Planar graphs, Connectivity, Directed graphs

Detailed Concepts, Examples, and Exercises

SDB

Spring 2025

Outline

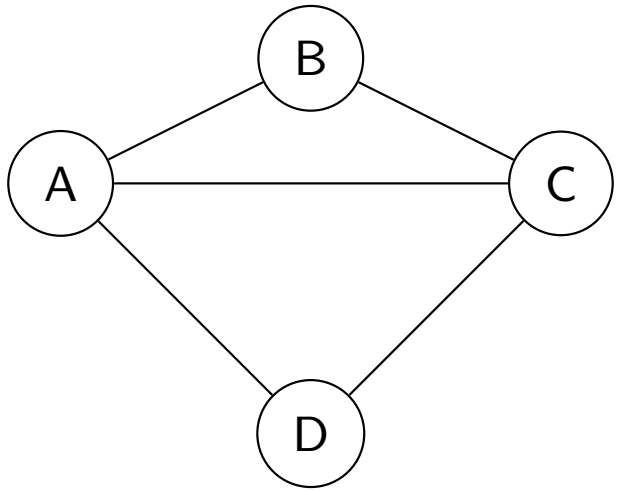
Introduction to Planar Graphs

Definition: A graph is **planar** if it can be drawn on a plane without edge crossings.

Key Properties:

- Planar graphs have a special structure that follows certain constraints.
- They are crucial in circuit design, geographic mapping, and network visualization.

Example:



Exercise: Show that K_4 is planar but K_5 is not.

Euler's Formula for Planar Graphs

Theorem: For a connected planar graph with V vertices, E edges, and F faces:

$$V - E + F = 2.$$

Proof Outline:

- Base Case: Holds for trees ($E = V - 1, F = 1$).
- Inductive Step: Adding an edge creates a new face, preserving $V - E + F$.

Exercise: Verify Euler's formula for a cube's graph representation.

Characterizations of Planar Graphs

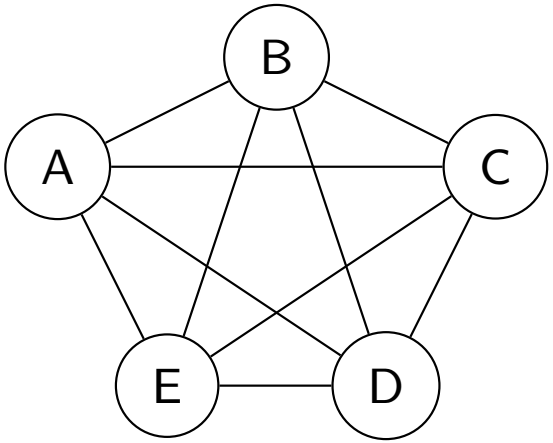
Kuratowski's Theorem: A graph is planar if and only if it does not contain a subgraph homeomorphic to K_5 or $K_{3,3}$.
Wagner's Theorem: A graph is planar if and only if it does not have K_5 or $K_{3,3}$ as a minor.
Exercise:

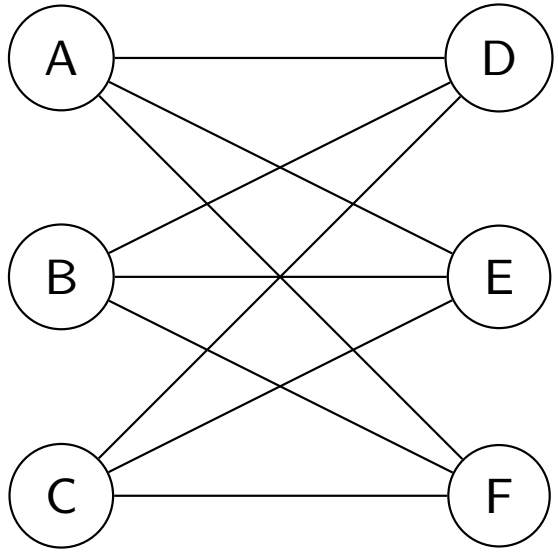
- Show that K_4 is planar but K_5 is not.
- Prove that every tree is planar.

Planarity Testing

Kuratowski's Theorem: A graph is non-planar if and only if it contains a subgraph homeomorphic to K_5 or $K_{3,3}$.

Wagner's Theorem: A graph is planar if and only if it does not have K_5 or $K_{3,3}$ as a minor.





Exercise:

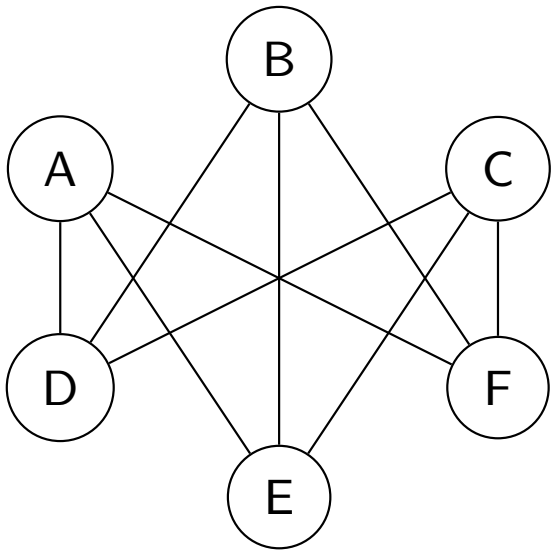
- Prove that K_5 is non-planar using Euler's formula.
- Show that $K_{3,3}$ is not planar.
- Find a minor in a given graph that makes it non-planar.

Kuratowski's Theorem vs. Wagner's Theorem on Planarity

Key Differences:

- **Homeomorphism (Kuratowski's Theorem):** - A graph is homeomorphic to another if it can be obtained by subdividing edges. - Kuratowski's theorem looks for these subdivisions explicitly.
- **Minors (Wagner's Theorem):** - A minor is obtained by deleting edges and contracting edges. - Wagner's theorem allows contractions, making it more general.

Example: A Graph Containing $K_{3,3}$ as a Minor



Exercise:

- Find a graph that is non-planar by Kuratowski's theorem but not by Wagner's theorem.
- Prove that every graph with a $K_{3,3}$ minor is non-planar.

Planarity Testing Algorithms

Key Algorithms:

- **Kuratowski's Theorem Approach:** Check for K_5 and $K_{3,3}$ subgraphs.
- **Hopcroft-Tarjan Planarity Algorithm:** Linear-time planarity testing.
- **PQ-Tree Algorithm:** Used for practical graph drawing.

Hopcroft and Tarjan's Algorithm Steps:

- Uses DFS to test planarity in $O(n)$ time.
- Identifies and removes Kuratowski subgraphs.

Exercise:

- Apply Hopcroft and Tarjan's Algorithm to test if a given graph is planar.
- Apply the Hopcroft-Tarjan algorithm to test the planarity of a given graph.

The Five-Color and Four-Color Theorems

Five-Color Theorem: Every planar graph can be properly colored with at most 5 colors.
Four-Color Theorem (Appel-Haken 1976): Every planar graph can be colored with at most 4 colors, but proof requires computer verification.
Exercise: Prove the Five-Color Theorem using induction.

5-Color Theorem

Theorem: Every planar graph can be colored with at most 5 colors.
Sketch of Proof:

- Uses induction and reducible configurations.
- Stronger than the 6-color theorem but weaker than the 4-color theorem.
- Base case: A simple cycle is 5-colorable.
- Inductive step: Remove a vertex, color remaining graph, reinsert vertex with valid coloring.

Exercise:

- Construct a planar graph that requires exactly 5 colors.
- Prove that every planar graph is 6-colorable.

Key Theorems on Planarity

Key Theorems:

- **Euler's Formula:** $V - E + F = 2$ for connected planar graphs.
- **Kuratowski's Theorem:** A graph is planar iff it has no K_5 or $K_{3,3}$ subgraphs.
- **Wagner's Theorem:** A graph is planar iff it has no K_5 or $K_{3,3}$ minors.
- **Five-Color and Four-Color Theorems:** Planar graphs can be colored with at most 4 colors.

Key Algorithms:

- Hopcroft-Tarjan planarity testing.
- PQ-tree method for efficient planarity verification.

Exercises:

- Prove that every planar graph has a vertex of degree at most 5.
- Find an example of a non-planar graph and explain why.

Outline

Graph Connectivity – Basics

Definitions:

- A graph is **connected** if there is a path between any two vertices.
- **Vertex connectivity** $\kappa(G)$ is the minimum number of vertices needed to disconnect the graph.
- **Edge connectivity** $\lambda(G)$ is the minimum number of edges needed to disconnect the graph.

Exercise:

- Find $\kappa(G)$ and $\lambda(G)$ for small graphs.

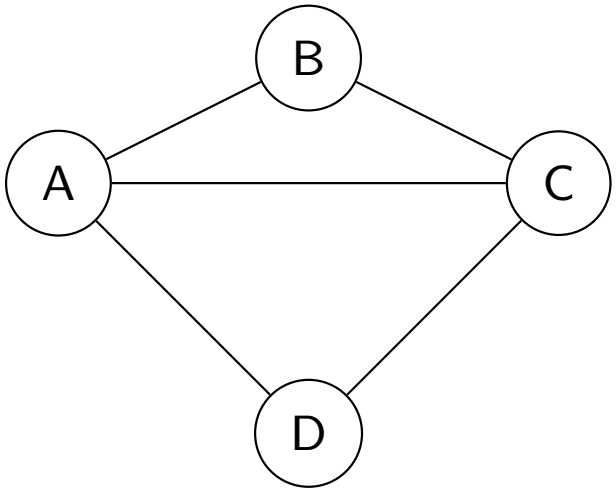
Introduction to Graph Connectivity

Definition: A graph is **connected** if there exists a path between every pair of vertices.

Types of Connectivity:

- **Vertex Connectivity** ($\kappa(G)$): Minimum number of vertices whose removal disconnects the graph.
- **Edge Connectivity** ($\lambda(G)$): Minimum number of edges whose removal disconnects the graph.

Example: A Connected Graph



Exercise:

- Find the vertex and edge connectivity of the graph.
- Construct a disconnected graph with at most 5 vertices.

Menger's Theorem

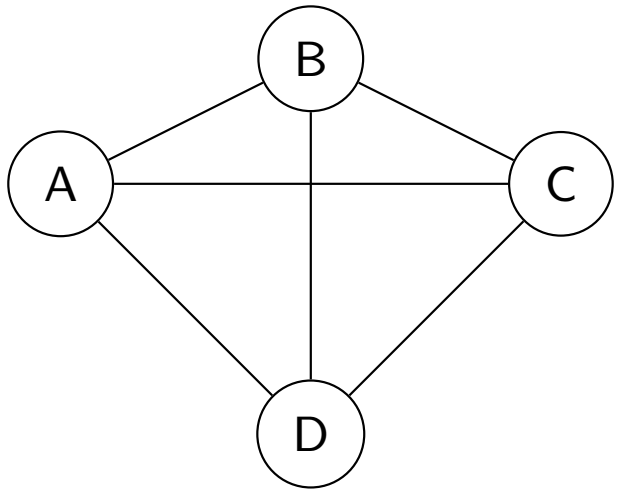
Theorem: The vertex connectivity $\kappa(G)$ is equal to the maximum number of internally disjoint paths between any two vertices.

Exercise:

- Prove Menger's theorem for small graphs.
- Identify disjoint paths in a given connected graph.

Menger's Theorem on Connectivity

Theorem: The vertex connectivity $\kappa(G)$ is equal to the maximum number of pairwise internally disjoint paths between any two vertices.
Edge Version: The edge connectivity $\lambda(G)$ is the maximum number of edge-disjoint paths between two vertices.
Example:



Exercise: Find the number of pairwise internally disjoint paths between A and C.

Whitney's Theorem on Connectivity

Theorem: If a graph is k-connected, then there exists a cycle passing through any k vertices.

Implications:

- Higher connectivity ensures larger cycles.
- Used in network resilience analysis.

Exercise: Construct a 3-connected graph and find a cycle that passes through three vertices.

Edge Connectivity Theorem

Theorem: A graph with edge connectivity $\lambda(G) \geq k$ remains connected after removing any $k-1$ edges.

Exercise:

- Show that $\lambda(K_n) = n - 1$.
- Find a graph where $\lambda(G) < \kappa(G)$.

Advanced Graph Connectivity – k-Connected Graphs

Definition: A graph is **k-connected** if it remains connected after removing fewer than k vertices.

Theorem: Every k-connected graph contains a cycle of length at least $k + 1$.

Exercise:

- Construct a 3-connected graph and verify its properties.

Expansion Properties and Connectivity

Definition: A graph is an **expander** if small vertex sets have many neighbors.

Cheeger's Inequality:

$$h(G) \leq 2\lambda_2(G),$$

where $h(G)$ is the edge expansion and $\lambda_2(G)$ is the second smallest eigenvalue of the Laplacian.

Exercise: Compute the Laplacian matrix of a small connected graph and find $\lambda_2(G)$.

Algorithms for Testing Connectivity

Common Algorithms:

- **Breadth-First Search (BFS):** Used for testing connected components.
- **Depth-First Search (DFS):** Efficiently finds articulation points.
- **Tarjan's Algorithm:** Finds strongly connected components in $O(V + E)$.
- **Stoer-Wagner Algorithm:** Computes global min-cut for edge connectivity.

Exercise: Implement BFS and DFS on a sample graph and determine its connectivity.

Summary of Key Theorems on Graph Connectivity

Key Theorems:

- **Menger's Theorem:** Relates vertex/edge connectivity to disjoint paths.
- **Whitney's Theorem:** High connectivity implies large cycles.
- **Cheeger's Inequality:** Expansion properties influence connectivity.

Key Algorithms:

- BFS and DFS for testing connectivity.
- Tarjan's algorithm for strongly connected components.
- Stoer-Wagner algorithm for edge connectivity.

Exercises:

- Prove Menger's theorem for small graphs.
- Compute the connectivity of a real-world network graph.

Outline

Basic Concepts of Directed Graphs

Definition: A directed graph (**digraph**) consists of vertices connected by directed edges.

Key Terms:

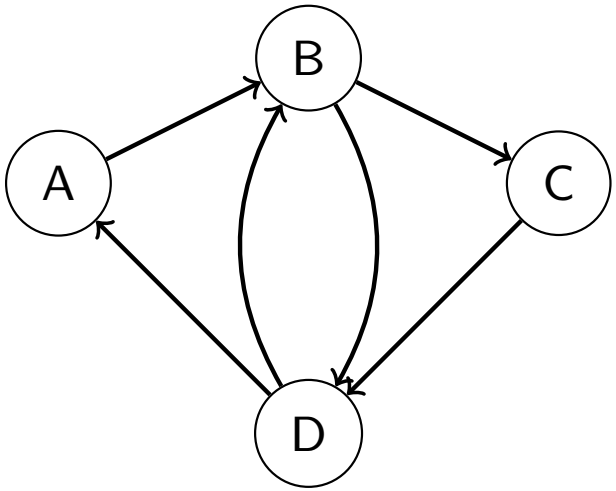
- **Out-degree**: Number of edges leaving a vertex.
- **In-degree**: Number of edges entering a vertex.
- **Underlying Graph**: The undirected version of a digraph.

Eulerian Directed Graphs

Definition: A directed graph is **Eulerian** if it contains an Eulerian circuit, a closed walk that visits every edge exactly once.

Theorem (Necessary and Sufficient Condition): A directed graph G has an Eulerian circuit if and only if:

- G is strongly connected.
- Every vertex has equal in-degree and out-degree.



Exercise:

- Check if the given directed graph is Eulerian.
- Construct a directed Eulerian graph with 5 vertices.

Algorithm for Finding an Eulerian Circuit

Fleury's Algorithm (Modified for Directed Graphs):

- Step 1: Start at any vertex with nonzero out-degree.
- Step 2: Follow edges one by one, removing them as they are used.
- Step 3: Avoid bridges unless necessary.
- Step 4: Continue until all edges are used.

Exercise:

- Apply Fleury's algorithm to a directed Eulerian graph.
- Modify Fleury's algorithm to work for semi-Eulerian digraphs.

Key Theorems and Results – Eulerian Directed Graphs

Fundamental Theorems:

- **Euler's Theorem (Directed Version):** A directed graph has an Eulerian circuit if and only if it is strongly connected and every vertex has equal in-degree and out-degree.
- **Necessary Condition for an Eulerian Path:** A directed graph has an Eulerian path (but not necessarily a circuit) if:
  - At most one vertex has out-degree $-$ in-degree $= 1$.
  - At most one vertex has in-degree $-$ out-degree $= 1$.
  - All other vertices have equal in-degree and out-degree.
- **Strong Connectivity and Eulerian Circuits:** If a directed graph contains an Eulerian circuit, it must be strongly connected.
- **Fleury's Algorithm (Adaptation for Digraphs):** An Eulerian circuit can be constructed by repeatedly traversing non-bridge edges.
- **De Bruijn Graphs:** Certain directed graphs (like De Bruijn graphs) always contain Eulerian circuits, useful in DNA sequencing.

Exercise:

- Prove why strong connectivity is necessary for an Eulerian circuit.
- Construct a directed graph with an Eulerian path but no Eulerian circuit.

Hamiltonian Directed Graphs

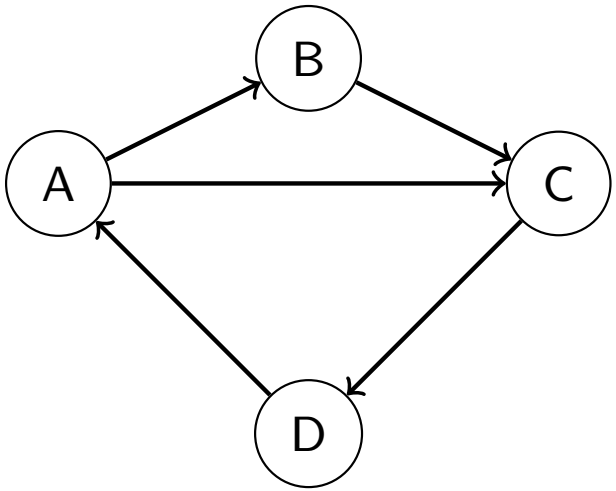
Definition: A directed graph is **Hamiltonian** if it contains a Hamiltonian cycle, a cycle that visits every vertex exactly once.

Dirac's Theorem for Digraphs: If a strongly connected directed graph with n vertices satisfies:

$$\text{For every vertex } v, \quad d^+(v), d^-(v) \geq \frac{n}{2},$$

then G has a Hamiltonian cycle.

Example: A Hamiltonian Digraph



Exercise:

- Verify Dirac's theorem on a directed graph with 6 vertices.
- Find a directed graph that is not Hamiltonian.
- Identify a Hamiltonian cycle in the given digraph.

Algorithm for Finding a Hamiltonian Path

Backtracking Algorithm:

- Step 1: Start at an arbitrary vertex.
- Step 2: Try extending the path by visiting an unvisited vertex.
- Step 3: If all vertices are visited exactly once, return the path.
- Step 4: If a dead-end is reached, backtrack.

Exercise:

- Implement the Hamiltonian path algorithm for a small directed graph.
- Show an example where the algorithm fails.

Key Theorems and Results – Hamiltonian Directed Graphs

Fundamental Theorems:

- **Dirac's Theorem (Directed Version):** A strongly connected directed graph with n vertices has a Hamiltonian cycle if for all v: $d^+(v), d^-(v) \geq \frac{n}{2}$.
- **Ghouila-Houri Theorem:** If G is a directed graph where: $d^+(v) + d^-(v) \geq n$ for every vertex v, then G has a Hamiltonian cycle.
- **Tournament Hamiltonicity:** Every tournament has a Hamiltonian path. A strongly connected tournament has a Hamiltonian cycle.
- **Chvátal's Theorem (Generalization for Digraphs):** A digraph satisfies a degree-based condition for Hamiltonicity if: $k \geq n/2 \Rightarrow d^+(v) + d^-(w) \geq n$ for any distinct vertices v, w.
- **NP-Completeness of Hamiltonian Cycle Problem:** Finding a Hamiltonian cycle in a directed graph is NP-complete.
- **Ore's Theorem for Directed Graphs:** If $d^+(u) + d^-(v) \geq n$ for all non-adjacent pairs, then the digraph has a Hamiltonian cycle.

Exercise:

- Apply Dirac's theorem to verify Hamiltonicity in a given digraph.
- Construct a tournament that has a Hamiltonian cycle.

Introduction to De Bruijn Graphs

Definition: A **De Bruijn graph** $B(k, n)$ is a directed graph whose vertices represent sequences of length n over an alphabet of size k, with edges denoting shifts of these sequences.
Example: $B(2, 3)$ represents all binary sequences of length 3.
Key Properties:

- Each vertex has exactly k incoming and k outgoing edges.
- It has an Eulerian circuit if and only if it is strongly connected.
- It has a Hamiltonian cycle (de Bruijn sequence) covering all length-n sequences.

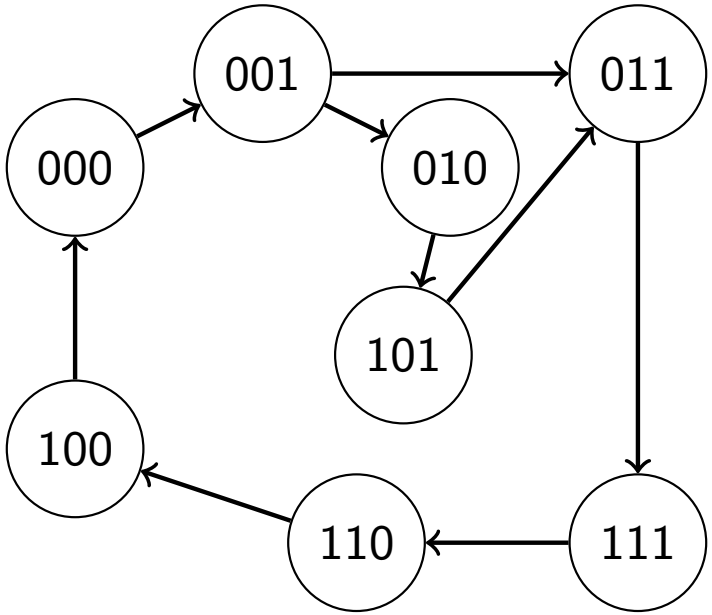
Exercise:

- Construct the De Bruijn graph $B(3, 2)$ for ternary strings of length 2.

Construction of De Bruijn Graphs (Example $B(2,3)$)

Example: The De Bruijn graph $B(2,3)$ consists of:

- Vertices: All binary strings of length 3 ($\{000, 001, 010, 011, 100, 101, 110, 111\}$).
- Directed edges: Connecting a sequence to the next by shifting left and appending a bit.



Exercise:

- Verify that every node has in-degree and out-degree 2.

Properties of De Bruijn Graphs

Key Properties:

- Each vertex represents a unique sequence of length n.
- Each vertex has exactly k incoming and k outgoing edges.
- It has an Eulerian circuit if and only if it is strongly connected.
- It has a Hamiltonian cycle (de Bruijn sequence) covering all length-n sequences.
- The total number of edges is k^n, and the total number of vertices is k^{n-1}.

Exercise:

- Prove that $B(k, n)$ has an Eulerian circuit.
- Show that $B(k, n)$ has a Hamiltonian path.

Eulerian and Hamiltonian Properties of De Bruijn Graphs

Eulerian Property: A De Bruijn graph is Eulerian since every vertex has equal in-degree and out-degree.
Hamiltonian Property: De Bruijn graphs contain a Hamiltonian cycle (de Bruijn sequence), which visits every possible sequence exactly once.
Theorem: The de Bruijn sequence of order n over an alphabet of size k corresponds to a Hamiltonian cycle in $B(k, n)$.
Exercise:

- Find a de Bruijn sequence for $B(2, 3)$.
- Prove that every Eulerian circuit in $B(k, n)$ can be transformed into a de Bruijn sequence.

Applications of De Bruijn Graphs

Key Applications:

- **DNA Sequencing:** Used to reconstruct genome sequences by aligning k-mers.
- **Networking:** De Bruijn graphs are used for efficient routing in peer-to-peer networks.
- **Cryptography:** De Bruijn sequences help in key generation and secure communications.

Example: DNA Assembly Using De Bruijn Graphs

- Given DNA reads of length k, construct the De Bruijn graph.
- Eulerian paths reconstruct the genome.

Exercise:

- Explain why Eulerian paths are preferred over Hamiltonian paths in DNA sequencing.

Key Takeaways of De Bruijn Graphs

- De Bruijn graphs represent sequences of length n over an alphabet of size k.
- Every De Bruijn graph is Eulerian and has a Hamiltonian cycle.
- The de Bruijn sequence corresponds to a Hamiltonian cycle in the graph.
- Applications in DNA sequencing, networking, and cryptography.

Exercises:

- Construct $B(2, 4)$ and find its Eulerian circuit.
- Find the shortest de Bruijn sequence for $k = 3, n = 2$.

Relationship Between Eulerian and Hamiltonian Directed Graphs

Key Differences:

- Eulerian graphs focus on edges (visiting every edge exactly once).
- Hamiltonian graphs focus on vertices (visiting every vertex exactly once).

Theorem: If a directed graph is both Eulerian and Hamiltonian, then it must satisfy:

- Strong connectivity.
- Equal in-degree and out-degree.
- Hamiltonian cycle existence conditions.

Exercise:

- Find a graph that is Eulerian but not Hamiltonian.
- Construct a digraph that is both Eulerian and Hamiltonian.

Eulerian and Hamiltonian Directed Graphs

Eulerian Directed Graphs:

- Contains an Eulerian circuit if it is strongly connected and has equal in-degree and out-degree.
- Fleury's algorithm helps in finding Eulerian circuits.

Hamiltonian Directed Graphs:

- Contains a Hamiltonian cycle if Dirac's theorem conditions hold.
- Finding Hamiltonian paths is NP-complete.

Key Differences:

- Eulerian graphs focus on edges, Hamiltonian graphs focus on vertices.
- Eulerian circuits are easy to find, while Hamiltonian cycles are difficult to determine.

Exercises:

- Find an example of a digraph that is Eulerian but not Hamiltonian.
- Construct a directed graph that satisfies Dirac's theorem.

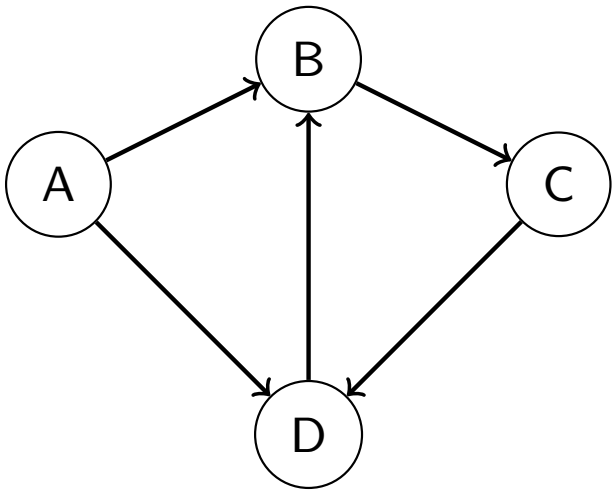
Tournaments in Directed Graphs

Definition: A **tournament** is a directed graph obtained by assigning a direction to each edge in a complete graph.

Key Properties:

- Every tournament has a directed Hamiltonian path.
- A tournament has a unique king (a vertex that can reach all others in at most two steps).

Example: A Tournament Graph



Exercise:

- Find a Hamiltonian path in the tournament.
- Identify the king in the tournament.

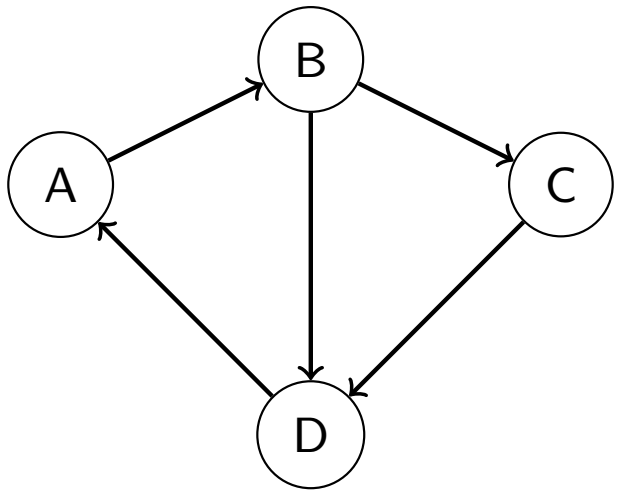
Definition of a King in a Tournament

Definition: In a tournament (a directed complete graph), a vertex v is called a **king** if it can reach every other vertex within at most two steps.

Key Observations:

- Every tournament has at least one king.
- A king may not be the overall winner (dominant vertex).

Example:



Exercise: Identify the kings in this tournament.

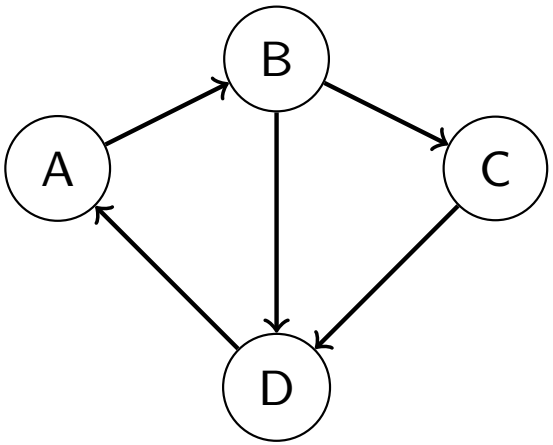
Theorem: Every Tournament Has at Least One King

Theorem: In any tournament, there exists at least one king.

Proof Idea:

- Consider the vertex with the highest out-degree.
- If this vertex is not a king, another vertex must dominate it and be a king.

Example: Tournament with Multiple Kings



Exercise:

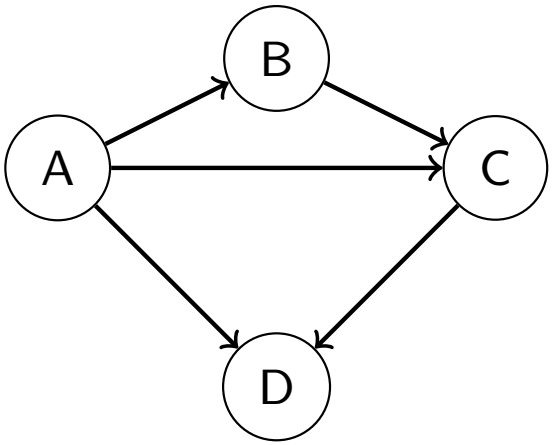
- Prove that this tournament has a king.
- Find an example of a tournament with exactly one king.

Dominance Order and Kings in Tournaments

Definition: The **dominance order** of a tournament arranges vertices such that if u dominates v, then u is ranked higher.

Key Theorem: If a vertex has the second-highest out-degree, it must be a king.

Example: A Tournament with a Clear King



Exercise:

- Identify the king in this tournament.
- Prove that the second-highest out-degree vertex is a king.

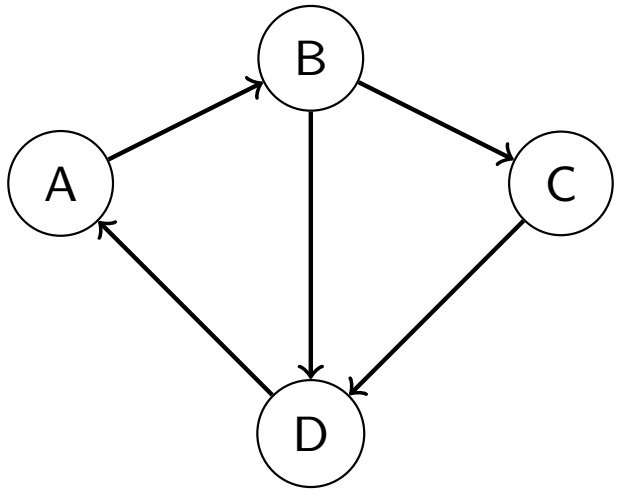
Hamiltonian Paths in Tournaments

Theorem: Every tournament has a Hamiltonian path.

Proof Sketch:

- Construct a sequence of vertices by repeatedly inserting each vertex in its correct position.
- Since every pair of vertices has a directed edge, a Hamiltonian path always exists.

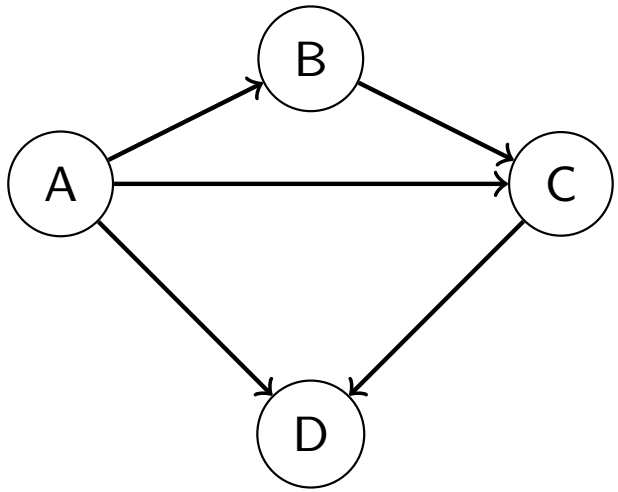
Example:



Exercise: Find a Hamiltonian path in this tournament.

Transitivity in Tournaments

Definition: A tournament is **transitive** if it has a vertex ordering such that if $u \to v$ and $v \to w$, then $u \to w$.
Theorem: A tournament is transitive if and only if it is acyclic.
Example:



Exercise: Show that this tournament is transitive.

Strong Connectivity in Tournaments

Theorem: Every tournament is strongly connected if and only if it contains a directed cycle.

Proof Sketch:

- Every tournament has a directed Hamiltonian path.
- If a tournament is not strongly connected, then it must be transitive.

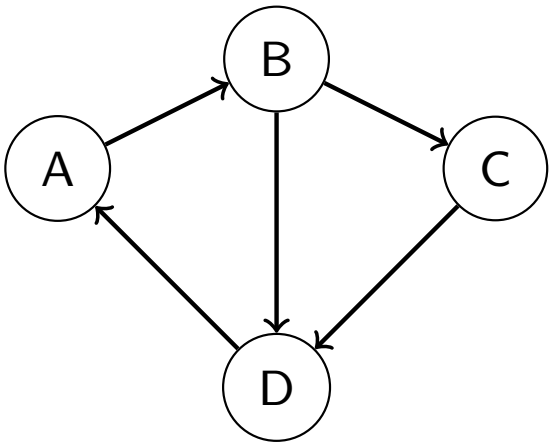
Exercise: Prove that a non-transitive tournament is always strongly connected.

Kings in Tournaments – Further Properties

Key Theorems:

- A king in a tournament may not be the overall champion.
- Every tournament has at least one vertex that is a king.
- The second-highest out-degree vertex is always a king.

Example: A Tournament with Multiple Kings



Exercise:

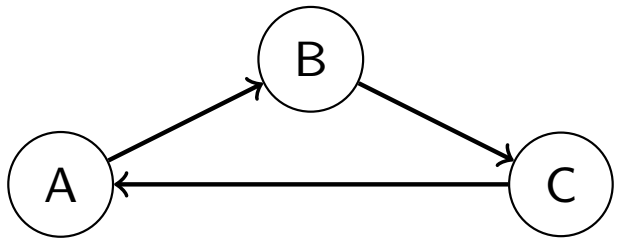
- Identify all kings in this tournament.
- Show that the second-highest out-degree vertex is always a king.

Cycles in Tournaments

Key Properties:

- Every tournament has at least one directed cycle.
- If a tournament is not transitive, then it has a directed triangle.

Example:



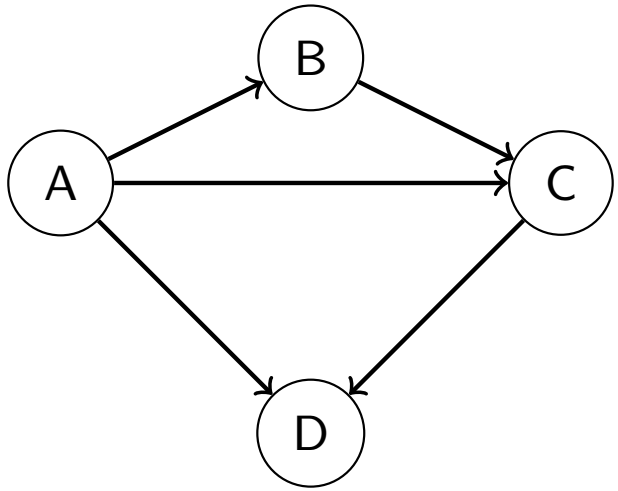
Exercise: Find a directed cycle in the given tournament.

Transitivity in Tournaments

Definition: A tournament is **transitive** if there exists a total ordering of the vertices such that for every directed edge (u, v) and (v, w), we also have (u, w).

Theorem: A tournament is transitive if and only if it contains no directed cycles.

Example:



Exercise: Prove that a transitive tournament has a unique Hamiltonian path.

Dominating Sets in Tournaments

Definition: A **dominating set** in a tournament is a set of vertices such that every vertex outside the set is dominated by at least one vertex inside the set.

Theorem: Every tournament has a dominating set of size at most $\lceil n/2 \rceil$.

Proof Idea:

- Partition the vertices into two groups: those with high out-degree and those with low out-degree.
- Show that one of these groups dominates the tournament.

Exercise: Find the smallest dominating set in a given tournament.

Fixed-Point Properties in Tournaments

Definition: A fixed point in a tournament is a vertex v such that for every subset of vertices containing v, the tournament restricted to that subset retains its structural properties.

Key Result: In every tournament, there exists at least one vertex with a unique minimal out-degree.

Implications:

- This result helps in ranking-based decision models.
- It guarantees that some player in a round-robin tournament is never the worst performer.

Exercise: Identify a fixed point in a given tournament.

Key Results on Tournaments

Key Theorems and Results:

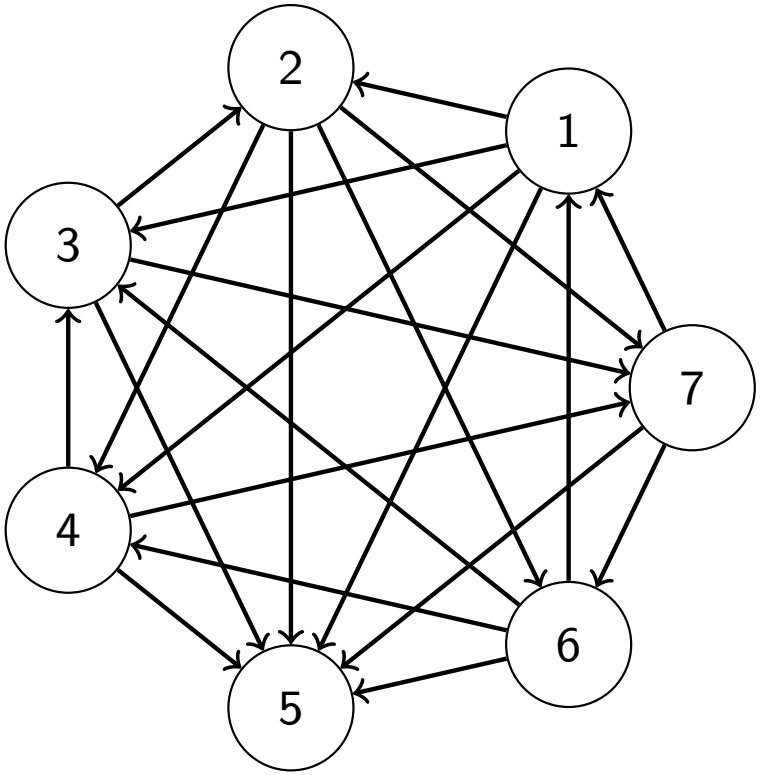
- Every tournament has a Hamiltonian path.
- Transitive tournaments are acyclic.
- A tournament is strongly connected if and only if it contains a cycle.
- Every tournament has at least one king.
- The second-highest out-degree vertex is always a king.
- Transitive tournaments correspond to strict total orders.
- Every transitive tournament has a unique Hamiltonian path.
- A tournament is transitive if and only if it has no directed cycles.
- Every tournament has a dominating set of size at most $\lceil n/2 \rceil$.
- Tournaments have fixed-point properties that influence ranking decisions.

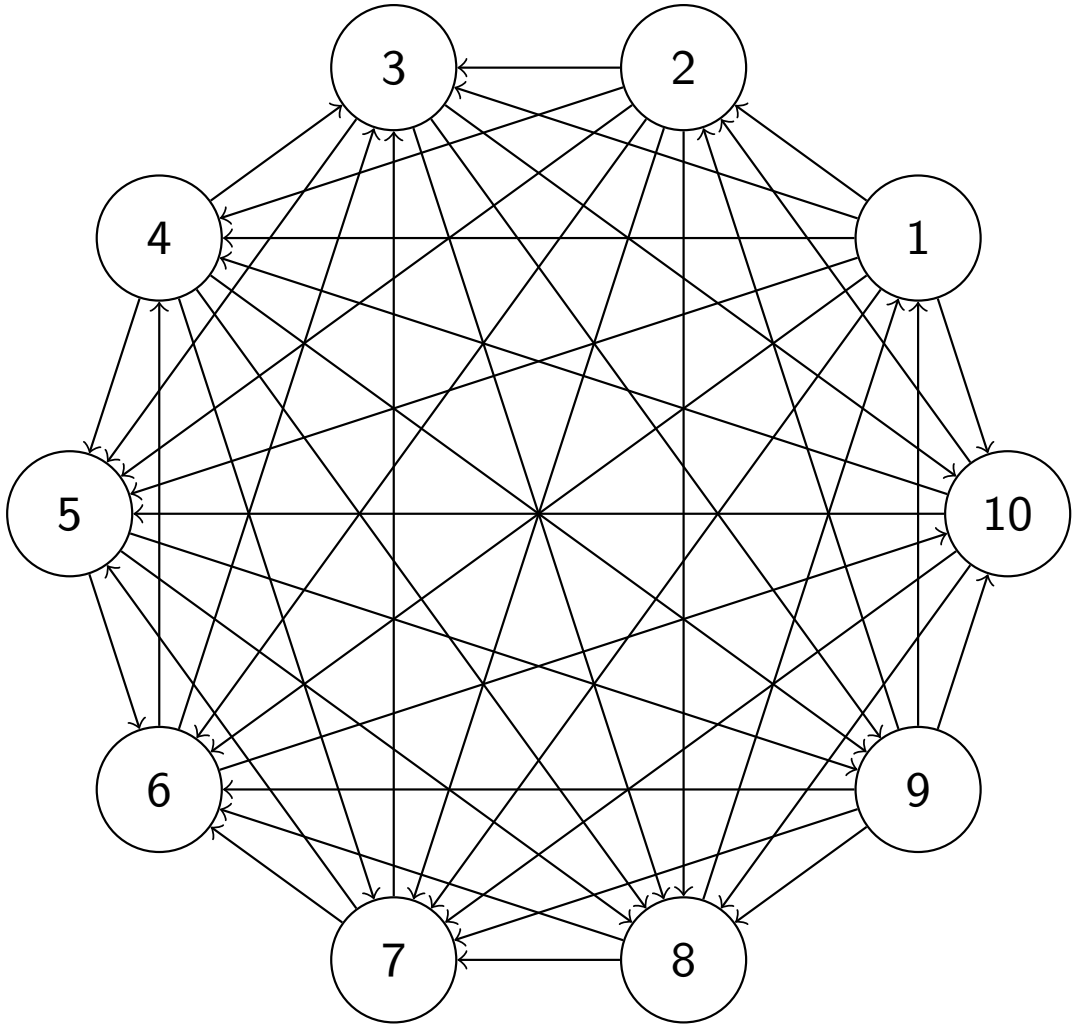
Exercises:

- Construct a tournament with exactly one king.
- Show that every tournament has a Hamiltonian path.
- Identify a tournament that is both transitive and has a king.
- Prove that a tournament with an even number of vertices has a dominating set of size $n/2$.
- Find an example of a tournament that is not transitive but has a unique Hamiltonian path.

Larger Tournament Graphs

Test out the Theorems





Comprehensive Summary I

Planar Graphs:

- Definition and examples.
- Euler's Formula: $V - E + F = 2$.
- Kuratowski's Theorem: Non-planar graphs contain K_5 or $K_{3,3}$.
- 5-Color Theorem: Every planar graph is 5-colorable.

Directed Graphs:

- Definitions of in-degree, out-degree, and underlying graph.
- Eulerian Directed Graphs: Strongly connected and equal in-degree/out-degree.
- Hamiltonian Directed Graphs: Dirac's Theorem and NP-completeness.
- Tournaments: Every tournament has a Hamiltonian path.

Outline

4. Appendix
   - Background
   - More on Planarity
   - 4-color Theorem
   - More on Connectivity

Outline

Graph Homomorphism – Definition & Examples

Definition: A **homomorphism** from a graph $G = (V_G, E_G)$ to a graph $H = (V_H, E_H)$ is a function:

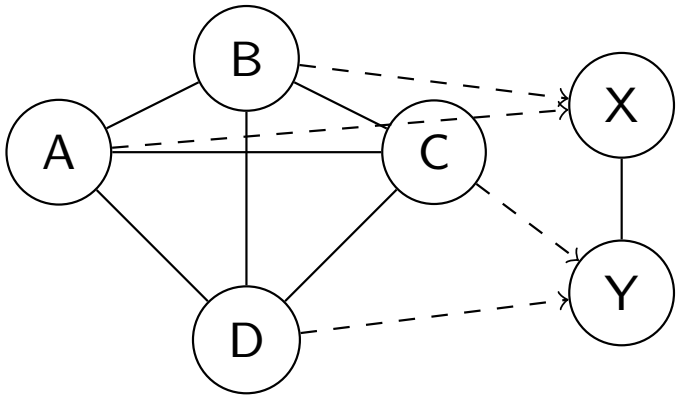
$$f : V_G \to V_H$$

such that if $(u, v) \in E_G$, then $(f(u), f(v)) \in E_H$.

Key Properties:

- **Preserves adjacency** but does not necessarily preserve non-adjacency.
- Used in **graph colorings**, constraint satisfaction problems, and algebraic graph theory.

Example: A homomorphism from K_4 to K_2.



Exercise:

- Find a homomorphism from C_6 (cycle graph) K_3.

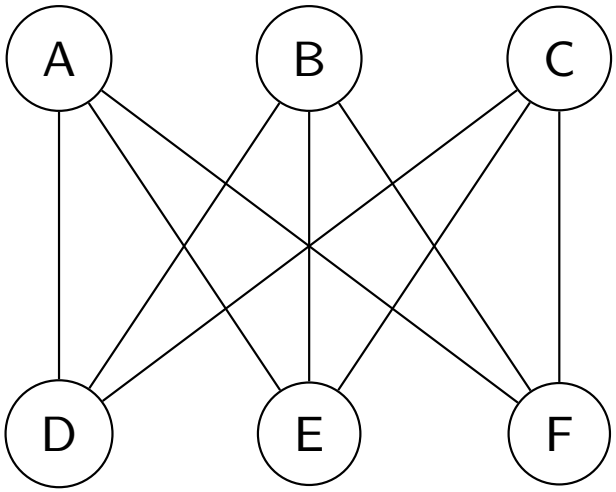
Graph Homeomorphism – Definition & Examples

Definition: Two graphs G and H are **homeomorphic** if one can be obtained from the other by **subdividing edges** (inserting degree-2 vertices).

Key Properties:

- **Topological equivalence** of graphs.
- Used in **planarity testing** and **graph embeddings**.

Example: $K_{3,3}$ and a homeomorphic version with subdivision.



Exercise:

- Subdivide edges in K_5 to create a homeomorphic graph.

Homeomorphism vs. Homomorphism in Graphs

Key Differences:

- Homeomorphism relates to **subdivision of edges**, while homomorphism relates to **mapping vertices**.
- Homeomorphism preserves **topological structure**, while homomorphism preserves **adjacency**.

Exercise:

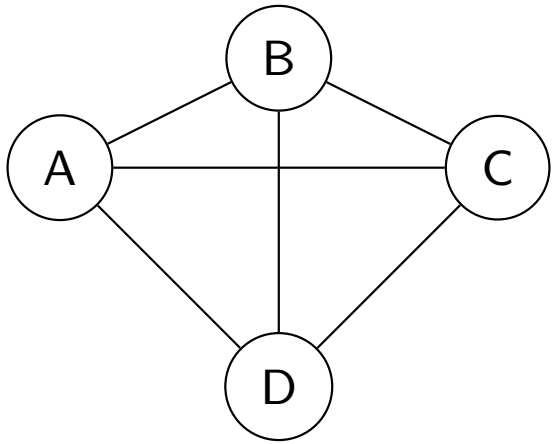
- Find a homomorphism from K_4 to K_2.
- Show that two graphs with a subdivision of the same edge are homeomorphic.

Graph Minors and Edge Contractions

Definition:

- A graph H is a **minor** of G if it can be obtained from G by deleting edges, deleting vertices, and contracting edges.
- **Edge Contraction:** Replacing an edge (u, v) with a single vertex merging u and v.

Example: Contraction of K_5 leads to a graph minor.



Exercise: Find a minor of K_5 by contracting edges.

Graph Minors and Edge Contractions

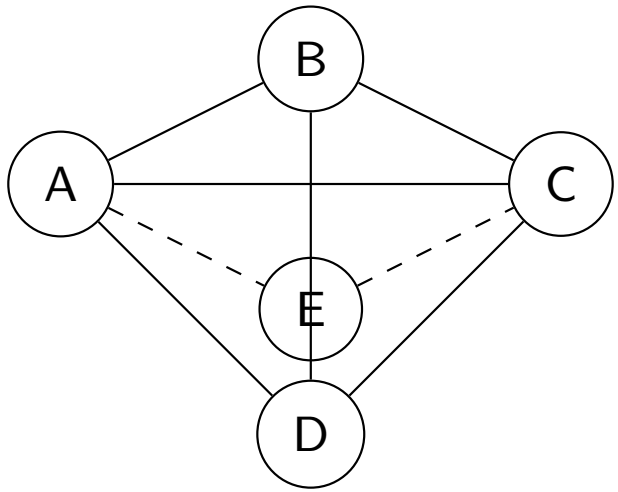
Definition: A graph H is a **minor** of G if it can be obtained from G by:

- **Deleting vertices**,
- **Deleting edges**, or
- **Contracting edges** (merging endpoints into one).

Key Properties:

- Graph minors are used in **Wagner's theorem** for planarity testing.
- The **Graph Minor Theorem (Robertson-Seymour)** states that any infinite sequence of graphs contains a minor of another.

Example: K_5 contracting to a smaller minor.



Exercise:

Applications of Graph Minors and Contractions

Why Are Graph Minors Important?

- Used in **Kuratowski's Theorem** to determine planarity.
- **Graph Minor Theorem (Robertson-Seymour Theorem)** states that for any infinite sequence of graphs, one is a minor of another.
- Edge contractions help in **network simplifications**.

Exercise: Apply edge contractions to simplify a given network graph.

Summary of Graph Transformations

Key Concepts:

- **Homomorphism:** Maps vertices while preserving adjacency.
- **Homeomorphism:** Graphs equivalent under edge subdivisions.
- **Minors:** Obtained by vertex/edge deletions and contractions.
- **Contractions:** Merging adjacent vertices into one.

Key Applications:

- Planarity testing (Kuratowski's & Wagner's theorems).
- Graph transformations in network reductions.
- Algebraic graph theory and constraint problems.

Exercises:

- Show that every homeomorphic transformation is a minor but not vice versa.
- Construct an example where homomorphism exists but homeomorphism does not.

Outline

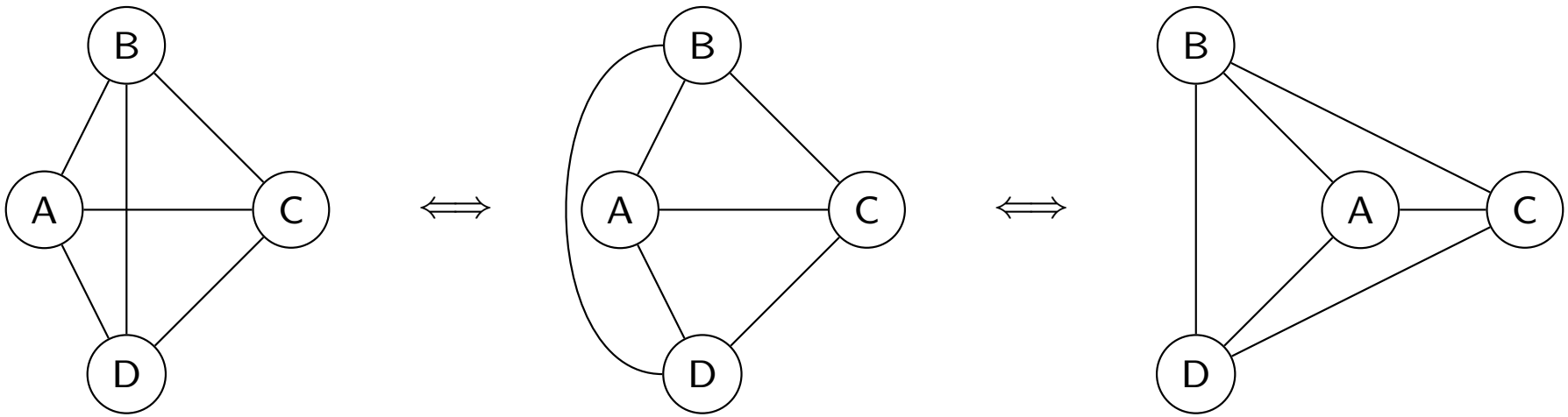
Fáry's Theorem – Planar Graphs with Straight-Line Edges

Theorem: Every planar graph can be drawn in the plane with straight-line edges.

Proof Idea:

- Uses induction on the number of vertices.
- Applies triangulation and barycentric placement.

Example:



Exercise:

- Draw a planar graph using only straight-line edges.

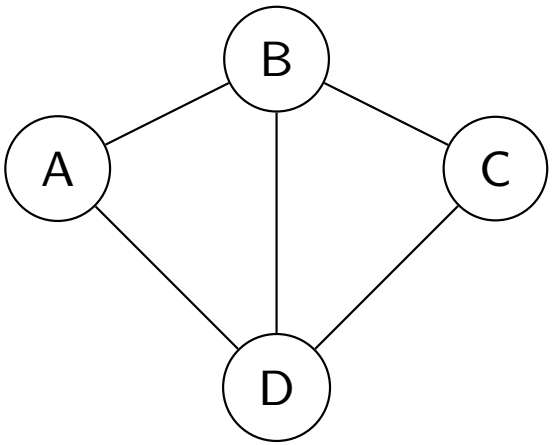
Whitney's Duality Theorem

Theorem: Every connected planar graph has a unique dual graph (up to isomorphism).

Key Concepts:

- The dual graph G^* is formed by placing a vertex in each face of G.
- Edges of G^* correspond to edges crossing between faces in G.

Example: Dual of a Planar Graph



Exercise:

- Find the dual of a given planar graph.
- Prove that the dual of a tree is always a cycle.

Crossing Number and Beyond Planarity

Definition: The crossing number $\mathrm{cr}(G)$ of a graph G is the minimum number of edge crossings required in any drawing of G.
Key Theorem (Turán's Brick Factory Problem):

$$\mathrm{cr}(K_n) \geq \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor$$

Exercise:

- Find the crossing number of K_5 and $K_{3,3}$.
- Show that every planar graph has $\mathrm{cr}(G) = 0$.

Outline

Introduction to the 4-Color Theorem

Statement: Every planar graph can be properly colored using at most 4 colors.

Why is this Important?

- Solves the problem of coloring maps on a plane.
- First major theorem proved using a computer.

Exercise: Show that the Petersen graph is non-planar, so the 4-Color Theorem does not apply.

Historical Background of the 4-Color Theorem

Key Milestones:

- 1852 – First observed by Francis Guthrie while coloring maps.
- 1879 – Alfred Kempe published an incorrect proof.
- 1890 – Percy Heawood corrected Kempe's mistake but proved the 5-Color Theorem.
- 1976 – Kenneth Appel and Wolfgang Haken proved the theorem using a computer.

Exercise: Research and summarize why Appel and Haken's proof was controversial.

Proof Idea – Appel & Haken's Approach

Key Strategy:

- Reduce the problem to a finite set of "unavoidable configurations."
- Use a computer to check all cases.

Challenges:

- Proof relied on 1,200 hours of computer calculations.
- Initially criticized due to lack of human-verifiable steps.

Exercise: Explain why reducible configurations are important in proving the 4-Color Theorem.

Implications & Applications

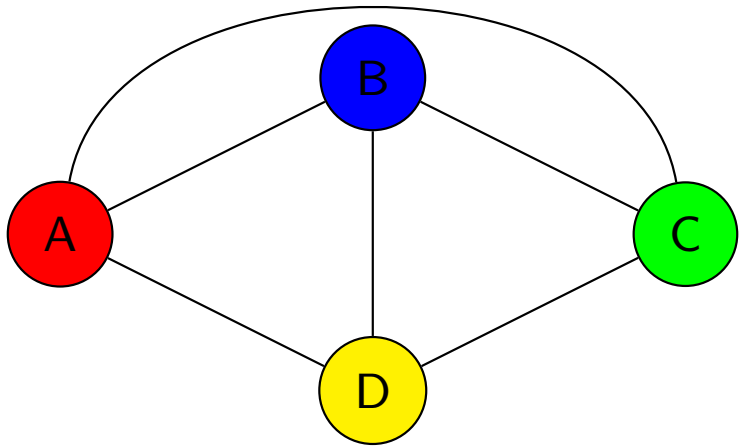
Why Does This Matter?

- **Graph Theory:** Strengthens understanding of planar graphs.
- **Map Coloring:** Helps in real-world geographical problems.
- **Computer Science:** Led to the development of automated theorem proving.

Exercise: Find a real-world problem where 4-coloring is applicable.

Example: A 4-Colorable Planar Graph

Graph Representation:



Exercise: Try coloring the graph with only 3 colors. Is it possible?

Outline

k-Connected Graphs and Dirac's Theorem

Definition: A graph is k-connected if it remains connected after removing any $k-1$ vertices.

Dirac's Theorem: A k-connected graph with n vertices has a cycle of length at least $k+1$.

Exercise:

- Show that K_n is $(n-1)$-connected.
- Construct a 3-connected graph and find its largest cycle.

Global and Local Connectivity Relations

Menger's Theorem: The vertex connectivity $\kappa(G)$ is equal to the maximum number of internally disjoint paths between any two vertices.

Exercise:

- Find the connectivity of a given graph.
- Prove Menger's theorem for a simple case.

Expander Graphs and Spectral Graph Theory

Definition: An expander graph is a sparse graph that has strong connectivity properties.

Cheeger's Inequality:

$$h(G) \leq 2\lambda_2(G)$$

where $h(G)$ is the edge expansion and $\lambda_2(G)$ is the second smallest eigenvalue of the Laplacian matrix.

Exercise:

- Compute the Laplacian matrix of a given graph.
- Show how eigenvalues relate to connectivity.

Edge Disjoint Spanning Trees

Theorem: A graph G contains k edge-disjoint spanning trees if and only if it remains connected after removing any $k-1$ edges.

Exercise:

- Find two edge-disjoint spanning trees in a given graph.
- Prove that a 3-connected graph has at least two edge-disjoint spanning trees.

Summary

Planar Graphs:

- Fáry's Theorem – Planar graphs with straight-line edges.
- Whitney's Duality – Relationship between graphs and their duals.
- Crossing Number – Generalization of planarity.

Graph Connectivity:

- k-Connected Graphs – Dirac's Theorem.
- Menger's Theorem – Relationship between paths and connectivity.
- Expander Graphs – Spectral properties and applications.
- Edge Disjoint Spanning Trees – Structural connectivity properties.