Linear Graph Theory

1. A Branch of Topology
   focus on connections

2. Graphs
   nodes, edges, matrices
   degree of a node
   paths
   components

3. Directed Graphs
   nodes, edges
   indegree, outdegree
   paths and semi-paths
   n-connectedness
   cyclomatic number
   strong and weak components
   Directed Acyclic Graphs (DAG)

4. Program Graphs
   DD-Paths
   short form
Linear Graph Theory

• A Branch of Topology (focus on connections)
• Graphs
  – nodes, edges, matrices
  – degree of a node
  – Paths
  – Components
• Directed Graphs
  – nodes, edges
  – indegree, outdegree
  – paths and semi-paths
  – n-connectedness
  – cyclomatic number
  – strong and weak components
  – Program Graphs
Linear Graph

1. A graph $G = (V, E)$ consists of:

   a finite set $V = \{ n(1), n(2), ..., n(k) \}$ of nodes (or vertices), and
   a set $E = \{ (n(i), n(j)) \}$ where $i \neq j$ and $1 \leq i, j \leq k$ of edges.

In English, a graph is a set of nodes and a set of edges that connect pairs of nodes.

$$V = \{ n1, n2, n3, n4, n5, n6, n7 \}$$

$$E = \{ e1, e2, e3, e4, e5 \}$$

$$= \{(n1, n2), (n1, n4), (n3, n4), (n2, n5), (n4, n6)\}$$
2. The degree of a node in a graph is the number of distinct edges that have the node as an endpoint. We write

$$\text{deg}(n)$$

```
\begin{align*}
\text{deg}(n_1) &= 2 \\
\text{deg}(n_2) &= 2 \\
\text{deg}(n_3) &= 1 \\
\text{deg}(n_4) &= 3 \\
\text{deg}(n_5) &= 1 \\
\text{deg}(n_6) &= 1 \\
\text{deg}(n_7) &= 0
\end{align*}
```
Incidence Matrix

3. The incidence matrix of a graph \( G = (V, E) \) with \( m \) nodes and \( n \) edges is an \( m \times n \) matrix, where the element in row \( i \), column \( j \) is a 1 if and only if node \( i \) is an endpoint of edge \( j \), otherwise the element is 0.

The row sum is the degree of the node, and every column sum must be 2.

Why must every column sum be 2?

\[
\begin{array}{cccccc}
& e_1 & e_2 & e_3 & e_4 & e_5 \\
n_1 & 1 & 1 & 0 & 0 & 0 \\
n_2 & 1 & 0 & 0 & 1 & 0 \\
n_3 & 0 & 0 & 1 & 0 & 0 \\
n_4 & 0 & 1 & 1 & 0 & 1 \\
n_5 & 0 & 0 & 0 & 1 & 0 \\
n_6 & 0 & 0 & 0 & 0 & 1 \\
n_7 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
## Adjacency Matrix

4. The adjacency matrix of a graph $G = (V, E)$ with $m$ nodes is an $m \times m$ matrix, where the element in row $i$, column $j$ is a 1 if and only if there is an edge between node $i$ and node $j$, otherwise the element is 0.

The adjacency matrix is symmetric (element $i,j$ always equals element $j,i$). A row sum is the degree of the node.

What can we say about column sums?

```
<table>
<thead>
<tr>
<th></th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
<th>n5</th>
<th>n6</th>
<th>n7</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>n5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
5. A path is a sequence of edges such that, for any adjacent pair of edges $e(i)$, $e(j)$ in the sequence, the edges share a common (node) endpoint.

Paths can be described either as sequences of edges, or as sequences of nodes; node sequences is the more common choice.

Some paths:
- between $n1$ and $n5$
- between $n6$ and $n5$
- between $n3$ and $n2$
- between $n4$ and $n6$
6. Nodes $n(i)$ and $n(j)$ are connected if and only if they are in the same path.

"Connectedness" is an equivalence relation (the reflexive part comes from the convention that every node is connected to itself). Recall that an equivalence relation imposes a partition on the set on which it is defined.

7. A component of a graph is a maximal set of connected nodes; nodes in the equivalence classes are components of the graph.

Components:

\{n1, n2, n3, n4, n5, n6\}

\{n7\}
8. Given a graph $G = (V, E)$, its condensation graph is formed by replacing each component by a condensing node.

There are no paths in a condensation graph of an ordinary (undirected) graph. Why?

Components:

$S_1 = \{n_1, n_2, n_3, n_4, n_5, n_6\}$

$S_2 = \{n_7\}$

Condensation Graph
9. A directed graph (or digraph) $D = (V, E)$ consists of:

- A finite set $V = \{ n(1), n(2), ..., n(k) \}$ of nodes (or vertices), and
- A set $E = \{ <n(i), n(j)> \}$, where $i \neq j$ and $1 \leq i, j \leq k$ of edges.

Notice that the (undirected) edges of an ordinary graph become ordered pairs of nodes. We say that a directed edge goes from its starting node to its finish node.

\[ V = \{ n_1, n_2, n_3, n_4, n_5, n_6, n_7 \} \]

\[ E = \{ e_1, e_2, e_3, e_4, e_5 \} \]

\[ = \{ <n_1, n_2>, <n_1, n_4>, <n_3, n_4>, <n_2, n_5>, <n_4, n_6> \} \]
Indegrees and Outdegrees

10. The indegree of a node in a directed graph is the number of distinct edges that have the node as an endpoint. We write \( \text{indeg}(n) \)

11. The outdegree of a node in a directed graph is the number of distinct edges that have the node as an start point. We write \( \text{outdeg}(n) \)

\[
\begin{align*}
\text{indeg}(n1) &= 0 & \text{outdeg}(n1) &= 2 \\
\text{indeg}(n2) &= 1 & \text{outdeg}(n2) &= 1 \\
\text{indeg}(n3) &= 0 & \text{outdeg}(n3) &= 1 \\
\text{indeg}(n4) &= 2 & \text{outdeg}(n4) &= 1 \\
\text{indeg}(n5) &= 1 & \text{outdeg}(n5) &= 0 \\
\text{indeg}(n6) &= 1 & \text{outdeg}(n6) &= 0 \\
\text{indeg}(n7) &= 0 & \text{outdeg}(n7) &= 0
\end{align*}
\]
Types of Nodes

12. A node with indegree = 0 is a source node.
13. A node with outdegree = 0 is a sink node.
14. A node with indegree ≠ 0 and outdegree ≠ 0 is a transfer node. (AKA an interior node)

n1, n3 and n7 are source nodes, n5, n6 and n7 are sink nodes, n2 and n4 are transfer nodes.
15. The adjacency matrix of a directed graph $D = (V, E)$ with $m$ nodes is an $m \times m$ matrix $A = (a(i, j))$, where $a(i, j)$ is a 1 if and only if there is an edge from node $i$ and node $j$, otherwise the element is 0.

The adjacency matrix of a directed graph is not necessarily symmetric. A row sum is the outdegree of the node; a column sum is the indegree of a node.
16. A (directed) path is a sequence of edges such that, for any adjacent pair of edges $e(i)$, $e(j)$ in the sequence, the end node of the first edge is the start node of the second edge.

17. A (directed) semi-path is a sequence of edges such that, for at least one adjacent pair of edges $e(i)$, $e(j)$ in the sequence, the start node of the first edge is the start node of the second edge or the end node of the first edge is the end node of the second edge.

There is a path from $n_1$ to $n_6$, there is a semi-path between $n_1$ and $n_3$, there is a semi-path between $n_2$ and $n_4$, there is a semi-path between $n_5$ and $n_6$. 
18. The reachability matrix of a directed graph \( D = (V, E) \) with \( m \) nodes is an \( m \times m \) matrix \( R = (r(i, j)) \), where \( r(i, j) \) is a 1 if and only if there is a true path from node \( i \) and node \( j \), otherwise the element is 0.

The reachability matrix of a directed graph can be calculated from the adjacency matrix as follows:

\[
R = I + A + A^2 + A^3 + \ldots + A^k,
\]

where \( k \) is the length of the longest path in \( D \) (and a simplified version of binary arithmetic is used).

<table>
<thead>
<tr>
<th></th>
<th>n1</th>
<th>n2</th>
<th>n3</th>
<th>n4</th>
<th>n5</th>
<th>n6</th>
<th>n7</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>n2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>n3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>n4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>n5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>n7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
n-Connectedness

19. Two nodes $n(i)$ and $n(j)$ in a directed graph are:

- 0-connected iff there is no path between $n(i)$ and $n(j)$,
- 1-connected iff there is a semi-path but no path between $n(i)$ and $n(j)$,
- 2-connected iff there is a path between $n(i)$ and $n(j)$,
- 3-connected iff there is a path from $n(i)$ to $n(j)$ and a path from $n(j)$ to $n(i)$

There are no other degrees of connectedness.

![Diagram showing connectedness relationships between nodes](image-url)
3-connectedness imposes an equivalence relation on the nodes in a directed graph. The partition imposed by this equivalence relation defines strong components of a directed graph.

20. A strong component of a directed graph is a maximal set of 3-connected nodes.

Strong Components:

{ n3, n4 n6 }

(n7)
21. Given a directed graph $D = (V, E)$, its condensation graph is formed by replacing each strong component by a condensing node.

22. A directed graph with no 3-connected nodes is a directed acyclic graph (DAG).

The condensation graph of a directed graph is always a directed acyclic graph. Why?

Strong Components:

$S_1 = \{ n_3, n_4, n_6 \}$

$S_2 = \{ n_7 \}$

Condensation Graph
Cyclomatic Number of a Graph

23. The cyclomatic number of a graph (undirected or a digraph) is given by

\[ V(G) = e - n + 2p, \]

where

- \( e \) is the number of edges in \( G \),
- \( n \) is the number of nodes in \( G \), and
- \( p \) is the number of components in \( G \).

\( V(G) \) is the number of distinct regions in a graph.

\[ V(G) = 6 - 7 + 2(2) = 3 \]
24. Given a program written in an imperative programming language, its program graph is a directed graph in which:

(Traditional Definition) nodes are program statements, and edges represent flow of control (there is an edge from node i to node j iff the statement corresponding to node j can be executed immediately after the statement corresponding to node i).

(Improved Definition) nodes are either entire statements or fragments of a statement, and edges represent flow of control (there is an edge from node i to node j iff the statement or statement fragment corresponding to node j can be executed immediately after the statement or statement fragment corresponding to node i).