Connectivity

A path of length $n$ from $u$ to $v$ is a sequence of vertices $x_0, x_1, \ldots, x_n$, where $x_{i-1}$ to $x_i$ is an edge for each $i$, $1 \leq i \leq n$.

A circuit or cycle is a path that begins and ends at the same vertex.

A path is simple if it does not contain the same vertex more than once.

A simple graph is connected if there is a path between every pair of vertices.

A directed graph is strongly connected if there is a path between every pair of vertices.

A directed graph is weakly connected if there is a path between every pair of vertices in the underlying simple graph.

Let $G$ be a graph with adjacency matrix $A$. The number of paths of length $n$ from $v_i$ to $v_j$ is the $(i, j)$ entry of $A^n$.

Here is a proof by mathematical induction.
Predicate $P(n)$: The number of paths of length $n$ from $v_i$ to $v_j$ is the $(i, j)$ entry of $A^n$.

Basis $P(1)$: True for $A^1$ because $a_{ij} = 1$ iff there is an edge from $v_i$ to $v_j$.

Induction: Prove $P(k) \rightarrow P(k + 1)$

Assume $P(k)$. Show $P(k + 1)$.

Proof: Let $B = A^k$. $A^{k+1} = B A$, so the $(i, j)$ entry of $A^{k+1}$ is $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots$.

This adds the paths of length $k$ from $v_i$ which can be followed by an edge to $v_j$.

---

**Weighted Graphs**

A weighted graph is a graph in which each edge $(u, v)$ has a weight $w(u, v)$. Each weight is a real number. Weights can represent distance, cost, time, capacity, etc.

The *length* of a path in a weighted graph is the sum of the weights on the edges.

Dijkstra’s Algorithm finds the shortest path between two vertices.
\begin{center}
\begin{tikzpicture}
    \node (a) at (0,0) [circle,draw] {a};
    \node (b) at (1,1) [circle,draw] {b};
    \node (c) at (2,1) [circle,draw] {c};
    \node (d) at (3,0) [circle,draw] {d};
    \node (e) at (4,0) [circle,draw] {e};
    \node (f) at (3,-1) [circle,draw] {f};
    \node (g) at (2,-2) [circle,draw] {g};
    \node (h) at (1,-1) [circle,draw] {h};
    \node (i) at (0,-2) [circle,draw] {i};

    \draw (a) -- (b) node[midway,above] {1};
    \draw (b) -- (c) node[midway,below] {4};
    \draw (b) -- (i) node[midway,left] {3};
    \draw (c) -- (d) node[midway,right] {8};
    \draw (c) -- (i) node[midway,left] {7};
    \draw (d) -- (e) node[midway,right] {5};
    \draw (d) -- (f) node[midway,above] {7};
    \draw (f) -- (g) node[midway,above] {8};
    \draw (g) -- (i) node[midway,below] {3};
    \draw (h) -- (i) node[midway,right] {4};

\end{tikzpicture}
\end{center}

\begin{center}
\begin{tikzpicture}
    \node (a) at (0,0) [circle,draw] {FW};
    \node (b) at (1,1) [circle,draw] {D};
    \node (c) at (2,0) [circle,draw] {A};
    \node (d) at (1,-1) [circle,draw] {EP};
    \node (e) at (0,-2) [circle,draw] {SA};
    \node (f) at (2,-1) [circle,draw] {H};
    \node (g) at (3,-2) [circle,draw] {CC};

    \draw (a) -- (b) node[midway,above] {30};
    \draw (a) -- (c) node[midway,above] {604};
    \draw (a) -- (d) node[midway,above] {191};
    \draw (b) -- (c) node[midway,above] {191};
    \draw (b) -- (f) node[midway,above] {201};
    \draw (c) -- (e) node[midway,above] {590};
    \draw (c) -- (f) node[midway,above] {163};
    \draw (d) -- (e) node[midway,above] {582};
    \draw (e) -- (f) node[midway,above] {201};
    \draw (f) -- (g) node[midway,above] {146};
    \draw (g) -- (c) node[midway,above] {217};
\end{tikzpicture}
\end{center}
procedure shortest_path
(G: weighted simple graph; a, z: vertices in G)
{Assume that a is connected to z}
{L(v) is min. known distance from a to v}
{S is the set of processed vertices}
for each vertex v in G
   L(v) := \infty
L(a) := 0
S := \emptyset
loop
   u := vertex not in S with min. L value
if u = z then exit loop
S := S \cup \{u\}
N := (neighbors of u) \setminus S
for each vertex v in N
   if L(u) + w(u, v) < L(v)
      then L(v) := L(u) + w(u, v)
return L(z)