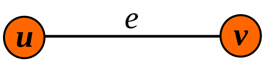
Chapter IV – Graph Theory

Terminology of Graph

Graphs

A *graph* G is a discrete structure consisting of nodes (called *vertices*) and lines joining the nodes (called *edges*). Two vertices are *adjacent to* each other if they are joint by an edge. The edge joining the two vertices is said to be an edge *incident with* them. We use $V(G)$ and $E(G)$ to denote the set of vertices and edges of G respectively.

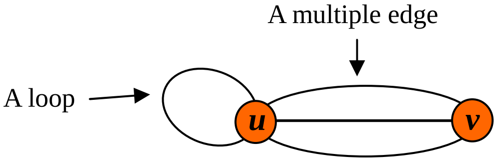
Example



u and v are adjacent vertices; e is an edge incident with u and v. e can also be denoted by uv or vu.

Loops and Multiple Edges

An edge joining only one vertex is called a *loop*. If there are more than one edge joining u and v of G, then all edges joining u and v form a *multiple edge* of G.



Simple Graph

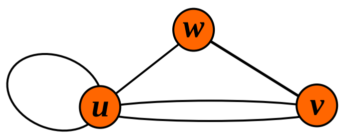
A *simple graph* is a graph containing no loops and multiple edges.

Degrees of Vertices

The degree of a vertex is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by $\deg(v)$ or $d(v)$.

Example

$d(u)=5$, $d(v)=3$ and $d(w)=2$



Theorem [The handshaking theorem] Let G be a graph with e edges. Then

$$2e = \sum_{v \in V(G)} d(v)$$

Theorem The number of vertices of odd degree in a graph G is even.

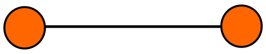
Complete Graphs

The *complete graph* on n vertices, denoted by K_n, is the simple graph in which any pair of vertices are adjacent.

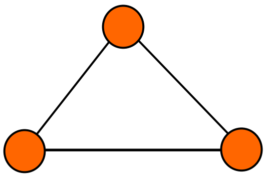
Examples

I) K_1

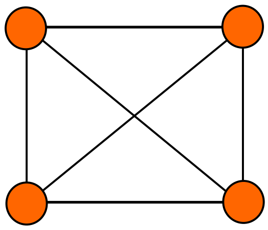
II) K_2



III) K_3



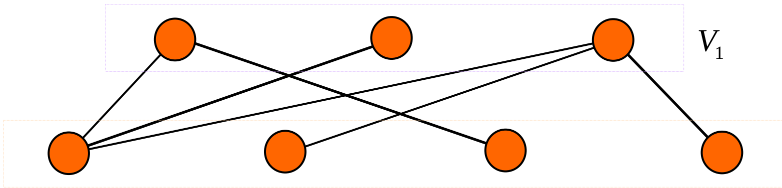
IV) K_4



Bipartite Graphs

A simple graph G is called *bipartite* if its vertex set $V(G)$ can be partitioned into two disjoint sets of V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2. In other words, any two vertices in the same set V_1 (and V_2) are not adjacent.

Example



V_1

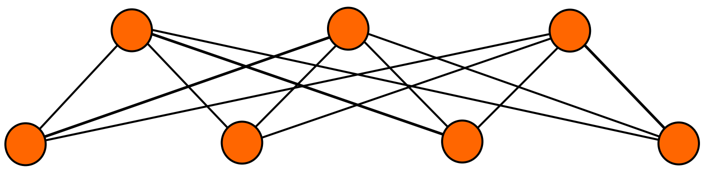
V_2

Complete Bipartite Graphs

A *complete bipartite graph*, denoted by $K_{m,n}$, is the bipartite graph with $|V_1| = m$ and $|V_2| = n$, and $vu \in E(G)$ if and only if $v \in V_1$ and $u \in V_2$.

Example

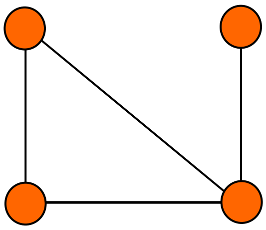
$K_{3,4}$



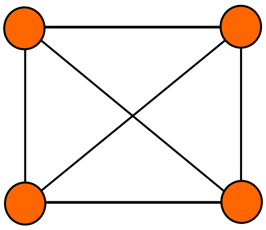
Subgraphs

A *subgraph* of a graph G is a graph H where $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Example



is a subgraph of

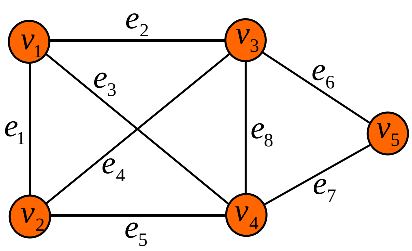


Incidence Matrices

Let G be an graph. Suppose $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G. Then the *incidence matrix* with respect to this ordering of $V(G)$ and $E(G)$ is the $n \times m$ matrix

$$M = [m_{ij}], \text{ where } m_{ij} = \begin{cases} 1 & \text{when } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

Example



The corresponding incidence matrix is

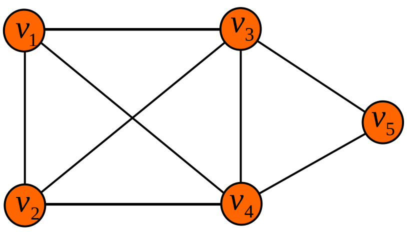
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Adjacency Matrices

Let G be an graph. Suppose $v_1, v_2, \ldots, v_n$ are the vertices of G. Then the *adjacent matrix* with respect to this ordering of $V(G)$ is the $n \times n$ matrix $M = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when } v_i \text{ and } v_j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases}$$

Example



The corresponding adjacency matrix is

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

Paths and Cycles

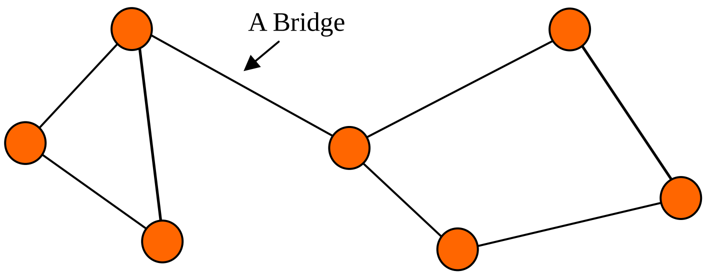
In a graph *G*, a *path of length n* from *u* to *v* is a sequence of $n+1$ vertices $v_1, v_2, \ldots, v_n, v_{n+1}$ where $v_1 = u$ and $v_{n+1} = v$, and $v_i v_{i+1} \in E(G)$ for $i = 1, 2, \ldots, n$. A path is called a *circuit* when $u = v$. A path or a circuit is *simple* if it does not contain the same edge more than once. A *cycle* is a circuit in which all vertices are distinct.

Connected Graphs

A graph is *connected* if there is a path between every pair of distinct vertices of the graph. An edge *uv* in a connected graph *G* is called a *bridge* if $G - uv$, the graph obtained by deleting *uv* from *G*, is not connected.

Example

A connected graph



Euler Circuit and Euler Path

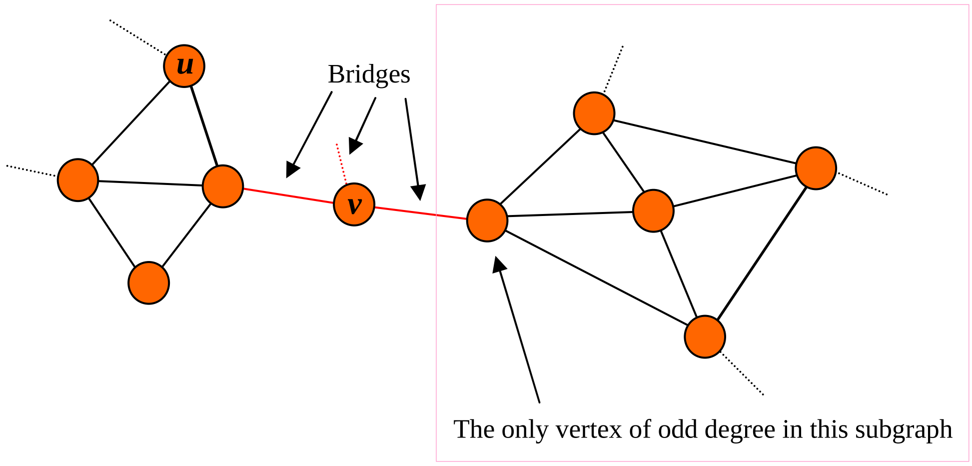
An *Euler circuit* in a graph *G* is a simple circuit containing every edge of *G*. An *Euler path* in *G* is a simple path containing every edge of *G*.

Theorem A connected graph has an Euler circuit if and only if each of its vertices has even degree.

Algorithm [Fleury's Algorithm] to Construct an Euler Circuit

procedure Euler Circuit
circuit $:= u, v$, any $uv \in E(G)$
$G := G - uv$
while $E(G) \neq \varnothing$
circuit := *circuit* with *w*, $vw \in E(G)$ and *v* is the last vertex in *circuit* (*vw* should not be a bridge in *G* unless there is no other vertices in $E(G)$ adjacent to *v*.)
$G := G - vw$
end {*circuit* is an Euler circuit}

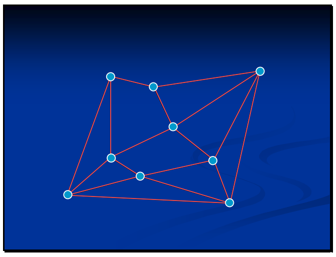
Proof Suppose that we follow the algorithm above started from u to extend the circuit in G and have just reached a vertex v. We also assume the next move must be a bridge. It is noted that all vertices in G except u and v must be of even degree. We shall show in the following that there is only one remaining edge incident with v. Suppose contrary that there are more than one remaining edge incident with v. Then, in $G-v$ (the graph obtained by deleting v and all its incident edges from G), there are more than one connected subgraph and least one of them does not contain u. In the connected subgraph not containing u, the only vertex of odd degree is the vertex adjacent to v in G. This is a contradiction to the fact that there should be even number of odd vertices in a graph. Hence, any moves over a bridge will separate the graph into one connected subgraph and an isolated vertex. After finitely many steps, all edges will be included in the circuit.



Corollary A connected graph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree.

Proof Let G be a connected graph. If G contains an Euler path $P=v,v_1,v_2,\ldots,v_n,u$, then $v,v_1,v_2,\ldots,v_n,u,v$ is an Euler circuit in $G+uv$. From the theorem above, all vertices in $G+uv$ are of even degree; hence there are exactly two odd vertices u and v of G. Conversely, if there are exactly two odd vertices u and v of G, then all vertices in $G+uv$ (a graph obtained by adding uv to G) must be of even degree. It follows that $G+uv$ has an Euler circuit $C=v,v_1,v_2,\ldots,v_n,u,v$; hence, $v,v_1,v_2,\ldots,v_n,u$ is an Euler path in G.

Example



Hamilton Paths and Hamilton Cycles

A path $v_1, v_2, \ldots, v_n$ containing all vertices in G and $v_i \neq v_j$ for all $1 \le i \ne j \le n$ is called *Hamilton path*. A cycle $v_1, v_2, \ldots, v_n, v_1$ in G is a *Hamilton cycle* if $v_1, v_2, \ldots, v_n$ is a Hamilton path.

Theroem [Ore's Theorem] If G is a simple graph with n vertices with $n \ge 3$ such that $d(u)+d(v) \ge n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton cycle.

Proof http://www.ecp6.jussieu.fr/pageperso/bondy/research/papers/shortproofs.ps

Theorem [Dirac's Theorem] If G is a simple graph with n vertices $n \ge 3$ such that the degree of every vertex in G is at least $n/2$, then G has a Hamilton cycle.

Proof For any $u, v \in V(G)$, $d(u)+d(v) \ge \frac{n}{2} + \frac{n}{2} = n$. From Ore's Theorem, G has a Hamilton cycle.

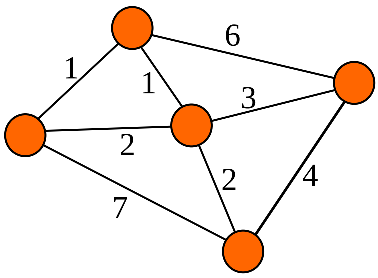
Shortest Path Problems

Weighted Graphs

A graph has a number (weight) assigned to each edge is called a *weighted graph*.

Example

A weighted graph



Shortest Path

A path in a weighted graph such that the sum of the weights of the edges in this path is a minimum over all paths between specified vertices is called a *shortest path*.

Algorithm [Dijkstra's Algorithm] for Producing Shortest Paths from a Fixed Vertex

procedure Dijkstra (G: a weighted connected simple graph, with all weights positive)

{G has vertices $v_0, v_1, v_2, \ldots, v_n$ and weights $w(v_i v_j)$. $w(v_i v_j) = \infty$ if $v_i v_j \notin E(G)$ }

for $i := 1$ **to** n

$\quad L(v_i) := \infty$

$L(v_0) := 0$

$S := \varnothing$

while $S \neq V(G)$

$\quad u :=$ a vertex not in S with $L(u)$ minimal

$\quad S := S \cup \{u\}$

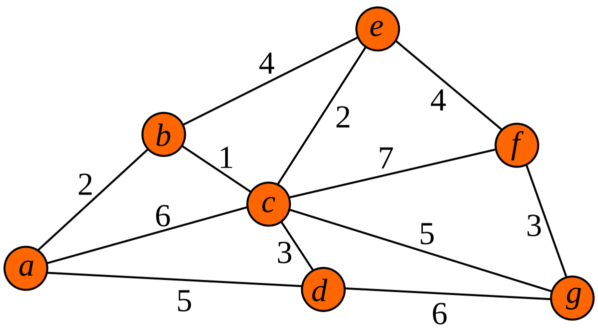
$\quad$ **for** all vertices v not in S

$\quad\quad$ **if** $L(u) + w(uv) < L(v)$ **then** $L(v) := L(u) + w(uv)$

end {the lengths of shortest paths from v_0 }

Example

Consider a to be the fixed vertex.



Tabular Method

S	$L(a)$	$L(b)$	$L(c)$	$L(d)$	$L(e)$	$L(f)$	$L(g)$	*Predecessor*
	$\boxed{0}$	∞	∞	∞	∞	∞	∞	
a		2	6	5	∞	∞	∞	
		$\boxed{2}$	6	5	∞	∞	∞	a
a,b			3	∞	6	∞	∞	
			$\boxed{3}$	5	6	∞	∞	b
a,b,c				6	5	10	8	
				$\boxed{5}$	5	10	8	a
a,b,c,d					∞	∞	11	
					$\boxed{5}$	10	8	c
a,b,c,d,e						9	∞	
						9	$\boxed{8}$	c
a,b,c,d,e,g						11		
						$\boxed{9}$		e
a,b,c,d,e,g,f								