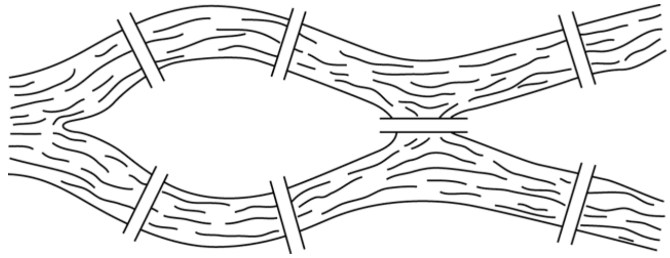
COMP 2121
DISCRETE MATHEMATICS

Lecture 14



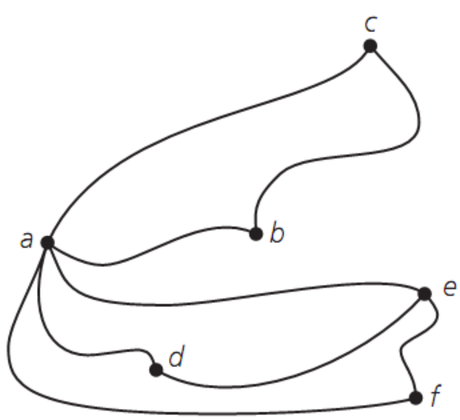
The people of Konigsberg wanted to know if it were possible to stroll in such a way that they could go over each bridge exactly once and return to the starting point.

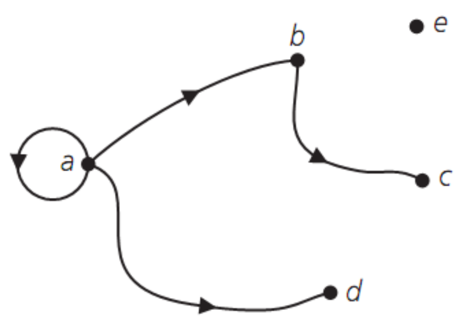
This problem was presented to famous mathematician, Leonard Euler, and his solution is often credited with being the beginning of graph theory.

Graphs

Graphs and trees are convenient visualizations to use in a variety of situations.

Definition. *A* ***graph*** *G, G = (V, E) consists of two finite sets: a set V(G) of* ***vertices*** *and a set of E(G) of* ***edges****, where each edge is associated with a set consisting of either one or two vertices. Two vertices that are connected by an edge are called* ***adjacent****. An edge with just one endpoint is called a* ***loop****. A vertex on which no edges are incident is called* ***isolated****.*





In a graph, the number of edges incident with a vertex v is called a **degree** of v and it is denoted as $\deg(v)$.

Observation: $\sum_{i=1}^{|V|} \deg(v_i) = 2 \cdot |E|$, or in plain English: In undirected graph, the sum of the degrees of the vertices equals twice the number of edges.

Example. Design a computer network with 7 computers, such that every computer is connected with 3 other computers.

Definition. *Let x, y be vertices in an undirected graph G. An x-y* ***walk*** *in G is a (loop-free) finite alternating sequence*

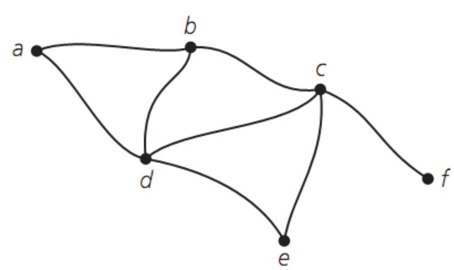
$$x = x_0, e_1, x_1, e_2, x_2, e_3, x_3, \ldots, e_{n-1}, x_{n-1}, e_n, x_n = y$$

of vertices and edges from G, starting at vertex x an ending at vertex y and involving the n edges.

The length of this walk is n, the number of edges in the walk. Any *x-y* walk where *x=y* is called closed walk.

If no edge in the *x-y* walk is repeated, then the walk is called an *x-y* trail. A closed *x-x* walk is called a circuit.

If no vertex of the *x-y* walk occurs more than once, then the walk is called an *x-y* **path**. When *x=y*, the term **cycle** is used to described a closed path.



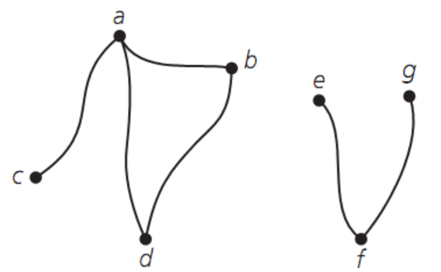
Example of walk:

Example of trail:

Example of path:

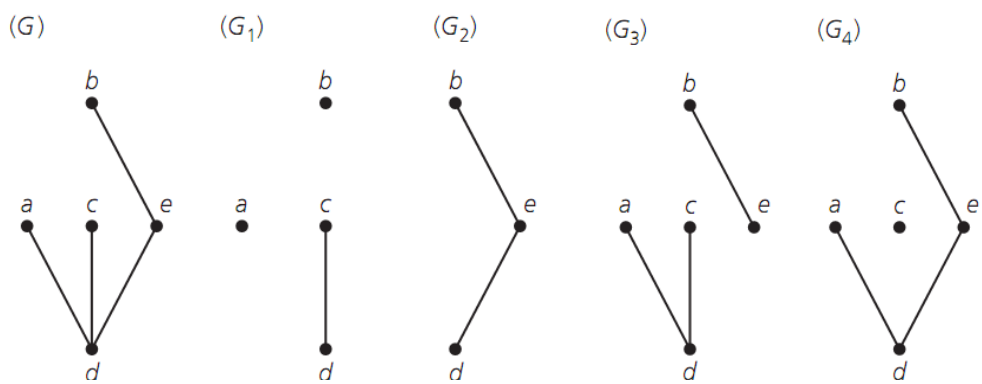
Example of cycle:

Definition. *Let G = (V, E) be an undirected graph. We call G* ***connected*** *if there is a path between any two distinct vertices of G. A graph that is not connected is called* ***disconnected****. The number of components of G is denoted $\kappa(G)$.*



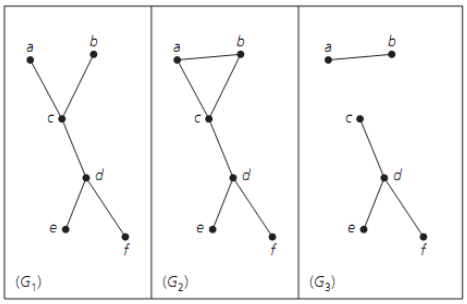
This graph has connected components.

Definition. *If $G=(V, E)$ is graph, then $G_1=(V_1, E_1)$ is called a* ***subgraph*** *of G if $V_1 \subseteq V$ and $E_1 \subseteq E$, where each edge in E_1 is incident with vertex in V_1. If $V_1=V$, then G_1 is called a* ***spanning subgraph*** *of G.*



Trees

Definition: *Let G=(V, E) be a loop free undirected graph. The graph G is called a **tree** if G is connected and contains no cycles. We denote tree with T = (V, E). If G is disconnected and each component of G is a tree, then G is called a **forest**.*

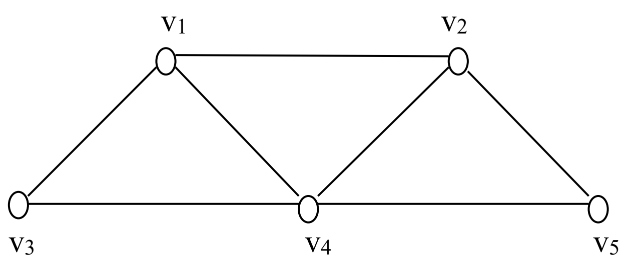


Observation: If *a* and *b* are distinct vertices in a tree *T=(V, E)* then there is a unique path that connects these vertices. (If there is more than one path then this implies cycle, which furthermore implies cycle in T – contradicting the definition of T.)

Theorem. In every tree T = (V, E), |V| = |E| + 1.

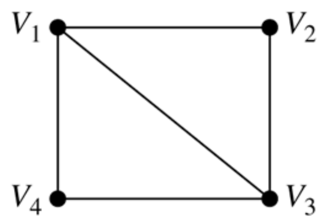
Observation: If G(V,E) is an undirected graph, then G is connected if and only if G has a spanning tree.

Example. Find spanning trees of the following graph:



Data representation of the graph is the adjacency matrix. Adjacency matrix of G = (V, E) is |V| x |V| matrix with 1 in entry v_iv_j if there is an edge connecting v_i and v_j, and 0 otherwise.

Example. What is the adjacency matrix of the following graph?



Example. Draw a graph that corresponds to the Adjacency Matrix A given below:

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

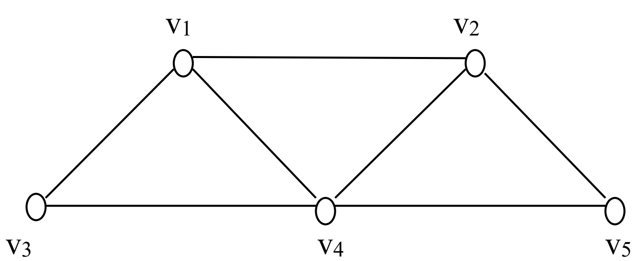
Breadth First Search Algorithm can be used to obtain a spanning tree of the graph. Hence, we can decide if graph is connected.

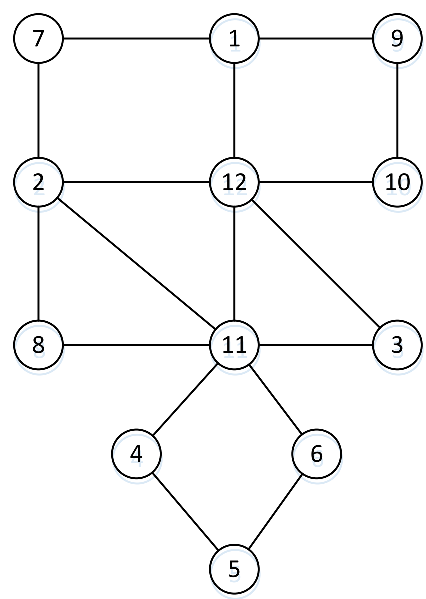
Breadth – First Search Algorithm.

Insert vertex v_x at the rear of (initially empty) queue Q and initialize T as the tree made up of this one vertex v_x. Visit v_x.

```
While (Q is not empty){
          Delete the vertex v from the front of the Q
          for each neighbor w of v  //in the increasing order
                    if w is unvisited
                    {
                    visit(w);
                    insert w at the rear of the Q;
                    add edge vw to tree T
                    }
}
```

BFS Example:





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

Algorhyme

I think that I shall never see
a graph more lovely than a tree.
A tree whose crucial property
is loop-free connectivity.
A tree that must be sure to span
so packet can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
then bridges find a spanning tree.

Radia Perlman