Graphs

$$\mathbf{G = (V, E)}$$

V - set of vertices, E - set of edges

Undirected graphs

Simple graph: V - nonempty set of vertices, E - set of unordered pairs of distinct vertices (no multiple edges or loops)

Multigraph: multiple edges allowed, loops not allowed

Pseudograph: multiple edges and loops allowed

Directed graphs

Directed graph: V - set of vertices, E - set of ordered pairs of vertices (loops allowed, multiple edges in the same direction not allowed)

Directed multigraph: loops and multiple directed edges allowed

Terminology: In undirected graphs vertex u and vertex v are called adjacent in undirected G iff $\{u, v\}$ is an edge in G. We say $\{u, v\}$ is incident on vertices u and v. The degree $d(v)$ of a vertex v is the number of edges incident on v.

Handshaking Theorem: For an undirected graph $G = (V, E)$:

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

(true even for graphs with multiple edges and loops)

Proof: It follows from the fact that each edge contributes 2 to the sum of degrees of vertices since it's incident to exactly 2 (possibly equal, i. e., loop) vertices.

Theorem: An undirected graph has an even number of vertices of odd degree.

Proof:

$2e = \sum_{v \in V} d(v) = \sum_{v \in V_1} d(v) + \sum_{v \in V_2} d(v)$

V_1 = set of odd degree vertices

V_2 = set of even degree vertices

The second term of the RHS is even, hence $\sum_{v \in V_1} d(v)$ must also be even. But $d(v)$ in this sum is odd, hence the number of terms in this sum, i.e. $|V_1|$ must be even.

In a directed graph: (u, v) is an edge, u is the initial vertex (adjacent to v), and v is the terminal vertex (adjacent from u).

$d^-(v)$ is in-degree of vertex v
(i. e., # of edges terminating at v).
$d^+(v)$ is out-degree of vertex v
(i. e., # of edges originating at at v).

<u>Theorem</u>: Let G be a directed graph. Then:

$$\sum_{v \in V} d^-(v) = \sum_{v \in V} d^+(v) = |E|$$

More terminology:

- **Complete graphs** on n vertices K_n: a simple graph with exactly one edge between any pair of distinct vertices.

- **Cycles** $C_n, n \geq 3$: simple graph with vertices $v_1, \ldots, v_n$ and edges $\{v_1, v_2\}, \ldots, \{v_n, v_1\}$.

- **Wheels** $W_n, n \geq 3$: add $(n+1)$-st vertex to C_n and connect it to each of n vertices in C_n.

- n-**Cubes** Q_n: simple graph with vertices representing 2^n bit strings of length $n, n \geq 1$ such that adjacent vertices have bit strings differing in exactly one bit position.

- **Bipartite graphs**: simple graphs such that V can be partitioned into 2 disjoint subsets V_1 and V_2 such that each edge connects a vertex in V_1 and a vertex in V_2, and no edges connect 2 vertices that are both in V_1 or in V_2.

- **Complete bipartite graphs** $K_{m,n}$: $|V_1| = m$, $|V_2| = n$, there is an edge between two vertices iff one vertex is in V_1 and the other in V_2.

- Local area networks.

Representing Graphs

Adjacency matrix: for simple graph $G = (V, E)$, $|V| = n$, is an nxn matrix A of $0's$ and $1's$, such that:

$$a_{ij} = \begin{cases} 1 & if\ (v_i, v_j) \in E \\ 0 & otherwise \end{cases}$$

Incident matrix:

$$m_{ij} = \begin{cases} 1 & if\ edge\ e_j\ is\ incident\ on\ v_i \\ 0 & otherwise \end{cases}$$

Examples: in class.

Isomorphism of graphs

Graphs with the same structure.

Definition: Two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if:

a) There is a one-to-one and onto function f from V_1 to V_2; and

b) Vertices a, b are adjacent in V_1 iff vertices $f(a), f(b)$ are adjacent in V_2 for all a, b in V_1.

Examples: in class.

It is difficult to determine if 2 graphs are isomorphic. There are $n!$ possibilities to check! However, it is simpler to show that two graphs are not isomorphic.

For isomorphism we have some invariant properties:

1. Two graphs must have the same number of vertices and the same number of edges.

2. $d(v_i)$ and $d(u_i)$ must be the same if $f(u_i) = v_i$.

3. Other invariant properties will come later.

<u>Examples</u>: in class.

Graph Connectivity

Path: A path of length n from u to v in an undirected graph is a sequence of edges $e_1, e_2, ..., e_n$ which starts at u and ends at v.

A path is simple if it does not contain the same edge twice.

Circuit: if $u = v$, the path from u to u is a circuit.

Connectedness: An undirected graph is connected if there exists a path between every pair of vertices.

Theorem: There is a simple path between every pair of vertices in a connected undirected graph.

Paths and isomorphism:

Many ways that paths and circuits can help to determine if 2 graphs are isomorphic.

Example: The existence of a simple circuit of a particular length is a useful invariant to show isomorphism.

Example: given in class.

Connectedness in directed graphs

Definition: A directed graph is strongly connected if there exists a path from a to b and from b to a, whenever $a, b \in V$.

Definition: A directed graph is weakly connected if there exists a path between any 2 vertices in the underlying undirected graph.

Theorem (Counting paths between vertices):
Let G be a graph with vertices $v_1, v_2, ..., v_n$ and adjacency matrix A . The number of paths of length r from v_i to v_j is equal to the (i, j) element of the power matrix A^r.

Proof and examples to be given in class.

Euler Paths and Euler Circuits

- An Euler circuit in G is a simple circuit (that does not cross the same edge twice) containing every edge of G.

- An Euler path in G is a simple path containing every edge of G.

Example: given in class.

Necessary and sufficient conditions for Euler circuits and Euler paths.

Theorem: If a connected graph has an Euler circuit then every vertex must have even degree.

Proof: in class.

Theorem: 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: in class.

Hamilton Paths and Circuits

- A Hamilton circuit is a (simple) circuit passing through all vertices only once.

- A Hamilton path is a (simple) path passing through all vertices only once.

Example: given in class.

There are no necessary and sufficient conditions for the existence of Hamilton paths and circuits. For sufficient conditions there are many.

Theorem: If G is a simple graph with $n \geq 3$ then G has a Hamilton circuit if the degree of each vertex is $\geq \lceil \frac{n}{2} \rceil$.

Example: Gray codes (an application of Hamilton circuit to coding).

The Shortest Path Problems

Find the shortest path between two vertices of a weighted graph.

Dijstra's algorithm: All weights are positive.
G is connected and simple graph.
$w(i,j) = \infty$ if (v_i, v_j) is not an edge.

Input: V, W

Denote: $a = v_0$ *(the starting vertex), and* $z = v_n$ *(the end vertex).*

for $i = 1$ *to* n
    $L(v_i) = \infty$
$L(a) = 0$
$S = \phi$
while $(z \notin S)$
    $\{$*u= a vertex not in S with L(u) minimal*
    $S = S \cup \{u\}$
    for all adjacent vertices v *not in* S
        if $L(u) + w(u, v) < L(v)$ *then*
        $L(v) = L(u) + w(u, v)\}$

$L(z)$= *length of shortest path from* a *to* z.