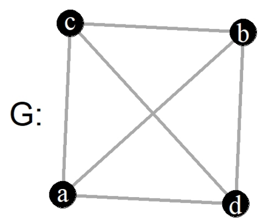


Figure 9: The complete graph K_4

4. **Trees and Forests:**

   A graph that is connected and acyclic is called a **tree**, while a forest is a collection of disjoint trees. Trees consist of two types of vertices: leaves (or leaf nodes), which have degree 1, and internal nodes of degree $\neq 1$ [15].

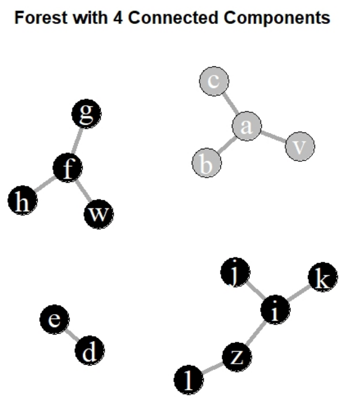


Figure 10: Forest, with tree G_1 in grey

Theorem 2.2. *If T is a tree of order n, then T has $|E| = n-1$*

Theorem 2.3. *If F is a forest of order n containing K connected componentd, then F has $|E| = n-K$ edges.*

Theorem 2.4. *A graph G of order n is a tree iff is is connected and contains $n-1$ edges*

Theorem 2.5. *A graph G of order n is a tree iff is is acyclic and contains $n-1$ edges*

Theorem 2.6. *Let T be a tree of order $n \geq 2$, then T has at least 2 leaves.*

5. **Planar and Non-planar Graphs:** Planar graphs are graphs that be drawn in a plane such that there's no edge crossing. A plane graph is a planar graph in its planar form (with no edge crossing).

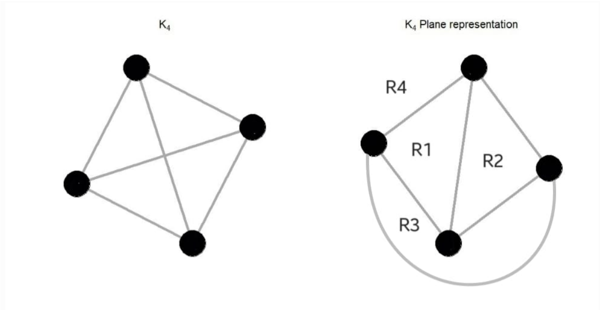


Figure 11: Planar graph representation of K_4

- Showing a graph G is **planar** is finding a planar isomorphic 2.14 graph to G, e.g K_4.
- Showing a graph G is **non-planar** is failing to find a planar isomorphic 2.14 graph to G, e.g $K_{3,3}$.

A planar graph divides the plane into regions. This means that if we remove the vertices and edges from the plane, we are left with disconnected pieces of the graph, each of which we call a **region or faces**.

The **boundary** $b(R)$ of a given region is all the edges and vertices incident to R and another region different than R. Notice that there is always one exterior region that contains all unbounded parts of the plane. [12]

Theorem 2.7 (Euler's Formula)**.** *For any connected planar graphs in the plane with V vertices, E edges and F faces, then V+E-F=2*

Proof. Let G be a connected simple planar graph with n vertices, m edges and r regions, then we need to prove that $n - m + r = 2$ holds for all types of graphs.

- Suppose G is a tree with n vertices, $(n-1)$ edges and $r = 1$ regions, then

$$n - m + r \Rightarrow n - (n-1) + 1 = 2$$

- (Contradiction) Suppose graph G connected of minimum size m such that $n - m + r \neq 2$. Let $e \in E(G)$ such and e lies in a cycle in G.

  Now consider $G-e$: $\because e$ lies in a cycle in G, $G-e$ is a connected Planar Graph (see Theorem 2.13). $G-e$ has n vertices, $(m-1)$ edges and $(r-1)$ region, hence

$$V(G) - E(G) + R(G) = 2 \Rightarrow n - (m-1) + (r-1) = 2 \Rightarrow n - m + r = 2$$

□

Theorem 2.8 (Upper bound for size of a planar graph)**.** *If G is a planar graph with $n \geq 3$ vertices and m edges then $m \geq 3n - 6$*

Proof. Consider the sum $B = \sum_{i=1}^{r \in R} b(R)$, $c \leq 2a$. Since each region is bounded by at least 3 edges, $B \geq 3r$, thus $3r \leq 2q$

$$3(2 + q - n) \leq 2q \Rightarrow m \leq 3n - 6$$

□

Theorem 2.9. *K_5 is non planar*

Theorem 2.10. *Every planar graph has a vertex of degree less than 5*

Proof. Let n be the order of G and m be the size of G. We can use the contrapositive to complete the proof

Assume G with $\delta(G) \geq 6$ is non planar. Since $\delta(G) \geq 6$, $\exists v \in G$ such that v is adjacent to 6 vertices $\therefore O(G) = 7$**.** From Theorem 2.8 we know that $m \leq 3n - 6$ for $n \geq 3$ and from Theorem 2.1, we know that $\sum_{v \in V(G)} \deg(v) = 2|E|$

Thus

$$\deg(v) = 2|E| \geq 6n \Rightarrow 2m \geq 6n \Rightarrow m \geq 3n > 3n - 6$$

Hence G is non-planar if $\delta(G) \geq 6$, hence G is planar if $\delta(G) < 5$. □

6. **Regular Graphs:**

   A graph G is said to be regular if each vertex $v \in V$ has the same degree k. We can say that G is a regular of degree k or is a $k - regular$ graph [22].

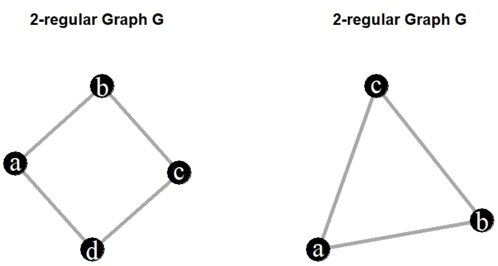


Figure 12: 2-regular graphs

7. **Cyclic Graphs:**

   A graph with n vertices (where, $n \geq 3$) and n edges forming a cycle of C_n with all its edges is known as a **cycle graph**.

   A cyclic graph may contain more than one cycle and is always 2-regular. See Figure 12. Graphs that are not cyclic are **Acyclic** [22].

8. **Bipartite Graphs:**

   A graph G is said to be bipartite when it can be partitioned into two sets of vertices V_1 and $V_2 \subseteq V$, with edges $e_i \in E$, $i \in I = \{1, 2, 3, \ldots, |E|\}$ of the graph connecting vertices in V_1 to those in V_2. A complete bipartite graph $K_{x,y}$ has an edge set of the form $E = \{xy \mid x \in X, y \in Y\}$.

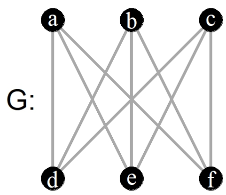


Figure 13: Bipartite graph $K_{3,3}$

   Theorem 2.11. *A graph with at least 2 vertices is bipartite if and only if it contains no odd cycles*

9. **Star Graphs:** A star graph is a complete bipartite graph in which n-1 vertices have degree 1 and a single vertex has degree (n - 1). This exactly looks like a star where (n - 1) vertices are connected to a single central vertex. A star graph with n vertices is denoted by S_n [22].

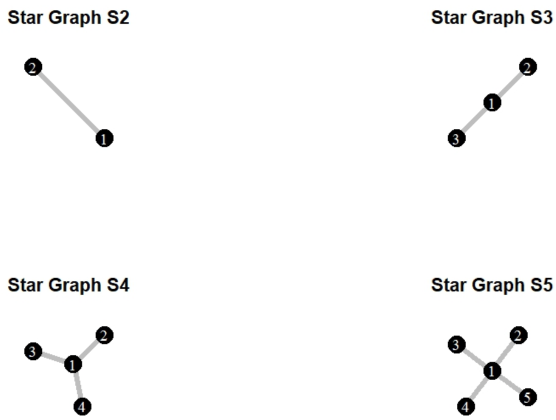


Figure 14: Star graphs

10. **Sub Graphs:**

    If $G(V, E)$ is a graph, then a subgraph $H(V', E')$ is a graph with the property that its sets of vertices and edges are subsets of the sets of vertices and edges of G, i.e. $V' \subseteq V$ and $E' \subseteq E$

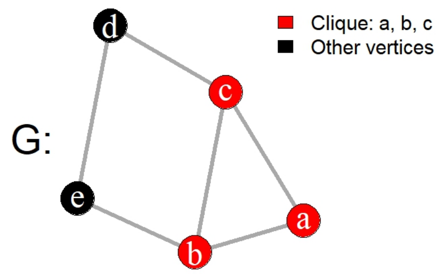


Figure 15: subgraphs and Clique

Remark 1. *In the example above* $G(V, E) = \{\{a, b, c, d, e\}, \{\mathtt{ab}, \mathtt{ac}, \mathtt{bc}, \mathtt{be}, \mathtt{cd}\}\}$. *Now if we let* $H(V', E') = \{\{a, b, c\}, \{\mathtt{ab}, \mathtt{ac}, \mathtt{bc}\}\}$, *then* H *is a subgraph of* G. *Moreover, since* H *is a complete graph,* H *is referred to as a* ***"clique"****.*

Types of cliques *A clique is a special type of subgraph where every pair of vertices is adjacent, forming a complete subgraph.*

- ***Clique****: A subset of vertices of a graph where each vertex is adjacent to every other vertex in the subset. In a complete subgraph, all possible edges are present between vertices.*
- ***Maximal Clique****: A clique that cannot be extended by adding an adjacent vertex. It's a clique that is not contained within any larger clique in the graph.*
- ***Maximum Clique****: A clique that contains the largest possible number of vertices in the graph. The size of the maximum clique is called the clique number, denoted as* $\omega(G)$.
- ***k-Clique****: A clique of exactly k vertices. For instance, a 3-clique is a triangle (a 3-vertex complete subgraph).*
- ***Clique Cover****: A partition of the graph's vertices into cliques such that each vertex belongs to exactly one clique. The smallest number of cliques required to cover the graph is known as the clique cover number.*

11. **Induced Sub Graphs:**

    (a) **Vertex induced Sub Graphs:** A subgraph, H of a graph G is vertex induced if $\forall u, v \in H \subseteq G$, the edges $u, v \in E(G)$ [7].

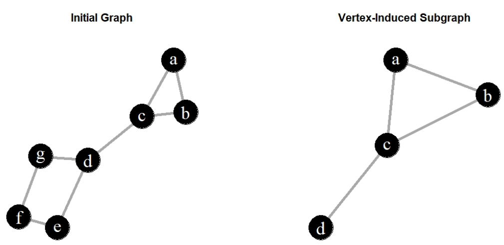


Figure 16: Vertex induced subgraphs

(b) **Edge induced Sub Graphs:** An edge-induced subgraph of a graph $G = (V, E)$ is a subgraph formed by selecting a subset of edges $E' \subseteq E$ from the original graph G and including only the vertices that are endpoints of these selected edges [7].

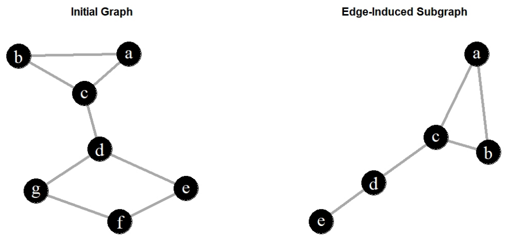


Figure 17: Edge induced subgraphs

Remark 2. • ***A vertex-induced subgraph*** *includes all edges that exist between the selected vertices in the original graph. It focuses on the selected vertices and the natural connections between them[7].*

• ***An edge-induced subgraph****, on the other hand, includes only the specified edges and the vertices incident to those edges, but not necessarily all connections between those vertices[7].*

As shown below, a vertex-induced subgraph is not always vertex-induced:

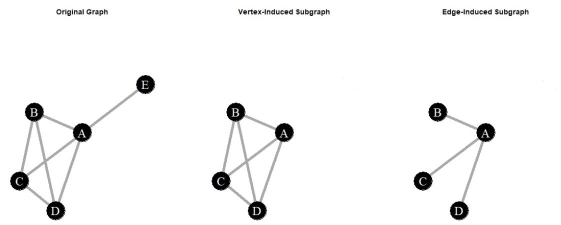


Figure 18: Vertex and Edge induced subgraphs

(c) **Spanning Sub Graphs:**

A spanning subgraph of a graph G is a subgraph that includes all the vertices of G but may include only a subset of its edges. In simpler terms, it is a subgraph that connects all the vertices of the original graph while potentially omitting some of the edges.

Formally, for a graph $G = (V, E)$, a subgraph $H = (V, E')$ is a spanning subgraph if:

$$V(H) = V(G) \quad \text{and} \quad E' \subseteq E(G)$$

12. **Weighted graphs**

A weighted graph is a type of graph where each edge is assigned a numerical value known as a weight. This makes it a specific kind of labeled graph, where the labels are numerical. The weight associated with each edge is denoted as $w(e)$, where e is an edge in the graph G.

Each edge typically has a corresponding non-negative integer weight. Weighted graphs can be either directed or undirected. The weight of an edge is commonly referred to as its "cost." In practical scenarios, the weight may represent factors such as the length of a route, the capacity of a connection, or the energy needed to move between locations along a route [2].

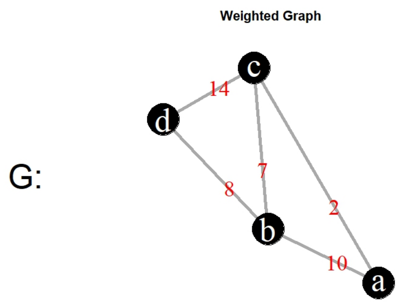


Figure 19: Weighted graph

Definition 2.4 (**Neighborhood and Adjacency**). *In a Graph, two vertices $a, b \in G$ are said to be* ***adjacent*** *if they share an edge $e \in E(G)$. The* ***neighbourhood*** *$N(v)$ of a vertex v is the set of all adjacent vertices. A vertex v is called an* ***"isolated vertex"****, when it is adjacent to no other vertex in the graph. Isolated vertices are* ***connected and complete****.*

For a given vertex v in a graph G, the **degree** $deg(v)$ is the number of adjacent vertices to v or vertices in the neighbourhood of vertex v. In simple graphs, $deg(v)$ is the cardinality of $N(v)$.

Definition 2.5 (**Path vs Trail**). *In graph theory, a walk is a sequence of non-distinct adjacent vertices. In any walk initial vertex and end vertex are called* ***terminal vertices****. There are 2 types of walks:*

1. ***Paths*** *P_n : are walks where the vertices in the sequences are distinct (no vertex repetition in the walk sequence). A closed path is called a* ***cycle*** *C_n*

   - ***A Hamiltonian path:*** *A Hamiltonian path is a path in a graph that visits every vertex exactly once. If such a path starts and ends at the same vertex, it is called a Hamiltonian cycle (or circuit). A Hamiltonian graph has a Hamiltonian cycle.*

2. ***Trails****: are walks where the edges in the sequences are distinct (no edge repetition in the walk sequence). A closed trail is called a* ***circuit****.*

   - ***Eulerian trail****: An Eulerian trail is a path in a graph that visits every edge exactly once. An Eulerian graph is a connected graph in which there exists a closed trail that visits every edge exactly once (known as an Eulerian circuit) [6].*

Every path is a trail, but not every trail is a path.

Theorem 2.12. *In a connected graph G with $u, v(G) \in G$, every $u - v$ walk of length l contains a path.*

Proof. Let $(u_0 = u, u_1, u_2, \ldots, u_k = v)$ be a shortest $u - v$ walk in G of length $k \leq l$

Claim: Suppose that the shortest $u - v$ walk is not a path (allows duplicate vertices), thus $u_i = u_j$ for some i and j where $0 \leq i < j \leq k$.

Delete $u_{i+1}, u_{i+2}, \ldots, u_j$, thus $(u_0 = u, u_1, u_2, \ldots, u_i, u_{j+1}, u_{j+2}, \ldots, u_k = v)$ is of length less than k, leading to a contradiction.$\Rightarrow\!\Leftarrow$ $\square$

Definition 2.6 (Vertex Deletion). *Given the graph G, if $v \in V(G)$, the graph $G - v$ is obtained by removing the vertex v and its incident edges. Furthermore, if S is a set of vertices v_i, the graph $G - S$ is the result of removing the vertices v_i and their incident edges.*

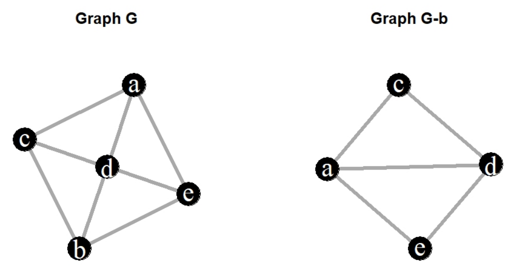


Figure 20: Vertex deletion

Definition 2.7 (Edge Deletion). *In a graph G, if $e \in E(G)$, the graph $G - e$ is obtained by only removing the edge e (incident vertices to e stay in the Graph). Similarly as in vertex deletion, if T is a set of edges e_i, then the graph $G - T$ after edge deletion is obtained by removing edges $e_i \in E(G)$.*

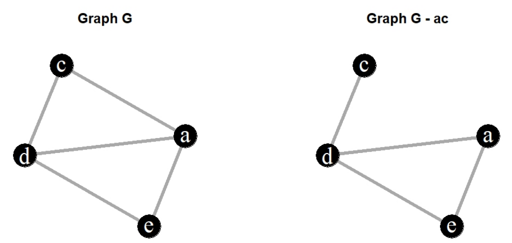


Figure 21: Edge deletion

Definition 2.8 (Cut vertex and cut set). *In a graph G, if a vertex deletion $G - v, v \in V(G)$ increases the number of connected components, then v is a **cut vertex**. If the deletion of a collection of vertices $S \subset G$ increases the number of connected components, then the set S is a **cut set**. Complete graphs K_n have no cut vertices [2].*

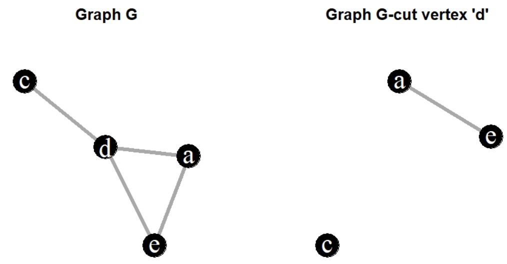


Figure 22: Cut vertex

Definition 2.9 (**Bridge, Isthmus or Cut edge**). *Let G be a connected graph. An edge e from G is called as isthmus or bridge or cut edge, if $G - e$ (edge deletion) results in an increased number of connected components in the graph. A graph is said to be bridgeless or isthmus-free if it contains no bridges [2].*

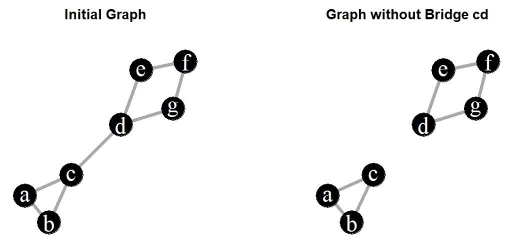


Figure 23: Cut vertex

Theorem 2.13. *In a graph G, an edge e is a bridge if and only if it lies in no cycles.*

Proof. We will prove the theorem by contrapositive.

$\Rightarrow$ If e is a bridge $\rightarrow$ e lies on no cycles.

Let $e = uv \in G$. Suppose e lies on a cycle of the connected component G_1 and there exists a $u - v$ path, P' not containing e. We want to show that $G_1 - e$ is connected.

Let $x, y \in V(G_1 - e)$ such that $\exists$ an $x - y$ path P in G_1. if e is not contained in P, then P exists in $G_1 - e$. Suppose e is on path P, replacing P with P' produces a $x - y$ walk that does not contain e.

Thus, there exists an $x - y$ walk in $G_1 - e$, hence $G_1 - e$ is connected as e is not a bridge of G_1.

By Theorem 2.12 since there exists a $x - y$ walk in $G_1 - e$, there exists a $x - y$ path in $G_1 - e$.

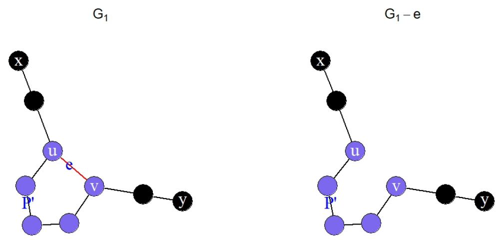


Figure 24: Visualisation of theorem

$\Leftarrow$ Suppose e lies on no cycles $\rightarrow$ e is a bridge.

Let $e = uv$ be an edge of G (say e is in the component G_1). Assume e is not a bridge of G. This means that $G_1 - e$ is connected.

Since $G_1 - e$ is connected, there exists a $u - v$ path, $P \in G_1 - e$. Therefore, P does not contain e. Combining the path P with the edge e produces a cycle C_n and e lies on $C_n \in G_1 \in G$.

This completes the proof. □

Definition 2.10 (Connectivity). *In graph theory, connectivity represents how connected a graph is according to its vertices or edges. There are mainly two types of connectivity in graph theory:*

Subdefinition 2.10.1 (Vertex connectivity). *The vertex connectivity of a graph G, denoted as $\kappa(G)$, is the smallest number of vertices that need to be removed from the graph to disconnect it or reduce it to a trivial graph. A graph is k-connected if $\kappa(G) = k$, meaning at least k vertices need to be removed to disconnect the graph [2].*

Bounds:

$$\text{In general } 0 \leq \kappa(G) \leq n-1 \quad \text{for a disconnected graph, where } n = |V|$$

$$1 \leq \kappa(G) \leq n-1 \quad \text{for a connected graph, where } n = |V|$$

Subdefinition 2.10.2 (Edge connectivity). *The edge connectivity of a graph G, denoted as $\lambda(G)$, is the smallest number of edges that need to be removed to disconnect the graph.*

Bounds:

$$\lambda(G) \geq 0 \quad \text{(for a disconnected graph, it's 0)}$$

$$\lambda(G) \leq n-1$$

[2].

Definition 2.11 (Graph Isomorphism). *There are an infinite number of ways that a given graph can be represented. So long as the vertex and edge set are identical, various iterations of a graph represent the same graph structure. Therefore, if two graphs have the same structure, they are the same graph.*

Let G and H be Graphs, then the graphs are isomorphic if we can define a function $\varphi : V(G) \to V(H)$ as $\varphi(v_G) \mapsto v_H$. We denote "G isomorphic to H" as $G \cong H$. If $G \cong H$ then if we consider a vertex $\varphi(v_G) \in G$, this vertex plays the same role as the vertex $v_H \in H$. The function φ is defined to be:

1. ***Injective:*** *$\forall v_1, v_2 \in G$, if $\varphi(v_1) = \varphi(v_2)$, then $v_1 = v_2$* ***i.e., no two distinct vertices** $v_1, v_2 \in V(G)$ **are mapped to the same vertex in** $V(H)$**.***

2. ***Surjective:*** *$\forall v_H \in H, \exists v_G \in V(G)$, such that $\varphi(v_G) = v_H$*

Theorem 2.14 (**Degree Invariance Under Isomorphism**)**.** *If $G \cong H$, then degrees of v_G, $deg(v_G) \in G$ are the same as degrees of v_H, $deg(v_H) \in H$*

Proof. Let $\varphi : V(G) \to V(H)$ be an isomorphism, and let $v \in V(G)$ such that $\varphi(v) = u \in V(H)$. Then v is adjacent to vertices $w_1, w_2, \ldots, w_k$ and not adjacent to vertices $x_1, x_2, \ldots, x_l$, where $k, l \in \mathbb{N}$. Since v is not adjacent to itself, v is included in the set of non-adjacent vertices $\{x_1, x_2, \ldots, x_l\}$, and $|V(G)| = k + l$. Therefore, $\deg_G(v) = k$.

Similarly, u is adjacent to $\varphi(w_1), \varphi(w_2), \ldots, \varphi(w_k) \in N(u)$ and not adjacent to $\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_l)$. $\therefore$ $|V(H)| = k + l$, and u has the same degree in H as v has in G. Hence, $\deg_H(u) = k$. $\square$

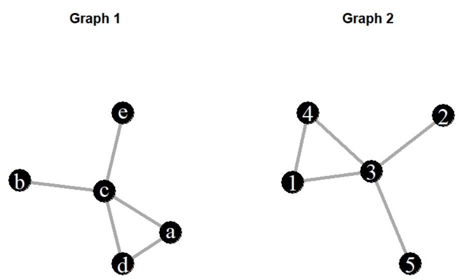


Figure 25: Isomorphic graphs

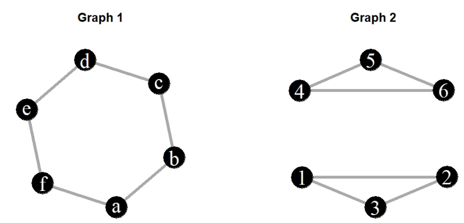


Figure 26: Non-Isomorphic graphs

Definition 2.12 (**Distance**)**.** *In mathematics, the distance function $M(x, y)$, or distance metric, is a function that defines the distance between two points in a space, say X. The distance function must satisfy the following properties:*

(M1) Non-negativity: $\forall x, y \in X$, $M(x, y) \geq 0$ and $M(x, y) = 0$ if and only if $x = y$

(M2) Symmetry: $\forall x, y \in X$, $M(x, y) = M(y, x)$

(M3) Triangle inequality: $\forall x, y, z \in X$, $M(x, y) \leq M(x, z) + M(z, y)$ [17]

In graph theory, the distance $d(u, v)$ between the vertices $u, v \in V(G)$ is the length of the shortest $u - v$ path in (G) [2]

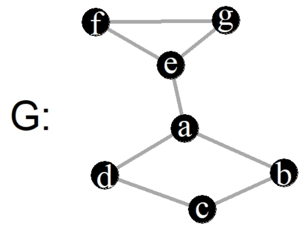


Figure 27: Graph G

Definition 2.13 (**Eccentricity**). *In graph theory, the eccentricity of a vertex v in a graph is defined as the greatest distance from v to any other vertex in the graph. As $d(v, u)$ represents the shortest path distance between v and another vertex u, the eccentricity $e(v)$ of v is given by:*

$$e(v) = \max\{d(v, u) \mid u \in V(G)\}$$

where $V(G)$ is the set of all vertices in the graph G [2].

Definition 2.14 (**Radius and Diameter**). *In graph theory, the radius of a graph G is defined as the minimum eccentricity of any vertex in the graph. As $e(v)$ denotes the eccentricity of a vertex v, the radius $r(G)$ is given by:*

$$r(G) = \min\{e(v) \mid v \in V(G)\}$$

The diameter of a graph G is the greatest eccentricity among all vertices in the graph. It represents the longest shortest path between any two vertices. Formally, the diameter $d(G)$ is given by:

$$d(G) = \max\{e(v) \mid v \in V(G)\}$$

where $V(G)$ is the set of all vertices in the graph G[2].

Definition 2.15 (**Center and Periphery**). *In graph theory, the center of a graph G is the set of all vertices that have the minimum eccentricity. Formally, the center $C(G)$ is defined as:*

$$C(G) = \{v \in V(G) \mid e(v) = r(G)\}$$

The periphery of a graph G is the set of vertices that have the maximum eccentricity. Formally, the periphery $P(G)$ is given by:

$$P(G) = \{v \in V(G) \mid e(v) = d(G)\}$$

where $d(G)$ is the diameter of the graph, $e(v)$ is the eccentricity of vertex v, and $r(G)$ is the radius of the graph [2].

Theorem 2.15. *For any connected graphs G, $rad(G) \leq diam(G) \leq 2rad(G)$*

Proof. Let u and v be two vertices in G such that $d(u, v) = diam(G)$, which is the maximum distance between any two vertices in G. Let w be a vertex in the center of G, denoted as $center(G)$. By definition,w is a vertex such that the eccentricity $ecc(w)$ is minimized. Hence, $ecc(w) = rad(G)$, where $rad(G)$ is the radius of G.

By property (M3), and using information from Figure 28 and item (M2), we have the following inequality:

$$diam(G) = d(u, v) \leq d(u, w) + d(w, v).$$

Since w is in the center of G, the distance from u to w and from w to v are at most the radius of G, which gives:

$$d(u, w) \leq rad(G) \quad \text{and} \quad d(w, v) \leq rad(G).$$

Therefore,

$$d(u, v) \leq d(u, w) + d(w, v) \leq rad(G) + rad(G) = 2rad(G).$$

Thus, we have:

$$diam(G) \leq 2rad(G).$$

□

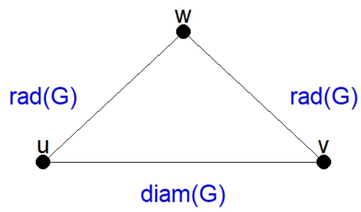


Figure 28: Triangle inequality in proof

For example, Figure 27 has the center, periphery, radius and diameter $C(G) = \{a\}$, $P(G) = \{c, f, g\}$, $r(G) = 2$, $d(G) = 4$, respectively. Furthermore, distances and eccentricities can be observed in the tables below.

Distance matrix for G

	a	b	c	d	e	f	g
a	0	1	2	1	1	2	2
b	1	0	1	2	2	3	3
c	2	1	0	1	3	4	4
d	1	2	1	0	2	3	3
e	1	2	3	2	0	1	1
f	2	3	4	3	1	0	1
g	2	3	4	3	1	1	0

Figure 29: Distance Matrix

Eccentricities for G

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

Figure 30: Eccentricities

3 Graph Coloring Definitions

3.1 Vertex Coloring

A proper coloring (or just coloring) of a graph G is an assignment of colors to the vertices of G such that adjacent vertices are colored differently. Another definition of a proper k-coloring is a function $K : V(G) \to \{1, 2, \ldots, k\}$ from the vertex set into the set $\{1, 2, \ldots, k\}$, where if $u, v \in V(G)$ are adjacent, $k(u) \neq k(v)$.

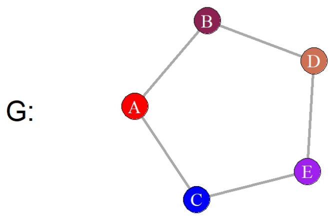


Figure 31: Coloring of C_5

3.1.1 Chromatic number $\chi(G)$ and Minimum coloring

The chromatic number of a graph, usually denoted by $\chi(G)$, is the smallest number of colors required to color the vertices of the graph such that no two adjacent vertices share the same color. In other words, it is the minimum number of distinct colors needed to properly color the vertices while ensuring that adjacent vertices are not the same color.[8]. A graph G with chromatic number exactly $\chi(G) = k$ is called k-chromatic or k-colorable and k is the minimum coloring.

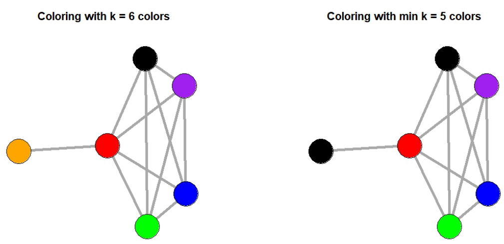


Figure 32: K-Coloring Vs Minimum K-Coloring

Rowland Leonard Brooks (1916–1993) was a British mathematician who studied mathematics at the University of Cambridge. Brooks worked closely with several prominent mathematicians of his time, contributing to various areas of mathematics, including combinatorics and graph theory. His career was marked by a deep analytical approach to mathematical problems, especially those related to the structure of graphs. pAmong his most famous contributions is Brooks' Theorem, which addresses the problem of vertex coloring in graphs. This theorem provides a powerful result in graph theory: for any connected simple graph that is neither a complete graph nor an odd cycle, the chromatic number is at most equal to the maximum vertex degree [21]

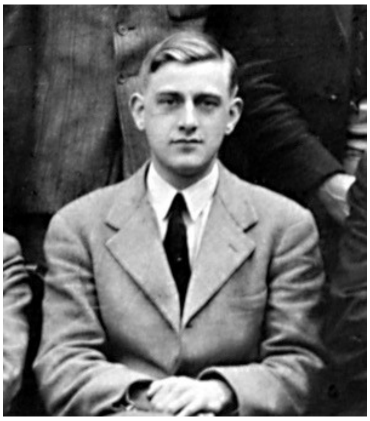


Figure 33: Leonard Brooks in 1938

Remark 3. *There exist predetermined values and bounds for $\chi(G)$, :*

- *For any connected graphs, we have G $\omega(G) \leq \chi(G) \leq \Delta(G)$* ***(Brooks' Theorem)***
- *For $G = C_{2n}$, $\chi(G) = 2$*
- *For $G = C_{2n} + 1$, $\chi(G) = 3$*
- *For $G = K_n$, $\chi(G) = n$*
- *For $G = P_n$, $\chi(G) = 2$ if $n \geq 2$ and $\chi(G) = 1$ if $n = 1$*
- *For bipartite graphs and complete bipartite $K_{m,n}$, $\chi(G) = 2$.* ***This ensues that 2-colorable graphs are bipartite!***
- *For $G = E_n$, $\chi(G) = 1$*

Definition 3.1 (**Independent vertex set I.V.S.**)**.** *An independent vertex set (also called an independent set) in a graph is a subset of vertices such that no two vertices in the subset are adjacent. In other words, if I is an independent set in a graph $G = (V, E)$, then for every pair of vertices $u, v \in I$, there is no edge$(u, v) \in E$ connecting them [8].*

Formally:

Let $G = (V, E)$ be a graph where V is the set of vertices and E is the set of edges. A subset $I \subseteq V$ is an independent vertex set if for all $u, v \in I$, $(u, v) \notin E$.

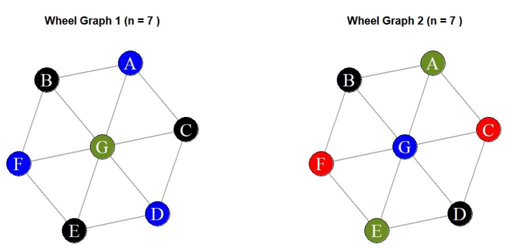


Figure 34: Coloring of W_7

Remark 4. *As graphed above, we can see that we can color the wheel Graph W_7 with 3 colors or more. However, the graph W_7 has chromatic number $chi(G) = 3$ bounded by $\omega(G) = 3 \leq \chi(G) \leq \Delta(G) = 5$. Thus W_7 has I_i independent vertex sets $\{G\}, \{A, D, F\}, \{B, C, E\}$, $\{ B, D\}, \{A, E\}, \{F, C\}$. For the graph W_7:*

- *$|I_4| = 2$ then I_i is maximal, meaning no vertex can be added to complete this independent set.*
- *$|I_2| = \alpha(G) = 3$ then I_i is maximum, meaning no vertex can be added to complete this independent set, where $\alpha(G)$ **is the independence number** [8].*

Vertex Coloring Algorithms

- **Greedy Coloring Algorithm** This is a straightforward approach where vertices are assigned the smallest possible color that has not been used by their adjacent vertices. It doesn't always yield the optimal solution but is often used due to its simplicity [6].

  Example: For a simple graph, you would color the first vertex, then proceed to the next, assigning the lowest available color that hasn't been used by adjacent vertices.

- **Backtracking Algorithm:**

  This is an exhaustive search method that tries every possible coloring combination by assigning colors to the vertices recursively **using state space trees**. If a conflict arises (i.e., two adjacent vertices have the same color), the algorithm backtracks to try a different coloring configuration. It guarantees an optimal solution but can be computationally expensive for large graphs [11].

3.2 Edge Coloring

Edge coloring is a concept in graph theory that involves assigning colors to the edges of a graph in such a way that no two edges that share a common vertex (i.e., adjacent edges) have the same color. The goal of edge coloring is to minimize the number of colors used, which is known as the chromatic index of the graph [6].

3.2.1 Chromatic index $\chi'(G)$ and Minimum coloring

The chromatic index of a graph, also known as the edge chromatic number, is the smallest number of colors needed to color the edges of the graph so that no two adjacent edges (edges that share a common vertex) have the same color. It is denoted as $\chi'(G)$ [6].

Definition 3.2 (Independent edge set or Matchings). *A matching in a graph is a set of edges such that no two edges share a common vertex. In other words, it is a collection of pairwise non-adjacent edges. Matchings can be categorized into various*

- ***Maximum Matching:*** *A matching that contains the largest possible number of edges.*

- ***Perfect Matching:*** *A matching that covers every vertex of the graph. This means every vertex is incident to exactly one edge in the matching.*
- ***Minimal, Minimum, Maximal, perfect, etc.***

In a graph, matching can have various effects such as exposing (un-saturating) and saturating vertices:

- **Saturated Vertices** A vertex is considered saturated if it is incident to an edge in a matching. In other words, a saturated vertex has been paired with another vertex through the matching edges [8].
- **Exposed or Unsaturated Vertices** A vertex is considered exposed if it is not incident to any edge in the matching. This means that it has not been paired with any other vertex through the edges of the matching [8].

Vizing's Theorem

Vadim Georgievich Vizing was a Russian mathematician whose pioneering contributions significantly advanced the field of graph theory, particularly in edge coloring. His most famous result, known as Vizing's Theorem, introduced a critical distinction between two classes of graphs based on their chromatic index. [9]

Figure 35: Vadim G. Vizing in 1975

The **Vizing's Theorem**, states that a graph can be edge-colored using either $\Delta(G)$ (the maximum degree of the graph) or $\Delta(G) + 1$ colorsm hence the 2 classes. [8].

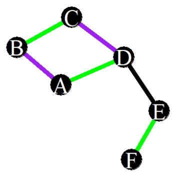


Figure 36: Edge coloring

Remark 5. *As graphed above, the graphs have chromatic index bounded by* $\Delta(G) + 1 \leq \chi'(G) \leq \Delta(G) + 1$*. Thus, G above has* M_i *matchings* $\{AB, CD, EF\}$*,* $\{AB, CD, EF\}$*,* $\{BC, AD\}$*,* $\{BA, DF\}$*,* $\{CD, EF\}$*, etc. For the graph G:*

- $|M_3| = 2$ *then* M_3 *is not maximal, meaning edges can be added to complete this independent set.*
- $|M_1| = |M_2| = 3$ *then* M_1 *and* M_2 *are maximal and maximum matchings, meaning no edges can be added to complete this independent set and it is one of the largest matching [8]. The graph is said to be perfect since it contains matchings that saturate all vertices in the graph.[6].*

Example: The algorithm starts with an initial improper coloring and iteratively adjusts the coloring until a proper edge coloring is achieved.

- **Blossom Algorithm for Edge Coloring:**

  The Blossom algorithm, originally developed for matching problems, can be adapted for edge coloring. It finds a maximum matching and then extends it to find an edge coloring. It's efficient for dense graphs and is based on finding augmenting paths to adjust existing matchings. This is particularly useful for bipartite graphs, where edge coloring is closely related to perfect matchings [20].

3.3 Total Coloring

Total coloring is an assignment of colors to both the vertices and edges of a graph such that no adjacent vertices, no adjacent edges, and no vertex and its incident edges share the same color [16].

3.3.1 Total Chromatic number $\chi''(G)$ and Minimum coloring

The total chromatic number of a graph, $\chi''(G)$, is the smallest number of colors needed to color both the vertices and edges of a graph such that no adjacent vertices, no adjacent edges, and no vertex and its incident edges share the same color. The upper bound for the total chromatic number is

$$\chi''(G) \leq \Delta(G) + 2$$

[16]

Mehdi Behzad's work on the *Total Coloring Conjecture* significantly advanced graph theory. It is this conjecture, that states that the total chromatic number $\chi''(G)$ of any graph G is at most $\Delta(G) + 2$, where $\Delta(G)$ is the highest degree of G. This realization has had a significant impact on graph coloring research and is still an open problem for many graph classes.

Figure 37: Medhi Bezhad

Behzad's and Vizing's work on edge coloring, often studied in combination, has laid the groundwork for the study of graph colorings in a variety of applications, ranging from network topology, scheduling, and frequency assignments to more intricate structural problems in theoretical computer science.

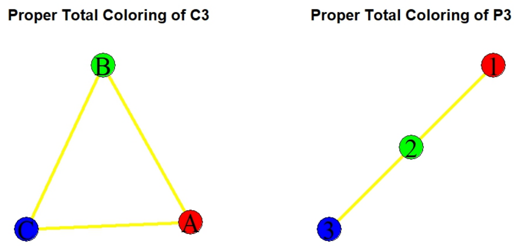


Figure 38: Total coloring

4 Applications

As the real-world applications of graph coloring became more complex, the need for efficient algorithms to find optimal colorings became clear. While graph coloring is easy to define, finding the minimal number of colors for a graph is a **NP-hard problem (non-deterministic polynomial time)**, meaning that there is no known efficient algorithm that solves all instances of the problem in a reasonable time. This led to the development of approximation algorithms and heuristics, which attempt to find good enough solutions even if they are not optimal [18].

Some application vertex or edge coloring below illustrate why graph coloring became Important:

1. **Map Coloring and Geography:** The problem of map coloring motivated the study of graph coloring. The Four Color Theorem, in particular, provided a solution to the issue of determining the minimum number of colors required to color regions on a map without adjacent regions sharing the same color [8].
2. **Scheduling Problems:** In the 20th century, the application of graph coloring exploded due to the growing need to solve scheduling problems. For example, in

university exam scheduling, certain exams need to be scheduled at different times to avoid conflicts for students taking multiple exams. Graph coloring can model this scenario by treating each exam as a vertex and drawing edges between exams that conflict [1].

3. **Register Allocation in Compilers:** In computer science, register allocation in compilers became another major application of graph coloring. When compiling a program, the computer must allocate variables to a limited number of processor registers. The graph coloring problem here involves assigning different variables to different registers, ensuring that variables that are used simultaneously do not share the same register [14].

4. **Frequency Assignment in Wireless Networks:** Graph coloring is also essential in telecommunications, particularly in frequency assignment problems, where adjacent radio towers or mobile stations must use different frequencies to avoid interference. This problem is modeled as a graph in which the vertices represent stations or towers, and the edges represent potential interference. [19]

5. **Sudoku and Puzzles:** Even in recreational mathematics, graph coloring has applications. In puzzles like Sudoku, you can treat the Sudoku grid as a graph and apply graph coloring principles to solve the puzzle. This connection to combinatorics and optimization problems increased interest in the field.[10]

6. **Parallel computing:** Ensuring tasks that must run concurrently are not assigned to the same processor.[3]

7. **Resource allocation:** Assigning resources like bandwidth, time slots, or frequencies in ways that avoid conflicts. Graph coloring games: Puzzle games and competitions based on graph theory.[13]

8. **GPS or Google Maps:** GPS or Google Maps are used to find the shortest route from one place to another. The places are presented by vertices and their connections are presented by edges. The best route is determined by the software picking the shortest distance.[10]

9. **Optimizing path designs based on Desire lines:** It can be used to reduce conflicts, improve safety, and enhance user experience by carefully assigning and optimizing paths in urban and open spaces. This approach blends the mathematical structure of graphs with real-world applications in urban planning [5].

Figure 39: Ohio State University

5 Conclusion

Graph coloring is an important method used in many areas, such as computer science, epidemiology, mathematics, scheduling, and network design. As research on graph coloring continues, coloring and optimization becomes more effective reducing conflicts. Graph Coloring's importance will continue to grow as technology advances and new challenges arise. This report lays ground in understanding the basics and theorems pertaining to graphs and their coloring, highlighting that no standard coloring algorithm works for every graph or that an optimization problem might require high computation power and time.

References

[1] Akbulut, A., & Yılmaz, G. (2013). "University Exam Scheduling System Using Graph Coloring." *International Journal of Innovation, Management and Technology*, vol. 4, no. 1, pp. 97–100.

[2] Anon (n.d.). Dr. D. Y. Patil Unitech Society. *Connected Graph.*

[3] Bogle I., Slota G. M., Boman E. G.,Devine K. D., and Rajamanickam S. (2022). "Parallel Graph Coloring Algorithms for Distributed GPU Environments," *Parallel Computing Systems and Applications*, ISSN 0167-8191, vol. 110.

[4] Calinger R., S. (2016) *Leonhard Euler: Mathematical Genius in the Enlightenment.* Princeton University Press.

[5] Davenport J. (2022). "The Meraki Spark". *Desire Paths: Ignore Challenge or Embrace?.*

[6] Diestel R. (2017). *Graph Theory*, 5th edition, Springer.

[7] Dr. Ameen. (2020). *Introduction to Graphs.*

[8] Durán G. (2007). "Departamento de Ingeniería Industrial, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, XII ELAVIO". *Graph Coloring Problems.* Itaipava, Brasil.

[9] European Mathematical Information Service (EMIS), (2000). *EMIS Newsletter No. 38*, pp. 22.

[10] Gangrade A., Agrawal B., Kumar S., and Mansuri (2022). "Graph Coloring in Statistics and Applied Mathematics," *International Journal of Statistics and Applied Mathematics*, ISSN 2394-6958, vol. 7, no. 2, pp. 51-53.

[11] GeeksforGeeks. (2024). "Backtracking Algorithms", `https://www.geeksforgeeks.org/backtracking-algorithms/` [15 November 2024]

[12] Glickenstein, D. (2012). *Math 443/543 Graph Theory Notes 5: Planar Graphs and Coloring.* Retrieved from University of Arizona website.

[13] Ivanov, A., Tonchev, K., Poulkov, V., Manolova, A., & Neshov, N. N. (2022). "Graph-Based Resource Allocation for Integrated Space and Terrestrial Communications". *Sensors 5778*, vol. 22, ISSN 15.

[14] Jain, M. (2015). "Register Allocation via Graph Coloring." *HCL Technologies*

[15] Jungnickel, D. (2013). *Graphs, Networks and Algorithms* (4th ed.). Springer.

[16] Krishnakanth G., J. (2019). "A Study on Total Coloring of Some Graphs," *Journal of Information and Computational Science*, vol. 9, no. 12, ISSN: 1548-7741.

[17] Makarychev Y., (n.d). *Basic Properties of Metric and Normed Spaces.*

[18] Naseera J., *P, NP, NP-Hard & NP-complete problems.*

[19] Noor M., A. (2010). *Frequency Assignment Problem in Cellular Networks.*

[20] Shoemaker, A., Vare, S. (2016). "Stanford University". *Edmonds' Blossom Algorithm.*

[21] Squaring.net, *Brooks, Smith, Stone, and Tutte: A History and Theory of Graph Coloring.*

[22] TutorialsPoint, *Graph Theory: Types of Graphs*. Available at: `https://www.tutorialspoint.com/graph_theory/types_of_graphs.htm`.