Overview

- HW6 due & HW7 assignment will be online!
- HW7 is due in 4 weeks on Nov 30 Tuesday
- Last Lecture: Completed Algorithms
  - Big-O notation of Growth of Complexity Function
  - Tractable vs Intractable Problems
  - Computability theory
  - Number theory: Division, Primes, GCD, LCM
  - Euclidean Algorithm for GCD
  - Modular Arithmetic: Properties of Congruence
- This Lecture: Graph Theory
  - Introduction of Graphs
  - Graph theory

Chapter 6.1 Theory and Applications of Graphs

- Coding
- Intuitive
- Theory
- Matrix
- Spreadsheet
- Visual
- Sets
Introduction to Graphs

1. Definition: A simple graph $G = (V, E)$ consists of vertex set $V$, a nonempty set of vertices/nodes, and edge set $E$, a set of edges or unordered pairs of vertices. So each edge $e \in E$ is a set; $e = \{u, v\}$ where $u, v \in V$. An edge $e$ is a self-loop if $e = \{u, u\}$ for some $u \in V$.

2. Definition: A undirected graph is a simple graph with no self loops and there is at most one edge between two vertices (multigraph may have self-loops and multi-edges in between two vertices).

3. Definition: A directed graph $G = (V, E)$ consists of a vertex set $V$ and an edge set $E$. Edges are ordered pairs or arrows of elements in $V$. For each $e \in E$, $e = (u, v)$ where $u, v \in V$.

In this chapter, we assume

“no multi-edges” unless we clearly specify “with multi-edges”

graphs mean undirected graphs

Graph Models

• Example I: How can we represent a network of (bi-directional) railways connecting a set of cities?

• We should use an undirected graph with an edge $\{a, b\}$ indicating a direct train connection between cities $a$ and $b$.
• 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 wins over which other team)?

• We should use a directed graph with an edge \((a, b)\) indicating that team \(a\) wins over team \(b\).

\[
G = (V, E) \quad \text{Note: directions of edges}
\]

\[
V = \{A, B, C, D\}
\]

\[
E = \{(A, C), (B, C), (D, A), (D, B), (D, C)\}
\]

**Graph Terminology (undirected graphs)**

• 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.

• A vertex of degree 0 is called **isolated**, since it is not adjacent to any vertex.
• **The Handshaking Theorem**: Let $G = (V, E)$ be an undirected graph with $|E|$ edges. Then $2|E| = \sum_{v \in V} \deg(v) \quad // \text{sum of degrees}$

• 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.

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

Proof: Let $V_1$ and $V_2$ be the set of vertices of even and odd degrees, respectively. Then by Handshaking theorem

$$2|E| = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v)$$

Since both $2|E|$ and $\sum_{v \in V_1} \deg(v)$ are even, $\sum_{v \in V_2} \deg(v)$ must be even $\Rightarrow |V_2|$ is even.

And for directed graphs:

- **Definition**: When $(u, v)$ is an edge of the directed graph $G$, $u$ 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.

- **Definition**: In a directed graph, the **in-degree** of a vertex $v$ is the number of edges with $v$ as their terminal vertex. The **out-degree** of $v$ 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.
• **Example:** What are the in-degrees and out-degrees of the vertices $a, b, c, d$ in this graph:

- $\text{deg}^\text{in}(a) = 1$
- $\text{deg}^\text{out}(a) = 2$
- $\text{deg}^\text{in}(b) = 4$
- $\text{deg}^\text{out}(b) = 2$
- $\text{deg}^\text{in}(c) = 0$
- $\text{deg}^\text{out}(c) = 2$
- $\text{deg}^\text{in}(d) = 2$
- $\text{deg}^\text{out}(d) = 1$

• **Theorem:** Let $G = (V, E)$ be a directed graph. Then: $\sum_{v \in V} \text{deg}^\text{in}(v) = \sum_{v \in V} \text{deg}^\text{out}(v) = |E|$

• 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 every pair of distinct vertices.
• **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.**

• **Definition:** A 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 $V_2$).

For example, consider a graph that represents mating partners of penguin in a colony. (March of Penguins!) This graph is **bipartite**, because the vertex set can be partitioned into female set and male set then each edge connects a vertex between the two sets.

**Example I:** Is graph $G$ below 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 graph $G$ below bipartite?

Yes, because we can display $G$ like this:
• **Definition:** The complete bipartite graph $K_{m,n}$ is the graph that has its vertex set partitioned into two subsets of $m$ and $n$ vertices, respectively. Two vertices are connected if and only if they are in different subsets.

\[ K_{3,2} \quad \quad \quad K_{3,4} \]

**Neural Network!**

---

### Operations on Graphs

- **Definition:** A subgraph of a graph $G = (V, E)$ is a graph $H = (W, F)$ where $W \subseteq V$ and $F \subseteq E$.

- **Note:** $H$ must be a valid graph. i.e. For each $\{u, v\}$ in $F$, $u$ and $v$ are vertices in $W$.

- **Example:**

\[ K_5 \quad \quad \quad \text{subgraph of } K_5 \]
• **Definition:** The union of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. 

$\{(v_1, \hat{v}_1), (v_2, \hat{v}_2)\}$

• The union of $G_1$ and $G_2$ is denoted by $G_1 \cup G_2$.

$G_1 \cup G_2 = K_5$

---

**Representing Graphs**

• **Definition:** Let $G = (V, E)$ be a graph with $|V| = n$. Suppose that the vertices of $G$ are listed in arbitrary order as $v_1, v_2, \ldots, v_n$.

• The adjacency matrix $A_G$ of $G$, with respect to this listing of the vertices, is the $n \times n$ zero-one-valued matrix with 1 as its $(i, j)$-th entry when $v_i$ and $v_j$ are adjacent, and 0 otherwise.

• In other words, for an adjacency matrix $A = [a_{ij}]$, $a_{ij} = 1$ if $\{v_i, v_j\}$ is an edge of $G$, $a_{ij} = 0$ otherwise.

• **Example:** What is the adjacency matrix $A_G$ for the following graph $G$ based on the order of vertices $a, b, c, d$?

**Solution:**

$$A_G = \begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}$$

$G = (V, E)$

$V = \{a, b, c, d\}$

$E = \{(a,b), (a,c), (a,d), (b,c), (b,d), (c,d)\}$
Definition: **Adjacency list** representation consists of a list of its vertices together with a separate list for each vertex that contains all the vertices adjacent to that vertex.

Example: let $1 = a$, $2 = b$, $3 = c$ and $4 = d$

<table>
<thead>
<tr>
<th>Vertex</th>
<th>List of adjacencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 3, 4</td>
</tr>
<tr>
<td>2</td>
<td>1, 4</td>
</tr>
<tr>
<td>3</td>
<td>1, 4</td>
</tr>
<tr>
<td>4</td>
<td>1, 2, 3</td>
</tr>
</tbody>
</table>

Note: Adjacency Matrix and Adjacency Lists can be used to defined directed graphs also.