Graphs

Hongfei Fu

John Hopcroft Center for Computer Science
Shanghai Jiao Tong University

Nov. 9th, 2019
Finishing Set-Theory Part

- functions
- equinumerosity
Today’s Topic

Graph Theory (图论)

- graph definitions
- basic terminologies
- special simple graphs
- graph operations
- isomorphism
Textbooks

- **main textbook:**

- **reference material:**
  《图论与代数结构》，戴一奇，清华大学出版社

- **reference material:**
  Reinhard Diestel, *Graph Theory*, 5th edition [RD]
  http://diestel-graph-theory.com/basic.html?
Graphs Definitions

main textbook, Page 641–649
Question

How can we extract only the connectivity and intersection information of roads?
Question
How can we extract only the connectivity and intersection information of roads?

A Solution
- road intersections as points/vertices （结点）
- connecting road segments as lines/edges （边）
A General Definition

A graph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges.
A General Definition

A graph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges.

In addition, we have:

- every edge has either one or two vertices associated with it, call its endpoints;
- an edge is said to connect its endpoints.
Simple Graphs （简单图）

A simple graph is an ordered pair \((V, E)\) where

- The vertex set \(V\) can be infinite.
- \(V\) focuses on finite graphs.
- An edge \(\{u, v\}\) connects \(u\) and \(v\).
- There is no edge \(\{u, u\}\) that connects \(u\) to itself.
- There is at most one edge \(\{u, v\}\) that connects \(u\) and \(v\).
Simple Graphs （简单图）

A simple graph is an ordered pair \((V, E)\) where

- The vertex set \(V\) can be infinite.
- We focus on finite graphs.
- An edge \({u, v}\) is said to connect \(u, v\).
- There is no edge \({u, u}\) that connects \(u\) to itself.
- There is at most one edge \({u, v}\) that connects \(u\) and \(v\).
A simple graph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges satisfying \(E \subseteq \{\{u, v\} | u, v \in V, u \neq v\}\).
A simple graph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges satisfying \(E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}\).
Simple Graphs (简单图)

A simple graph is an ordered pair \( (V, E) \) where

- \( V \) is a set of vertices;
- \( E \) is a set of edges satisfying \( E \subseteq \{ \{u, v\} \mid u, v \in V, u \neq v \} \).

Remarks

- The vertex set \( V \) can be infinite.
- We focus on finite graphs.
- An edge \( \{u, v\} \) is said to connect \( u, v \).
- There is no edge \( \{u, u\} \) that connects \( u \) to itself.
- There is at most one edge \( \{u, v\} \) that connects \( u \) and \( v \).
Drawing of Graphs

- **essential part**: vertices and edges
- **not important**: the shape of vertices and edges
Social Networks

source: from internet
Multiple Edges
A **multigraph** is an ordered pair \((V, E)\) where
Multigraphs (多重图)

A multigraph is an ordered pair \((V, E)\) where
A multigraph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
A multigraph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
A multigraph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
- \(E\) is a function from \(E\) into \(\{\{u, v\} \mid u, v \in V, u \neq v\}\).
Chemical Bonds

source: from internet
A pseudograph is an ordered pair \((V, E)\) where
A pseudograph is an ordered pair \((V, E)\) where
A pseudograph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
A pseudograph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
A pseudograph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
- \(E\) is a function from \(E\) into \(\{\{u, v\} \mid u, v \in V\}\).
A Single-Direction Road
A directed multigraph is an ordered pair \((V, E)\) where
A directed multigraph is an ordered pair \((V, E)\) where
Directed (multi-)graphs (Digraphs) (有向图)

A directed multigraph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
A directed multigraph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
A **directed multigraph** is an ordered pair \((V, E)\) where

- \(V\) is a set of **vertices**;
- \(E\) is a set of **edges**;
- \(E\) is a function from \(E\) into \(V \times V\);
Directed (multi-)graphs (Digraphs) (有向图)

A directed multigraph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a set of edges;
- \(\mathcal{E}\) is a function from \(E\) into \(V \times V\);
- an edge \(e\) such that \(\mathcal{E}(e) = (u, v)\) is said to start at \(u\) and end at \(v\).
State Machines

source: from internet
A simple directed graph is an ordered pair \((V, E)\) where
Simple Directed graphs （简单有向图）

A simple directed graph is an ordered pair \((V, E)\) where
A simple directed graph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
A simple directed graph is an ordered pair \((V, E)\) where

- \(V\) is a set of vertices;
- \(E\) is a subset of \(\{(u, v) \in V \times V \mid u \neq v\}\);
Flow Network

source: from internet
Mixed Graphs

A mixed graph is a graph that has both directed and undirected edges.
Graph Terminologies

main textbook, Page 651 – 654
**Adjacency and Incidence**

- $G = (V, E)$: an **undirected** pseudograph

Two vertices $u, v \in V$ are adjacent (or neighbours) in $G$ if there is an edge $e \in E$ such that the endpoints of $e$ are $u, v$.

If the endpoints of an edge $e$ are $u, v$, then $e$ is incident with $u, v$. 

**Neighbourhood**

The neighbourhood $N(v)$ is the set of all neighbours of $v$.

$N(A) := \bigcup_{v \in A} N(v)$ for $A \subseteq V$. 
Graph Terminologies

Adjacency and Incidence

- \( G = (V, E) \): an undirected pseudograph
- Two vertices \( u, v \in V \) are adjacent (or neighbours) in \( G \) if there is an edge \( e \in E \) such that the endpoints of \( e \) are \( u, v \).

Neighbourhood

The neighbourhood \( N(v) \) is the set of all neighbours of \( v \).

\[ N(A) := \bigcup_{v \in A} N(v) \] for \( A \subseteq V \).
Adjacency and Incidence

- $G = (V, E)$: an undirected pseudograph
- Two vertices $u, v \in V$ are adjacent (or neighbours) in $G$ if there is an edge $e \in E$ such that the endpoints of $e$ are $u, v$.
- If the endpoints of an edge $e$ are $u, v$, then $e$ is incident with $u, v$. 
Graph Terminologies

Adjacency and Incidence

- $G = (V, E)$: an undirected pseudograph
- Two vertices $u, v \in V$ are adjacent (or neighbours) in $G$ if there is an edge $e \in E$ such that the endpoints of $e$ are $u, v$.
- If the endpoints of an edge $e$ are $u, v$, then $e$ is incident with $u, v$.

Neighbourhoods

- The neighbourhood $\mathcal{N}(v)$ is the set of all neighbours of $v$.
- $\mathcal{N}(A) := \bigcup_{v \in A} \mathcal{N}(v)$ for $A \subseteq V$. 
Graph Terminologies

Degrees (度)

- $G = (V, E)$: an undirected pseudograph

Isolated and Pendant Vertices

A vertex $v \in V$ is isolated if $\deg(v) = 0$; pendant if $\deg(v) = 1$. 
Degrees (度)

- $G = (V, E)$: an undirected pseudograph
- The degree of a vertex $v \in V$ is the number of edges incident with it, for which a loop associated with $\{v, v\}$ contributes twice to the degree of $v$.
Degrees (度)

- \( G = (V, E) \): an undirected pseudograph

- The degree of a vertex \( v \in V \) is the number of edges incident with it, for which a loop associated with \( \{v, v\} \) contributes twice to the degree of \( v \).

- notation: \( \text{deg}(v) \)
Graph Terminologies

Degrees (度)

- $G = (V, E)$: an undirected pseudograph
- The degree of a vertex $v \in V$ is the number of edges incident with it, for which a loop associated with $\{v, v\}$ contributes twice to the degree of $v$.
- notation: $\text{deg}(v)$

Isolated and Pendant Vertices

A vertex $v$ is . . .

- isolated if $\text{deg}(v) = 0$;
- pendant if $\text{deg}(v) = 1$. 
Graph Terminologies

An Example

On Blackboard
The Handshaking Theorem

- \( G = (V, E) \): a finite undirected pseudograph

Then we have that \( 2 \cdot |E| = \sum_{v \in V} \deg(v) \).
The Handshaking Theorem

- \( G = (V, E) \): a finite undirected pseudograph

Then we have that \( 2 \cdot |E| = \sum_{v \in V} \deg(v) \).

Proof

The summation \( \sum_{v \in V} \deg(v) \) counts every edge exactly twice.
The Handshaking Theorem

- \( G = (V, E) \): a finite undirected pseudograph

Then we have that \( 2 \cdot |E| = \sum_{v \in V} \deg(v) \).

**Proof**

The summation \( \sum_{v \in V} \deg(v) \) counts every edge exactly twice.

**Corollary**

An undirected graph has an even number of vertices of odd degree.
The Handshaking Theorem

- $G = (V, E)$: a finite undirected pseudograph

Then we have that $2 \cdot |E| = \sum_{v \in V} \deg(v)$.

Proof

The summation $\sum_{v \in V} \deg(v)$ counts every edge exactly twice.

Corollary

An undirected graph has an even number of vertices of odd degree.

Exercise

An undirected graph has 10 vertices each with degree 3. How many edges does this graph have?
Directed Edges

- \( G = (V, E) \): a directed multigraph

- If \( e \) is an edge associated with \((u, v)\) in \( E \) then:
  - \( u \) is adjacent to \( v \), and
  - \( v \) is adjacent from \( u \);
  - \( u \) is the initial vertex of the edge, while
  - \( v \) is the terminal (or end) vertex of the edge.
- Observation: If a loop edge associated with \((u, u)\) is in \( E \), then \( u \) is both the initial and end vertex.
Directed Edges

- \( G = (V, E) \): a directed multigraph
- If \( e \) is an edge associated with \((u, v)\) in \( E \) then:
  - \( u \) is adjacent to \( v \), and \( v \) is adjacent from \( u \);
  - \( u \) is the initial vertex of the edge, while \( v \) is the terminal (or end) vertex of the edge.
### Directed Edges

- \( G = (V, E) \): a directed multigraph
- If \( e \) is an edge associated with \((u, v)\) in \( E \) then:
  - \( u \) is adjacent to \( v \), and \( v \) is adjacent from \( u \);
  - \( u \) is the initial vertex of the edge, while \( v \) is the terminal (or end) vertex of the edge.

### Observation

If a loop edge associated with \((u, u)\) is in \( E \), then \( u \) is both the initial and end vertex.
**Graph Terminologies**

### In and Out Degrees

- $G = (V, E)$: a directed multigraph

Then we define:

- \( \text{deg}^{-}(v) := |\{ e \in E | e \text{ is associated with } (u', v') \text{ and } v = v' \}| \) (in-degree);

- \( \text{deg}^{+}(v) := |\{ e \in E | e \text{ is associated with } (u', v') \text{ and } v = u' \}| \) (out-degree).

**Observation**

If a loop associated with \((u, u)\) is present, then it contributes 1 to both the in- and out-degree of $u$. 
### In and Out Degrees

- **G = (V, E):** a directed multigraph
- **v:** a vertex in V

Then we define:

- $\deg^-(v) := |\{e \in E \mid e \text{ is associated with } (u', v') \text{ and } v = v'\}|$ (in-degree);
- $\deg^+(v) := |\{e \in E \mid e \text{ is associated with } (u', v') \text{ and } v = u'\}|$ (out-degree).

**Observation**

If a loop associated with $(u, u)$ is present, then it contributes 1 to both the in- and out-degree of $u$. 
In and Out Degrees

- \( G = (V, E) \): a directed multigraph
- \( v \): a vertex in \( V \)

Then we define:

- \( \text{deg}^- (v) := |\{e \in E \mid e \text{ is associated with } (u', v') \text{ and } v = v'\}| \) (in-degree);
- \( \text{deg}^+ (v) := |\{e \in E \mid e \text{ is associated with } (u', v') \text{ and } v = u'\}| \) (out-degree).

Observation

If a loop associated with \((u, u)\) is present, then it contributes 1 to both the in- and out-degree of \(u\).
Theorem

- \( G = (V, E) \): a finite directed multigraph

Then we have that

\[
|E| = \sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v).
\]
Theorem

- $G = (V, E)$: a finite directed multigraph

Then we have that

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

Exercise

Prove this theorem.
Special Simple Graphs

main textbook, Page 654 – 663
Special Simple Graphs

- complete graphs (完全图)
- cycles
- cubes
- bipartite graphs (二分图)
A complete graph is a simple graph where there is an edge between every two distinct vertices.
A complete graph is a simple graph where there is an edge between every two distinct vertices.

$K_n$: the complete graph with $n$ vertices.
**Complete Graphs**

- A **complete graph** is a simple graph where there is an edge between every two distinct vertices.
- \( K_n \): the complete graph with \( n \) vertices

**Examples**

- \( K_1 \)
- \( K_2 \)
- \( K_3 \)
- \( K_4 \)
- \( K_5 \)
- \( K_6 \)
Special Simple Graphs

Complete Graphs

- **A complete graph** is a simple graph where there is an edge between every two distinct vertices.
- **$K_n$**: the complete graph with $n$ vertices

Examples

```
K_1  K_2  K_3  K_4  K_5  K_6
```

Question

How many edges does a complete graph $K_n$ of $n$ vertices have?
A **cycle** is a graph that consists of

- $n$ vertices $v_1, \ldots, v_n$, and
- $n$ edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.

$C_n$: the cycle with $n$ vertices ($n \geq 3$)
A cycle is a graph that consists of

- $n$ vertices $v_1, \ldots, v_n$, and
- $n$ edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.
- $C_n$: the cycle with $n$ vertices ($n \geq 3$)
A **cycle** is a graph that consists of

- $n$ vertices $v_1, \ldots, v_n$, and
- $n$ edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.
- $C_n$: the cycle with $n$ vertices ($n \geq 3$)

**Examples**

![Cycles](image)

$C_3$, $C_4$, $C_5$, $C_6$
Cubes

An \( n \)-cube is a graph where the vertex set consists of all bit (i.e., \( 0, 1 \)) sequences of length \( n \), and two bit sequences are adjacent iff they differ in exactly one bit position.
Cubes

An $n$-cube is a graph where the vertex set consists of all bit (i.e., 0, 1) sequences of length $n$, and two bit sequences are adjacent iff they differ in exactly one bit position.

- $Q_n$: the $n$-cube ($n \geq 1$)
Special Simple Graphs

Cubes
An $n$-cube is a graph where the vertex set consists of all bit (i.e., 0, 1) sequences of length $n$, and two bit sequences are adjacent iff they differ in exactly one bit position.

- $Q_n$: the $n$-cube ($n \geq 1$)

Examples

- $Q_1$
- $Q_2$
- $Q_3$
Special Simple Graphs

Cubes

An *n*-cube is a graph where the vertex set consists of all bit (i.e., 0, 1) sequences of length \( n \), and two bit sequences are adjacent iff they differ in exactly one bit position.

- \( Q_n \): the *n*-cube \(( n \geq 1 )\)

Examples

![Graphs](image)

Quiz

How many edges does an *n*-cube have?
Special Simple Graphs

Bipartite Graphs

- $G = (V, E)$: a simple graph

Examples
Bipartite Graphs

- \( G = (V, E) \): a simple graph
- \( G \) is a bipartite graph if there is a partition \((V_1, V_2)\) on \( V \) such that every edge connects a vertex from \( V_1 \) and a vertex from \( V_2 \).
- \((V_1, V_2)\): a bipartition of \( G \)
Special Simple Graphs

Bipartite Graphs

- $G = (V, E)$: a simple graph
- $G$ is a bipartite graph if there is a partition $(V_1, V_2)$ on $V$ such that every edge connects a vertex from $V_1$ and a vertex from $V_2$.
- $(V_1, V_2)$: a bipartition of $G$

Examples
A simple graph $G = (V, E)$ is bipartite iff one can assign two different colors to the vertices so that adjacent vertices have different colors.
A bipartite graph $G = (V, E)$ is **complete** if there exists a bipartition $(V_1, V_2)$ for the graph such that for every $u \in V_1$ and $v \in V_2$, $\{u, v\}$ is an edge in $E$.

$K_{m,n}$: the complete bipartite graph with a bipartition of $m$ and $n$ vertices.
A bipartite graph $G = (V, E)$ is complete if there exists a bipartition $(V_1, V_2)$ for the graph such that for every $u \in V_1$ and $v \in V_2$, $\{u, v\}$ is an edge in $E$.

$K_{m,n}$: the complete bipartite graph with a bipartition of $m$ and $n$ vertices.

Examples:

$K_{2,3}$

$K_{3,3}$

$K_{3,5}$

$K_{2,8}$
Special Simple Graphs

Matchings

$n$ boys and $n$ girls are to attend a party. Each boy prefers several girls and each girl prefers several boys. The preference is always between each other. Can the boys and girls be matched so that each pair of matched boy and girl prefer each other?
Matchings (匹配)

- $n$ boys and $n$ girls are to attend a party.
Matchings (匹配)

- $n$ boys and $n$ girls are to attend a party.
- Each boy prefers several girls and each girl prefers several boys.
- The preference is always between each other.
Special Simple Graphs

Matchings (匹配)

- $n$ boys and $n$ girls are to attend a party.
- Each boy prefers several girls and each girl prefers several boys.
- The preference is always between each other.

Can the boys and girls be matched so that each pair of matched boy and girl prefer each other?
Special Simple Graphs

**Matchings**

- $G = (V, E)$: a simple graph
Special Simple Graphs

Matchings

- $G = (V, E)$: a simple graph
- A matching is a subset $M \subseteq E$ of edges such that no two edges in $M$ are incident with the same vertex.
Special Simple Graphs

**Matchings**

- \( G = (V, E) \): a simple graph
- A matching is a subset \( M \subseteq E \) of edges such that no two edges in \( M \) are incident with the same vertex.
- A maximal matching is a matching with the largest number of edges.
### Complete Matchings

- $G = (V, E)$: a bipartite graph with a bipartition $(V_1, V_2)$

A **complete** matching from $V_1$ to $V_2$ is a matching $M$ such that every vertex in $V_1$ is incident with some edge in $M$.

### Remark

- the case that $|V_1| = |V_2|$
Problem

When does a bipartite graph with a bipartition \((V_1, V_2)\) have a complete matching from \(V_1\) to \(V_2\)?
Problem

When does a bipartite graph with a bipartition \((V_1, V_2)\) have a complete matching from \(V_1\) to \(V_2\)?

Some Answers
Problem
When does a bipartite graph with a bipartition \((V_1, V_2)\) have a complete matching from \(V_1\) to \(V_2\)?

Some Answers
- \(|V_1| \leq |\mathcal{N}(V_1)|\)
Problem

When does a bipartite graph with a bipartition \((V_1, V_2)\) have a complete matching from \(V_1\) to \(V_2\)?

Some Answers

- \(|V_1| \leq |\mathcal{N}(V_1)|\)
- \(\forall A \subseteq V_1 (|A| \leq |\mathcal{N}(A)|)\)
Hall’s Marriage Theorem

- $G = (V, E)$: a bipartite graph with a bipartition $(V_1, V_2)$

Then $G$ has a complete matching from $V_1$ to $V_2$ iff $|A| \leq |N(A)|$ for all subsets $A \subseteq V_1$. 

Proof (from [R], Page 659 and [RD], Chapter 2, Page 38)

By induction on $|V_1|$.

base step: $|V_1| = 1$. Straightforward

inductive step:

1. $\forall A \subsetneq V_1 (A \neq \emptyset \rightarrow |A| + 1 \leq |N(A)|)$

2. $\exists A \subsetneq V_1 (A \neq \emptyset \land |A| = |N(A)|)$
Hall’s Marriage Theorem

- \( G = (V, E) \): a bipartite graph with a bipartition \((V_1, V_2)\)

Then \( G \) has a complete matching from \( V_1 \) to \( V_2 \) iff \( |A| \leq |\mathcal{N}(A)| \) for all subsets \( A \subseteq V_1 \).

Proof (from [R], Page 659 and [RD], Chapter 2, Page 38)

By induction on \(|V_1|\).

- base step: \(|V_1| = 1\). Straightforward
- inductive step:
  1. \( \forall A \subsetneq V_1 \,(A \neq \emptyset \rightarrow |A| + 1 \leq |\mathcal{N}(A)|) \)
  2. \( \exists A \subsetneq V_1 \,(A \neq \emptyset \land |A| = |\mathcal{N}(A)|) \)
Special Simple Graphs

Base Step

- $|V_1| = 1$

Inductive Step (a)

$\forall A \subset V_1 (A \neq \emptyset \rightarrow |A| + 1 \leq |N(A)|)$.

Pick an edge \{a, b\} such that $a \in V_1$, $b \in V_2$.

Construct the smaller graph $G'$ by removing the edge \{a, b\}.

Inductive Step (b)

$\exists A \subset V_1 (A \neq \emptyset \land |A| = |N(A)|)$.

Construct the smaller bipartite graph $G''$ by including only the vertices and edges between $A \cup N(A)$.

Construct the smaller bipartite graph $G'$ by removing $G'$ from $G$. 
Special Simple Graphs

**Base Step**
- \(|V_1| = 1\)

**Inductive Step (a)**
- \(\forall A \subsetneq V_1 (A \neq \emptyset \rightarrow |A| + 1 \leq |N(A)|)\).
- Pick an edge \(\{a, b\}\) such that \(a \in V_1, b \in V_2\).
- Construct the smaller graph \(G'\) by removing the edge \(\{a, b\}\).
Special Simple Graphs

Base Step

- $|V_1| = 1$

Inductive Step (a)

- $\forall A \subset V_1 (A \neq \emptyset \rightarrow |A| + 1 \leq |\mathcal{N}(A)|)$.
- Pick an edge $\{a, b\}$ such that $a \in V_1$, $b \in V_2$.
- Construct the smaller graph $G'$ by removing the edge $\{a, b\}$.

Inductive Step (b)

- $\exists A \subset V_1 (A \neq \emptyset \land |A| = |\mathcal{N}(A)|)$.
- Construct the smaller bipartite graph $G'$ by including only the vertices and edges between $A \cup \mathcal{N}(A)$.
- Construct the smaller bipartite graph $G''$ by removing $G'$ from $G$. 
Subgraphs and Graph Operations

main textbook, Page 663 – 665
Subgraphs （子图）

**Definition**

- A subgraph of a graph $G = (V, E)$ is a graph $G' = (V', E')$ such that
  (i) $V' \subseteq V$ and (ii) $E' \subseteq E$.

- A subgraph $G'$ of $G$ is proper if $G \neq G'$.

**Induced Subgraphs**

$G = (V, E)$: a simple graph $W \subseteq V$: a subset of vertices

The subgraph induced by $W$ consists of all the vertices from $W$ and all the edges from $E$ whose endpoints both lie in $W$.
Definition

- A subgraph of a graph \( G = (V, E) \) is a graph \( G' = (V', E') \) such that (i) \( V' \subseteq V \) and (ii) \( E' \subseteq E \).
- A subgraph \( G' \) of \( G \) is proper if \( G \neq G' \).

Induced Subgraphs

- \( G = (V, E) \): a simple graph
- \( W \subseteq V \): a subset of vertices

The subgraph induced by \( W \) consists of all the vertices from \( W \) and all the edges from \( E \) whose endpoints both lie in \( W \).
Edges Removal and Addition

- $G = (V, E)$: a graph
- $e$: a edge (not necessarily in $E$) whose endpoints fall in $V$
Graph Operations

Edges Removal and Addition

- $G = (V, E)$: a graph
- $e$: a edge (not necessarily in $E$) whose endpoints fall in $V$

Then we define that

- $G - e = (V, E \setminus \{e\})$;
- $G + e = (V, E \cup \{e\})$;

Extension

- $F$: a subset of edges whose endpoints fall in $V$ (not necessarily in $E$)
## Edges Removal and Addition

- \( G = (V, E) \): a graph
- \( e \): a edge (not necessarily in \( E \)) whose endpoints fall in \( V \)

Then we define that
- \( G - e = (V, E \setminus \{e\}) \);
- \( G + e = (V, E \cup \{e\}) \);

## Extension

- \( F \): a subset of edges whose endpoints fall in \( V \) (not necessarily in \( E \))

Then we define that
- \( G - F = (V, E \setminus F) \);
- \( G + F = (V, E \cup F) \);
**Vertex Removal**

- \( G = (V, E) \): a graph
- \( v \): a vertex in \( V \)
Vertex Removal

- \( G = (V, E) \): a graph
- \( v \): a vertex in \( V \)

Then we define the graph \( G - v = (V', E') \) by:
- \( V' := V \setminus \{v\} \);
- \( E' := \{e' \in E \mid v \text{ is not an endpoint of } e'\} \).
**Vertex Removal**

- $G = (V, E)$: a graph
- $v$: a vertex in $V$

Then we define the graph $G - v = (V', E')$ by:

- $V' := V \setminus \{v\}$;
- $E' := \{e' \in E \mid v$ is not an endpoint of $e'\}$.

**Extension**

- $G = (V, E)$: a graph
- $V'$: a subset of $V$
**Vertex Removal**

- $G = (V, E)$: a graph
- $v$: a vertex in $V$

Then we define the graph $G - v = (V', E')$ by:

- $V' := V \setminus \{v\}$;
- $E' := \{e' \in E \mid v \text{ is not an endpoint of } e'\}$.

**Extension**

- $G = (V, E)$: a graph
- $V'$: a subset of $V$

Then we define the graph $G - V' = (V'', E'')$ by:

- $V'' := V \setminus V'$;
- $E'' := \{e' \in E \mid e' \text{ does not have an endpoint in } V'\}$. 
Graph Union

- $G_i = (V_i, E_i) \ (i = 1, 2)$: two simple graphs

Then we define the union graph $G_1 \cup G_2$ as the simple graph $(V', E')$:

- $V' := V_1 \cup V_2$;
- $E' := E_1 \cup E_2$. 

Hongfei Fu (SJTU JHC)
Graph Operations

Graph Union

- $G_i = (V_i, E_i) \ (i = 1, 2)$: two simple graphs

Then we define the union graph $G_1 \cup G_2$ as the simple graph $(V', E')$:

- $V' := V_1 \cup V_2$;
- $E' := E_1 \cup E_2$. 
Graph Isomorphism (图同构)

main textbook, Page 671 – 675
Graph Isomorphism

Intuition

Two graphs are isomorphic if they are essentially the same, although their vertices and edges may be organized in a different way.
Graph Isomorphism

**Intuition**

Two graphs are **isomorphic** if they are essentially the same, although their vertices and edges may be organized in a different way.

**Definition**

- $G_i = (V_i, E_i)$ ($i = 1, 2$): two simple graphs

We say that $G_1$, $G_2$ are **isomorphic** if there exists a bijection $f : V_1 \rightarrow V_2$ such that $\forall u, v \in V_1 (\{u, v\} \in E_1 \iff \{f(u), f(v)\} \in E_2)$. 
Graph Isomorphism

Intuition

Two graphs are **isomorphic** if they are essentially the same, although their vertices and edges may be organized in a different way.

Definition

- $G_i = (V_i, E_i) \ (i = 1, 2)$: two simple graphs

We say that $G_1, G_2$ are **isomorphic** if there exists a bijection $f : V_1 \rightarrow V_2$ such that $\forall u, v \in V_1 \ (\{u, v\} \in E_1 \leftrightarrow \{f(u), f(v)\} \in E_2)$.

Example

![Graphs](image.png)
A graph invariant is a graph property that is preserved under graph isomorphism.
Graph Isomorphism

Graph Invariant

A graph invariant is a graph property that is preserved under graph isomorphism.

Examples

- maximal degree
- bipartite graphs
- the existence of triangles
- ...
Summary

- graph models
- graph terminologies
- special simple graphs
- graph operations
- graph isomorphism
Textbooks

- **main textbook:**

- **reference material:**
  《图论与代数结构》，戴一奇，清华大学出版社

- **reference material:**
  Reinhard Diestel, *Graph Theory*, 5th edition [RD]
  http://diestel-graph-theory.com/basic.html?
[R], Page 641 – 675

(optional) 第一章，《图论与代数结构》，戴一奇
Homeworks

Homework on Textbook

- [R], Page 650, Exercise 3, 5, 7, 9
- [R], Page 665, Exercise 1, 2, 8, 21
- [R], Page 666, Exercise 24, 31, 32
- [R], Page 668, Exercise 64
- [R], Page 676, Exercise 36, 37
- [R], Page 677, Exercise 42, 43, 44

Note:

- You don’t need to count the number of edges or vertices in Exercise 1, 2, 8 on Page 665.
- For Exercise 32 on Page 666, You need to assume that there is no complete matching from $A_1$ to $A_2$. 
Homework Submission

- **submission time:** the start of the class on Nov. 19th
- **teaching assistant:**
  - Peixin Wang: peter007008@qq.com
  - Jinyi Wang: jinyi.wang@sjtu.edu.cn
  - Luhua Jin: 1097795310@qq.com
- **submission:**
  - written version (preferred): submit on the desk
  - electronic version: word or pdf version, send email with title

“离散数学+姓名+学号+第九周周六”

to the teaching assistants:
- Students from F1903001, F1903003 and F1903004, send to Luhua Jin.
- Students from F1903801 and F1903802, send to Jinyi Wang.
- All other students please send to Peixin Wang.