social media explosion: uses network to get as revenue (targeted ads)
so many companies hire people who know graph theory

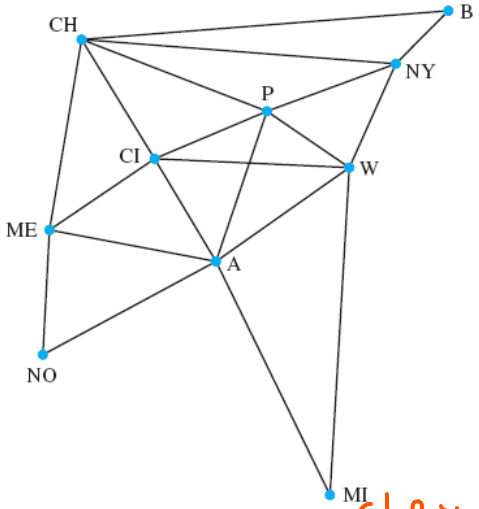
CHAPTER 6: GRAPHS

6.1 Graphs (does NOT mean graph of a function)

Example: The following illustration (called a graph) represents a data-communications network connecting major cities in the Midwest and East Coast, where a dot represents a city's communications center and each line segment represents cable wiring, i.e. a link, that connects the two cities. To ensure reliable and efficient communication within this network, it is natural to ask the following questions regarding how messages are routed between cities.

1. What route has the minimum number of links connecting two given cities? (For example Boston (B) and New Orleans (NO))
2. How many different routes connect any two given cities?
3. Is communication between any two cities possible even if a link was cut?

• vertices / nodes
— edges / links



vertex set

(1) efficient
↳ most: every destination has links to every destination

(2) reliable
↳ if wire breaks, how to still connect cities

Definition: A *graph* is a pair $G = (V, E)$ where V is a finite set of elements (called *vertices*) and E is a set of two-element subsets of V (called *edges*). The number of vertices, $v = |V|$, is called the *order* of G and the number of edges, $e = |E|$, is called the *size* of G.

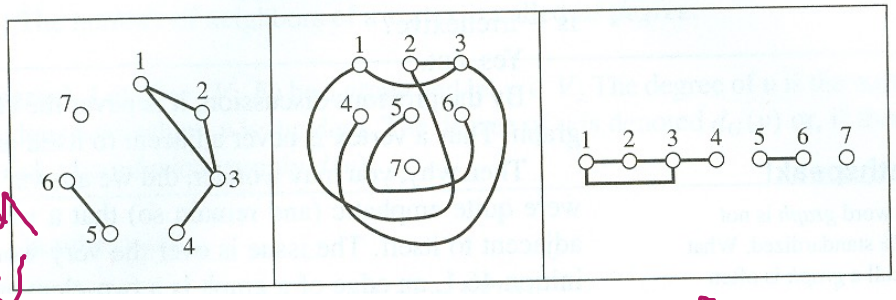
order doesn't matter
repetition not allowed

Example: Let

$$G = (V, E)$$
$$V = \{1,2,3,4,5,6,7\} \Rightarrow v = 7$$
$$E = \{\{1,2\},\{1,3\},\{2,3\},\{3,4\},\{5,6\}\} \Rightarrow e = 5$$

The following are three different (but isomorphic, i.e., equivalent) representations of this graph:

2 element subsets



G G G

graph's topological properties
position, length doesn't matter

Definition: Two graphs $G=(V,E)$ and $G'=(V',E')$ are said to be *isomorphic* if there is a bijection F (one-to-one and onto function) between V and V', which induces a bijection H between E and E' defined by $H(\{v_1,v_2\})=\{F(v_1),F(v_2)\}$.

Example: The following two graphs are isomorphic because of the bijections $F:V\to V'$ defined by

$$1\to c,\ 2\to f,\ 3\to b,\ 4\to d,\ 5\to a,\ 6\to g,\ 7\to e$$

and $H:E\to E'$ defined by

$$\{1,2\}\to\{c,f\},\ \{1,3\}\to\{c,b\},\ \{2,3\}\to\{f,b\},\ \{3,4\}\to\{b,d\},\ \{5,6\}\to\{a,g\}$$

Definition: (Adjacency) Let $G=(V,E)$ be a graph and $x,y\in V$.

(a) We say that x is *adjacent* to (or *joined* to or a *neighbor* of) y if they form an edge of G, i.e. $\{x,y\}\in E$, and write $x\sim y$ to denote this fact.

(b) If $\varepsilon=xy=\{x,y\}$ is the edge joining x and y, then x and y are called *endpoints* of ε and are also said to be *incident* on ε.

(c) The set of all neighbors of x is defined as $N(x)=\{y\in V: y\sim x\}$.

(d) The *degree* of x is the number of neighbors of x (or the number of edges with which v is incident) and is denoted by $d_G(x)$ or simply $d(x)$.

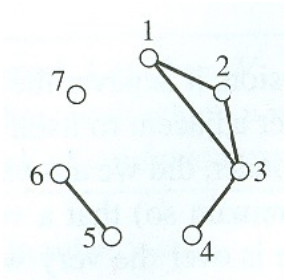
NOTES:

1. It is possible for an edge to join a vertex back to itself (called a loop).
2. A graph that has no more than one edge joining two vertices is called a *simple* graph. If there are more than one edge joining two vertices is called a *multiple-edged* graph.

Example: Consider the graph $G=(V,E)$ in the previous example:

$V=\{1,2,3,4,5,6,7\}$

$E=\{\{1,2\},\{1,3\},\{2,3\},\{3,4\},\{5,6\}\}$



Then the set of neighbors of each vertex and their corresponding degrees are

$$\begin{aligned} N(1)&=\{2,3\}, & d(1)&=2\\ N(2)&=\{1,3\}, & d(2)&=2\\ N(3)&=\{1,2,4\}, & d(3)&=3\\ N(4)&=\{3\}, & d(4)&=1\\ N(5)&=\{6\}, & d(5)&=1\\ N(6)&=\{5\}, & d(6)&=1\\ N(7)&=\varnothing, & d(7)&=0 \end{aligned}$$

Observe that

$$d(1)+d(2)+\ldots+d(7)=10=2\cdot 5=2|E|$$

This property is in fact true of all graphs and will be proven as a theorem later.

Sum of Degrees

Theorem: Let $G=(V,E)$. The sum of the degrees of the vertices of G is twice the number of edges, i.e.

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

Proof: (By induction on the number of edges) Let e denote the number of edges of G. We first prove that the formula is true for $e=1$, namely when G has only one edge, say x_0y_0. Then we have $d(x_0)=d(y_0)=1$ and $d(z)=0$ for all other vertices $z\in V$. It follows that

$$\sum_{x\in V} d(x)=d(x_0)+d(y_0)=2$$

which equals $2e=2$. Next, assume that the formula holds for any graph G' with $e-1$ edges:

$$\sum_{v\in V'} d_{G'}(v)=2(e-1)$$

We need to prove that the formula holds for G (with e edges). Pick any edge $\varepsilon=x_0y_0$ and consider the subgraph $G'(V',E')$ where ε is deleted from G so that $V'=V$ and $E'=E-\{\varepsilon\}$. Then observe that

$$\begin{aligned} d_G(x_0)&=d_{G'}(x_0)+1,\\ d_G(y_0)&=d_{G'}(y_0)+1,\\ d_G(z)&=d_{G'}(z) \end{aligned}$$

for all other vertices $z\in V$. It follows that

$$\sum_{x\in V} d_G(x) = \left(\sum_{x\in V'} d_{G'}(x)\right) + 2 = 2(e-1)+2 = 2e$$

Hence, by mathematical induction the formula holds for all integers $e \geq 1$.

Example: Suppose a graph G has 30 edges and it is known that every vertex has at least 5 neighbors. What is the maximum number of vertices that G can have?

Solution: Let $V = \{x_1, x_2, ..., x_n\}$ denote the vertices of G. Then on the one hand, we have

$$\sum_{x\in V} d(x) = d(x_1) + d(x_2) + ... + d(x_n) \geq \overbrace{5+5+...+5}^{n \text{ times}} = 5n$$

On the other hand, we have $\sum_{x\in V} d(x) = 2e = 2(30) = 60$. It follows that

$$60 = \sum_{x\in V} d(x) \geq 5n$$

$$\therefore n \leq 12$$

Thus, G has at most 12 vertices.

Example: Prove that any graph G must have an even number of vertices of odd degree.

Solution: Let $V = \{x_1, x_2, ..., x_n\}$ denote the vertices of G. Suppose on the contrary that G has an odd number of vertices of odd degree, say $v_1, v_2, ..., v_{2k+1}$, where the degrees $d(v_1), d(v_2), ..., d(v_{2k+1})$ are all odd. It follows that $\sum_{x\in V} d(x)$ is odd, being a sum of an odd part and an even part:

$$\sum_{x\in V} d(x) = \overbrace{d(x_1) + d(x_2) + ... + d(x_{2k+1})}^{\text{SUM OF ODD NUMBER OF ODDS = ODD}} + \overbrace{d(x_{2k+2}) + ... + d(x_n)}^{\text{SUM OF EVENS = EVEN}}$$

On the other hand, we have $\sum_{x\in V} d(x) = 2e$, which is even and hence a contradiction.

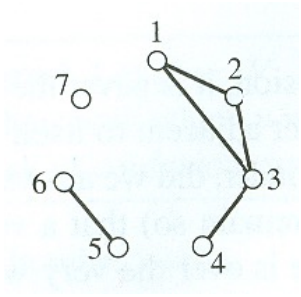
Adjacency Matrix

Definition: (Adjacency Matrix) Let G be a graph with a set of vertices $V = \{x_1, x_2, ..., x_n\}$. Given two vertices x_i, x_j, we assign it the value a_{ij} defined

$$a_{ij} = \begin{cases} 1 & \text{if } x_i \sim x_j \text{ (neighbors)} \\ 0 & \text{otherwise} \end{cases}$$

The table of values (a_{ij}) is called the *adjacency matrix* of G.

Example: The adjacency matrix for the graph

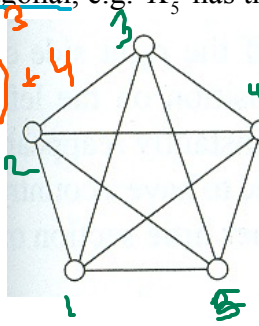


is given by

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	1	0	0	0	0
v_2	1	0	1	0	0	0	0
v_3	1	1	0	1	0	0	0
v_4	0	0	1	0	0	0	0
v_5	0	0	0	0	0	1	0
v_6	0	0	0	0	1	0	0
v_7	0	0	0	0	0	0	0

Definition: A *complete* graph on n vertices, denoted by K_n, is one where all pairs of distinct vertices are adjacent, i.e. every vertex is adjacent to every other vertex.

NOTE: The adjacency matrix of a complete graph is one whose entries consists of all 1's except for those along the main diagonal, e.g. K_5 has the following adjacency matrix:



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

Exercise: Determine a direct formula for the number of edges in the complete graph K_n.

Subgraphs and Edge/Vertex Deletion

Special subgraphs can be constructed from a graph by deleting either a subset of edges only (edge deletion) or a subset of vertices and their corresponding edges (vertex deletion). This leads to the following notions.

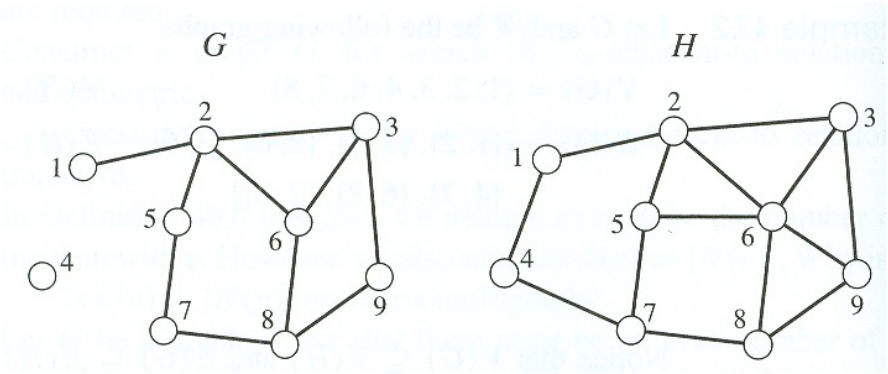
Definition: (Spanning Subgraph) Let G and H be graphs. If G is a subgraph of H with $V(G)=V(H)$ (edge deletion only), then G is called a *spanning subgraph* of H.

Example: Let G and H be the following graphs:

$$V(G)=\{1,2,3,4,5,6,7,8,9\}$$
$$E(G)=\{\{1,2\},\{2,3\},\{2,5\},\{2,6\},\{3,6\},\{3,9\},\{5,7\},\{6,8\},\{7,8\},\{8,9\}\}$$

$$V(H)=\{1,2,3,4,5,6,7,8,9\}$$
$$E(H)=\{\{1,2\},\{1,4\},\{2,3\},\{2,5\},\{2,6\},\{3,6\},\{3,9\},$$
$$\{4,7\},\{5,6\},\{5,7\},\{6,8\},\{6,9\},\{7,8\},\{8,9\}\}$$

Here, G is not only a *subgraph* of H but also a spanning subgraph of H since $E(G)\subseteq E(H)$ and $V(G)=V(H)$. Observe that G can be obtained from H by deleting 4 edges.



Definition: (Induced Subgraph) Let H be a graph and A be a subset of the vertices of H, i.e. $A\subseteq V(H)$. The *subgraph of H induced on A* is the graph $H[A]$ defined by the following set of vertices and edges:

a) $V(H[A])=A$

b) $E(H[A])=\{xy\in E(H): x\in A \text{ and } y\in A\}$

In other words, the graph $H[A]$ has A as its set of vertices and contains exactly those edges in H that are joined by vertices in A.

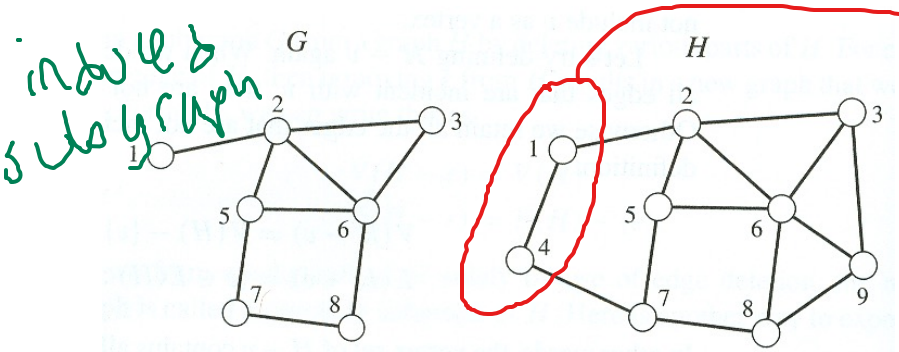
Example: Let H be the graph defined by

$$V(H)=\{1,2,3,4,5,6,7,8,9\}$$
$$E(H)=\{\{1,2\},\{1,4\},\{2,3\},\{2,5\},\{2,6\},\{3,6\},\{3,9\},$$
$$\{4,7\},\{5,6\},\{5,7\},\{6,8\},\{6,9\},\{7,8\},\{8,9\}\}$$

Consider the subset of vertices $A=\{1,2,3,5,6,7,8\}\subset V(H)$. Then the induced subgraph $H[A]$ is given by

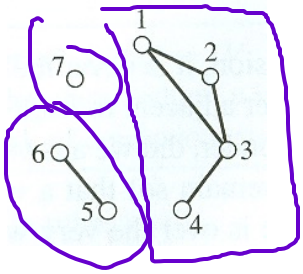
$$V(H[A])=\{1,2,3,5,6,7,8\}$$
$$E(H[A])=\{\{1,2\},\{2,3\},\{2,5\},\{2,6\},\{3,6\},\{5,6\},\{5,7\},$$
$$\{6,8\},\{7,8\}\}$$

Observe that $H[A]$ can be obtained from H by deleting 2 vertices and their corresponding 5 edges.



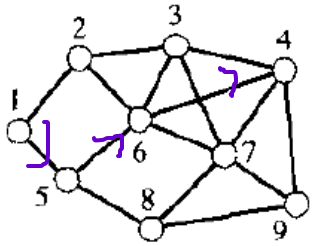
Connectivity

The following is a graph that seems 'disconnected' and has three 'components'. In this section we formally develop the notion of a connected graph.



Definition: (Walk) Let G be a graph.
(a) A *walk of length l* in G is a sequence of $(l+1)$ vertices, with each vertex adjacent to the next, i.e. $W = (x_0, x_1, ..., x_l)$ with $x_0 \sim x_1 \sim x_2 ... \sim x_l$.
(b) A walk W is *closed* if it begins and ends at the same vertex. Also, the reversal of W is $W^{-1} = (x_l, x_{l-1}, ..., x_0)$.

Example: Here are some examples of walks in G given below:



Walk of length 3: $W_1 = 1 \sim 5 \sim 6 \sim 4$
Walk of length 6: $W_2 = 4 \sim 9 \sim 8 \sim 5 \sim 6 \sim 3 \sim 2$
Walk of length 0: $W_3 = 9$

Definition: (Concatenation) Let G be a graph. Suppose two walks are given:

$$W_1 = x_0 \sim x_1 \sim x_2 \ldots \sim x_l$$
$$W_2 = y_0 \sim y_1 \sim y_2 \ldots \sim y_k$$

where $x_l = y_0$. Their *concatenation* is defined to be the walk (of length $l+k$)

$$W_1 + W_2 = x_0 \sim x_1 \sim x_2 \ldots \sim (x_l = y_0) \sim y_1 \sim y_2 \ldots \sim y_k$$

Example: The concatenation of the two walks in the previous example becomes

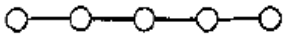
$$W_1 + W_2 = 1 \sim 5 \sim 6 \sim 4 \sim 9 \sim 8 \sim 5 \sim 6 \sim 3 \sim 2$$

Definition: (Path) A *path* in a graph is a walk in which no vertex is repeated. Equivalently, a path P can also be viewed as a graph in itself with vertex set $V = \{x_1, x_2, \ldots, x_n\}$ and edge set

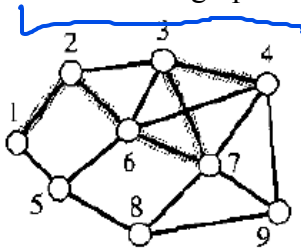
$$E = \{x_0x_1, x_1x_2, \ldots, x_{n-1}x_n\}$$

We also call P a (x_0, x_n)-path. A path on n vertices is denoted $P_n = (x_0, x_{n-1})$.

A P_5 graph:



Example: The walk $W = 1 \sim 2 \sim 6 \sim 7 \sim 3 \sim 4$ in the graph below is a path.



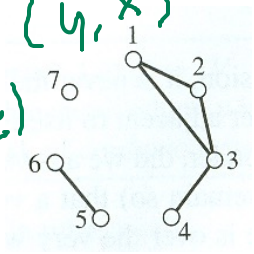
NOTE: The concatenation $W_1 + W_2 = 1 \sim 5 \sim 6 \sim 4 \sim 9 \sim 8 \sim 5 \sim 6 \sim 3 \sim 2$ described in the previous example is NOT a path.

Proposition: Let P be a path in a graph G. Then P does not traverse any edge of G more than once.

Definition: (Connected) Let G be a graph and $x, y \in V(G)$. Then x is said to be *connected to* y if there is a (x,y)-path. If this holds for all $x, y \in V(G)$, then G is said to be a *connected graph*.

'Connected to' as an Equivalence Relation

Example: Consider the following graph G, which has three *components* where each component is a subgraph consisting of a set of vertices that are connected to each other:

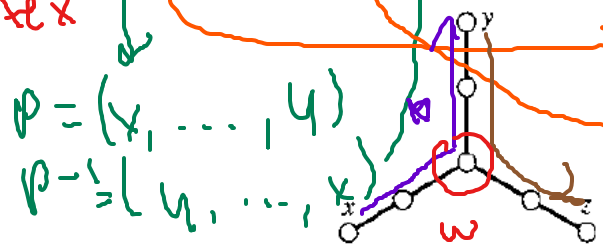


{1, 2, 3, 4}, {5, 6}, {7}

This leads to the notion of 'connected to' as an equivalence relation defined on $V(G)$.

Theorem: The notion of 'connected to' is an equivalence relation on $V(G)$:

a) Each vertex x is connected to itself.
b) If x is connected to y, then y is connected to x.
c) If x is connected to y and y is connected to z, then x is connected to z.



WARNING: Concatenation of two paths does NOT necessarily produce a path (demonstrate this with the graph above).

Lemma: Let G be a graph and $x, y \in V(G)$. If there is an (x,y)-walk in G, then there is an (x,y)-path in G.

Components

Since the 'connected to' equivalence relation divides the set of vertices $V(G)$ into distinct equivalence classes, this in turn divides the graph G into corresponding subgraphs.

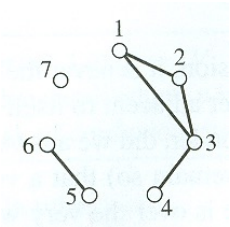
Definition:

a) (Equivalence class) Given a vertex $x \in V(G)$, define $C = [x]$ to be the *connectivity class* of x, i.e. the equivalence class of vertices of $V(G)$ that are connected to x, i.e.

$$C = [x] = \{y \in V(G) : y \text{ connected to } x\}$$

b) (Component) Let C be a connectivity class. A *connected component* of G is an induced subgraph $G[C] = (C, E(C))$ with vertex set C and edge set $E(C)$, which consists of edges whose endpoints belong to C.

Example: Let us determine the components of the graph G below.



First we determine the connectivity classes C on the set of vertices of G:

$[1] = \{1, 2, 3, 4\}$
$[2] = \{1, 2, 3, 4\}$
$[3] = \{1, 2, 3, 4\}$
$[4] = \{1, 2, 3, 4\}$
$[5] = \{5, 6\}$
$[6] = \{5, 6\}$
$[7] = \{7\}$

Thus there are three equivalence classes: $C = \{1, 2, 3, 4\}$, $\{5, 6\}$, and $\{7\}$. The corresponding components of G are the induced subgraphs $G[\{1,2,3,4\}]$, $G[\{5,6\}]$, and $G[\{7\}]$.

Definition: A graph G is *connected* if it has exactly one connected component.

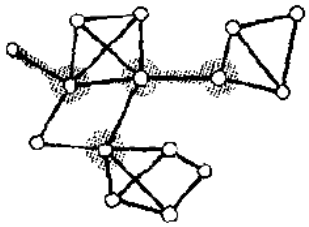
Disconnnection

Definition: (Cut Vertex/Edge) Let G be a graph.

a) A vertex $x \in V(G)$ is called a *cut vertex* provided $G - x$ has more components than G, i.e. removing x increases the number of components of G.
b) Similarly, an edge $\varepsilon \in E(G)$ is called a *cut edge* if $G - \varepsilon$ has more components than G, i.e. removing ε increases the number of components of G.

NOTE: Cut vertices and cut edges makes a graph more disconnected.

Example: The graph below has 4 cut vertices and 2 cut edges:

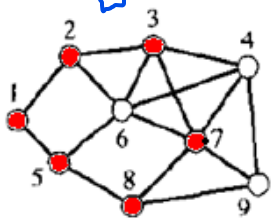


Theorem: Let G be a connected graph and e a cut edge of G. Then $G - e$ has exactly two components.

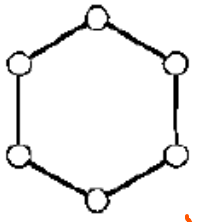
Cycles and Trees

Definition: (Cycle) A *cycle* is a walk of length at least three in which the first and last vertex are the same (closed walk), but no other vertices are repeated.

Example: The following walk is a cycle: $W = 1 \sim 2 \sim 3 \sim 7 \sim 8 \sim 5 \sim 1$

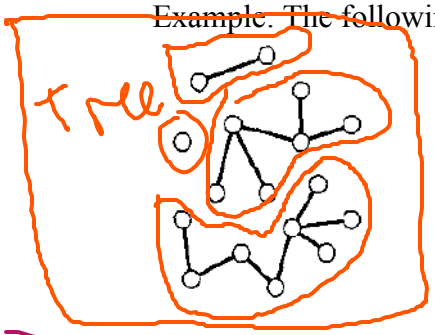


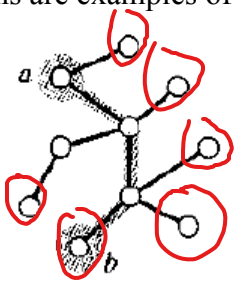
NOTE: A cycle (graph) on n vertices is denoted C_n.



Definition: (Forest/Tree) Let G be a graph.
a) If G contains no cycles, then G is said to be *acyclic* or a *forest*.
b) If G is a connected, acyclic graph, then G is said to be a *tree*.

Example: The following graphs are examples of trees:





Theorem: A graph T is a tree if and only if for any two vertices a and b in $V(T)$, there is a unique (a,b)-path.

Theorem: Let G be a connected graph. Then G is a tree if and only if every edge of G is a cut edge.

Leaves

Definition: (Leaf) Let G be a graph. If a vertex v of G has degree 1, then v is said to be a *leaf*.

Theorem: Every tree with at least two vertices has a leaf.

Theorem: Let T be a tree and v be a leaf of T. Then $T - v$ is a tree.

Theorem: Let T be a tree with $n \geq 1$ vertices. Then T has $n-1$ edges.

Proof: (By induction on n)

6.2 Spanning Trees and Rooted Trees

Motivating Question: Given a graph G, what is the minimum set of edges needed to connect all the vertices of G?

Definition: (Spanning Subgraph) Let H be a subgraph of G. If $V(G)=V(H)$ (edge deletion only), then H is called a *spanning subgraph* of G.

Spanning Trees

Definition: Let T be a subgraph of G. Then T is called *spanning tree* of G if
(a) T is a spanning subgraph of G.
(b) T is a tree.

Follow up questions:
1. Does every graph G have a spanning tree T?
2. Develop an algorithm to find a spanning tree T of G.

Lemma: Each finite connected graph G has a spanning tree.

Proof: (Induction) Let c denote the number of cycles in G. We prove that G has a spanning tree T by strong induction on c:
1. (Base case) Assume that $c=0$. Then G is a tree and clearly has a spanning tree, namely set $T=G$ (itself).
2. (Inductive step) Assume that every graph with fewer than c cycles has a spanning tree. Choose a cycle and delete an edge of that cycle. Then the resulting subgraph G' has a spanning tree T since G' has fewer than c cycles. But then T is also a spanning tree of G. This completes the proof.

Algorithm: Constructing Spanning Tree T of G (connected) with edge set E and vertex set V:
1. Set $E'=\{\}$ (empty) and $S=\{x_0\}$ where $x_0 \in V$.
2. If $S=V$, then return (quit).
3. Else choose edge $e \in E$ having one of its endpoints in S and the other endpoint x in $V-S$ (such an edge exists since G is connected). Append e to E' and x to S. Go back to Step 2.

The resulting graph $T=(E',V)$ is a tree.

Run-time: $O(ve)$, where $v=|V|$ and $e=|E|$.

Theorem: Let G be a connected graph with $n \geq 1$ vertices. Then G is a tree if and only if G has exactly $n-1$ edges.

Proof: ($\Rightarrow$) Proven in a previous theorem.
($\Leftarrow$) Let G be a graph with n vertices and $n-1$ edges. By the lemma above, G has a spanning tree T with $V(T)=V(G)=n$. It follows that T and G also have the same number of edges:

$$|E(T)|=|V(T)|-1=n-1=|E(G)|$$

Thus, $G = T$ and proves that G is a tree.

Rooted Trees

Definition:
(a) A *rooted tree* is a tree with a selected vertex x_0 (called the root).
(b) Vertices adjacent to the root are called *children*; in that case, the root is called the *parent*.
(c) By induction, given a vertex x, any vertex y that is adjacent to x (except for the parent of x) is called a child of x; in that case, x is called the parent of y.
(d) If there is a (unique) path between two vertices x and y, and the path contains the parent of y, then y is called a *descendant* of x. Moreover, x is called an *ancestor* of y.
(e) A vertex with no children is called an *external* (or *leaf*) vertex. Vertices with children are called *internal* vertices.

Lemma: Every vertex has at most one parent.

Binary Trees

Definition 1: (Inductive)
1. The smallest binary tree is the empty rooted tree $T_0 = (\varnothing, \varnothing)$ with no vertices and no edges.
2. Let $T^L = (V^L, E^L)$ and $T^R = (V^R, E^R)$ be two binary trees with roots x^L and x^R (assuming they exist). Define the rooted tree $T = T^L \oplus x \oplus T^R = (V, E)$, where x is a root which ties T^L and T^R together so that $V = \{x\} \cup V^L \cup V^R$ and $E = \{xx^L, xx^R\} \cup E^L \cup E^R$. Then T is a binary tree.

Examples: We use the definition above to inductively construct larger binary trees.
a) $T_1 = T_0 \oplus x_1 \oplus T_0 = (\{x_1\}, \varnothing)$
b) $T_2 = T_1^L \oplus x_2 \oplus T_1^R = \{x_2, x_1^L, x_1^R\}, \{x_2x_1^L, x_2x_1^R\})$
c) $T_{1,0} = T_1^L \oplus x_2 \oplus T_0^R = (\{x_2, x_1^L\}, \{x_2x_1^L\})$ and $T_{0,1} = T_0^L \oplus x_2 \oplus T_1^R = (\{x_2, x_1^R\}, \{x_2x_1^R\})$
d) $T_4 = T_2^L \oplus x_4 \oplus T_2^R = (\{x_4\} \cup V_2^L \cup V_2^R, \{x_2x_1^L, x_2x_1^R\} \cup E_2^L \cup E_2^R)$

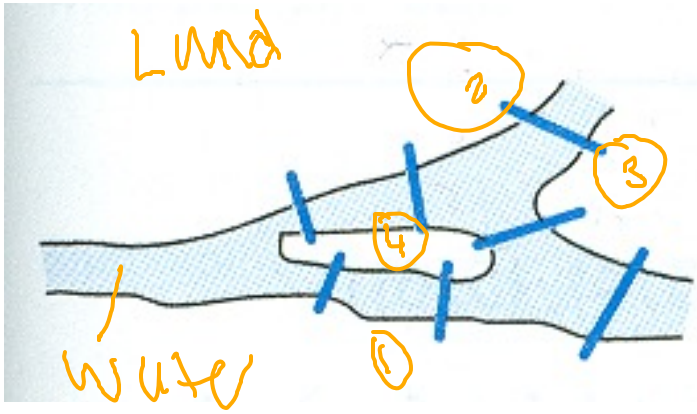
Definition 2: A *binary tree T* is a rooted tree where each vertex, viewed as a parent, has at most two children. If each vertex has exactly 2 children (or none, in which case it is called a leaf or an external vertex), then T is called a *full binary tree*.

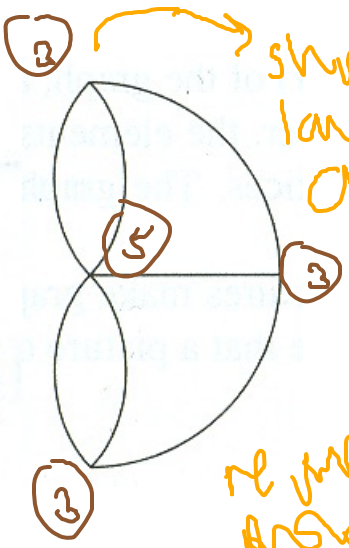
Exercise: Let T be a full binary tree.
a) What do you notice about the number of vertices of T? Prove your answer.
b) How does the number of internal vertices of T compare to the number of external vertices? Prove your answer.
HINT: To prove both parts use induction by removing the root vertex.

6.3 Eulerian and Hamiltonian Graphs

Seven Bridges of Konigsburg (Leonard Euler, 1736)

The town of Konisburg in Russia has seven bridges shown below. Its residents enjoy taking strolls across each of the bridges. A natural question was posed by its residents: Is it possible to take a stroll along a ~~path~~ that starts and ends at the same point and crosses each bridge exactly once. Can you find such a ~~path~~?





Eulerian Tours and Trails

Definition: (Eulerian trail/tour) Let G be a graph.

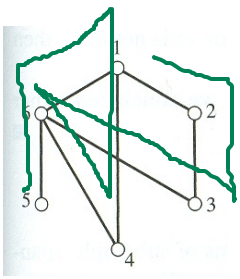
a) A walk in G that traverses every edge exactly once is called an *Eulerian trail*.

b) If a walk traverses every edge exactly once, but begins and ends at the same vertex, then it is said to be an *Eulerian tour* (or *circuit*).

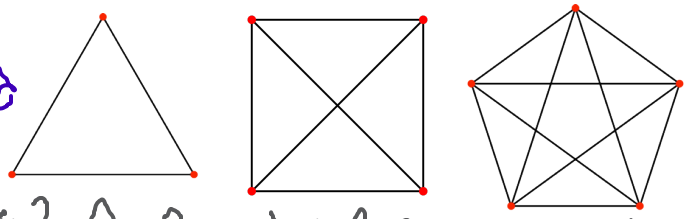
c) If G has an Eulerian tour, then G is simply said to be *Eulerian*.

Example:

a) The walk $W = 1 \sim 2 \sim 3 \sim 6 \sim 4 \sim 1 \sim 6 \sim 5$ is an Eulerian trail of the graph G below. However, G has no Eulerian tour.



b) The graphs K_3 and K_5 are Eulerian, but K_4 is not.



Necessary Conditions for Eulerian Trails/Tours

Lemma:

a) If G is Eulerian, then G has at most one nontrivial component.

b) If G has an Eulerian trail, then it has at most two vertices of odd degree.

c) If G has an Eulerian trail that begins at a vertex x and ends at a vertex y ($x \neq y$), then vertices x and y have odd degree. (See converse below)

d) If G is Eulerian, then all vertices of G have even degree. (See converse below)

e) If G is a connected Eulerian graph, then G has an Eulerian tour that begins and ends at any vertex.

Necessary and Sufficient Conditions for Eulerian Trails/Tours

Theorem: A graph G has a Eulerian tour if and only if it is connected and every vertex has even degree.

Proof: (p. 392) The previous lemma proves the forward direction $\Rightarrow$. We prove the reverse direction $\Leftarrow$ (converse):

1. Construct a closed walk $C = x_0 \sim x_1 \sim \ldots \sim x_N$ that starts and ends at a vertex x_0 but does not repeat any edge. This is possible since every vertex has even degree.
2. Delete the edges of C from G to obtain a graph $G' = (V, E')$ of smaller size.
3. Let $K_0, K_1, \ldots, K_k$ be the connected components of G'. Every vertex in each component K_i has even degree. Moreover, K_i contains a vertex x_{n_i} of C. If there is more than one such vertex, choose n_i to be the smallest index. Moreover, assume $x_0 = x_{n_0}$ belongs to K_0.
4. By (strong) induction, each K_i has a Eulerian tour T_i that starts and ends at x_{n_i}.
5. A Eulerian tour T of G can now be constructed as follows:
   (i) Start at x_0 and take the Eulerian tour of the connected component containing x_0. Walk to x_1 (along C).
   (ii) If x_1 is NOT in the connected component of x_0 (with respect to G'), then take the Eulerian tour of the connected component containing x_1. Walk to x_2 (along C).
   (iii) Repeat for each vertex x_n. If x_n is not any of the connected components containing $x_0, x_1, \ldots, x_{n-1}$, then take the Eulerian tour of the connected component containing x_n. Walk to x_{n+1} (along C).
   (iv) Stop when we reach x_0 again.
6. We claim that T includes every edge of G but does not repeat any edge.

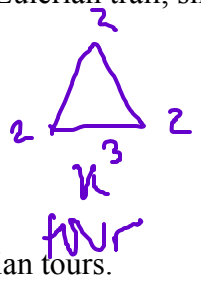
Theorem: A graph G has a Eulerian trail that begins at vertex x and ends at vertex y if and only if all vertices have even degree except for x and y, which have odd degree.

Proof: (p. 393) We prove only reverse direction $\Leftarrow$ (converse) since the forward direction $\Rightarrow$ was proven in the previous lemma.

1. Add the edge $e' = xy$ to G to obtain a graph G'. Then every vertex of G' has even degree.
2. By the previous theorem, G' has a Eulerian tour T' that starts and ends at say x.
3. Now delete the edge e' from T' to obtain a Eulerian trail T of G.

Example: The theorem above shows that the Seven Bridges of Konisburg problem has no solution, namely there is no Eulerian tour and no Eulerian trail, since there are more than 2 vertices of odd degree.

Exercise: For which values of n is K_n Eulerian?



Finding Eulerian Tours

Here is a constructive algorithm for finding Eulerian tours.

ALGORITHM for finding an Eulerian Tour of a Graph $G(V, E, x_0)$ that starts and ends at x_0.

// Assumptions: G is a connected graph whose vertices all have even degree.

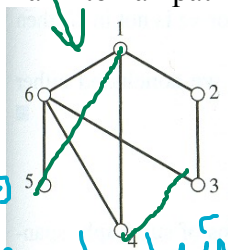
(1) y = any vertex adjacent to x_0
(2) W = CreateWalk(x_0, y)
(3) RemoveEdge(x_0, y, E)
(4) While ($y \neq x_0$)
(5) $x = y$
(6) y = any vertex adjacent to x
(7) AppendToWalk(W, y)
(8) RemoveEdge(x, y, E)
(9) $W_1 = W$
(10) For (x in W)
(11) While (Degree(x, E) > 0)
(12) W_2 = FindEulerianTour(V, E, x)
(13) SpliceWalks(W_1, x, W_2)
(14) return W_1

Hamiltonian Paths and Cycles

Definition: (Hamiltonian paths/cycles) Let G be a graph.

d) A path in G that traverses every vertex exactly once is called a *Hamiltonian path*.
e) A cycle in G that traverses every vertex exactly once is called a *Hamiltonian cycle*.
f) If G has a Hamiltonian cycle, then G is simply said to be *Hamiltonian*.

Example: Does the graph below have a Hamiltonian path or a Hamiltonian cycle?



Exercise: For which values of n does the complete graph K_n have a Hamiltonian path? Which ones have a Hamiltonian cycle?

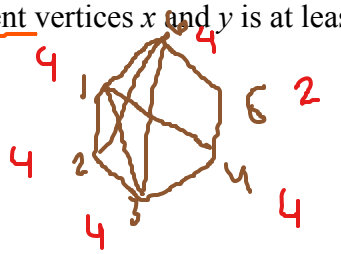
Dirac's Theorem: Let G be a simple graph with v vertices where $v \geq 3$. If every vertex of G has degree at least $v/2$, then G has a Hamiltonian cycle.

Proof: (p. 398) (Contradiction)

1. Suppose on the contrary that there exists a graph G_1 with no Hamiltonian cycle and each vertex has degree at least $v/2$.
2. Observe that if we start adding edges to G_1 one at a time, then the degree of each vertex can only increase. Moreover, at some point we will obtain a graph that does have a Hamiltonian cycle. This is true because we will eventually reach a complete graph.
3. Suppose we now add edges one at a time in such a way that we obtain a graph G_2 with the following property: adding ANY further edge to G_2 gives a graph that has Hamiltonian cycle, but G_2 itself does NOT have a Hamiltonian cycle. We call G_2 *maximal* in this regard.
4. Let $x = x_1$ and $y = x_v$ be two non-adjacent vertices in G_2. Then adding the edge x_1x_v to G_2 yields a Hamiltonian cycle, say $x_1 \sim x_2 \sim x_3 \sim \ldots \sim x_v \sim x_1$. But then $x_1 \sim x_2 \sim x_3 \sim \ldots \sim x_v$ is a Hamiltonian path for G_2.
5. Given any vertex x_i adjacent to x_1, we claim that x_{i-1} is NOT adjacent to x_v. Otherwise, the walk $x_1 \sim x_i \sim x_{i+1} \sim \ldots \sim x_v \sim x_{i-1} \sim x_{i-2} \sim \ldots \sim x_1$ would be a Hamiltonian cycle for G_2.
6. By assumption, $x = x_1$ has degree $d(x) \geq v/2$, which means it is adjacent to at least $v/2$ of the vertices $x_2, x_3, \ldots, x_{v-1}$.
7. It follows that $y = x_v$ is non-adjacent to at least $v/2$ vertices, but if we include y itself, then $v/2+1$. Thus, the maximum number of vertices that y is adjacent to equals
$$d(y) = v - (v/2+1) = v/2 - 1 < v/2$$
8. But this contradicts the fact every vertex of G_2, including y, has degree $d(y) \geq v/2$.

NOTE: Dirac's Theorem can be extended by only requiring that the degrees of x and y sum to v.

Ore's Theorem: Let G be a simple graph with v vertices where $v \geq 3$. If G has the property that the sum of the degrees of any two non-adjacent vertices x and y is at least v, i.e. $d(x) + d(y) \geq v$, then G has a Hamiltonian cycle.



Other Topics in Graph Theory

1. Finding Shortest Paths (Spanning Tree)

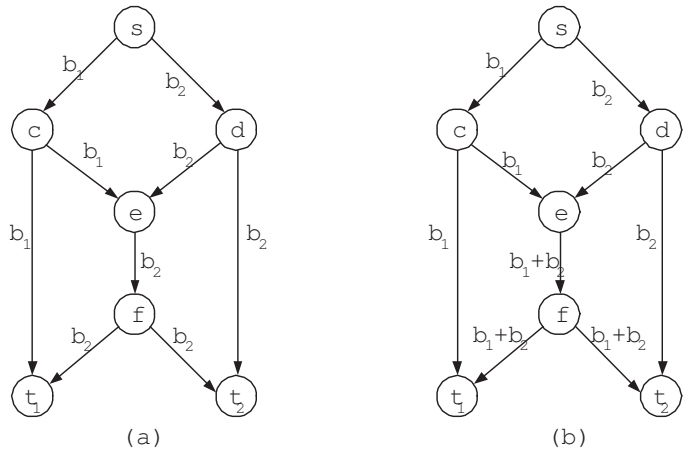
Dijkstra's Algorithm: https://en.wikipedia.org/wiki/Dijkstra's_algorithm

Example: http://www.di-mgt.com.au/docs/dijkstra.pdf

2. Network Coding

http://iest2.ie.cuhk.edu.hk/~whyeung/post/thesis/Ngai.pdf

Butterfly network



3. Traveling Salesman Problem

Wikipedia: The **travelling salesman problem** (**TSP**) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

- NP-hard problem
- Nearest-neighbor (greedy) algorithm: yields a path that is 25% longer on average than the shortest path
- Christofides algorithm: yields a path that is less than 1.5 times the shortest path (no more than 50%)

Recent progress:
Jan 2013: Computer Scientists Find New Shortcuts for Infamous Traveling Salesman Problem