CS062
DATA STRUCTURES AND ADVANCED PROGRAMMING

33: Intro to Undirected Graphs

Alexandra Papoutsaki
Lectures

Mark Kampe
Labs

GRAPHS
Lecture 33: Intro to Undirected Graphs

- Undirected Graphs

Some slides adopted from Algorithms 4th Edition or COS226
Graphs

- **Graphs**: mathematical abstractions that model a set of vertices connected pairwise by edges.

- **Why study graphs?**
  - Thousands of practical applications.
  - Hundreds of graph algorithms.
  - Interesting and widely applicable abstraction.
  - Core branch of computer science and discrete math.
Example: (Fake) LA subway map

- **Vertices:** stations.
- **Edges:** route.

Source: LA Weekly
Example: Social networks

- **Vertices**: people. **Edges**: “friendships”.

Source: Paul Butler
Example: Protein-protein networks

- **Vertices**: proteins.
- **Edges**: interactions.

Source: Macmillan Magazines Ltd.
# Graph Applications

<table>
<thead>
<tr>
<th>Graph</th>
<th>Vertex</th>
<th>Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Telephone, computer</td>
<td>Cable</td>
</tr>
<tr>
<td>Circuit</td>
<td>Gate, register, processor</td>
<td>Wire</td>
</tr>
<tr>
<td>Financial</td>
<td>Stock</td>
<td>Transaction</td>
</tr>
<tr>
<td>Transportation</td>
<td>Intersection</td>
<td>Street</td>
</tr>
<tr>
<td>Game</td>
<td>Board</td>
<td>Legal move</td>
</tr>
<tr>
<td>Neural network</td>
<td>Neuron</td>
<td>Synapse</td>
</tr>
<tr>
<td>Molecule</td>
<td>Atom</td>
<td>Bond</td>
</tr>
<tr>
<td>Schedule</td>
<td>Job</td>
<td>Constraint</td>
</tr>
</tbody>
</table>
Graph Terminology

- **Graph**: set of vertices \( V \) connected pairwise by a set of edges \( E \).
  - E.g., \( V = \{A, B, C, D\} \), \( E = \{\{A,B\}, \{A,C\}, \{A,D\}, \{B,D\}\} \).

- **Path**: sequence of vertices connected by edges, with no repeated edges.
  - A simple path is a path with no repeated vertices.

- **Cycle**: Path with at least one edge whose first and last vertices are the same.
  - A simple cycle is a cycle with no repeated vertices (other than the first and last).

- The length of a cycle or a path is its number of edges.
Graph Terminology

- **Self-loop**: an edge that connects a vertex to itself.
- Two vertices are **connected** if there is a path between them.
- Two edges are **parallel** if they connect the same pair of vertices.
- When an edge connects two vertices, we say that the vertices are **adjacent** to one another and that the edge is **incident** on both vertices.
- The **degree** of a vertex is the number of edges incident on it.
- A **subgraph** of a graph is a subset of a graph’s edges and their associated vertices.
Graph Terminology

- A graph is **connected** if there is a path from every vertex to every other vertex.

- A graph that is not connected consists of a set of connected components, which are maximal connected subgraphs.

- An **acyclic** graph is a graph with no cycles.

- A **tree** is an acyclic connected graph.

- A **forest** is a disjoint set of trees.
Graph Terminology

- Anatomy of a graph
- Acyclic
- 19 vertices
- 18 edges
- Connected
- Cycle of length 5
- Vertex of degree 3
- Path of length 4
- Connected components
- A tree
Popular graph problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s-t path</td>
<td>Is there a path between s and t?</td>
</tr>
<tr>
<td>Shortest s-t path</td>
<td>What is the shortest path between s and t?</td>
</tr>
<tr>
<td>Cycle</td>
<td>Is there a cycle in the graph?</td>
</tr>
<tr>
<td>Euler cycle</td>
<td>Is there a cycle that uses each edge exactly once?</td>
</tr>
<tr>
<td>Hamilton cycle</td>
<td>Is there a cycle that uses each vertex exactly once?</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Is there a path between every pair of vertices?</td>
</tr>
<tr>
<td>Biconnectivity</td>
<td>Is there an vertex whose removal disconnects the graph?</td>
</tr>
</tbody>
</table>
Lecture 33: Intro to Undirected Graphs

- Undirected Graphs
Readings:

- Textbook: Chapter 4.1 (Pages 515-521)
- Website:
  - https://algs4.cs.princeton.edu/41graph/