CSSE 230 Day 28

Graphs and their representations

Check out from SVN: RandomGraphs

Graphs

- Terminology
  Representations
  Algorithms

Example Graph

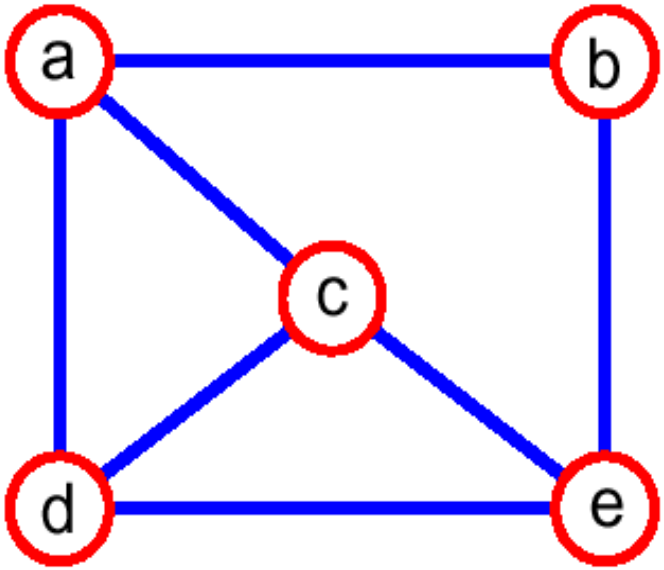
A **graph G** = (**V**,**E**) is composed of:

V: set of *vertices*

E: set of *edges* connecting the *vertices* in **V**

An **edge e** = (u,v) is a pair of vertices

Example:



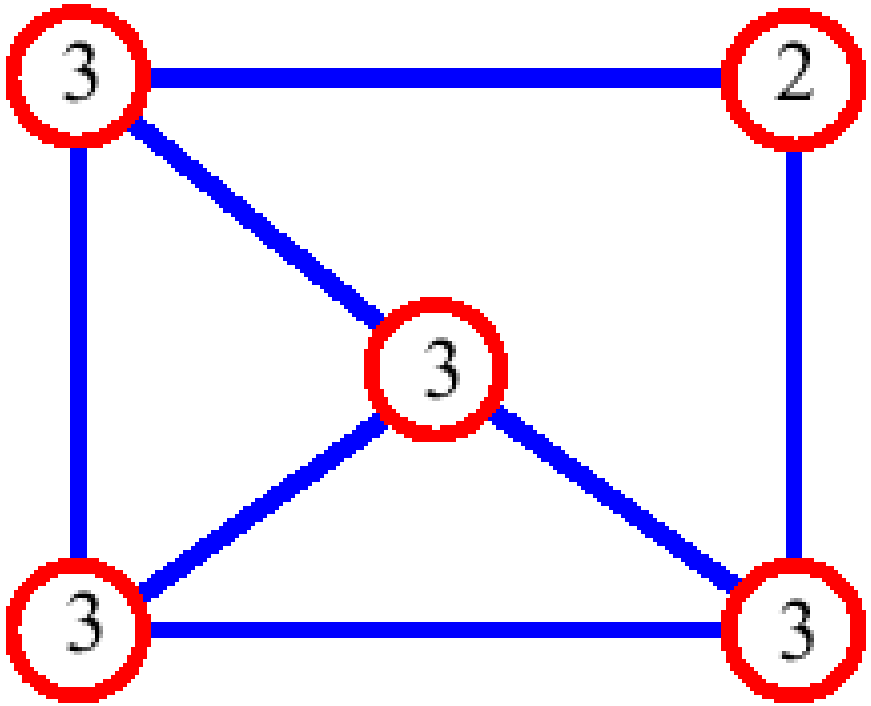
V= {a,b,c,d,e}

E=
{(a,b),(a,c),(a,d),
(b,e),(c,d),(c,e),
(d,e)}

Graph Terminology

also called "neighbors"

- adjacent vertices: connected by an edge
- degree (of a vertex): # of adjacent vertices

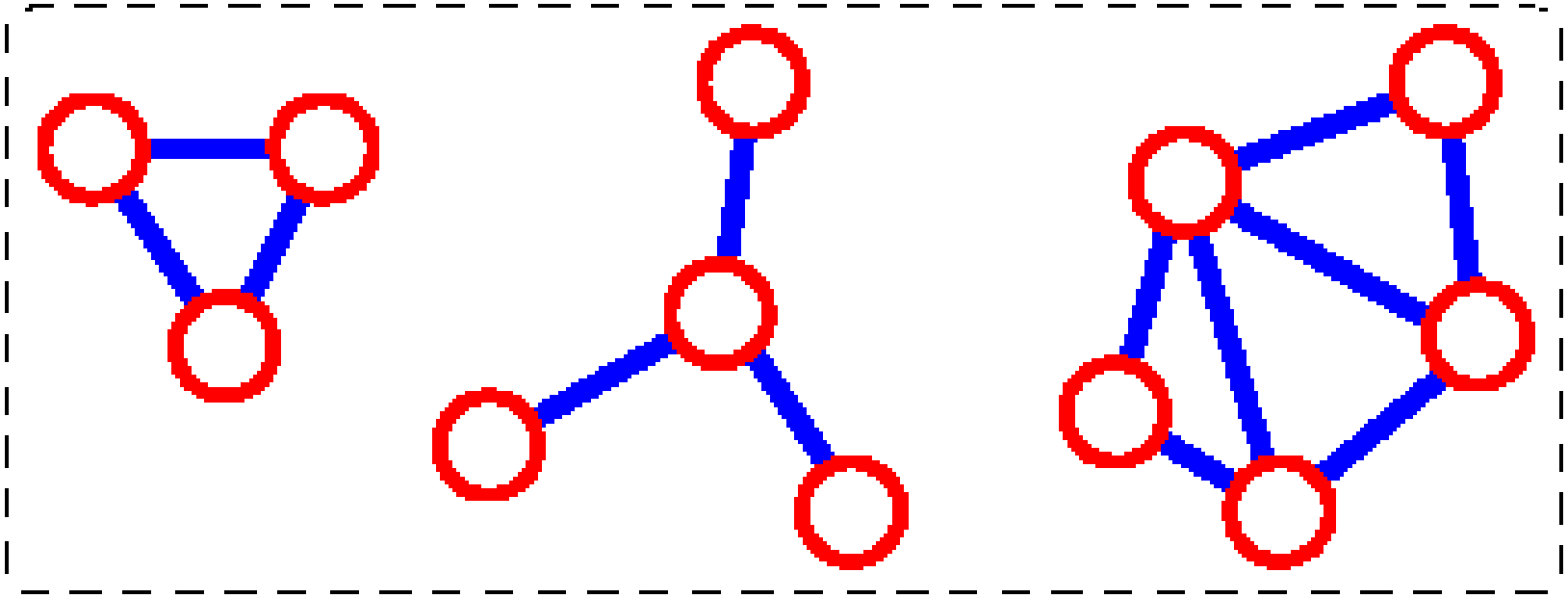


$$\sum_{v \in V} \deg(v) = 2(\#\text{ edges})$$

- Since adjacent vertices each count the adjoining edge, it will be counted twice

Continuing Graph Terminology

connected component: maximal connected subgraph. E.g., the graph below has 3 connected components.



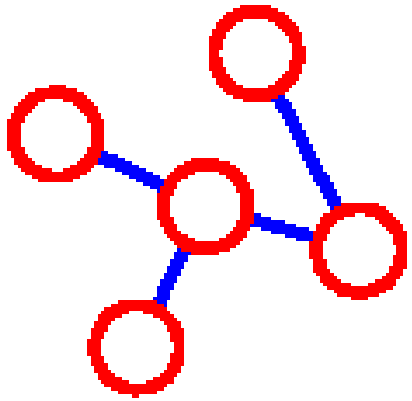
More Connectivity

n = #vertices

m = #edges

For a tree **m** = **n** - 1

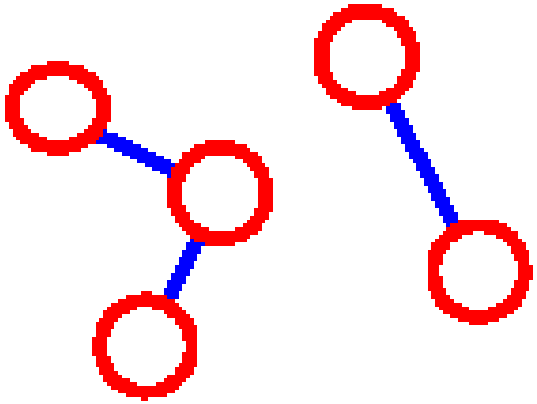
A necessary but not sufficient condition for a graph to be a tree.



n = 5

m = 4

If **m** < **n** - 1, G is not connected



n = 5

m = 3

We represent vertices using a collection of objects

- Each Vertex object contains information about itself
- Examples:
  - City name
  - IP address
  - People in a social network

There are many options for representing edges of a graph

- Adjacency matrix
- Adjacency list. Each vertex stores…
  - pointers to other vertices?
  - named vertices using a HashMap<Name,Vertex>
  - An index into an array of the Vertex objects In each case, we need a way to store the vertex collection
- Edge list

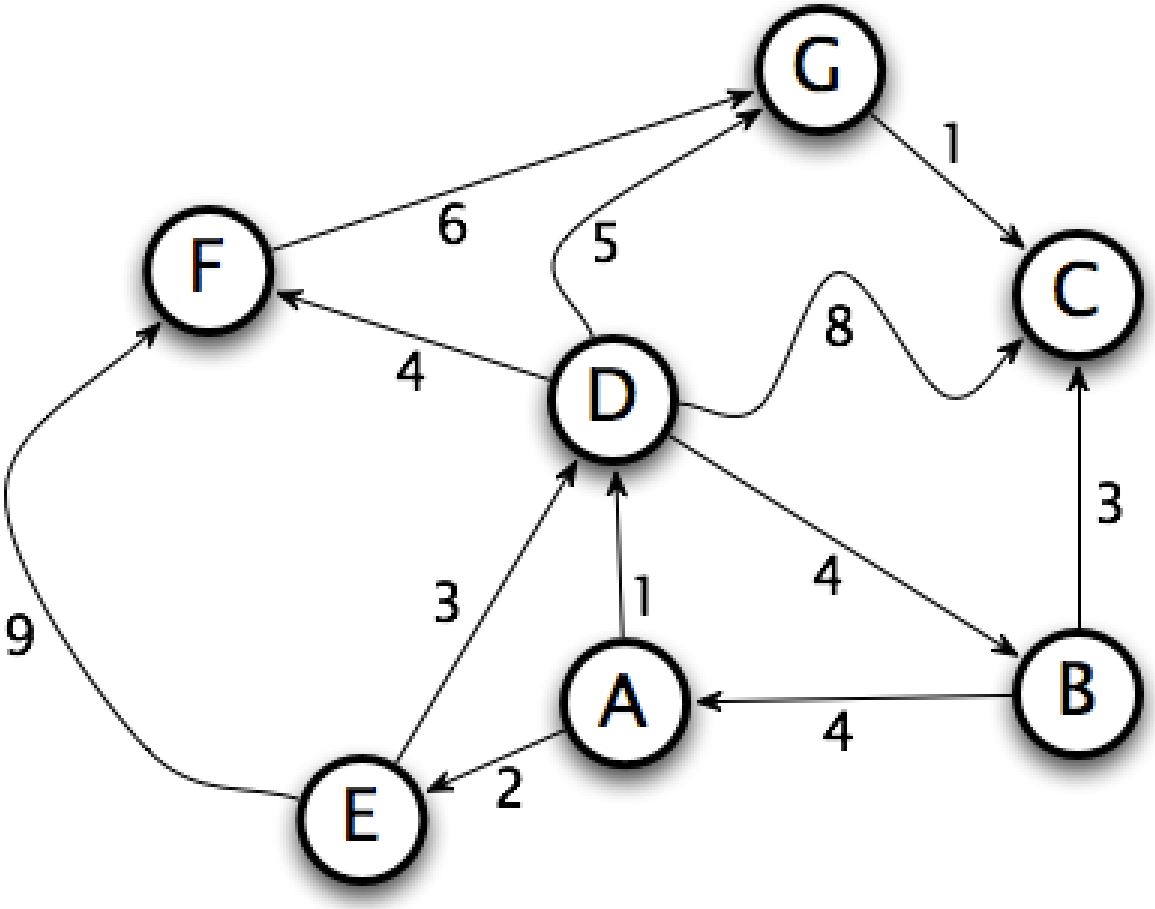
To consider:

Why not just use a triangular “matrix”?

Does a boolean adjacency matrix make sense?

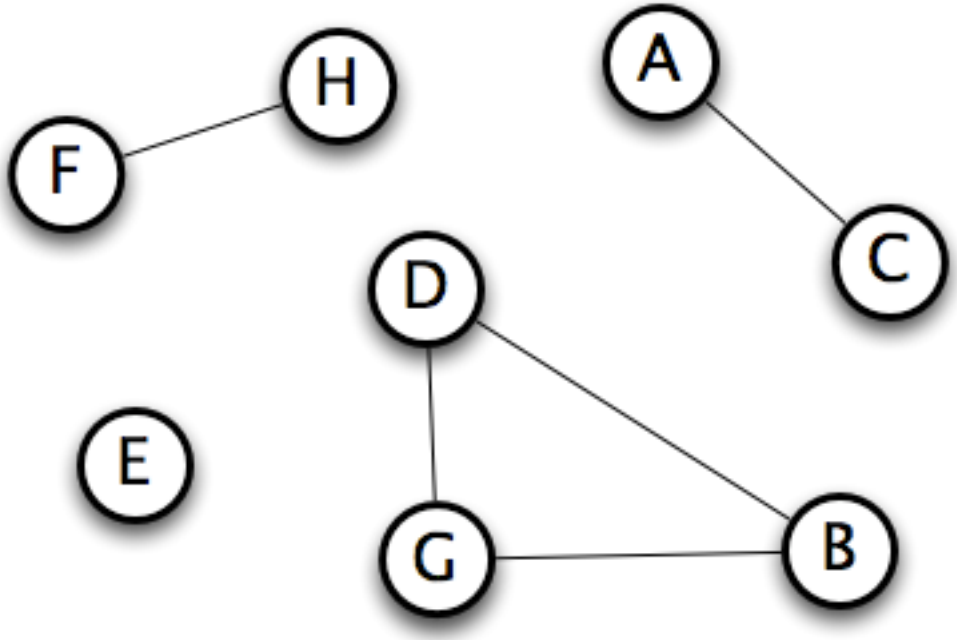
Sample graph problem: Weighted Shortest Path

- What's the cost of the shortest path from A to each of the other nodes in the graph?



Largest Connected Component

- What's the size of the largest connected component?



For much more on graphs, take MA/CSSE 473 or MA 477