What Is a Graph?

Graphs, Part I
What Is a Graph?

- Set of nodes (or vertices)
- Set of edges between pairs of nodes
LOTS of situations can be viewed as graphs.
→ Any situation that involves relationships between pairs of things.

Examples:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message routing in network</td>
<td>computers, cities, people</td>
<td>wires, highways</td>
</tr>
<tr>
<td>Transportation</td>
<td></td>
<td>friendships</td>
</tr>
<tr>
<td>Social relationships</td>
<td></td>
<td>friendships</td>
</tr>
<tr>
<td>Exam scheduling</td>
<td>exams, courses</td>
<td>shared student</td>
</tr>
<tr>
<td>Planning courses to take</td>
<td>board configurations</td>
<td>prerequisites</td>
</tr>
<tr>
<td>Chess</td>
<td></td>
<td>moves</td>
</tr>
</tbody>
</table>
Types of Graphs

Undirected

Directed

Graphs, Part I
Types of Graphs

Unweighted

Weighted
A graph can be represented in a computer as an *adjacency matrix* of boolean values.

(1 = true and 0 = false.)
Create an $n \times n$ two-dimensional array of booleans:

```java
boolean A[][] = new boolean[n][n]
```

To access an entry representing the edge from node $i$ to node $j$, use $A[i][j]$.

Remember that array indices will be $0..n-1$.

If you want to number the nodes from 1 to $n$, create an $(n+1) \times (n+1)$ array and just don’t use index 0.
A graph can be represented in a computer as an adjacency matrix of boolean values.

(We treat each edge as if it goes in both directions.)
A *weighted* graph can be represented in a computer as an adjacency matrix of weights.

```
  1 2 3 4 5
F 0 6 0 0 2
R 0 0 5 0 1
O 0 0 0 3 0
M 0 2 0 0 0
5 3 0 0 4 0
```
When to Use Adjacency Matrix

Most of the edges are present in the graph.

If graph has fewer edges, use adjacency lists (next week).

Note: If graph has 1000 nodes, adjacency matrix uses about 1 MB of memory.
Another flavour of graph is the bipartite graph.

Useful for situations where there are two *types* of things. Relationships are between things of opposite types.
When two people dance together, one leads and the other follows. At a dance, there are 5 leaders and 4 followers. The graph indicates which pairs of people are willing to dance with each other.
Bipartite graphs can also be represented as an adjacency matrix.

\[
\begin{align*}
&\text{LEFT} \\
&\text{1} \\
&\text{2} \\
&\text{3} \\
&\text{4} \\
&\text{5} \\
&\text{RIGHT} \\
&\begin{pmatrix}
1 & 2 & 3 & 4 \\
L & 1 & 0 & 0 & 0 \\
E & 2 & 0 & 1 & 0 & 0 \\
F & 3 & 1 & 1 & 1 & 0 \\
T & 4 & 0 & 0 & 1 & 0 \\
5 & 0 & 1 & 0 & 1 & 0
\end{pmatrix}
\]

Graphs, Part I
Some Basic Graph Terminology

- **Path**: sequence of nodes $v_1, v_2, \ldots, v_k$ where $v_i \rightarrow v_{i+1}$ is an edge for all $i$.

- **Cycle**: Path that starts and ends at the same node.
Some Basic Graph Terminology

- **Degree of a node** $v$: number of undirected edges that connect $v$ to other nodes.

- **In-degree of a node** $v$: number of directed edges that point to $v$.

- **Out-degree of a node** $v$: number of directed edges that point out of $v$.

In-degree of $v$ is 2
Out-degree of $v$ is 3

Degree of $v$ is 4