Let us switch to a new topic:

Graphs

Introduction to Graphs

Definition: A simple graph \( G = (V, E) \) consists of a nonempty set \( V \) of vertices, and \( E \), a set of unordered pairs of distinct elements of \( V \) called edges.

A simple graph is just like a directed graph, but with no specified direction of its edges.

Sometimes we want to model multiple connections between vertices, which is impossible using simple graphs.

In these cases, we have to use multigraphs.

Definition: A multigraph \( G = (V, E) \) consists of a set \( V \) of vertices, a set \( E \) of edges, and a function \( f \) from \( E \) to \( \{ \{u, v\} | u, v \in V, u \neq v\} \).

The edges \( e_1 \) and \( e_2 \) are called multiple or parallel edges if \( f(e_1) = f(e_2) \).

Note:
- Edges in multigraphs are not necessarily defined as pairs, but can be of any type.
- No loops are allowed in multigraphs (\( u \neq v \)).

Example: A multigraph \( G \) with vertices \( V = \{a, b, c, d\} \), edges \( \{1, 2, 3, 4, 5\} \) and function \( f \) with \( f(1) = \{a, b\} \), \( f(2) = \{a, b\} \), \( f(3) = \{b, c\} \), \( f(4) = \{c, d\} \) and \( f(5) = \{c, d\} \):

```
1  2  3  4  5
A B C D
```

If we want to define loops, we need the following type of graph:

Definition: A pseudograph \( G = (V, E) \) consists of a set \( V \) of vertices, a set \( E \) of edges, and a function \( f \) from \( E \) to \( \{ \{u, v\} | u, v \in V\} \).

An edge \( e \) is a loop if \( f(e) = \{u, u\} \) for some \( u \in V \).

Here is a type of graph that we already know:

Definition: A directed graph \( G = (V, E) \) consists of a set \( V \) of vertices and a set \( E \) of edges that are ordered pairs of elements in \( V \).

… leading to a new type of graph:

Definition: A directed multigraph \( G = (V, E) \) consists of a set \( V \) of vertices, a set \( E \) of edges, and a function \( f \) from \( E \) to \( \{(u, v) | u, v \in V\} \).

The edges \( e_1 \) and \( e_2 \) are called multiple edges if \( f(e_1) = f(e_2) \).
Introduction to Graphs

Example: A directed multigraph $G$ with vertices $V = \{a, b, c, d\}$, edges $\{1, 2, 3, 4, 5\}$ and function $f$ with $f(1) = (a, b)$, $f(2) = (b, a)$, $f(3) = (c, b)$, $f(4) = (c, d)$ and $f(5) = (c, d)$:

```
\[ a \rightarrow 1 \rightarrow b \rightarrow 2 \rightarrow a \]
\[ b \rightarrow 3 \rightarrow c \rightarrow 4 \rightarrow d \]
\[ c \rightarrow 5 \rightarrow c \]
```

Types of Graphs and Their Properties

<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>dir. multigraph</td>
<td>directed</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Graph Models

Example I: How can we represent a network of (bi-directional) railways connecting a set of cities? We should use a **simple graph** with an edge $\{a, b\}$ indicating a direct train connection between cities $a$ and $b$.

```
New York  --  Boston
    \|    \|    \|    \|    \|
    \|    \|    \|    \|    \|
Toronto  \|    Maple Leafs
```

Example II: In a round-robin tournament, each team plays against each other team exactly once. How can we represent the results of the tournament (which team beats which other team)? We should use a **directed graph** with an edge $(a, b)$ indicating that team $a$ beats team $b$.

```
Bruins  --  Penguins
    \|    \|    \|    \|    \|
    \|    \|    \|    \|    \|
Maple Leafs  \|    Lübeck Giants
```

Graph Terminology

**Definition:** Two vertices $u$ and $v$ in an undirected graph $G$ are called **adjacent** (or **neighbors** in $G$ if $\{u, v\}$ is an edge in $G$.

If $e = \{u, v\}$, the edge $e$ is called **incident with** the vertices $u$ and $v$. The edge $e$ is also said to **connect** $u$ and $v$.

The vertices $u$ and $v$ are called **endpoints** of the edge $\{u, v\}$.

**Definition:** The **degree** of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.

In other words, you can determine the degree of a vertex in a displayed graph by **counting the lines** that touch it.

The degree of the vertex $v$ is denoted by $\text{deg}(v)$.
Graph Terminology

A vertex of degree 0 is called isolated, since it is not adjacent to any vertex.

Note: A vertex with a loop at it has at least degree 2 and, by definition, is not isolated, even if it is not adjacent to any other vertex.

A vertex of degree 1 is called pendant. It is adjacent to exactly one other vertex.

Graph Terminology

Example: Which vertices in the following graph are isolated, which are pendant, and what is the maximum degree? What type of graph is it?

Solution: Vertex f is isolated, and vertices a, d, and j are pendant. The maximum degree is $\deg(g) = 5$. This graph is a pseudograph (undirected, loops).

Graph Terminology

Let us look at the same graph again and determine the number of its edges and the sum of the degrees of all its vertices:

Result: There are 9 edges, and the sum of all degrees is 18. This is easy to explain: Each new edge increases the sum of degrees by exactly two.

Graph Terminology

The Handshaking Theorem: Let $G = (V, E)$ be an undirected graph with $e$ edges. Then

$$2e = \sum_{v \in V} \deg(v)$$

Note: This theorem holds even if multiple edges and/or loops are present.

Example: How many edges are there in a graph with 10 vertices, each of degree 6?

Solution: The sum of the degrees of the vertices is $6 \cdot 10 = 60$. According to the Handshaking Theorem, it follows that $2e = 60$, so there are 30 edges.

Graph Theorems

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

Idea: There are three possibilities for adding an edge to connect two vertices in the graph:

Before:
- Both vertices have even degree
- Both vertices have odd degree
- One vertex has odd degree, the other even

After:
- Both vertices have even degree
- Both vertices have odd degree
- One vertex has even degree, the other odd

Graph Theorems

There are two possibilities for adding a loop to a vertex in the graph:

Before:
- The vertex has even degree
- The vertex has odd degree

After:
- The vertex has even degree
- The vertex has odd degree
Graph Terminology

So if there is an even number of vertices of odd degree in the graph, it will still be even after adding an edge.

Therefore, since an undirected graph with no edges has an even number of vertices with odd degree (zero), the same must be true for any undirected graph.

Graph Terminology

Definition: When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to v, and v is said to be adjacent from u.

The vertex u is called the initial vertex of (u, v), and v is called the terminal vertex of (u, v).

The initial vertex and terminal vertex of a loop are the same.

Graph Terminology

Definition: When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to v, and v is said to be adjacent from u.

The vertex u is called the initial vertex of (u, v), and v is called the terminal vertex of (u, v).

Graph Terminology

Definition: In a graph with directed edges, the in-degree of a vertex v, denoted by \( \deg^-(v) \), is the number of edges with v as their terminal vertex. The out-degree of v, denoted by \( \deg^+(v) \), is the number of edges with v as their initial vertex.

Question: How does adding a loop to a vertex change the in-degree and out-degree of that vertex?

Answer: It increases both the in-degree and the out-degree by one.

Graph Terminology

Theorem: Let G = (V, E) be a graph with directed edges. Then:

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

This is easy to see, because every new edge increases both the sum of in-degrees and the sum of out-degrees by one.

Special Graphs

Definition: The complete graph on n vertices, denoted by \( K_n \), is the simple graph that contains exactly one edge between each pair of distinct vertices.

K₁, K₂, K₃, K₄, K₅
Special Graphs

**Definition:** The cycle $C_n$, $n \geq 3$, consists of $n$ vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}.$

![C3, C4, C5, C6](image)

**Definition:** We obtain the wheel $W_n$ when we add an additional vertex to the cycle $C_n$, for $n \geq 3$, and connect this new vertex to each of the $n$ vertices in $C_n$ by adding new edges.

![W3, W4, W5, W6](image)

**Definition:** The n-cube, denoted by $Q_n$, is the graph that has vertices representing the $2^n$ bit strings of length $n$. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position.

![Q1, Q2, Q3](image)

**Definition:** A simple graph is called bipartite if its vertex set $V$ can be partitioned into two disjoint nonempty sets $V_1$ and $V_2$ such that every edge in the graph connects a vertex in $V_1$ with a vertex in $V_2$ (so that no edge in $G$ connects either two vertices in $V_1$ or two vertices in $V_2$).

**Example I:** Is $C_3$ bipartite?

No, because there is no way to partition the vertices into two sets so that there are no edges with both endpoints in the same set.

**Example II:** Is $C_6$ bipartite?

Yes, because we can display $C_6$ like this: