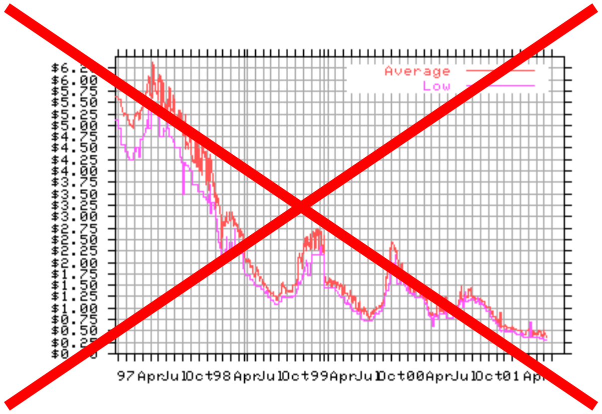
CSE 373

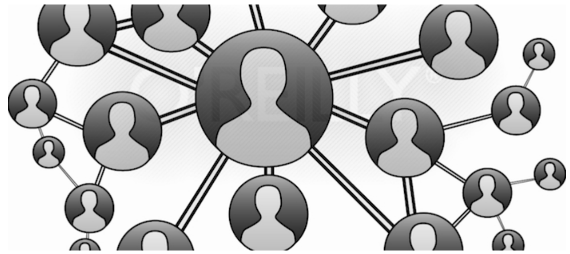
Graphs 1: Concepts,
Depth/Breadth-First Search

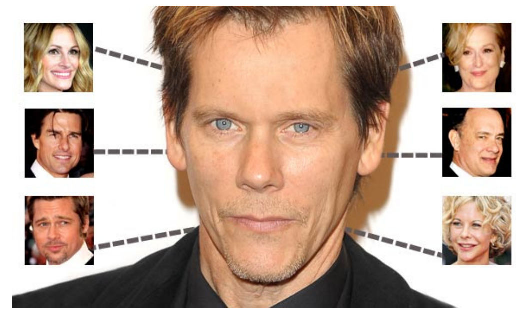
reading: Weiss Ch. 9

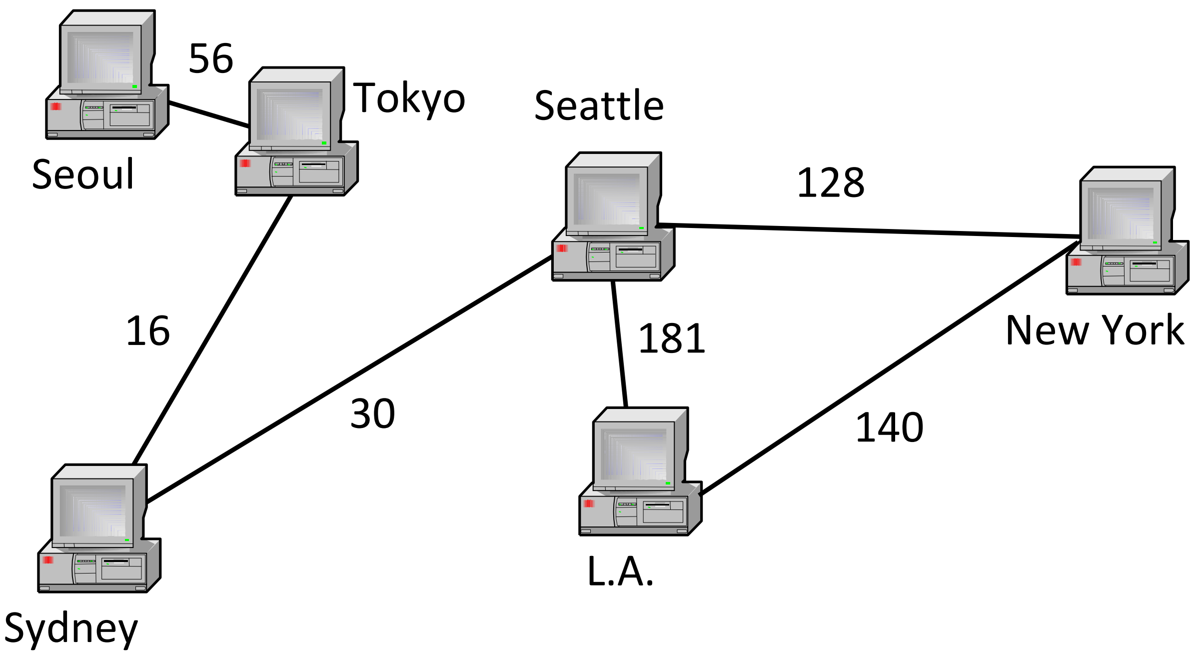
slides created by Marty Stepp
http://www.cs.washington.edu/373/

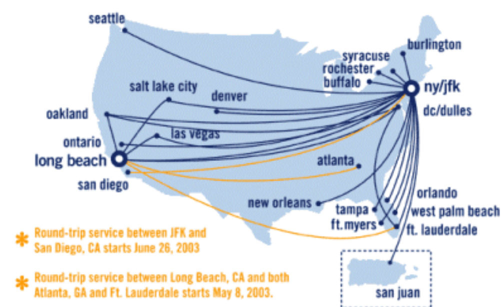
What is a graph?





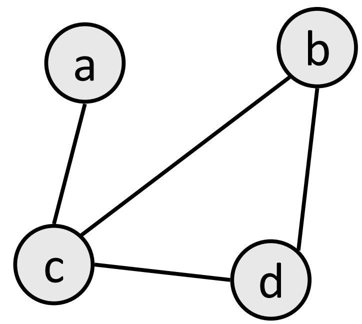






Graphs

- **graph**: A data structure containing:
  - a set of **vertices** *V*, *(sometimes called nodes)*
  - a set of **edges** *E*, where an edge represents a connection between 2 vertices.
    - Graph $G = (V, E)$
    - an edge is a pair (v, w) where v, w are in V

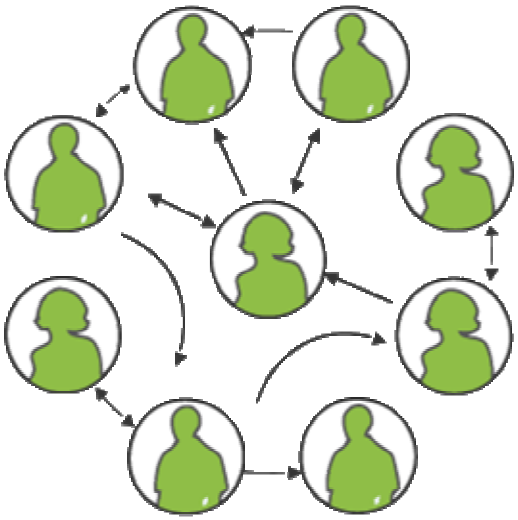


- the graph at right:
  - $V = \{a, b, c, d\}$
  - $E = \{(a, c), (b, c), (b, d), (c, d)\}$

- **degree**: number of edges touching a given vertex.
  - at right: a=1, b=2, c=3, d=2

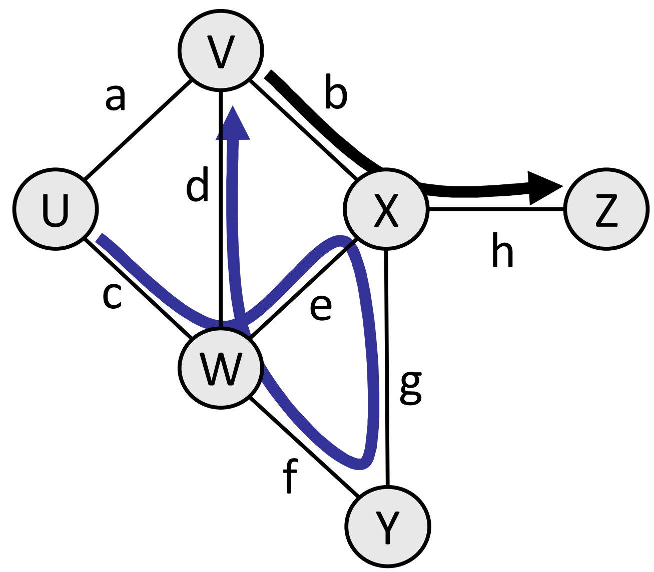
Graph examples

- For each, what are the vertices and what are the edges?
  - Web pages with links
  - Methods in a program that call each other
  - Road maps (e.g., Google maps)
  - Airline routes
  - Facebook friends
  - Course pre-requisites
  - Family trees
  - Paths through a maze



Paths

- **path**: A path from vertex *a* to *b* is a sequence of edges that can be followed starting from *a* to reach *b*.
  - can be represented as vertices visited, or edges taken
  - example, one path from *V* to *Z*: {b, h} or {V, X, Z}
  - What are two paths from U to Y?

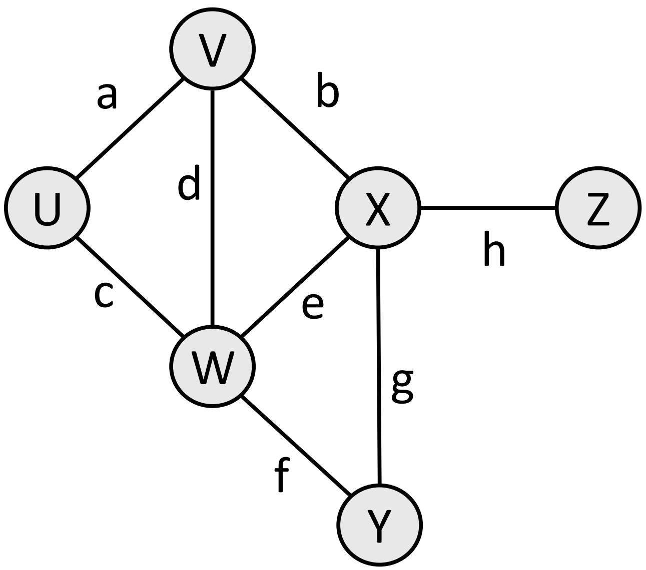


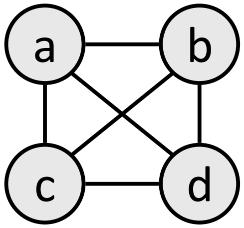
- **path length**: Number of vertices or edges contained in the path.

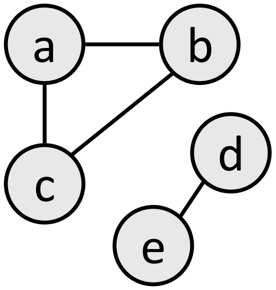
- **neighbor** or **adjacent***:* Two vertices connected directly by an edge.
  - example: V and X

Reachability, connectedness

- **reachable**: Vertex *a* is *reachable* from *b* if a path exists from *a* to *b*.
- **connected**: A graph is *connected* if every vertex is reachable from any other.
  - Is the graph at top right connected?
- **strongly connected**: When every vertex has an edge to every other vertex.

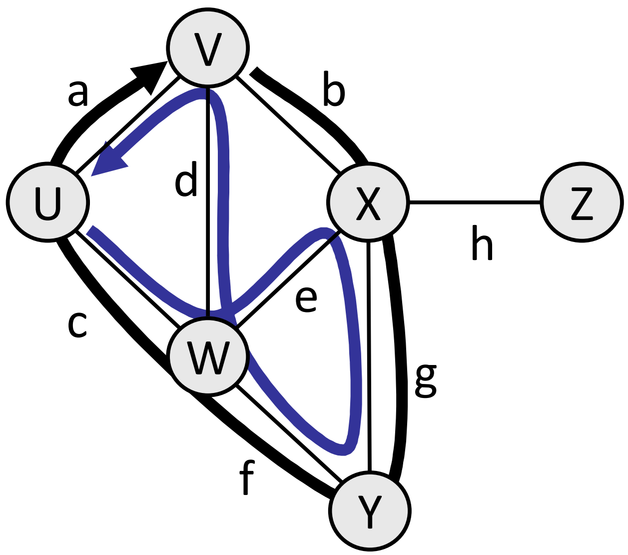






Loops and cycles

- **cycle**: A path that begins and ends at the same node.
  - example: {b, g, f, c, a} or {V, X, Y, W, U, V}.
  - example: {c, d, a} or {U, W, V, U}.
  - **acyclic graph**: One that does not contain any cycles.

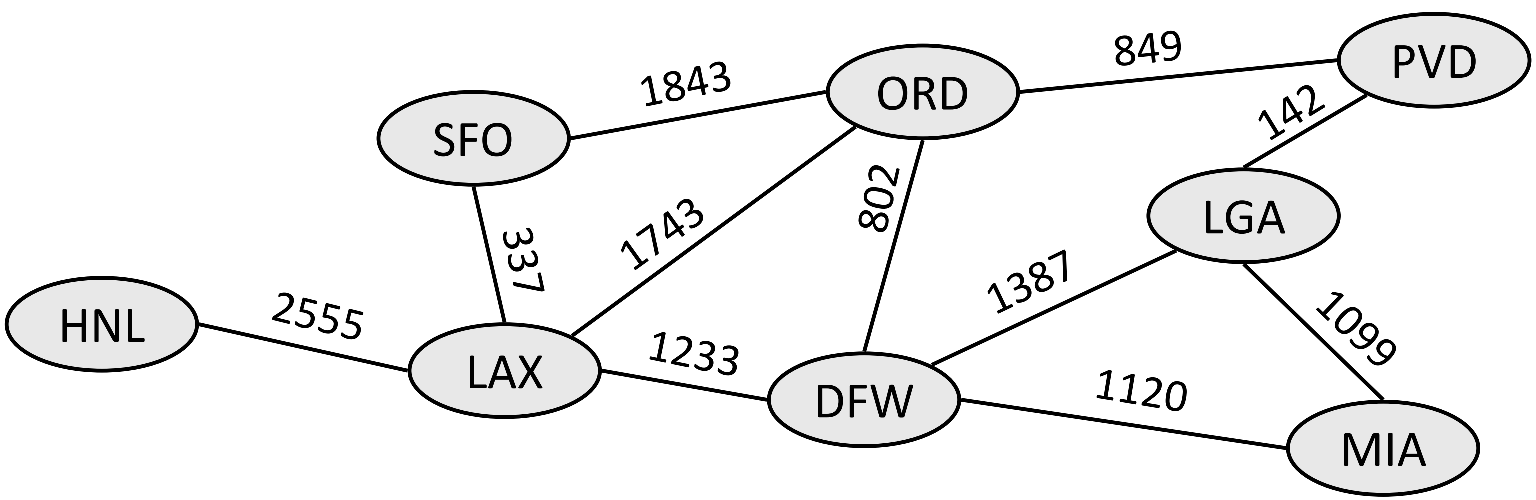


- **loop**: An edge directly from a node to itself.
  - Many graphs don't allow loops.

Weighted graphs

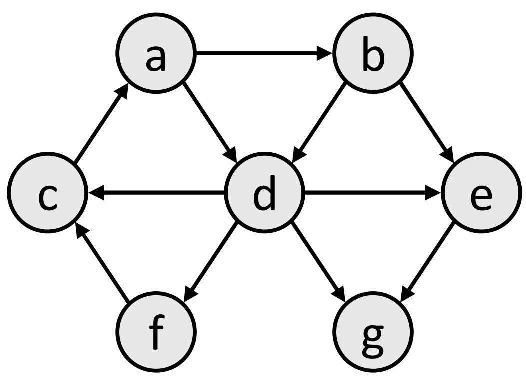
- **weight**: Cost associated with a given edge.
  - Some graphs have weighted edges, and some are unweighted.
  - Edges in an unweighted graph can be thought of as having equal weight (e.g. all 0, or all 1, etc.)
  - Most graphs do not allow negative weights.

- *example*: graph of airline flights, weighted by miles between cities:



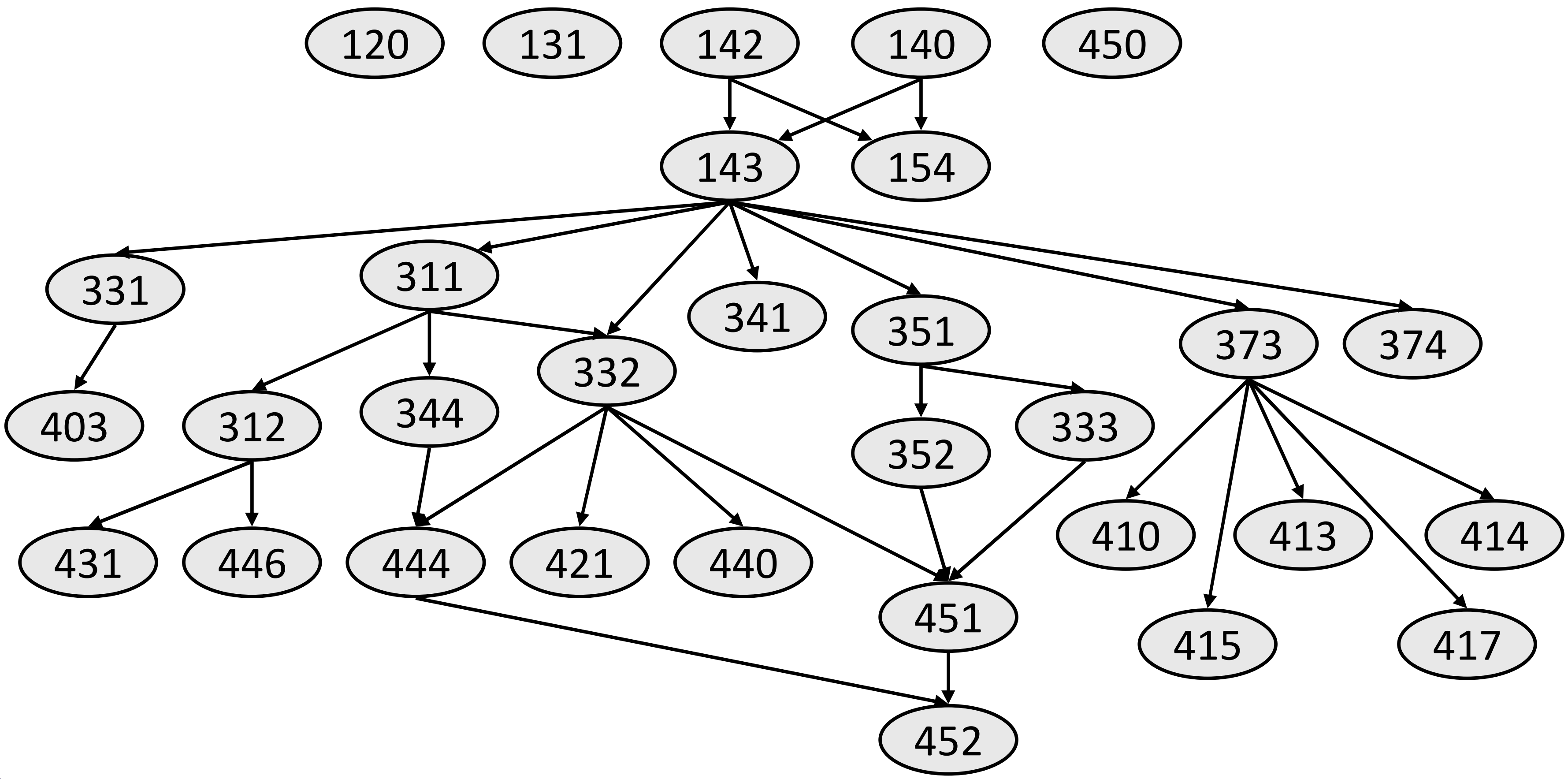
Directed graphs

- **directed graph** ("digraph"): One where edges are *one-way* connections between vertices.
  - If graph is directed, a vertex has a separate in/out degree.
  - A digraph can be weighted or unweighted.
  - Is the graph below connected? Why or why not?



Digraph example

- Vertices = UW CSE courses (incomplete list)
- Edge (a, b) = *a* is a prerequisite for *b*



Linked Lists, Trees, Graphs

- A *binary tree* is a graph with some restrictions:
  - The tree is an unweighted, directed, acyclic graph (DAG).
  - Each node's in-degree is at most 1, and out-degree is at most 2.
  - There is exactly one path from the root to every node.

- A *linked list* is also a graph:
  - Unweighted DAG.
  - In/out degree of at most 1 for all nodes.