Discrete Mathematics 2025 Spring

同济经管
TONGJI SEM

魏可佶 kejiwei@tongji.edu.cn

Chapter 6: Graphs

- 6.3.1 Incidence Matrix of an Undirected Graph
- **6.3.2 Incidence Matrix of a Directed Acyclic Graph**
- **6.3.3 Adjacency Matrix of a Directed Graph, Adjacency Matrix of an Undirected Graph**

  The Number of Paths and Cycles in a Graph
- **6.3.4 Reachability Matrix of a Graph**

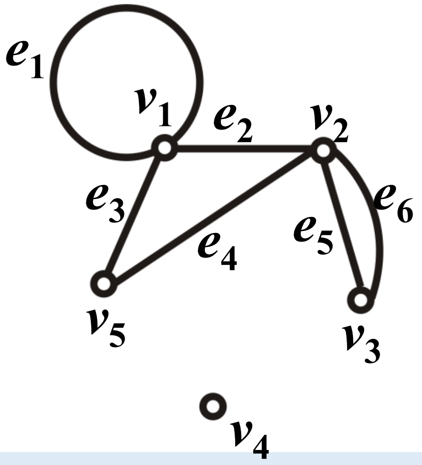
6.3.1 Incidence Matrix of an Undirected Graph

↳Definition of the Incidence Matrix

- Let G=<V,E>, V={v_1, v_2, …, v_n}, E={e_1, e_2, …, e_m}.
  - Let m_{ij} be the incidence of vertex v_i with edge e_j, the matrix $(m_{ij})_{n\times m}$ is called the ***incidence matrix*** of G, denoted as $M(G)$. The possible values of m_{ij}: 0,1,2

Example:

$$M(G)=\begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$



↳ Properties of the Incidence Matrix

(1) $$\sum_{i=1}^{n} m_{ij} = 2, \quad j = 1, 2, \ldots, m$$

(2) $$\sum_{j=1}^{m} m_{ij} = d(v_i), \quad i = 1, 2, \ldots, n$$

(3) $$\sum_{i,j} m_{ij} = 2m$$

(4) e_j and e_k are ***parallel edge***
⇔ the j-th column and the k-th column are identical.

(5) V_i is an ***isolated vertex*** ⇔ i-th row is all zeros.

(6) E_j is ***loop*** ⇔ The first element in column j is **2**, others are all **0.**

- 6.3.1 Incidence Matrix of an Undirected Graph
- 6.3.2 Incidence Matrix of a Directed Acyclic Graph
- **6.3.3 Adjacency Matrix of a Directed Graph, Adjacency Matrix of an Undirected Graph**
  The Number of Paths and Cycles in a Graph
- **6.3.4 Reachability Matrix of a Graph**

↳ The Incidence Matrix of a DAG

- Let the *directed acyclic graph* $D=<V,E>$, $V=\{v_1, v_2, \dots, v_n\}$, $E=\{e_1, e_2, \dots, e_m\}$.

$$\text{Let } m_{ij} = \begin{cases} 1, & v_i \text{ is the starting point of } e_j \\ 0, & v_i \text{ is not incident to } e_j \\ -1, & v_i \text{ is the endpoint of } e_j \end{cases}$$

The matrix $(m_{ij})_{n\times m}$ called the incidence matrix of ***D***, called $M(D)$.

- **Properties of a DAG:**

(1) **Each column** contains exactly one 1 and one −1.

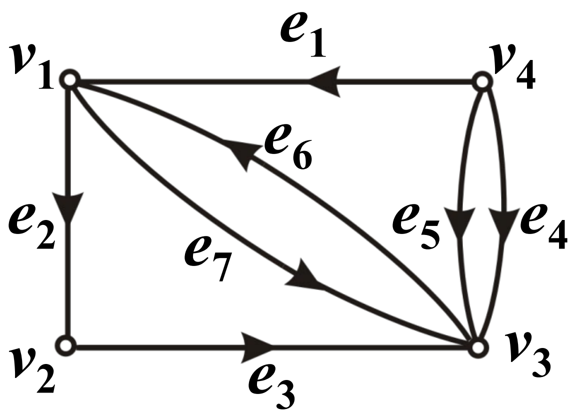
(2) The total number of **1's** is equal to the total number of **−1's**, which equals the number of edges.

(3) The number of **1's** in the i-th row equals $d^+(v_i)$, and the number of −1's in the i-th row equals $d^-(v_i)$.

(4) E_j *and* e_k are parallel edges ⇔ the **j-th** column and the **k-th** column are identical.

6.3.2 Incidence Matrix of a Directed Acyclic Graph

↳ The Incidence Matrix of a DAG(e.g.)



$$M(D)=\begin{bmatrix} -1 & 1 & 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

- 6.3.1 Incidence Matrix of an Undirected Graph
- 6.3.2 Incidence Matrix of a Directed Acyclic Graph
- 6.3.3 Adjacency Matrix of a Directed and Undirected Graph

  The Number of Paths and Cycles in a Graph
- **6.3.4 Reachability Matrix of a Graph**

6.3.3 Adjacency Matrix of a Directed and Undirected Graph

↳ Adjacency Matrix of a Directed Graph

- Let ***D***=⟨***V***,***E***⟩, where $V=\{v_1, v_2, \ldots, v_n\}$ and $E=\{e_1, e_2, \ldots, e_m\}$,
  Let the number of edges from vertex $\boldsymbol{v_i}$ to vertex $\boldsymbol{v_j}$ be denoted as $a_{ij}^{(1)}$. The matrix $(a_{ij}^{(1)})_{m\times n}$ is called the ***adjacency matrix*** of ***D***, denoted as ***A*(*D*)**, or simply ***A***.

- **Properties of *A*(*D*):**

(1) $$\sum_{j=1}^{n} a_{ij}^{(1)} = d^{+}(v_i), \quad i = 1, 2, \ldots, n$$

The sum of the rows equals the outdegree of the graph.

(2) $$\sum_{i=1}^{n} a_{ij}^{(1)} = d^{-}(v_j), \quad j = 1, 2, \ldots, n$$

The sum of the columns equals the indegree of the graph.

6.3.3 Adjacency Matrix of a Directed and Undirected Graph

↳ Adjacency Matrix of a Directed Graph

- **Properties of A(D):**

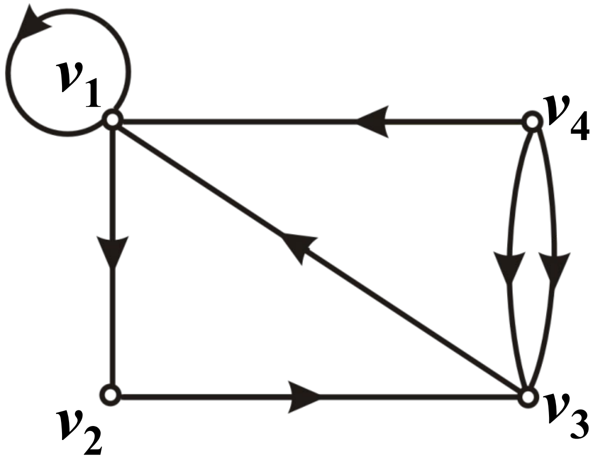
(3) $$\sum_{i,j} a_{ij}^{(1)} = m$$

The sum of all the elements equals the number of edges.

(4) $$\sum_{i=1}^{n} a_{ii}^{(1)} = \textit{the number of loops at } D$$

The sum of the diagonal elements equals the number of vertex loops.

- **Example:**



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

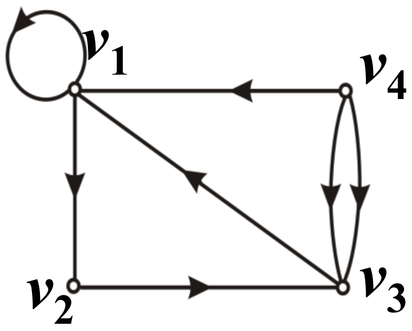
↳ Counting Directed Paths via Adjacency Matrices

- **Theorem 6.4:** Let ***A*** be the adjacency matrix of the n-order directed graph ***D***. Then, the elements of $A^l (l \geq 1)$:
  - $a_{ij}^{(l)}$ are the number of *paths of length l* from vertex v_i to vertex v_j in ***D***.
  - $a_{ii}^{(l)}$ is the number of *cycles of length l* starting and ending at vertex v_i .
  - $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(l)}$ is the *total number of paths of length l* (including cycles) in ***D***.
  - $\sum_{i=1}^{n} a_{ii}^{(l)}$ is the *total number of cycles of length l* in ***D***.

- **Corollary:** Let $B_l=A+A^2+\ldots+A^l(l\geq 1)$, Then, the elements :
  - $b_{ij}^{(l)}$ are the **number of paths** (including cycles) of length *less than or equal to l* from vertex v_i to vertex v_j in ***D***.
  - $b_{ii}^{(l)}$ is the **number of cycles** in *D* whose length from v_i to v_i *less than or equal to l*.
  - $\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^{(l)}$ is the **number of paths** (including cycles) in ***D*** whose length is *less than or equal to l*.
  - $\sum_{i=1}^{n} b_{ii}^{(l)}$ is the **number of cycles** in ***D*** whose length is *less than or equal to l*.

6.3.3 Adjacency Matrices of Directed and Undirected Graphs

↳ Counting Directed Paths via Adjacency Matrices (e.g.)



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

$$A^2=\begin{bmatrix}1&1&1&0\\1&0&0&0\\1&1&0&0\\3&1&0&0\end{bmatrix}$$

$$A^3=\begin{bmatrix}2&1&1&0\\1&1&0&0\\1&1&0&0\\3&3&1&0\end{bmatrix}$$

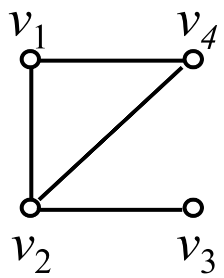
$$A^4=\begin{bmatrix}3&2&1&0\\1&1&0&0\\2&1&1&0\\4&3&1&0\end{bmatrix}$$

- There is 1 path of length 3 from v_1 to v_2.
- There is 1 path of length 3 from v_1 to v_3.
- There are 2 cycles of length 3 from v_1 to itself.
- There are a total of 15 paths of length 3 in D, of which 3 are cycles.

6.3.3 Adjacency Matrix of a Directed and Undirected Graph

↳ Adjacency Matrix of a Undirected Graph

- Let $G=\langle V,E\rangle$ be an undirected simple graph, where $V=\{v_1,v_2,\ldots,v_n\}$. Let $a_{ij}^{(1)}$ denote the number of edges between vertices v_i and v_j. The matrix $(a_{ij})_{n\times n}$ is called the *adjacency matrix of G*, denoted as $A(G)$.
- **Example**: Write the adjacency matrix of an undirected graph, and find the number of paths of length **3** from v_i to v_2 and the number of cycles of length 3 from v_1 to v_1.



$$A=\begin{bmatrix}0&1&0&1\\1&0&1&1\\0&1&0&0\\1&1&0&0\end{bmatrix}\qquad A^2=\begin{bmatrix}2&1&1&1\\1&3&0&1\\1&0&1&1\\1&1&1&2\end{bmatrix}\qquad A^3=\begin{bmatrix}2&3&1&3\\3&2&3&4\\1&3&0&1\\3&4&1&2\end{bmatrix}$$

There are **3 paths** of length 3 from $v_1$to v_2: $v_1v_2v_1v_2$, $v_1v_2v_3v_2$, $v_1v_4v_1v_2$.

There are **2 cycles** of length 3 from v_1 to v_1: $v_1v_2v_4v_1$, $v_1v_4v_2v_1$.

- 6.3.1 Incidence Matrix of an Undirected Graph
- 6.3.2 Incidence Matrix of a Directed Acyclic Graph
- 6.3.3 Adjacency Matrix of a Directed and Undirected Graph
  The Number of Paths and Cycles in a Graph
- 6.3.4 Reachability Matrix of a Graph

↳ Reachability Matrix of Graph

- Let the graph (either undirected or directed) G=<V,E>, V={v_1, v_2, …, v_n},

  let $$p_{ij} = \begin{cases} 1\ , & v_i \textbf{ can reach } v_j \\ 0\ , & \textbf{otherwise} \end{cases}$$

  The matrix $(p_{ij})_{n\times n}$ is called the ***reachability matrix*** of ***G***, denoted as $P(G)$, or simply ***P***.

- **Property:**

  (1) All the elements on the *main diagonal* of ***P***(***G***) are **1**.

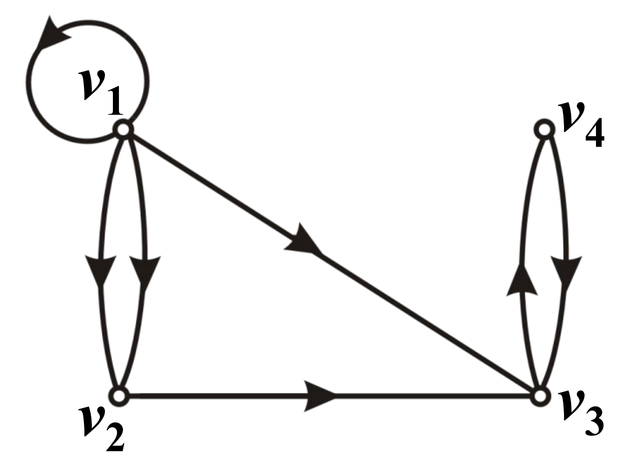
  (2) The reachability matrix of an undirected graph is *symmetric*.

  (3) An *undirected graph* ***G*** is **connected** if and only if all elements of ***P***(***G***) are 1. A *directed graph* ***D*** is **strongly connected** if and only if all elements of ***P***(***D***) are **1**.

  (4) For an *n-order graph*, p_{ij}=1 ⇔ $b_{ij}^{(n-1)} > 0$, $i \neq j$.

↳ Reachability matrix of Graph(e.g.)

- **Example:**



(1) How many paths of length 3 are there from v_1 to v_4, and from v_4 to v_1 ?

(2) How many cycles of length 1, 2, 3, and 4 are there from v_1 to itself?

(3) How many paths of length 4 are there in total ? How many of them are cycles ?

(4) How many cycles of length less than or equal to 4 are there in total ?

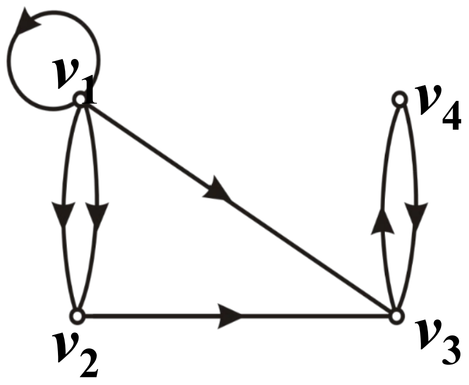
(5) Write the reachability matrix of *D*, and is *D* strongly connected?

- **Solution ?**

6.3.4 Reachability Matrix of a Graph

↳ Reachability matrix of Graph(e.g.)

- **Solution:**



$$A=\begin{bmatrix}1&2&1&0\\0&0&1&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\quad A^2=\begin{bmatrix}1&2&3&1\\0&0&0&1\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$A^3=\begin{bmatrix}1&2&4&3\\0&0&1&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\quad A^4=\begin{bmatrix}1&2&6&4\\0&0&0&1\\0&0&1&0\\0&0&0&1\end{bmatrix}\quad P=\begin{bmatrix}1&1&1&1\\0&1&1&1\\0&0&1&1\\0&0&1&1\end{bmatrix}$$

(1) There are **3** paths of length **3** from v_1 to v_4. There are 0 paths of length 3 from v_4 to v_1.

(2) There are 1 cycles of length 1, 2, 3, and 4 from v_1 to itself.

(3) There are 16 paths of length 4, of which 3 are cycles.

(4) There are 8 cycles of length less than or equal to 4.

(5) Reachability Matrix: *P*(*G*)

6.3 Matrix Representations of Graphs• Brief summary

Objective :

Key Concepts ：

Discrete Mathematics 2025 Spring

同济经管
TONGJI SEM

魏可佶　kejiwei@tongji.edu.cn

CAMEA 中国高质量MBA教育认证
AACSB ACCREDITED
EQUIS ACCREDITED

Chapter 6: Graphs

- 6.1 Basic Concepts of Graphs
- 6.2 Connectivity of Graphs
- 6.3 Matrix Representations of Graphs
- 6.4 Several Special Types of Graphs

6.4 Several Special Types of Graphs

- **6.4.1 Bipartite Graphs**

  Necessary and sufficient conditions for a graph to be bipartite matching, maximal matching, maximum matching, complete matching, perfect matching
- **6.4.2 Eulerian Graphs**

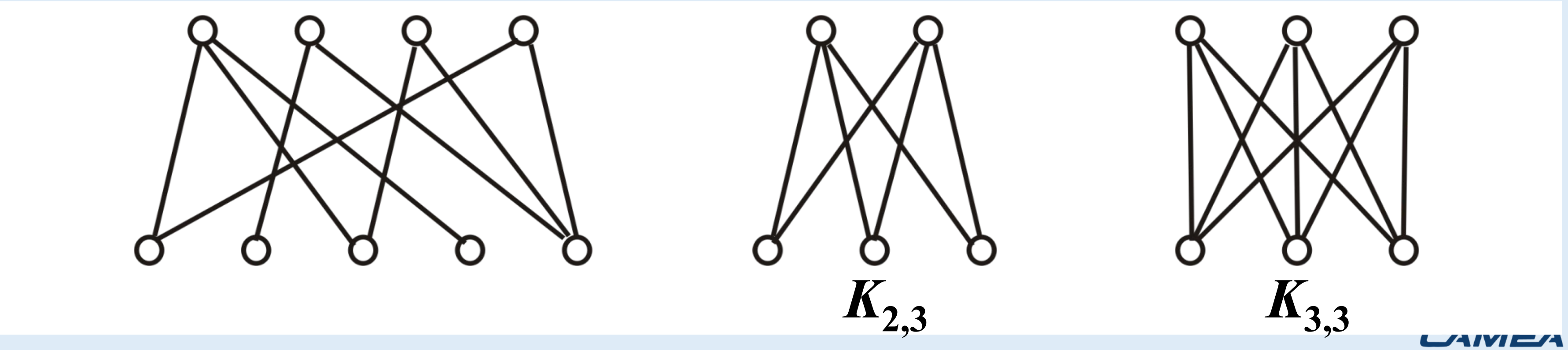
  Eulerian circuits (paths) and their necessary and sufficient conditions for existence
- **6.4.3 Hamiltonian Graphs**

  Hamiltonian circuits (paths) and the necessary and sufficient conditions for their existence
- **6.4.4 Planar Graphs**

6.4.1 Bipartite Graphs

↳ Bipartite Graph and Complete Bipartite Graph

- Let G=⟨V,E⟩,G=⟨V,E⟩ be an undirected graph. If it is possible to partition V into two sets V_1 and V_2 such that: $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \varnothing$, Each edge in G has one endpoint in V_1 and the other in V_2 ,then G is called a ***bipartite graph***, denoted as $<V_1, V_2, E>$, and V_1 and V_2 are called ***complementary vertex subsets***.
  - Furthermore, if G is a simple graph and every vertex in V_1 is adjacent to every vertex in V_2 , then G is called a ***complete bipartite graph***, denoted as $K_{r,s}$, where $r=|V_1|$ and $s=|V_2|$.



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

↳ Cycle Characterization of Bipartite Graphs

- **Theorem 6.5:** An undirected graph ***G***=<***V***,***E***> is a bipartite graph if and only if it contains no odd-length cycles.

- **Proof:**

(1) **Necessity:** Let $G=<V_1,V_2,E>$ be a bipartite graph. Each edge can only connect V_1 to V_2 or V_2 to V_1, so any cycle in ***G*** must have even length.

(2) **Sufficiency:** Assume ***G*** has at least one edge and is connected. Take any vertex ***u***, and define:

$V_1=\{v \mid v\in V$ and the distance from ***v*** to ***u*** is even}

$V_2=\{v \mid v\in V$ and the distance from v to ***u*** is odd}

Then, $V_1\cup V_2=V$, $V_1\cap V_2=\varnothing$.

↳ Cycle Characterization of Bipartite Graphs

Proof:

(2) **Sufficiency:** Assume G has at least one edge and is connected. Take any vertex u, and define:

$V_1=\{v \mid v \in V$ and the distance from v to u is even$\}$

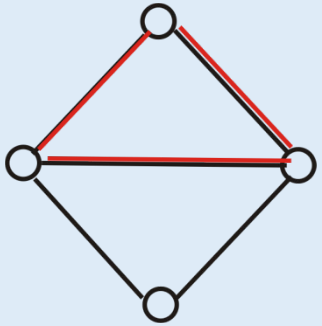
$V_2=\{v \mid v \in V$ and the distance from v to u is odd$\}$

Then, $V_1 \cup V_2=V$, $V_1 \cap V_2=\varnothing$.

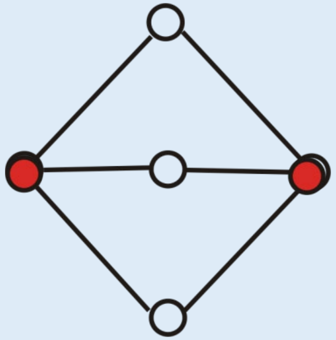
- First, we prove that no two vertices in V_1 are adjacent. Suppose there exist $s,t \in V_1$ such that $e=(s,t) \in E$. Let Γ_1 and Γ_2 be the shortest paths from u to s and u to t, respectively. Then, $\Gamma_1 \cup e \cup \Gamma_2$ forms a cycle of odd length, which contradicts the assumption.
- Similarly, we can prove that no two vertices in V_2 are adjacent.

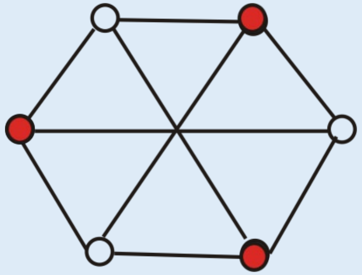
6.4.1 Bipartite Graphs

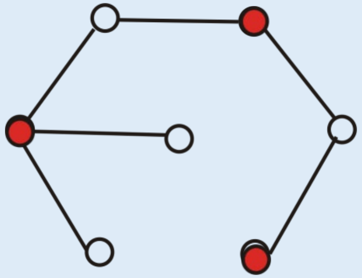
↳ Cycle Characterization of Bipartite Graphs(e.g.)

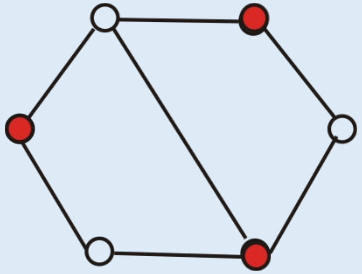


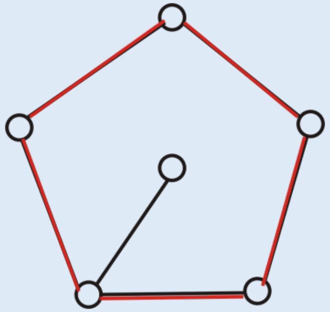
Non-bipartite graph











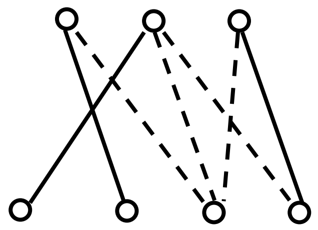
Non-bipartite graph

↳ Matchings and Complete Matchings in Bipartite Graphs

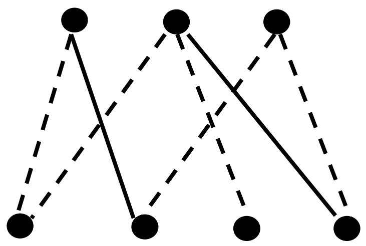
- **Definition 6.11:** Let $G=<V_1,V_2,E>$ be a bipartite graph, where $E'\subseteq E$. If the edges in E' are pairwise non-adjacent, then E' is called a ***matching*** in G. If adding any edge to E' results in a set of edges that is no longer a matching, then E' is called a ***maximal matching*** in G. The matching in G with the maximum number of edges is called the ***maximum matching*** of G.
  - Moreover, suppose $|V_1|\leq|V_2|$ and E' is a matching in G. If $|E'|=|V_1|$, then E' is called a ***complete matching*** from V_1 to V_2 .
  - When $|V_1|=|V_2|$, a complete matching is called a ***perfect matching***.

6.4.1 Bipartite Graphs

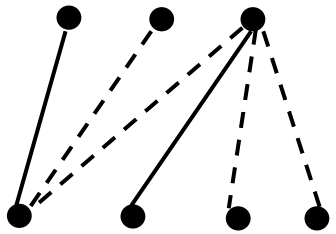
↳ Matchings and Complete Matchings in Bipartite Graphs(e.g.)



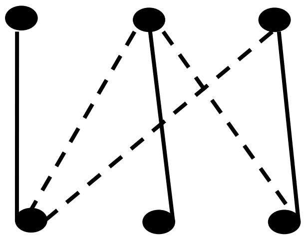
Maximum Matching and Complete Matching



Maximal Matching



Maximum Matching



Perfect Matching

↳ Complete Matching Existence Theorem

- **Theorem 6.6 (Hall's Theorem):**

  Let G=<V_1,V_2,E> be a bipartite graph with $|V_1|=|V_2|$. There ***exists a complete matching*** from V_1 to V_2 in G if and only if, for any k (where $1 \le k \le |V_1|$), the set of k vertices in V_1 is adjacent to at least k vertices in V_2 (the ***distinctness condition***).

- **Theorem 6.7:**

  Let G=<V_1,V_2,E> be a bipartite graph with $|V_1| \le |V_2|$. If there exists a positive integer t such that each vertex in V_1 is connected to at least t edges, and each vertex in V_2 is connected to at most t edges (the **t-condition**), then there ***exists a complete matching*** from V_1 to V_2 in G.

↳ Application Examples of Complete Matchings

- **Example:**

  A middle school has three extracurricular activity groups: the Math Group, the Computer Group, and the Biology Group. There are five students: Zhao, Qian, Sun, Li, and Zhou. In each of the following three cases, determine whether it is possible to select three students to serve as group leaders, one for each group:

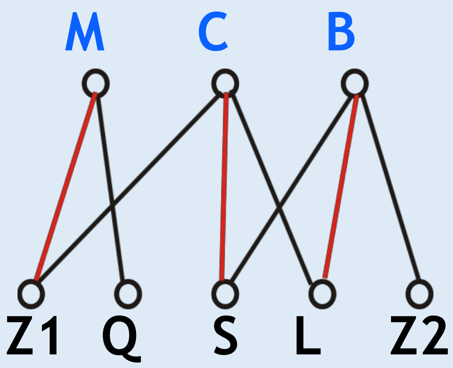
(1) Zhao and Qian are members of the Math Group; Zhao, Sun, and Li are members of the Computer Group; Sun, Li, and Zhou are members of the Biology Group.

(2) Zhao is a member of the Math Group; Qian, Sun, and Li are members of the Computer Group; Qian, Sun, Li, and Zhou are members of the Biology Group.

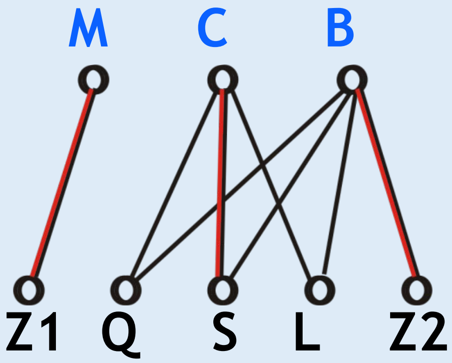
(3) Zhao is a member of both the Math and Computer Groups; Qian, Sun, Li, and Zhou are members of the Biology Group.

↳ Application Examples of Complete Matchings

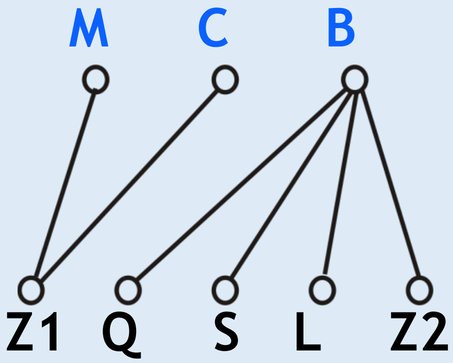
M: Math Group
C: Computer Group
B: Biology Group
Z1: Zhao
Q: Qian
S: Sun
L: Li
Z2: Zhou



(1)



(2)



(3)

A complete matching corresponds to a feasible assignment.

(1) and (2) admit complete matchings, with multiple possible assignments.

(3) does not satisfy the distinctness condition, so no complete matching exists.

6.4 Several Special Types of Graphs

- **6.4.1 Bipartite Graphs**

  Necessary and sufficient conditions for a graph to be bipartite

  Matching, maximal matching, maximum matching, complete matching, perfect matching

- **6.4.2 Eulerian Graphs**

  Eulerian circuits (paths) and their necessary and sufficient conditions for existence

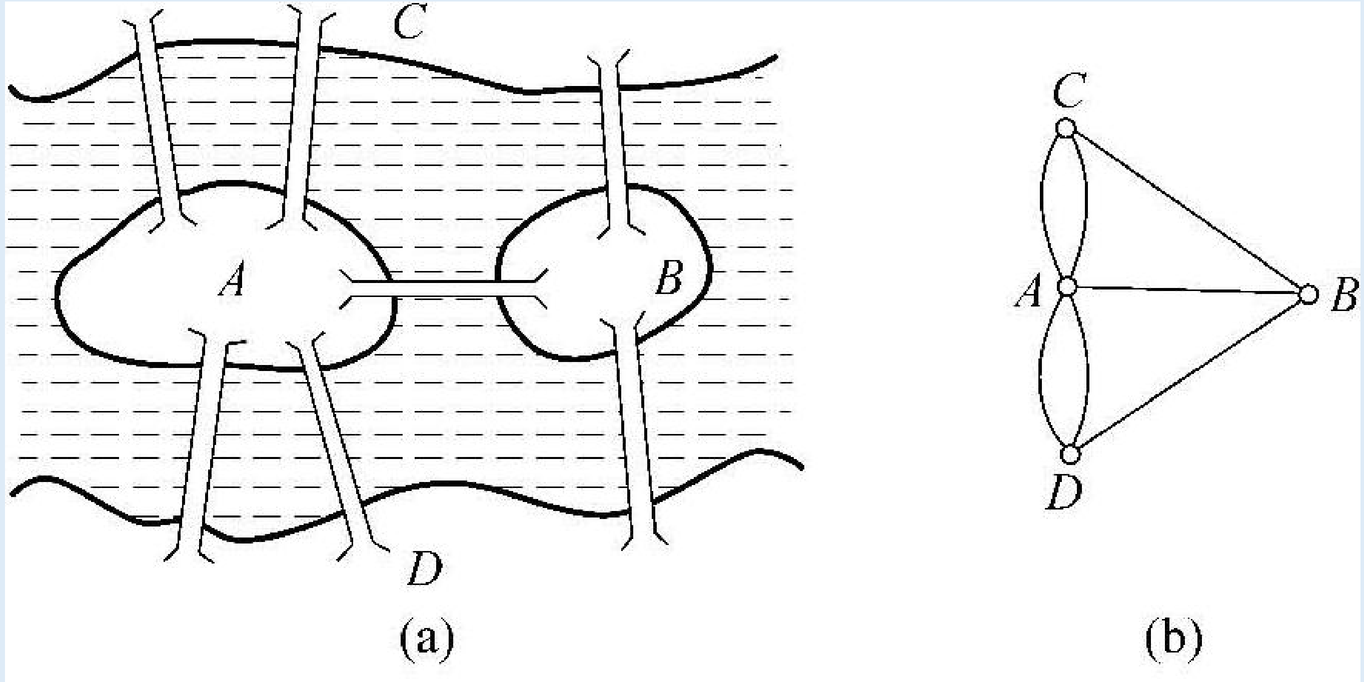
- **6.4.3 Hamiltonian Graphs**

  Hamiltonian circuits (paths) and the necessary and sufficient conditions for their existence

- **6.4.4 Planar Graphs**

↳ Seven Bridges of Königsberg → Birth of Graph Theory

Diagram of the Seven Bridges of Königsberg



↳ Eulerian Path (Circuit) and Eulerian Graph

- *Eulerian Path*: A path that passes through all vertices and **each edge exactly once**.
- *Eulerian Circuit*: A circuit that passes through all vertices and **each edge exactly once**.
- *Eulerian Graph*: A graph that contains an Eulerian circuit.
- Notes:
  - The above definitions apply to both **undirected** and **directed** graphs.
  - A **trivial graph** is considered an Eulerian graph.
  - An Eulerian path is a **simple path**, and an Eulerian circuit is a **simple circuit**.
  - **Loops (self-edges)** do **not affect** the Eulerian property of a graph.

↳ Eulerian Graph Theorem (for undirected graphs)

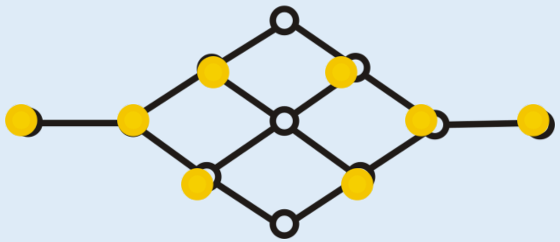
- **Theorem 6.8:**

  (1) An undirected graph ***G*** has an ***Eulerian circuit*** if and only if ***G*** is connected and has no vertices of odd degree.

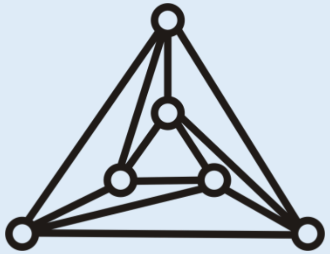
- (2) An undirected graph ***G*** has an ***Eulerian path*** but not an Eulerian circuit if and only if ***G*** is connected and has exactly two vertices of odd degree, with all other vertices having even degree. These two odd-degree vertices are the endpoints of every Eulerian path.

6.4.2 Eulerian Graphs

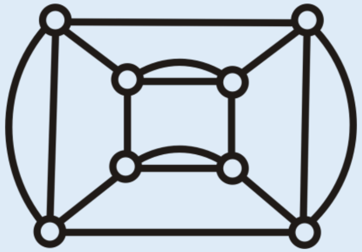
↳ Eulerian Graph Theorem (for undirected graphs)(e.g.)



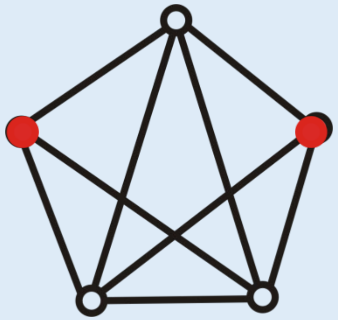
No Eulerian Path



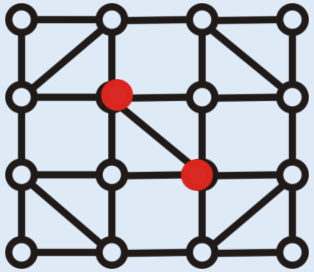
Eulerian Graph



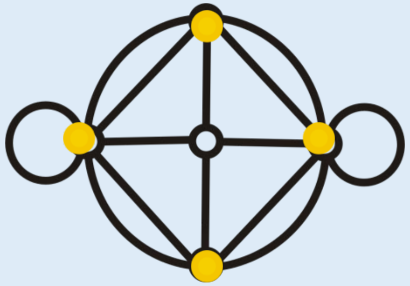
Eulerian Graph



Eulerian Path, not Eulerian Graph



Eulerian Path, not Eulerian Graph



No Eulerian Path

↳ Eulerian Graph Theorem (for directed graphs)

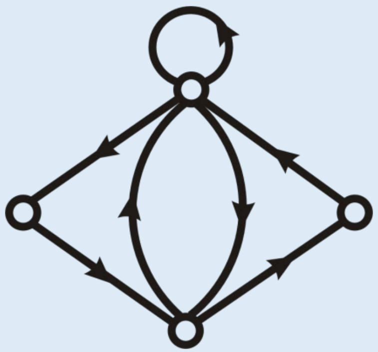
- **Theorem 6.9:**

(1) A directed graph ***D*** has an ***Eulerian circuit*** if and only if ***D*** is connected and the **in-degree equals the out-degree** for every vertex.

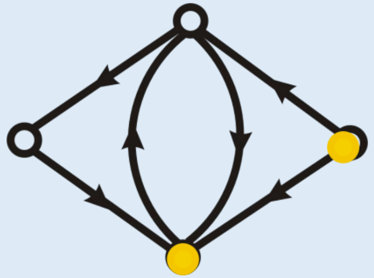
(2) A directed graph D has an ***Eulerian path*** but not an Eulerian circuit if and only if ***D*** is connected and there is **one vertex** whose in-degree exceeds its out-degree by 1, and **one vertex** whose out-degree exceeds its in-degree by 1, with all other vertices having equal in-degree and out-degree.

6.4.2 Eulerian Graphs

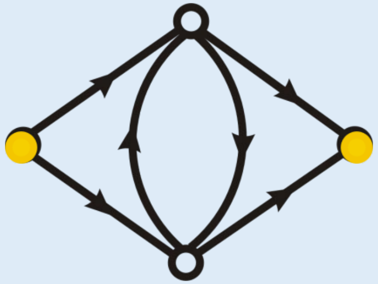
↳ Eulerian Graph Theorem (for directed graphs)(e.g.)



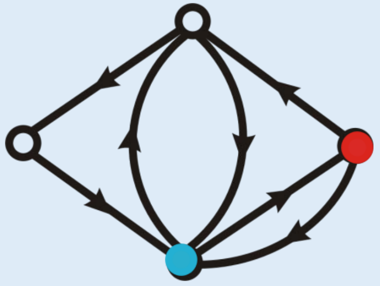
Eulerian Graph



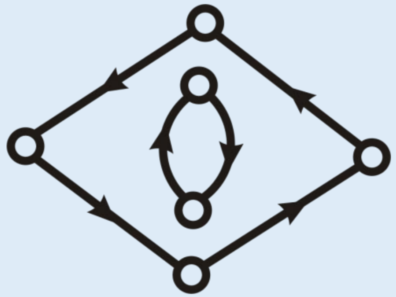
No Eulerian Path



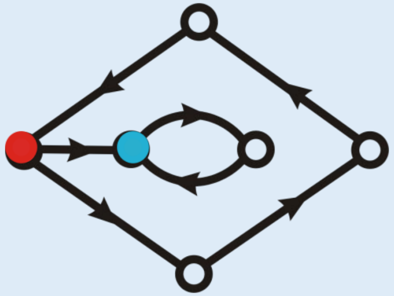
No Eulerian Path



Eulerian Path, not Circuit



No Eulerian Path



Eulerian Path, not Circuit

6.4 Several Special Types of Graphs

- 6.4.1 Bipartite Graphs

  Necessary and sufficient conditions for a graph to be bipartite

  Matching, maximal matching, maximum matching, complete matching, perfect matching
- 6.4.2 Eulerian Graphs

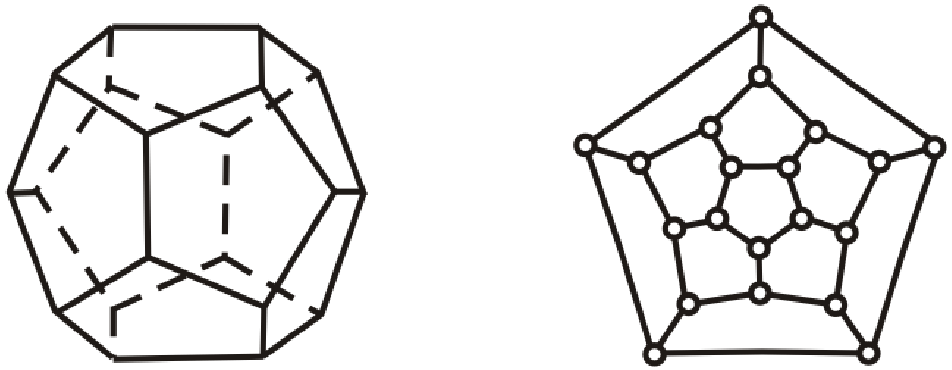
  Eulerian circuits (paths) and their necessary and sufficient conditions for existence
- 6.4.3 Hamiltonian Graphs

  Hamiltonian circuits (paths) and the necessary and sufficient conditions for their existence
- 6.4.4 Planar Graphs

6.4.3 Hamiltonian Graphs

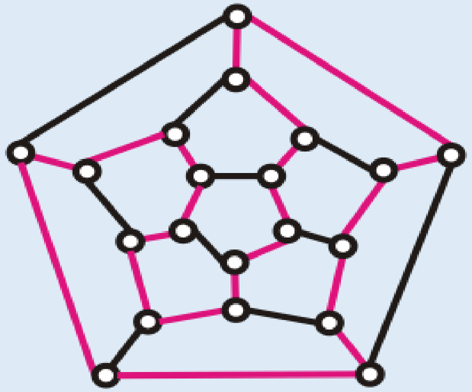
↳ The "Hamiltonian Game" and the Hamiltonian Circuit

W.Hamilton, 1859



↳ Hamiltonian Path (Circuit) and Hamiltonian Graph

- ***Hamiltonian Path***: A **path** that visits every vertex in the graph **exactly once.**
- ***Hamiltonian Circuit***: A **circuit** that visits every vertex in the graph **exactly once.**
- ***Hamiltonian Graph***: A graph that **contains a Hamiltonian circuit.**
- **Notes:**
  - A Hamiltonian path is a ***elementary*** **path.**
  - A Hamiltonian circuit is a ***elementary*** **circuit.**
  - A graph having a Hamiltonian path does **not necessarily** have a Hamiltonian circuit.
  - **Loops** and **parallel edges** do **not affect** the Hamiltonian property of a graph.



↳ Necessary Condition for Hamiltonian Graphs (Undirected only)

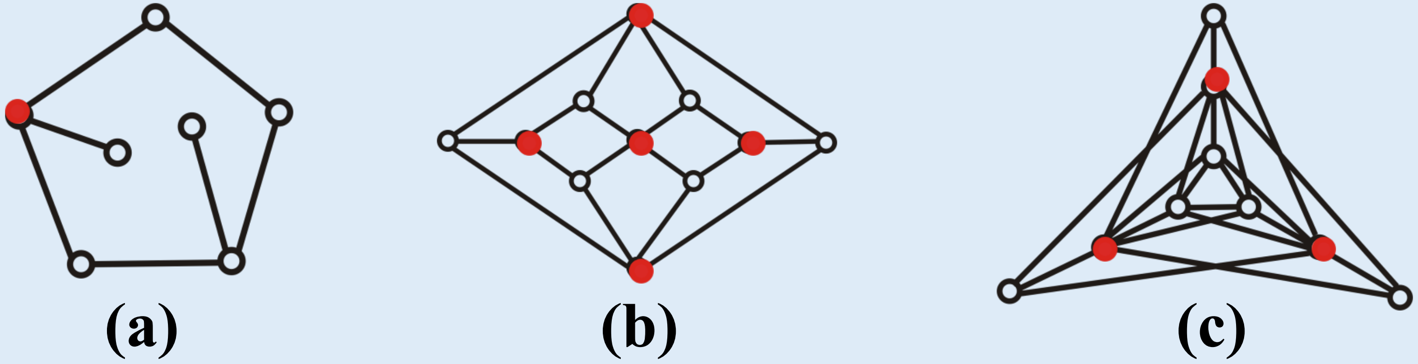
- **Theorem 6.10:** If an undirected graph $G=<V,E>$ is a Hamiltonian graph, then for any non-empty proper subset $V_1 \subset V$, The number of connected components in $G-V_1$ satisfies: $p(G-V_1) \leq |V_1|$.
- **Proof:**

  Let C be a Hamiltonian circuit in G. Then: p(C−V1)≤|V1|, **since** $p(C-V_1) \leq |V_1|$. And because $C \subseteq G$, Hence, $p(G-V_1) \leq p(C-V_1) \leq |V_1|$.
- **Corollary:**

  A graph with a **cut vertex** is not a Hamiltonian graph.

↳ Necessary Condition for Hamiltonian Graphs (e.g.)

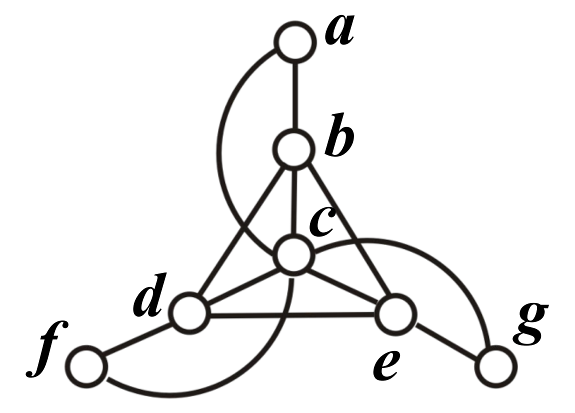
- **Example:** Prove that each of the following graphs is not a Hamiltonian graph.



There exists a Hamiltonian path in (c).

↳ Necessary Condition for Hamiltonian Graphs (e.g.)

- **Example:** Prove that the graph on the right is not a Hamiltonian graph.



- **Proof:**
  - Assume there exists a Hamiltonian circuit. ***a*,*f*,*g*** are the node of degree 2, edge **(*a*,*c*)**, **(*f*,*c*)** and **(*g*,*c*)** must all be included in the Hamiltonian circuit. As a result, vertex ***c*** would appear three times, which is a **contradiction**.
  - Moreover, the graph satisfies the condition of **Theorem 6.10**, which shows that the condition is **necessary but not sufficient**. The graph has a Hamiltonian path.

↳ Sufficient Conditions for Hamiltonian Graphs (Undirected Case)

- **Theorem 6.11：**

  Let G be a simple undirected graph of order n ($n \geq 3$).

  - If the sum of the degrees of any two non-adjacent vertices is at least $n-1$, then G contains a ***Hamiltonian path***.
  - If the sum is at least n, then G contains a ***Hamiltonian circuit***, i.e., G is a ***Hamiltonian graph***.

- **Corollary：**

  - Let G be a simple undirected graph of order n ($n \geq 3$), If $\delta(G) \geq n/2$, then G is a Hamiltonian graph.
  - When $n \geq 3$, the complete graph K_n is Hamiltonian; when $r=s \geq 2$, the complete bipartite graph $K_{r,s}$ is Hamiltonian.

↳ Detecting Hamiltonian Paths via Complete Underlying Graphs

- **Theorem 6.12:**

  Let ***D*** be a **directed graph** of order n (n≥2).
  If the underlying undirected graph (obtained by ignoring the directions of all edges) contains a subgraph K_n , then ***D*** contains a ***Hamiltonian path*.**

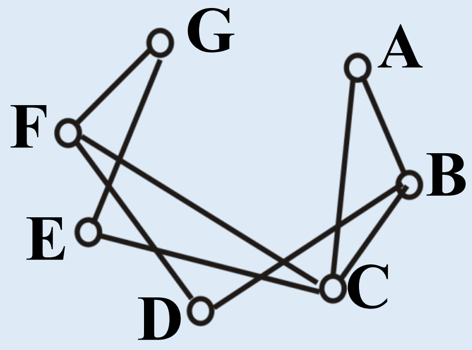
- **Example:** There are 7 people:
  - A speaks English.
  - B speaks English and Chinese.
  - C speaks English, Italian, and Russian.
  - D speaks Japanese and Chinese.
  - E speaks German and Italian.
  - F speaks French, Japanese, and Russian.
  - G speaks French and German

Can they be seated around a round table so that **each person can communicate with both neighbors?**

↳ Hamiltonian Circuits: Roundtable Seating for Communication

- **Solve:**

(1) Construct an undirected graph where each person is a vertex, and there is an edge between two people **if and only if** they speak a common language.



(2) ACEGFDBA is a *Hamiltonian circuit*; they can be seated in this order around the table.

6.4 Several Special Types of Graphs

- **6.4.1 Bipartite Graphs**

  Necessary and sufficient conditions for a graph to be bipartite

  Matching, maximal matching, maximum matching, complete matching, perfect matching
- **6.4.2 Eulerian Graphs**

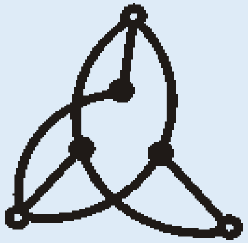
  Eulerian circuits (paths) and their necessary and sufficient conditions for existence
- 6.4.3 Hamiltonian Graphs

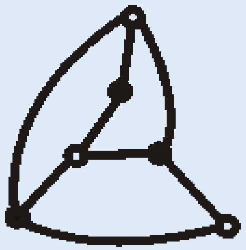
  Hamiltonian circuits (paths) and the necessary and sufficient conditions for their existence
- **6.4.4 Planar Graphs**

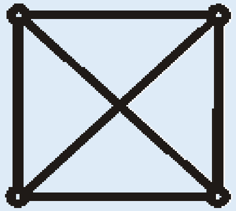
6.4.4 Planar Graphs

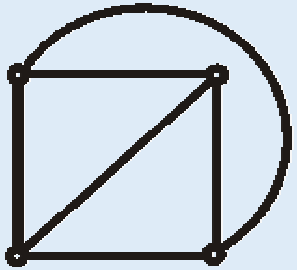
↳ Planar Graphs and Planar Embeddings

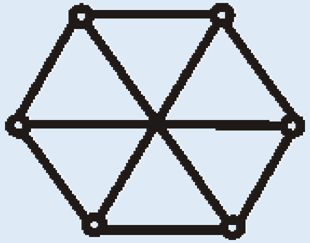
- **Definition 6.12:** A graph ***G*** is called a ***planar graph*** if it can be drawn in the plane such that its edges do not intersect except at the vertices. The drawing of the graph with no edge intersections is called a ***planar embedding*** of ***G***. A graph that does not have a planar embedding is called a ***non-planar graph***.
- **Example**: Determine whether the following graph is a planar graph.

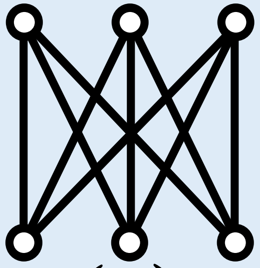

(1)


(2)


(3)


(4)

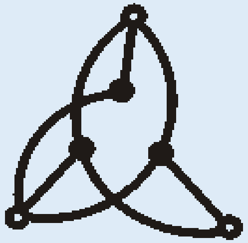

(5)


(6)

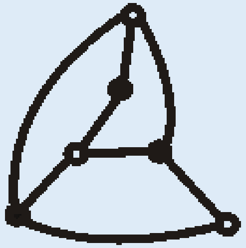
6.4.4 Planar Graphs

↳ Planar Graphs and Planar Embeddings(e.g.)

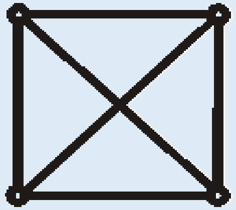
- **Example**: Determine whether the following graph is a planar graph.



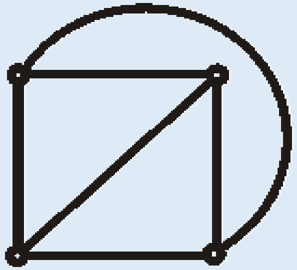
(1)



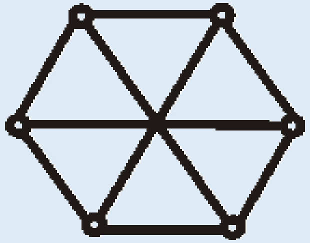
(2)



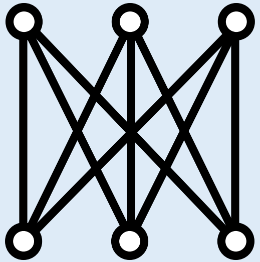
(3)



(4)



(5)



(6)

- **Solution:**
  - The graphs **(1)** to **(4)** are ***planar graphs***. **(2)** is a ***planar embedding*** of **(1)**, and **(4)** is a ***planar embedding*** of **(3)**.
  - **(5)** is the **complete graph** K_6, which is a typical ***non-planar grap***h.
  - **(6)** is the **complete bipartite graph** $K_{3,3}$, which is a typical ***non-planar graph***.

↳Properties of Planar Embeddings: Faces, Boundaries, Degrees

- Let *G* be a planar embedding.
  - *Faces* **of G**: Each region into which the plane is divided by the edges of *G*.
  - *Infinite face* **(outer face)**: The face with infinite area, denoted by R_0 .
  - *Finite faces* **(inner faces)**: Faces with finite areas, denoted by R_1, R_2,…, R_k.
  - *Boundary* **of face R_i** : The set of loops formed by the edges that enclose R_i.
  - *Degree* **of face R_i** : The length of the boundary of R_i, denoted by **deg(R_i)**.
- **Note**: The **boundary** of a face may consist of simple **loops**, elementary **cycles**, or even more complex loops, and in some cases, it may be the union of disconnected loops.

↳Planar Embeddings: Faces, Boundaries, Degrees(e.g.)

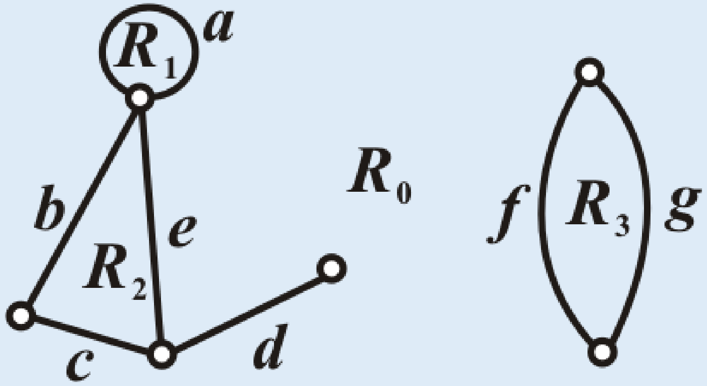
- **Example:** The diagram on the right has 4 faces.

R_1 Boundary: a

R_2 Boundary: bce

R_3 Boundary: fg

R_0 Boundary: $abcdde$, fg



deg(R_1)= 1

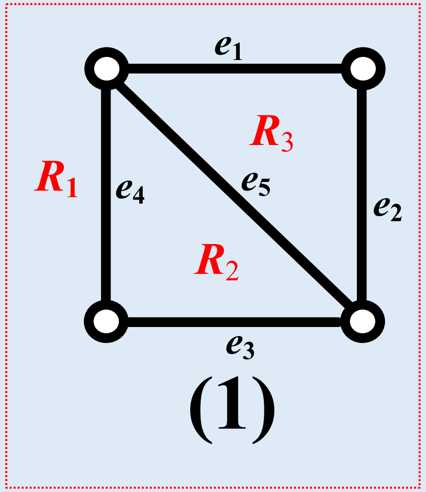
deg(R_2)= 3

deg(R_3)= 2

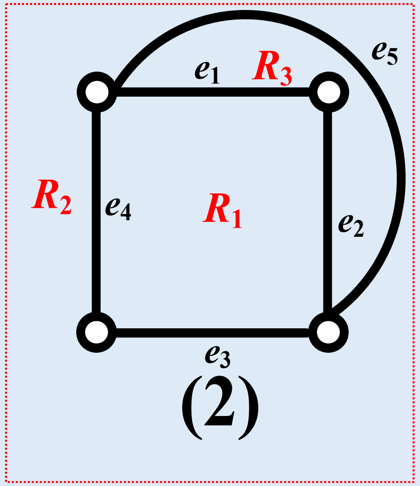
deg(R_0)= 8

↳Planar Embeddings: Faces, Boundaries, Degrees(e.g.)

- **Example:** The two diagrams on the right are planar embeddings of the same planar graph.



R_1 is the **outer face** in **(1)** and the inner face in **(2)**.
R_2 is the **inner face** in **(1)** and the outer face in **(2)**.

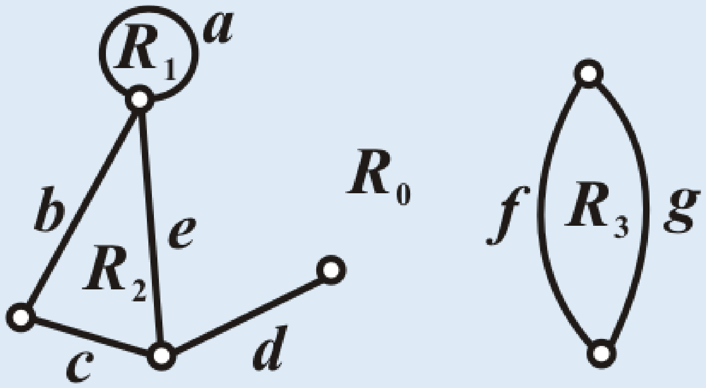


- **Explanation:**
  - (1) A planar graph can have **multiple different forms of planar embeddings**, all of which are isomorphic.
  - (2) Any face of a planar graph can be considered the outer face through a **transformation** (such as geodesic projection).

6.4.4 Planar Graphs

↳Theorem on the Sum of Face Degrees in a Planar Embedding

- **Theorem 6.13**: The *sum of the degrees* of all faces in a planar graph is equal to **twice the number of edges**.
- **Proof**: An edge either serves as a common boundary for two faces or appears twice in the boundary of a single face. When calculating the sum of the degrees of all faces, each edge is counted exactly twice.
- **For example:** In the diagram below, the sum of the degrees of the faces is equal to: $\boldsymbol{\sum_{i=0}^{3} deg(R_i) = 8+1+3+2 = 2|\{a,b,c,d,e,f,g\}|}$

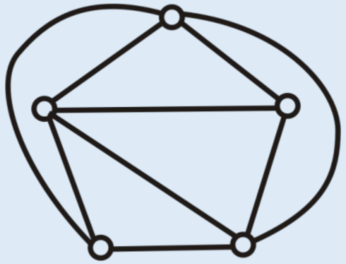


同济经管
TONGJI SEM

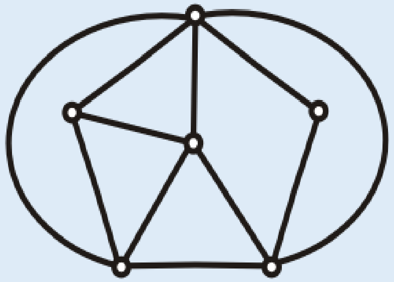
CAMEA
中国高质量MBA教育认证
AACSB ACCREDITED
EQUIS ACCREDITED

↳Maximal planar graph

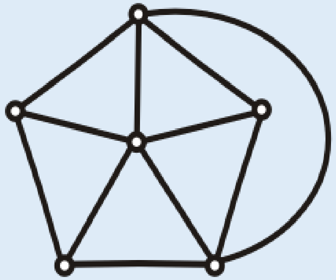
- **Definition 6.13:** If ***G*** is a simple planar graph, and the graph obtained by adding a new edge between any two non-adjacent vertices is non-planar, then ***G*** is called a ***maximal planar graph***.
- **Example:**
  - K_1, K_2, K_3, K_4 are all maximal planar graphs.
  - **(1)** is K_5 with one edge removed, which is a maximal planar graph. **(2)** and **(3)** are not.



(1)



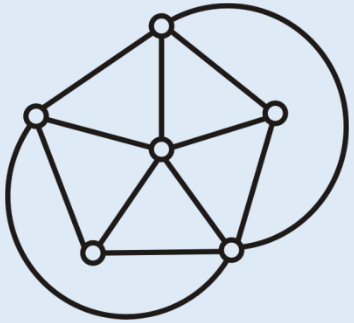
(2)



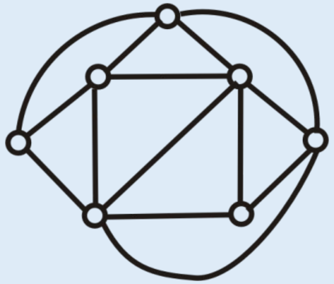
(3)

↳Properties of Maximal Planar Graphs: Connected and triangular.

- A maximal planar graph is ***connected***.
- Let ***G*** be a simple graph of order ***n*** (***n***≥3). A necessary and sufficient condition for ***G*** to be a maximal planar graph is that the ***degree of each face in G is 3***. （Triangulation）
- **Example:**



Maximal planar graph



The degree of the outer face is 4.
It is a non-maximal planar graph.

↳Euler’s Formula for Connected Planar Graphs

- **Theorem 6.14**: Let ***G*** be a *connected planar graph* with ***n*** vertices, ***m*** edges, and ***r*** faces. Then, $n-m+r=2$.
- **Proof**: (by induction on ***m***).
  - **Base case**: When $m=0$, ***G*** is a trivial graph and $n-0+r=2$ holds.
  - **Inductive Hypothesis:** Assume the formula holds for all graphs with $m=k$ edges, i.e., $n-k+r=2$.
  - **Inductive step**: We now prove it for $m=k+1$ by considering two cases:

①If ***G*** contains *no cycle*, then ***G*** must have a vertex ***v*** of degree **1**. By deleting ***v*** and its incident edge, we get a new graph ***G′***, which is connected, has $n-1$ vertices, ***k*** edges, and ***r*** faces. By the inductive hypothesis, $(n-1)-k+r=2$, so $n-(k+1)+r=2$, which proves the conclusion for $m=k+1$.

②If ***G*** contains a *cycle*, we can delete one edge from the cycle, resulting in a new graph ***G′*** with ***n*** vertices, ***k*** edges, and $r-1$ faces. Again, by the inductive hypothesis, $n-k+(r-1)=2$, so $n-(k+1)+r=2$, proving the conclusion for $m=k+1$.

- Thus, by induction, the theorem holds.

↳Corollary: Euler’s Formula for Disconnected Planar Graphs

- **Corollary**: Let G be a planar graph with p connected components ($p \geq 2$). Then, $n-m+r=p+1$ where n, m, and r are the number of vertices, edges, and faces of G, respectively.
- **Proof**: Let the ***i-th*** connected component have n_i vertices, m_i edges, and r_i faces.
  - By Euler's formula for each connected component, we have: $n_i - m_i + r_i = 2$, $i = 1, 2, \ldots, p$.
  - Summing these equations gives: $(n_1 + n_2 + \cdots + n_p) - (m_1 + m_2 + \cdots + m_p) + (r_1 + r_2 + \cdots + r_p) = 2p$.
  - Note that the total number of faces $r = r_1 + \ldots + r_p - p + 1$, so we obtain: $n - m + r = p + 1$.
- Thus, the *corollary is proven*.

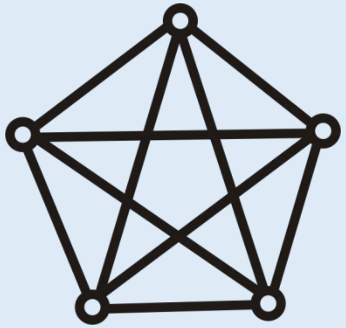
↳Edge Bound for Connected Planar Graphs

- **Theorem 6.15**: Let *G* be a *connected planar graph* with *n* vertices and *m* edges, where the degree of each face is at least *l* (*l* ≥3), Then $m \le \frac{l}{l-2}(n-2)$.
- **Proof**: In the planar graph *G*, the sum of the degrees of all faces is 2*m*. Let the number of faces be *r*.
  - Since the degree of each face is at least *l*， we have $r \cdot l \le 2m$， By Euler's formula， $n-m+r=2$, thus $r=2+m-n$, Substituting this into the inequality：$2m \ge l(2+m-n)$， $2m - lm \ge 2l - ln$， $m(2-l) \ge 2l - ln$.
  - Since $l \ge 3, 2-l<0$, thus dividing by $2-l$ reverses the inequality: $m \le \frac{l}{l-2}(n-2)$.
- Thus, the inequality is proven.

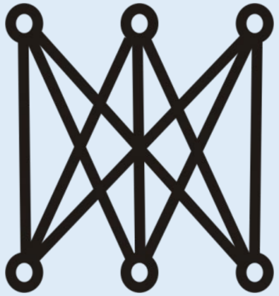
6.4.4 Planar Graphs

↳Edge Bound for Connected Planar Graphs（e.g.）

- **Example:** Prove that the complete graph K_5 and the complete bipartite graph $K_{3,3}$ are not planar graphs.



- **Proof**: We use proof by contradiction. Assume that they are planar graphs.
  - **For K_5 : n=5, m=10, l=3，** Assuming the graph satisfies Theorem 6.15 , $m \le \frac{l}{l-2}(n-2)$， **10 <= 9.**
  - **For $K_{3,3}$: n=6, m=9, l=4 ，** Similarly, we get: **9 <= 8.**
  - This also leads to a contradiction. Therefore, the assumption is incorrect, meaning **K_5 and $K_{3,3}$** are ***not planar graphs***.



- **Note**：$K_{3,3}$ there are no simple cycles of length 1 or 2. Any closed path must pass through an even number of edges, so each face is surrounded by at least 4 boundary edges, which means the degree of each face **l ≥4**。

↳ Graph Homeomorphism and Homeomorphic Transformations

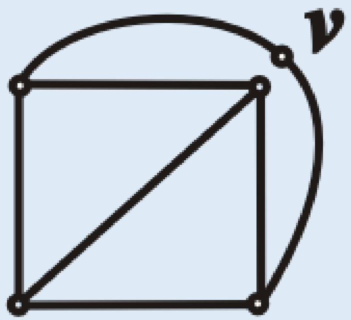
- ***Homotopy***: Homotopy focuses on the isomorphism between two graphs after inserting or **removing 2-degree vertices.**
  - Homotopy helps in understanding whether two different graphs are "***essentially the same***" and aids in recognizing the fundamental similarities or equivalences between different structures.
  - *Homotopy transformations* are typically a concept in topology, and graph transformations are considered homotopy transformations in the graph's topological structure (such as inserting or removing 2-degree vertices).

↳ Graph Contraction and Contraction Transformations

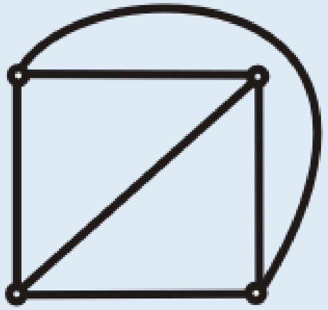
- *Contraction*: Contraction simplifies a graph by removing an edge and replacing the two original vertices with a new vertex.
  - Contraction helps reduce the complexity of a problem, making it easier to analyze and solve. It can assist in solving complex optimization problems such as finding the minimum cut, network flow, and graph coloring.
  - Contraction is one of the graph ***transformation operations*** that simplifies a graph by merging edges while maintaining its topological structure.

↳ Homeomorphism and Contraction Operations on Graphs

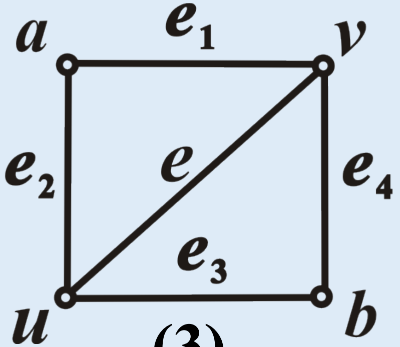
- *Delete a 2-degree vertex* **v**: As shown, from **(1)** to **(2)**.
- *Insert a 2-degree vertex* **v** : As shown, from **(2)** to **(1)**.
- **G_1 and G_2 are** *homotopic*: G_1 and G_2 are **isomorphic**, or they become isomorphic after repeatedly **inserting or removing 2-degree vertices**.
- *Edge contraction* **e**: As shown, from(3) to (4)



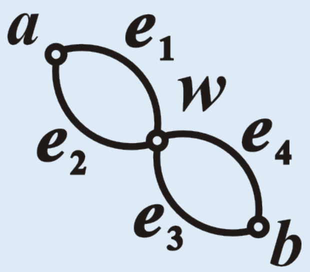
(1)



(2)



(3)



(4)

6.4.4 Planar Graphs • Kuratowski's Theorem:
↳ Necessary and Sufficient Condition for Planar Graphs

- **Theorem 6.16**: A graph is planar if and only if it contains neither a subgraph **homeomorphic to K_5** nor a subgraph **homeomorphic to $K_{3,3}$**.
- **Theorem 6.17**: A graph is planar if and only if it contains neither a subgraph that can be **contracted to K_5** nor a subgraph that can be **contracted to $K_{3,3}$**.
- **Note:** K_5 (the complete graph with five vertices) and $K_{3,3}$ (the complete bipartite graph with two sets of three vertices) are **typical non-planar graphs**.

6.4.4 Planar Graphs • Kuratowski's Theorem:
↳ Necessary and Sufficient Condition for Planar Graphs

- **Explanation:**

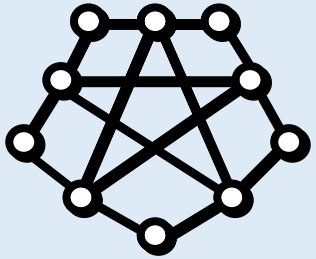
①A subgraph homeomorphic to K_5 or $K_{3,3}$ refers to a graph obtained by performing ***Homotopy*** transformations (adding or deleting 2-degree vertices) on K_5 or $K_{3,3}$. These *transformations do not change the non-planarity or bipartiteness* of K_5 or $K_{3,3}$.

②Theorem 6.17 emphasizes that if, after any ***edge contraction*** (deleting edges, merging vertices), the graph ***cannot be simplified to*** K_5 ***or*** $K_{3,3}$, then the graph is planar.

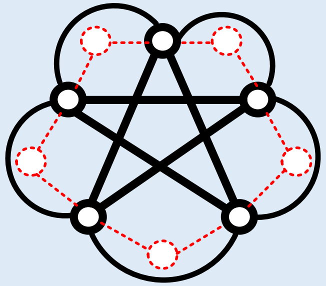
③Homotopy focuses on ***edge subdivision*** (inserting a 2-degree vertex) and the removal of 2-degree vertices, while contraction focuses on ***edge merging*** and vertex merging. These two operations are equivalent when determining the planarity of a graph.

6.4.4 Planar Graphs • Kuratowski's Theorem:
↳ Examples of Planar Graph Proofs

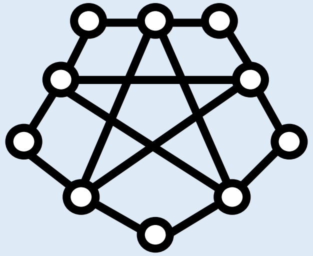
- **Example:** Prove that the following graph is non-planar.



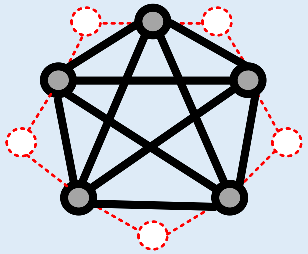
Removing five 2-degree vertices results in K_5 ⇒



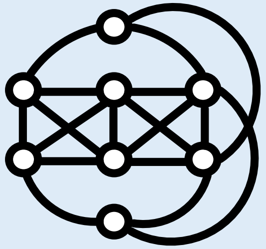
⇒ **Homeomorphic to K_5** , a **non-planar graph.**



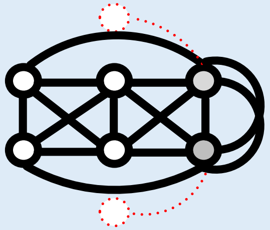
Contracting 5 edges results in K_5 ⇒



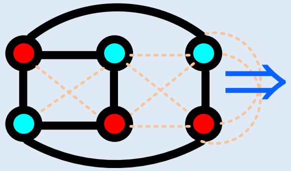
⇒ **Contracting to K_5** , a **non-planar graph.**



Contracting 2 edges ⇒



Extract $K_{3,3}$ subgraph ⇒



⇒ A **non-planar graph** that is **homeomorphic to $K_{3,3}$**

↳ Construction Method of the Dual Graph of a Planar Graph

Definition 6.14:

Let ***G*** be a planar graph with ***n*** vertices, ***m*** edges, and ***r*** faces. The *dual graph* $G^*=<V^*,E^*>$ is constructed as follows:

- For each face R_i of ***G***, choose an arbitrary point v_i^* within R_i to serve as a vertex of G^*, $V^*=\{ v_i^* \mid i=1,2,\ldots,r \}$.
- For each edge e_k in ***G***:

  If e_k lies *on the common boundary* of faces R_i and R_j, create an edge $e_k^*=(v_i^*,v_j^*)$, in G*, such that e_k^* intersects e_k .

  If e_k lies only *on the boundary of a single face* R_i, create a loop $e_k^*=(v_i^*,v_i^*)$. $E^*=\{ e_k^* \mid k=1,2, \ldots,m \}$.

↳ Planar Graph ⇒ Dual Graph: Lost Information

- **Details of the planar graph *lost* in the dual graph:**
  - **Original layout of vertices and edges:**
    In the dual graph, faces of the original planar graph become vertices, and adjacency between faces becomes edges. However, the original layout, including the positions, placements, and relative distances of vertices and edges, is no longer directly preserved in the dual graph。
  - **Vertex degrees and edge connections (such as edge crossings or winding patterns):**
    The degree of vertices and the specific ways edges connect (e.g., crossings or how edges wrap around certain regions) are also not directly reflected in the dual graph.

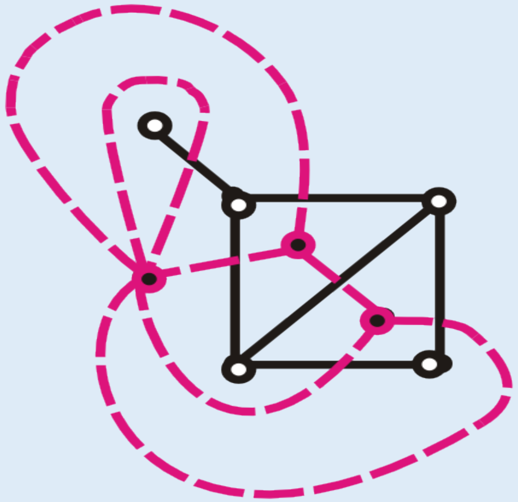
↳ Planar Graph ⇒ Dual Graph: Preserved Planar Graph Properties

- **Properties of the planar graph *preserved* in the dual graph :**
  - **Connectivity**
  - **Cycles and cut sets:** Cycles in the original graph correspond to cut sets in the dual graph.
  - **Planarity**
  - **Satisfaction of the same Euler's formula as the original graph.**
  - **The original graph and its dual have the same number of edges**

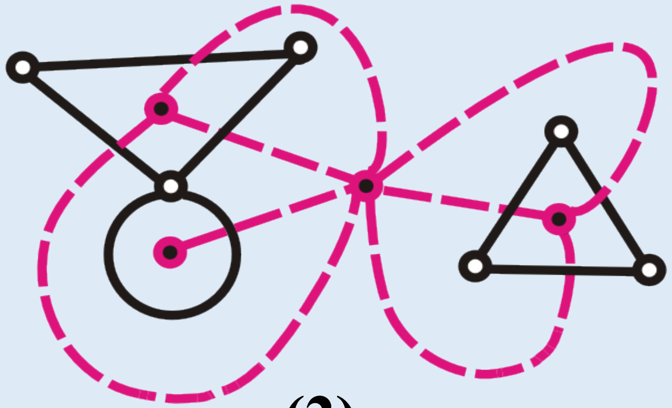
6.4.4 Planar Graphs

↳ Planar Graph ⇒ Dual Graph（e.g.）

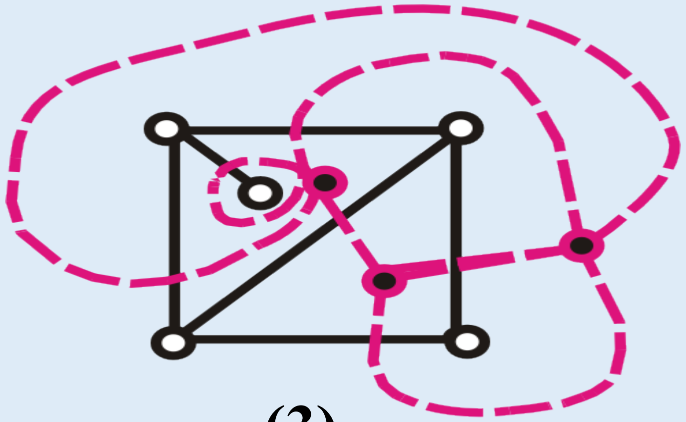
- **Example:** The black solid lines represent the original planar graph, and the red dashed lines represent its dual graph.



(1)



(2)



(3)

↳ Properties of the Dual Graph of a Planar Graph

- The dual graph ***G**** is a planar graph and a planar embedding.
- The dual graph ***G**** is ***connected***.
- If an edge ***e*** forms a ***cycle*** in ***G***, then the corresponding edge ***e**** in ***G**** is a ***cut-edge*** (bridge); if ***e*** is a bridge in ***G***, then the corresponding edge ***e**** in ***G**** forms a cycle.
- The dual graphs of isomorphic planar graphs are ***not necessarily isomorphic***.

For example, in the previous illustration, planar graphs (1) and (3) are isomorphic, but their dual graphs are not isomorphic.

↳ Relationship Between *G* and *G**

- **Theorem 6.18:** Let ***G**** be the dual graph of a connected planar graph ***G***, ***n****, ***m****, ***r**** and ***n***, ***m***, ***r*** denote the number of vertices, edges, and faces of ***G**** and ***G***, respectively.

(1) $n^* = r$

(2) $m^* = m$

(3) $r^* = n$

(4) If the vertex v_i^* of ***G**** lies in the face R_i of ***G***, then $d(v_i^*) = \deg(R_i)$.

6.4.4 Planar Graphs

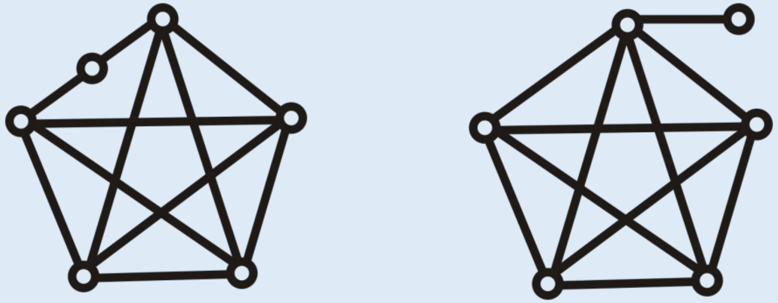
↳ Constructing a Simple Connected Non-Planar Graph (e.g.)

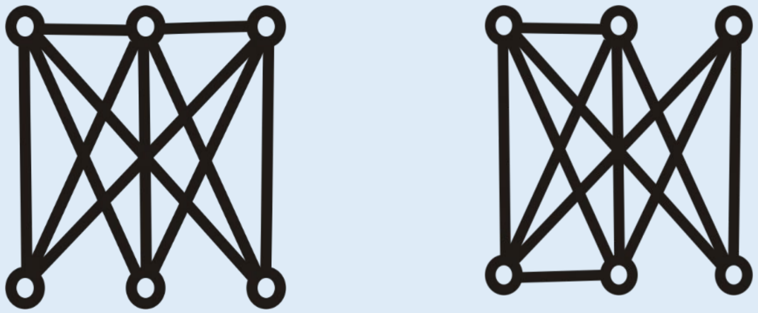
- **Example:** Draw all non-isomorphic simple connected non-planar graphs with 6 vertices and 11 edges.

- **Solution:**

  (1) Add one vertex and one edge to K_5 (the complete graph with 5 vertices and 10 edges).

  (2) Add two edges to $K_{3,3}$ (the complete bipartite graph with 6 vertices and 9 edges).





↳ Four Color Theorem and Graph Coloring of Planar Graphs

- The core objective of *graph coloring problems* is to avoid using the same color for adjacent or related elements (such as vertices, edges, or faces) under specific constraints, while using as few colors as possible.
- The *Four ColorTheorem* applies to the face coloring of planar graphs, whereas vertex coloring and edge coloring follow different rules and theoretical frameworks.
- *Four Color Theorem*: For any planar graph, it is possible to color all its faces using no more than four colors, in such a way that any two faces sharing a common boundary do not have the same color (i.e., *every planar graph is 4-face-colorable*).
- The *map coloring problem* can be regarded as a specific instance of the face coloring problem for planar graph.

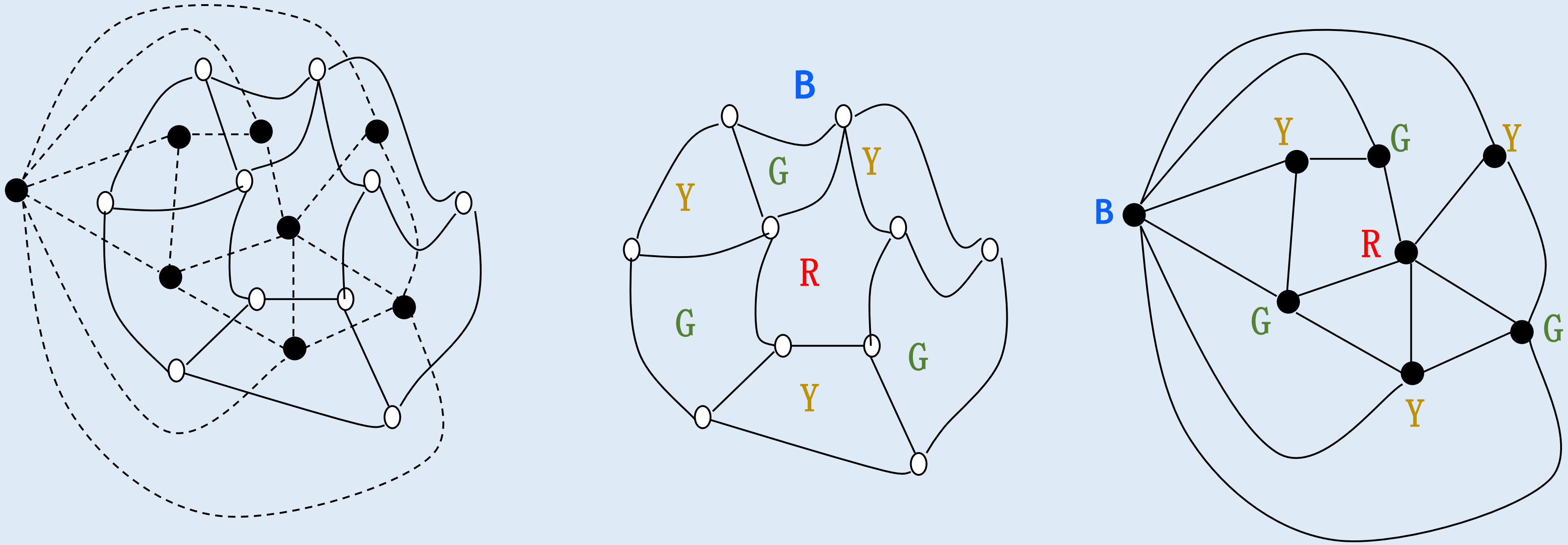
↳ Map Coloring as Vertex Coloring of Planar Graphs

- **Map:** A planar embedding of a connected, ***bridgeless planar graph***, where each face represents a country. Two countries are said to be adjacent if they share a common boundary.
- **Map coloring (face coloring):** Assign a color to each country on the map such that ***adjacent countries receive different colors***.
- **Map coloring problem:** Color the map using ***as few colors as possible***.
- Map coloring can be ***transformed into the vertex coloring of a planar graph***.When G has no bridges, its dual graph G^* has no loops.Faces of G correspond to vertices of G^*, and two faces of G are adjacent if and only if the corresponding vertices in G^* are adjacent.Thus, ***face coloring of G is equivalent to vertex coloring of G^****.

6.4.4 Planar Graphs

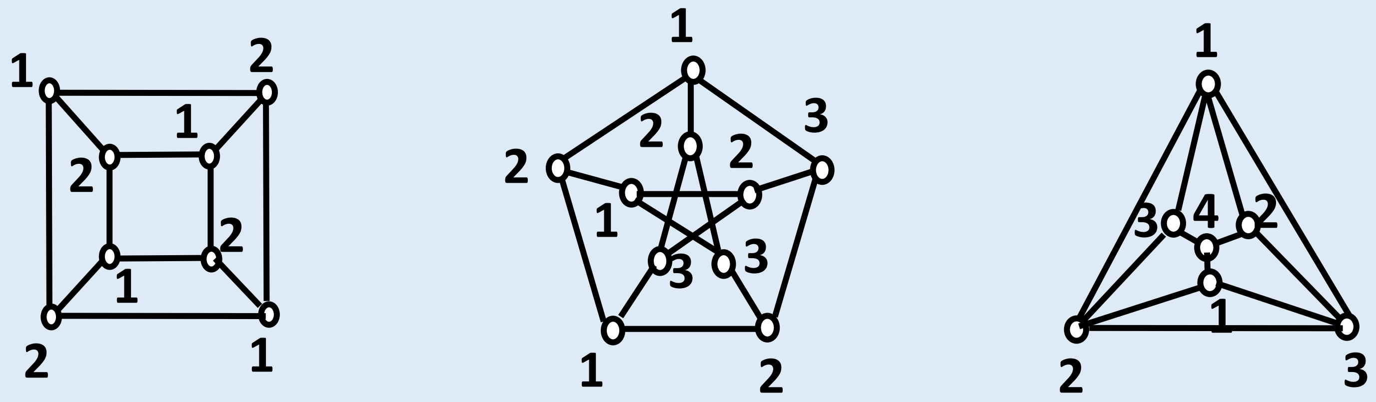
↳ Map Coloring (e.g.)

- **Example:** Map Coloring and Vertex Coloring of Planar Graphs.



↳ Graph coloring (e.g.)

- **Example:** Provide a coloring using as few colors as possible.



↳ Graph Coloring Example: Variable Register Allocation

- **Example：** A program has six variables x_i for i=1,2,…,6, where the following pairs of variables need to be used simultaneously: x_1 with x_4, x_1 with x_5 , x_2 with x_5 , x_2 with x_6 , x_3 with x_4, x_3 with x_6, x_4 with x_5 , and x_5 with x_6. Assign each variable to a register. Variables that need to be used simultaneously cannot be assigned to the same register. **Question:** What is the minimum number of registers needed? How should the variables be assigned?

- **Solution:**

  ① The problem is ***transformed into a vertex coloring problem*** of a graph: each variable x_i is treated as a vertex, and the "simultaneous usage" relationship between variables indicates the presence of an edge between the corresponding vertices.

↳ Graph Coloring Example: Variable Register Allocation

■ **Solution:**

① The problem is ***transformed into a vertex coloring problem*** of a graph: each variable $\boldsymbol{x}_i$ is treated as a vertex, and the "simultaneous usage" relationship between variables indicates the presence of an edge between the corresponding vertices.

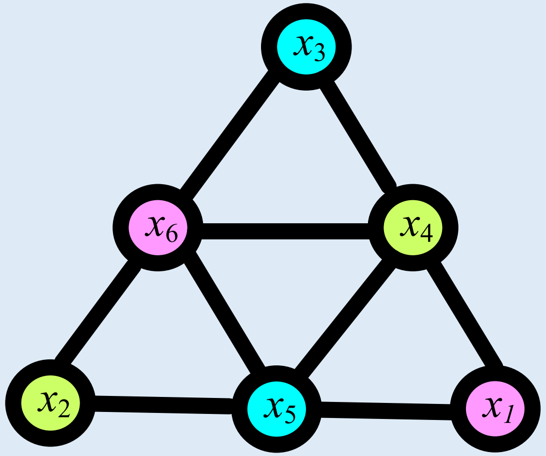
② ***Construct the graph*** by defining the vertex set and the edge set.

③ Build the graph and apply the principle of ***vertex coloring*** to determine the **chromatic number** (the minimum number of colors needed), ensuring that adjacent vertices are assigned different colors.

④ Based on the chromatic number, determine the ***minimum number of registers required*** and the corresponding assignment scheme.

↳ Graph Coloring Example: Variable Register Allocation

- **Result:**



The register allocation scheme using three registers is as follows:

Register 0: Assigned to variables $\mathbf{x_4}$ and $\mathbf{x_2}$.

Register 1: Assigned to variables $\mathbf{x_5}$ and $\mathbf{x_3}$.

Register 2: Assigned to variables $\mathbf{x_6}$ and $\mathbf{x_1}$.

↳ Four Color Theorem: Every planar graph is 4-colorable

- **Four Color Conjecture (1850s)**
  - **→ Five Color Theorem (Heawood, 1890)**
    - **→ Four Color Theorem (Appel and Haken, 1976)**
- **Theorem (Four Color Theorem):** ***Every planar graph is 4-colorable*.**
- The Four Color Theorem ***guarantees the existence of a four-coloring scheme for any planar graph***, but finding a specific coloring usually relies on concrete algorithms and techniques.
- Common ***coloring algorithms*** include greedy algorithms, backtracking algorithms, and heuristic search methods such as simulated annealing and genetic algorithms.

6.4 Special Types of Graphs• Brief summary

Objective :

Key Concepts ：