

# CSSE 230 Day 28

## Graphs and their representations

After this lesson, you should be able to ...  
... explain what makes a graph different than a tree  
... implement simple graph algorithms

<https://www.google.com/maps/preview#!data=!1m4!1m3!1d989355!2d-87.4496039!3d38.8342589!4m2!6!3m1!7!1m5!1sRose-Hulman+Institute+of+Technology%2C+5500+Wabash+Ave%2C+Terre+Haute%2C+IN+47803!2s0x886d6e421b703737%3A0x96447680305ae1a4!3m2!3d39.482156!4d-87.322345!1m1!1sHoliday+World+%26+Splashin'+Safari%2C+Santa+Claus%2C+IN!3m8!1m3!1d245622!2d-86.923997!3d39.3256455!3m2!1i1920!2i955!4f13.1!5m2!13m1!1e1!7m4!11m3!1m1!1e1!2b1&fid=0>

# Announcements

- ▶ Hardy Part 2 partner evaluation due
  - Complete during worktime today
  
- ▶ But first...

# Graphs

Terminology

Representations

Algorithms

# Example Graph

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

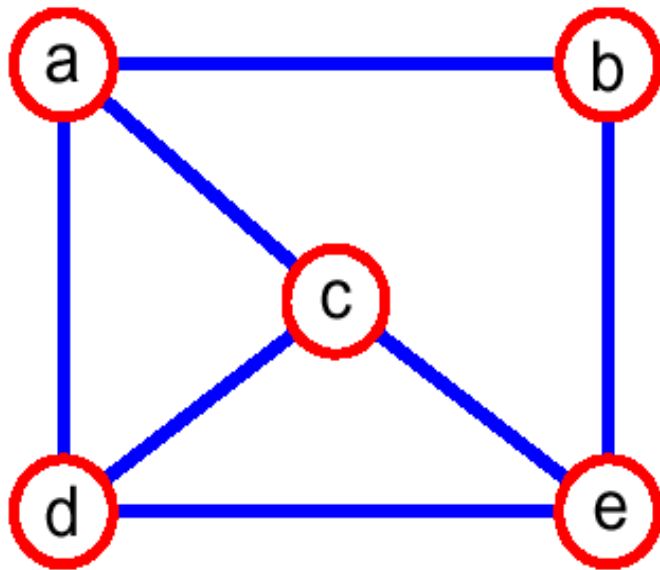
$V$ : set of *vertices* (singular: vertex)

$E$ : set of *edges*

An *edge* is a pair of *vertices*. Can be

unordered:  $e = \{u, v\}$  (*undirected* graph)

ordered:  $e = (u, v)$  (*directed* graph or *digraph*)



Example:

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

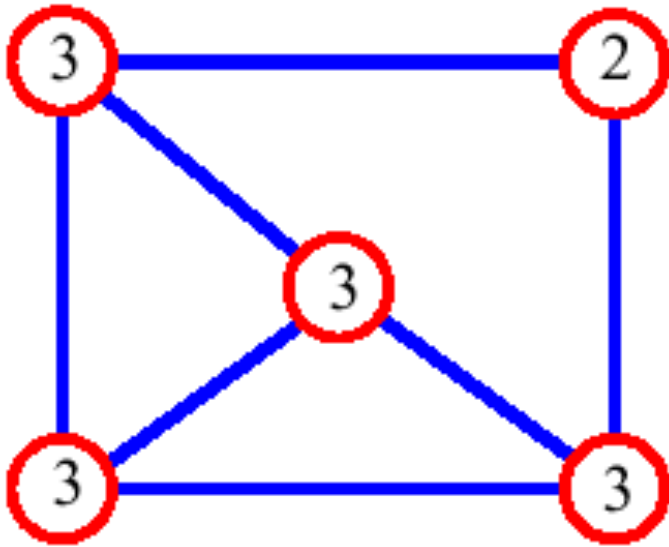
$E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}\}$

# Graph Terminology

- ▶ Size? Edges or vertices?
- ▶ Usually take size to be  $n = |V|$  (# of vertices)
- ▶ But the runtime of graph algorithms often depend on the number of edges,  $|E|$
- ▶ Relationships between  $|V|$  and  $|E|$ ?

# Graph Terminology

- *Adjacent* vertices or *neighbors*: connected by an edge
- Degree (of a vertex): # of incident edges

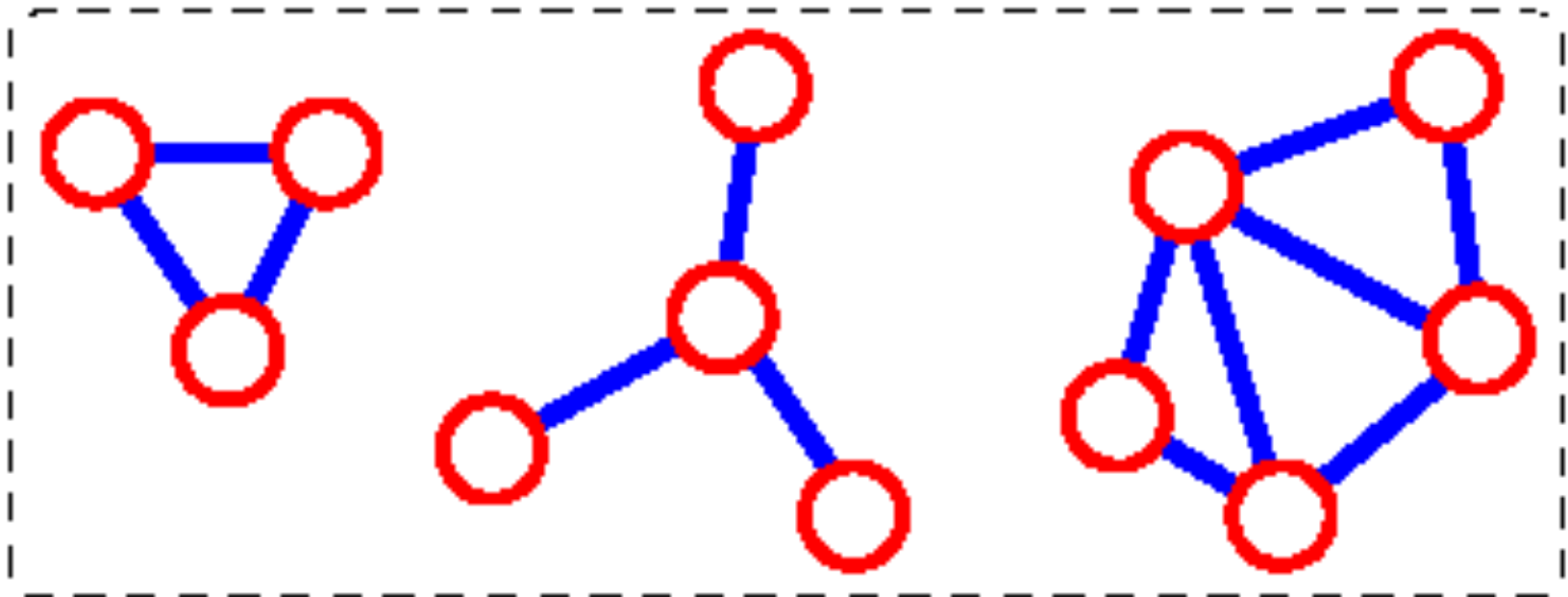


Fact:

$$\sum_{v \in V} \deg(v) = 2|E|$$

(Why?)

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



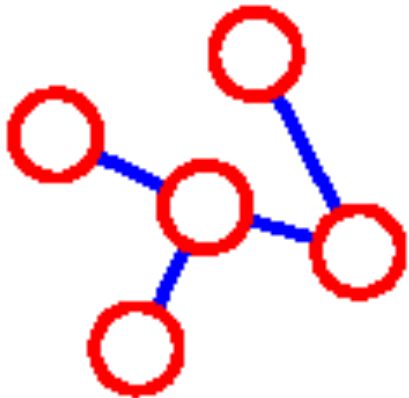
## More Connectivity

**n** = #vertices

**m** = #edges

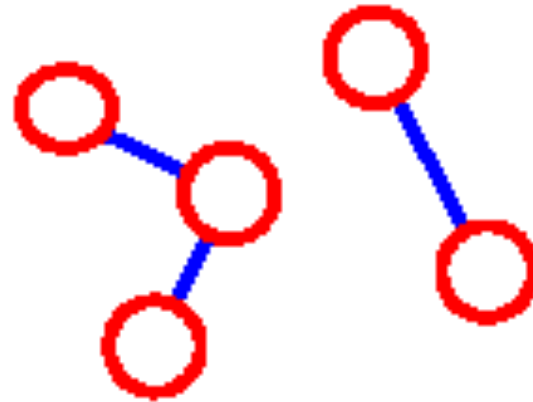
For a tree **m** = **n** - 1

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
- ▶ Observations about real graphs?

# There are many options for representing edges of a graph

- ▶ **Adjacency matrix**
  - 2D array ( $\#V \times \#V$ ) of booleans, or ints (weighted)
  - True or nonzero denotes edge
- ▶ **Adjacency list**
  - Collection of named vertices: `HashMap<String,Vertex>`
  - Each vertex stores a List of adjacent vertices
- ▶ **Edge list**
  - Graph object contains the collection of vertices and the collection of edges

To consider:

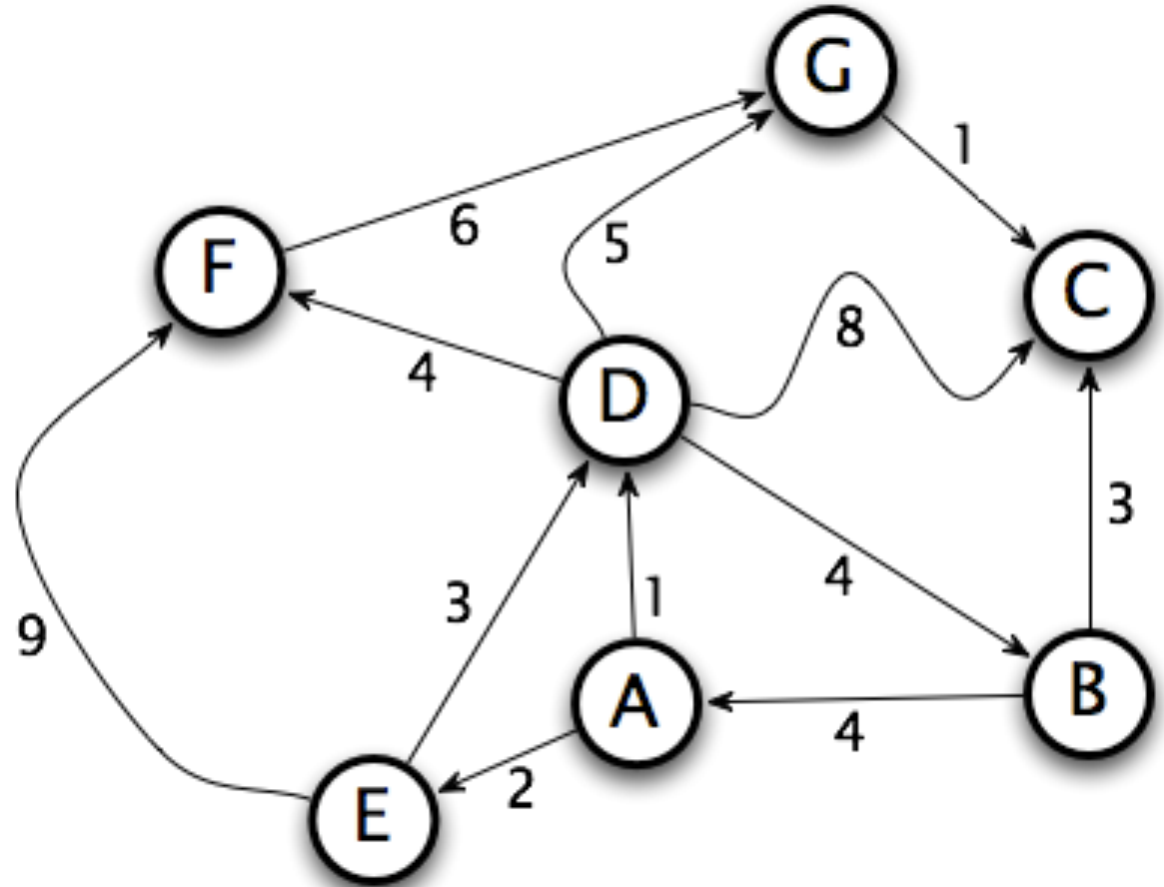
Why not just use a triangular “matrix”?

Does a boolean adjacency matrix make sense?

# Sample graph problems

## Weighted Shortest Path

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

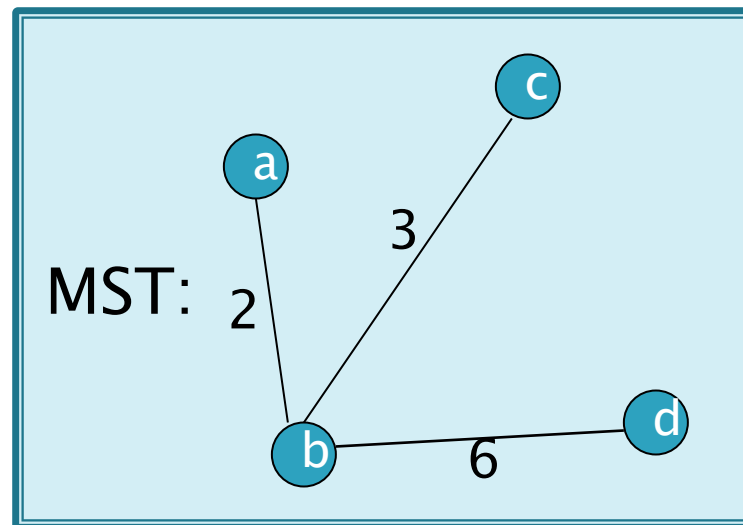
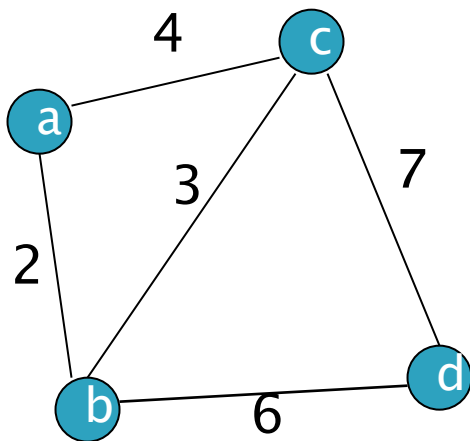


For much more on graphs, take MA/CSSE 473 or MA 477

# Minimum Spanning Tree

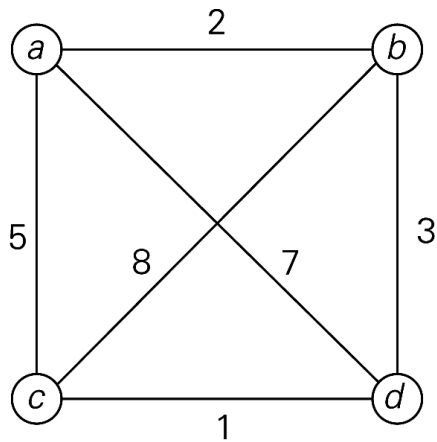
- ▶ *Spanning tree*: a connected acyclic subgraph that includes all of the graph's vertices
- ▶ *Minimum spanning tree* of a weighted, connected graph: a spanning tree of minimum total weight

Example:



# Traveling Salesman Problem (TSP)

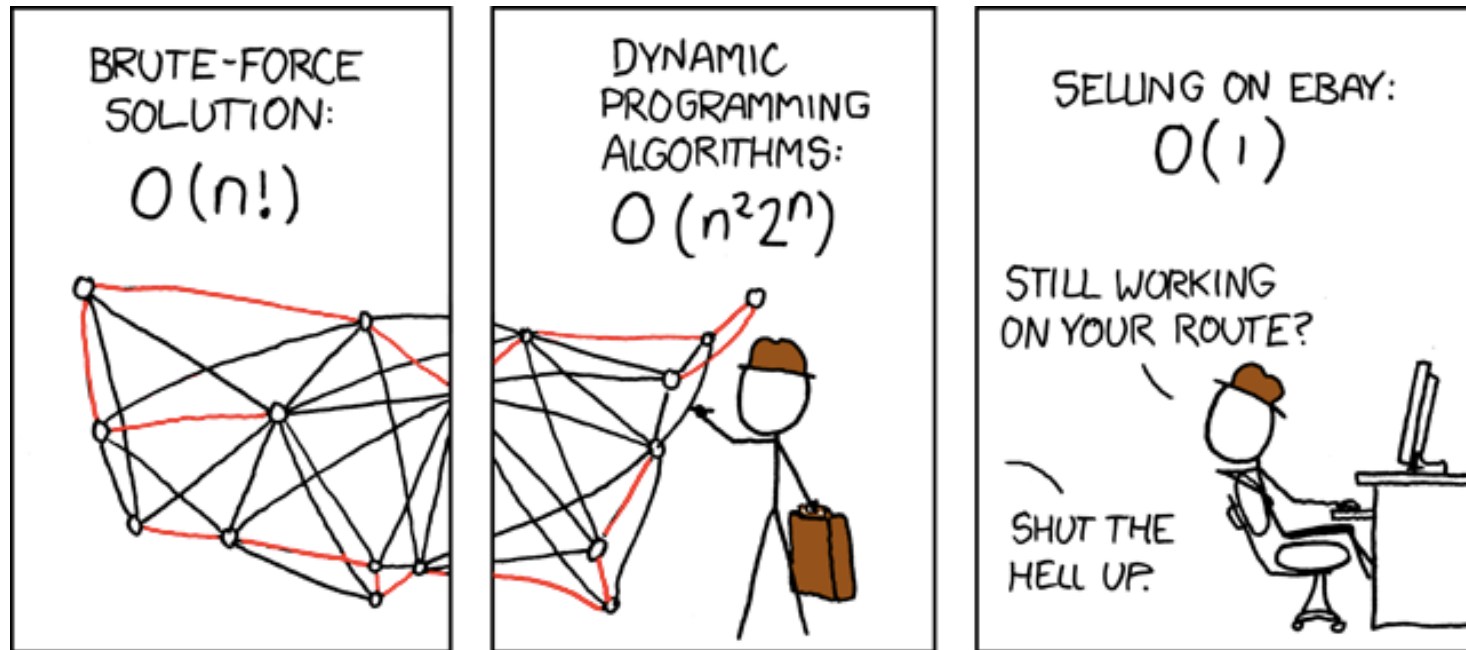
- ▶  $n$  cities, weights are travel distance
- ▶ Must visit all cities (starting & ending at same place) with shortest possible distance



<u>Tour</u>	<u>Length</u>	
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow a$	$l = 2 + 8 + 1 + 7 = 18$	
$a \rightarrow b \rightarrow d \rightarrow c \rightarrow a$	$l = 2 + 3 + 1 + 5 = 11$	optimal
$a \rightarrow c \rightarrow b \rightarrow d \rightarrow a$	$l = 5 + 8 + 3 + 7 = 23$	
$a \rightarrow c \rightarrow d \rightarrow b \rightarrow a$	$l = 5 + 1 + 3 + 2 = 11$	optimal
$a \rightarrow d \rightarrow b \rightarrow c \rightarrow a$	$l = 7 + 3 + 8 + 5 = 23$	
$a \rightarrow d \rightarrow c \rightarrow b \rightarrow a$	$l = 7 + 1 + 8 + 2 = 18$	

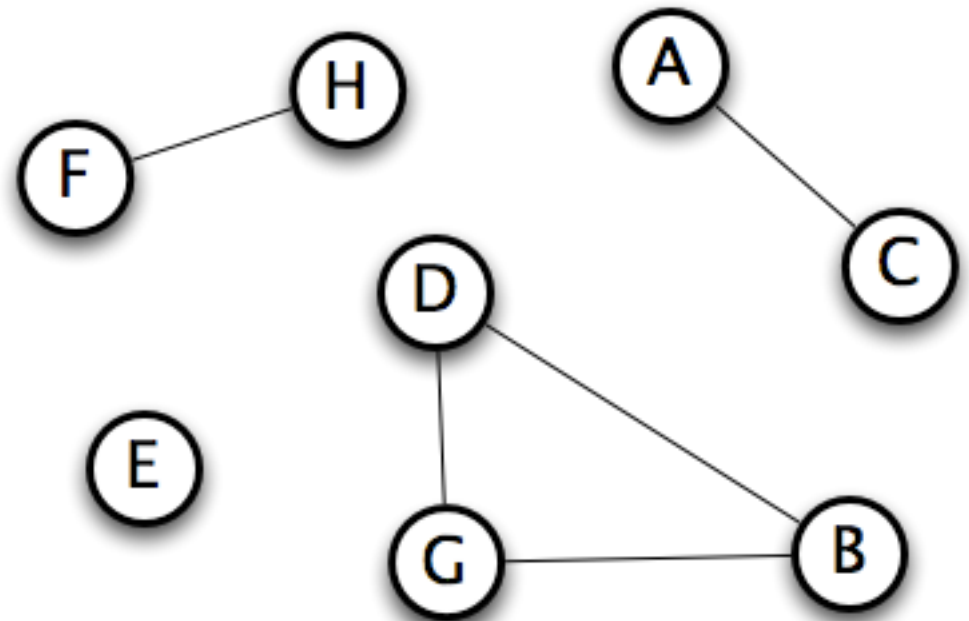
- Exhaustive search: how many routes?
- $(n-1)!/2 \in \Theta((n-1)!)$

# Traveling Salesman Problem



- ▶ Online source for all things TSP:
  - <http://www.math.uwaterloo.ca/tsp/>

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



In SVN: RandomGraphs