Functions and Graphs

CSC 1300 – Discrete Structures

Villanova University

Major Themes

- Functions
  - \( f : A \rightarrow B \)
  - domain/co-domain
  - range
  - one-to-one
  - onto
  - bijection

- Graphs
  - vertices
  - edges
  - path
  - cycle
  - Handshaking theorem
  - Isomorphism

Functions: Basic terminology

A function \( f \) from \( A \) to \( B \) assigns exactly one element of \( B \) to each element of \( A \).

- We write \( f(x) = y \) if the function \( f \) assigns \( y \) to \( x \).
- The range of \( f \) is the set of all images of elements of \( A \).

NB: Range can be smaller than co-domain!

Functions: How many?

Suppose \(|A| = m, |B| = q\).
- How many functions are there from \( A \) to \( B \)? \( q^m \)
- one-to-one functions? \( q(q-1)(q-2)...(q-(m-1)) \)
- onto? this question is much harder, see #24 in Chapter 6
One-to-One Functions

- A function \( f : X \rightarrow Y \) is one-to-one (or injective) iff for each \( y \in Y \) there is at most one \( x \in X \) with \( f(x) = y \)
- Examples:
  \[ \{(1,5), (2,3), (4,5)\} \]
  \( f(x) = x^2 \) for \( x \in \mathbb{Z} \)

Onto Functions

- A function \( f : X \rightarrow Y \) is onto (or surjective) if for each \( y \in Y \) there exists an \( x \in X \) with \( f(x) = y \)
  - (co-domain = range)

\[ \{(1,2), (2,4), (3,6), (4,6)\} \]

\( f(x) = x^2 \) for \( x \in \{(1,0.2) \) and \( y \in \{1,0.2\} \)

\( f(x) = x^2 \) for \( x \in \{(1,0.2) \) and \( y \in \{1,0.4\} \)

Bijections

- A function \( f \) from \( X \) to \( Y \) is a bijection (or a one-to-one correspondence) if it is both injective and surjective.

Example. Let \( f(x) = x+1 \). Is \( f \) a bijection?
- if the domain and codomain are \( \mathbb{N} \) - no
- if the domain and codomain are \( \mathbb{Z} \) - yes
- if the domain and codomain are \( \mathbb{R} \) - yes

Graphs

- Graphs are discrete structures consisting of vertices and edges that connect these vertices.
- Graphs can be used to model:
  - computer systems/networks
  - mathematical relations
  - logic circuit layout
  - jobs/processes

Questions
- isomorphism
- shortest paths
- cycles/paths
- planarity
- coloring
Graphs: Basic Terminology

• A graph is defined as \( G = (V, E) \) with the set of vertices \( V \) and a set of edges \( E \).
• Two vertices \( u \) and \( v \) in an undirected graph \( G \) are adjacent (or neighbors) if \( \{u, v\} \) is an edge of \( G \).
  – The edge \( e \) is said to connect (or to be incident with) \( u \) and \( v \).

\[
G = \{(v_1, v_2, v_3), \{e_1, e_2\}\}
\]

\( e_2 \) connects \( v_2 \) and \( v_3 \)

Directed Graphs

• By definition, the edges of a directed graph are ordered pairs.
• In a directed graph, if we have edge \( e = (u, v) \), then
  – \( u \) is said to be adjacent to \( v \), the terminal vertex
  – \( v \) is said to be adjacent from \( u \), the initial vertex

Types of Graphs

<table>
<thead>
<tr>
<th>Type</th>
<th>Edges</th>
<th>Multiple Edges?</th>
<th>Loops?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple graph</td>
<td>Undirected</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Multigraph</td>
<td>Undirected</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Pseudograph</td>
<td>Undirected</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Directed graph</td>
<td>Directed</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Directed multigraph</td>
<td>Directed</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Degree of a vertex

• The degree of a vertex \( v \) is the number of edges incident on \( v \).
  – Denoted \( \text{deg}(v) \)
  – A loop on \( v \) contributes 2 to the \( \text{deg}(v) \)

• Example:

\[
\text{deg}(a) = 3 \\
\text{deg}(b) = 1 \\
\text{deg}(c) = 5 \\
\text{deg}(d) = 0 \\
\text{deg}(e) = 1
\]
**Handshaking Lemma**

In an undirected graph, the sum of the degrees of the vertices is twice the number of edges. Therefore, the sum of the degrees of all the vertices is even.

\[ \sum_{v \in V} \deg(v) = 2|E|. \]

- **Corollary**: An undirected graph has an even number of vertices of odd degree.

**Path**

- A **path** begins at vertex \( v_0 \), follows an edge \( e_1 \) to \( v_1 \), follows another edge to \( v_2 \) ...
  - A path is represented without edges (especially when there are no parallel edges)
    - \( (v_0, v_1, v_2, \ldots, v_n) \)
  - Said to be of length \( n \)
    - A path on a vertex itself is of length 0
  - A **simple path** from \( v_0 \) to \( v_n \) is a path with no repeated edges.

**Cycle**

- A **cycle** (or **circuit**) is a path of nonzero length from \( v \) to \( v \).

- A **simple circuit** is a circuit from \( v \) to \( v \) with no repeated edges

**Connected Graph**

- A graph \( G \) is connected if given any vertices \( v_1 \) and \( v_2 \) in \( G \), there is a path from \( v_1 \) to \( v_2 \).

- **Tree**: a connected graph with no cycles
**Bipartite Graph**

- A simple graph is called **bipartite** if its vertex set $V$ can be partitioned into 2 disjoint sets $V_1$ and $V_2$ such that every edge in the graph connects a vertex in $V_1$ to a vertex in $V_2$.

**Special Graphs**

- **Complete Graph** - $K_n$
  - Simple graph that contains exactly 1 edge between each pair of $n$ distinct vertices.
- **Complete Bipartite Graph** – $K_{n,m}$
  - Simple graph that contains exactly 1 edge between each pair of $n$ distinct vertices to $m$ distinct vertices.
- **Cycles** – $C_n$
  - For $n \geq 3$, consists of $n$ vertices and edges $(v_1,v_2), (v_2,v_3) \ldots (v_{n-1},v_n)$
- **Wheels** – $W_n$
  - Cycle $C_n$ with an additional vertex added and an edge from the new vertex to each of the $n$ vertices

**Representation of Graphs**

- Adjacency list
  - $V = \{a, b, c, d, e\}$,
  - $\{\{a\}, \{b, b\}, \{c, d\}, \{a, e\}, \{e, d\}, \{a, e\}\}$
- Adjacency matrix
  - $\begin{pmatrix}
  0 & 1 & 0 & 1 & 0 \\
  1 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 1 & 0 \\
  0 & 0 & 1 & 0 & 0 \\
  1 & 0 & 0 & 0 & 0 
  \end{pmatrix}$

**Isomorphism...**

- It means two graphs are essentially the same (maybe drawn differently and re-labeled)
  - Same number of vertices
  - Same number of edges
  - Same degree sequence

**3.8 Problem 3:** Are these essentially the same graph?

- 3.8 Problem 3: are these essentially the same graph?
Isomorphism

• Graphs are **isomorphic** if they have the same structure
  – 1-to-1 mapping of vertices that preserves edges

![Graph Isomorphism Example](image)

Isomorphic graphs will have the same adjacency matrix under some reordering of the vertices

![Adjacency Matrix Example](image)

Special Graphs

• Connected, simple graph
  – Cycle $C_n$
  – Wheels $W_n$
  – Complete graph $K_n$
  – The Complement Graph $\bar{G}$
  – A Complete Bipartite $K_{n,m}$
  – Tree: a connected graph with no cycles

![Special Graphs Examples](image)

More Graph Terminology

Let $G = (V, E)$

- The **complement** $\bar{G} = (V, E')$
  - where $E' = \{(u,v) \mid u \in V$ and $v \in V$ and $\{u,v\} \notin E\}$

- A graph $G' = (V', E')$ is a **subgraph** of $G$ iff $V' \subseteq V$ and $E' \subseteq E$

- A **connected component** of a graph is a subgraph that is connected.

**Tree**: a connected graph with no cycles

**Forest**: graph with no cycles

![More Graph Terminology Diagram](image)