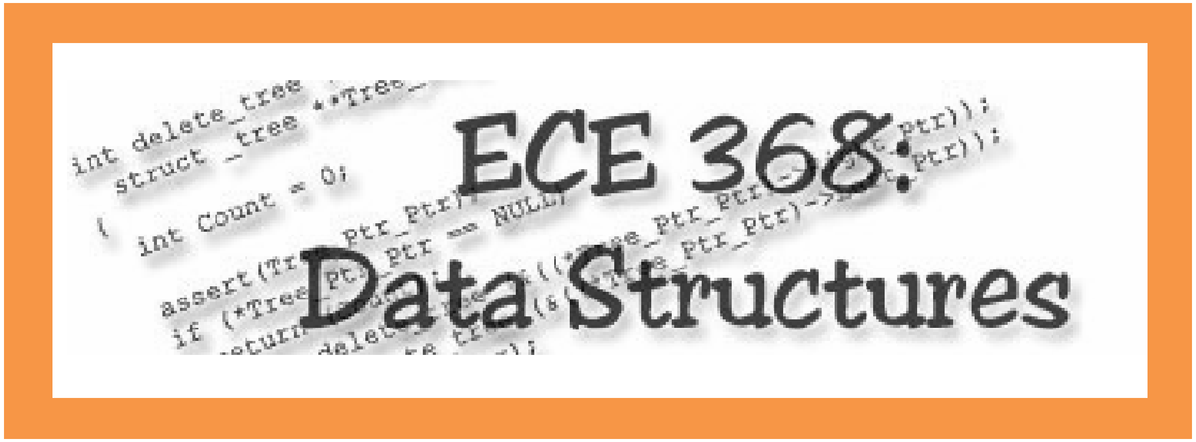


Felix Xiaozhu Lin

Lecture 21: Introduction to Graphs

Graph Definitions and Examples:
http://www.personal.kent.edu/~rmuhamma/GraphTheory/MyGraphTheory/defEx.htm

Slides Courtesy: Prof. Vijay Raghunathan

Introduction to Graphs

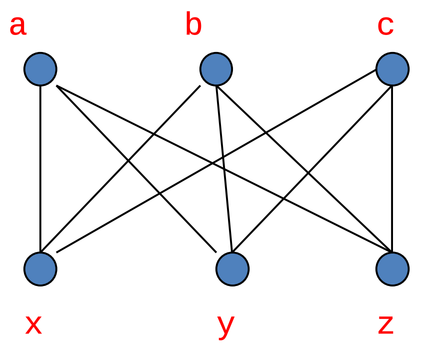
- Graph: Set of vertices (also called nodes) that are connected by edges

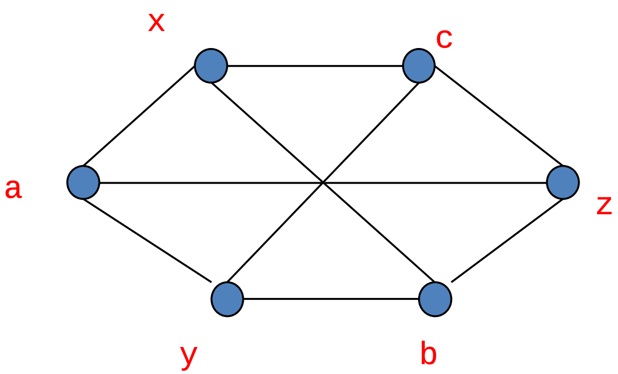
- *Graph $G = (V, E)$ consists of two sets:*
  - V is a finite set of *vertices (nodes)*
  - E is a set of *edges*, where each edge is a pair of vertices

- A vertex (node) is represented by a point or circle and an edge by a line joining the relevant pair of points

Introduction to Graphs

- A graph can be drawn in different ways





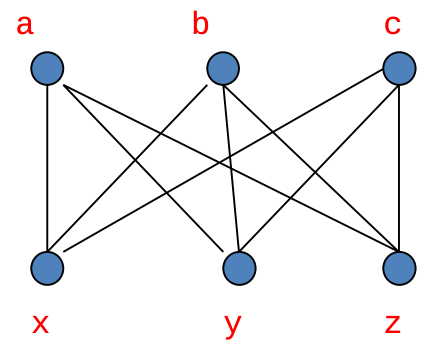
Two drawings of the *same* graph G

- Any binary relation is a graph, so a graph can be used to represent essentially any relationship

- To understand many important problems, we must think of in terms of graphs!

Graph Terminology

- Two vertices are *neighbors* if there exists an edge between them. In the below graph, vertices x, y, and z are *neighbors* of a since (a, x) $\in$ E, (a, y) $\in$ E, and (a, z) $\in$ E, whereas vertices a & b are *non-neighbors*



- Friendship Graph: The vertices are people and an edge exists between two people if and only if they are friends

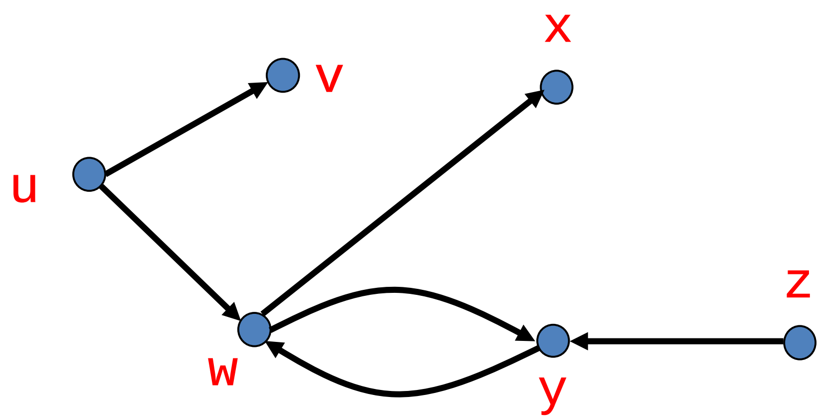
- This friendship graph is well-defined on any set of people: 368 students, Purdue students, people living in Indiana, etc.

- What questions can we ask about this?

Directed Graphs

- If I am your friend, does that mean that you are my friend?

- A graph is a *directed graph* (*digraph*) if
  - each edge in E is an *ordered pair* (x, y) of elements of V
  - edges are drawn as *directed* lines
  - E.g.:



- In a digraph, $(u, v) \in E$ *does not* mean that $(v, u) \in E$.

- If $(u, v) \in E$ implies that $(v, u) \in E$, then the graph is undirected

- Heard-of graph: Directed/Undirected?

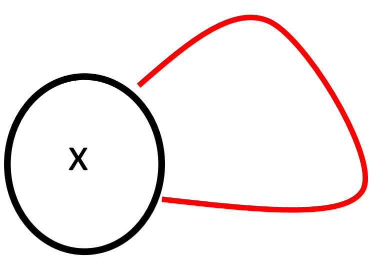
Simple Graphs

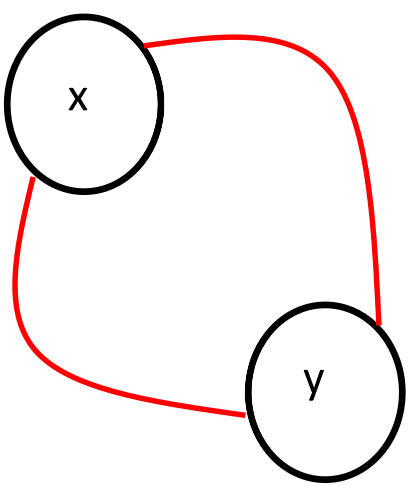
- Am I my own friend?

- An edge (x, x) is said to be a loop

- If x is y's friend many times over, that could be modeled using multiple edges between x and y

- A graph that has no loops and no multi-edges is called a simple graph

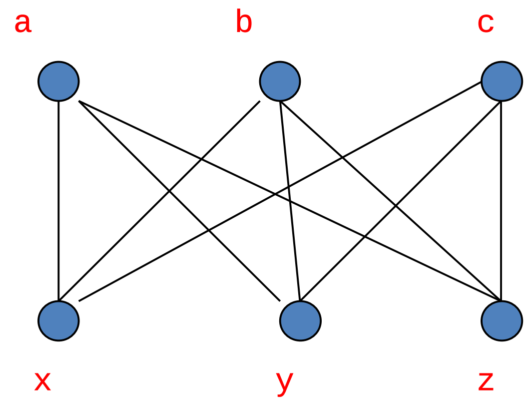




Paths and Cycles

- Is there a friendship link between me and President Obama?

- A *path* is a sequence of edges connecting two vertices. In the below graph, a - x - b - y - c is a path of length 4



- How long will it take for some gossip that I started to get back to me?

- A *cycle* is a path that starts and ends at the same vertex. In the above graph, a - x- c- z- a is a cycle of length 4

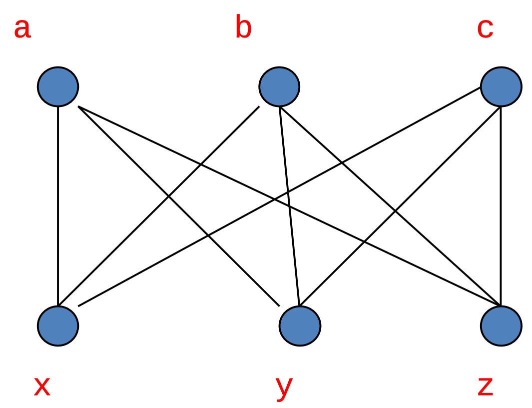
- A cycle in which no vertex repeats is called a simple cycle.

- In a digraph, paths and cycles must follow edge directions

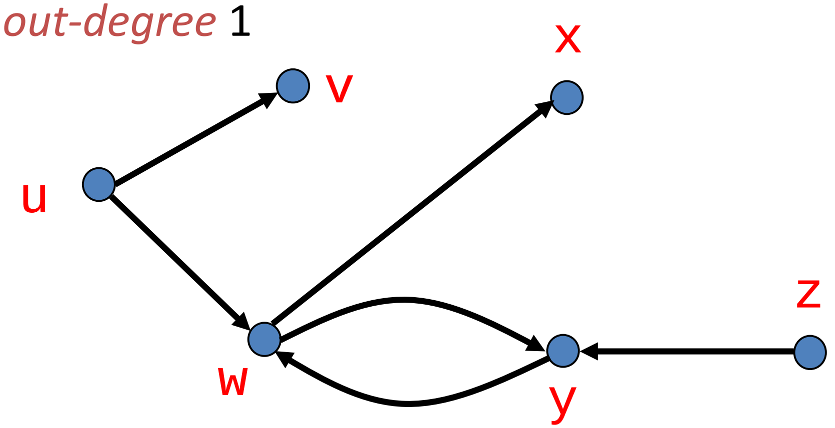
Degree of a vertex

- How many friends do I have?

- The *degree* of a vertex is the number of edges attached to it. In the below graph, every vertex has a degree of 3

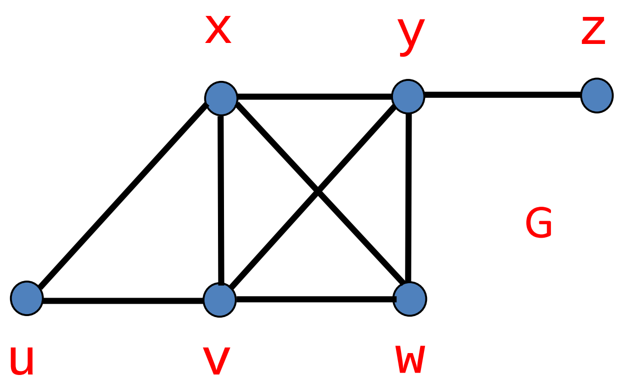


- The *in-degree* of a vertex is the number of edges entering the vertex
- The *out-degree* of a vertex is the number of edges leaving a vertex
- In below, y has *in-degree* 2 & *out-degree* 1



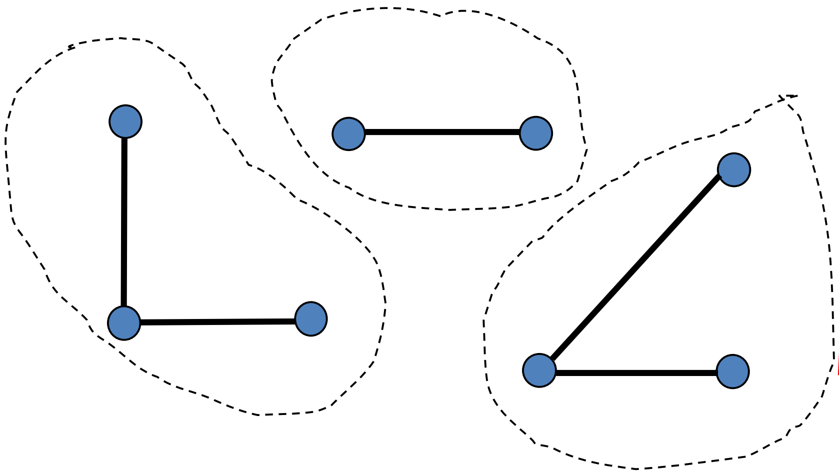
An undirected graph is:

- *Connected*, if every pair of vertices is joined by a path, e.g.,

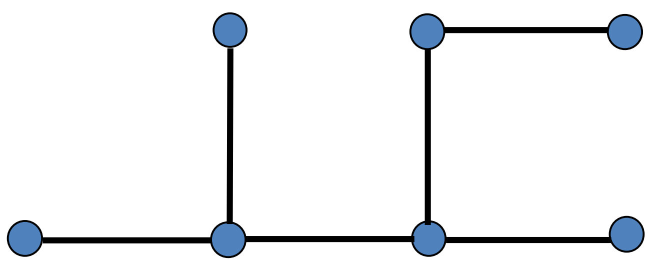


An undirected graph is:

- A non-connected graph has two or more *connected components*, *e.g.*,



- A *free tree* if it is connected and *acyclic*

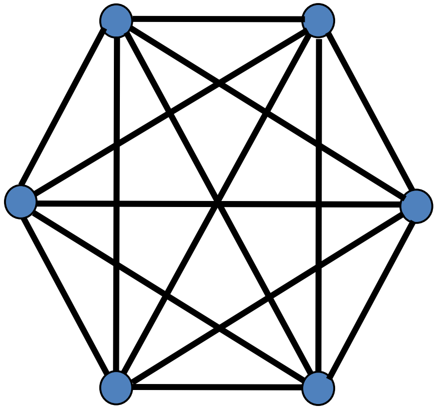


A free tree with n vertices has $n-1$ edges

- A *forest* if it is *acyclic*
  - Its connected components are free trees

A graph is:

- *Complete*, or *a clique* if every pair of vertices is joined by an edge



HANDSHAKE LEMMA:

- In any graph G = (V, E), the *sum of the degrees* of all the vertices in V is *even* and is equal to *2 × |E|*

- For a directed graph D = (V, E), the *sum of the in-degrees* of all vertices is equal to *|E|* and the *sum of the out-degrees* of all vertices is equal to *|E|*

Summary of Graph Concepts

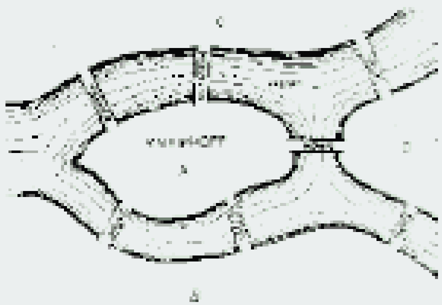
- Edge/vertex
- Neighbors
- Directed/undirected graphs
- Loop/path/cycles
- Component/tree/forest
- Degree
- Connected/complete

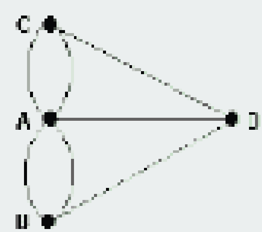
A brief history of graphs

- First known application of graph theory

The Seven Bridges of Königsberg. [Leonhard Euler 1736]

"... in Königsberg in Prussia, there is an island *A*, called the Kneiphof; the river which surrounds it is divided into two branches ... and these branches are crossed by seven bridges. *Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he could cross each bridge once and only once...*"



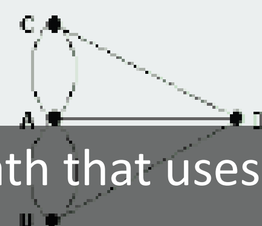


A brief history of graphs

- First known application of graph theory

The Seven Bridges of Königsberg. [Leonhard Euler 1736]

"... in Königsberg in Prussia, there is an island *A*, called the Kneiphof; the river which surrounds it is divided into two branches ... and these branches are crossed by seven bridges. Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he could cross each bridge once and only once..."

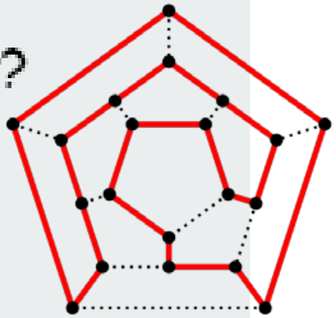


Is there a cyclic path that uses each edge exactly once?

Answer: Yes, if and only if the graph is connected and *all* vertices have an *even* degree. Such a cycle is called an “Euler tour” or “Euler cycle”

A few graph questions

Path. Is there a path between s to t?
Shortest path. What is the shortest path between s and t?
Longest path. What is the longest simple path between s and t?



Cycle. Is there a cycle in the graph?
Euler tour. Is there a cycle that uses each edge exactly once?
Hamilton tour. Is there a cycle that uses each vertex exactly once?

Connectivity. Is there a way to connect all of the vertices?
MST. What is the best way to connect all of the vertices?
Biconnectivity. Is there a vertex whose removal disconnects the graph?

Planarity. Can you draw the graph in the plane with no crossing edges?

Representation: undirected graph G

Adjacency matrix for G

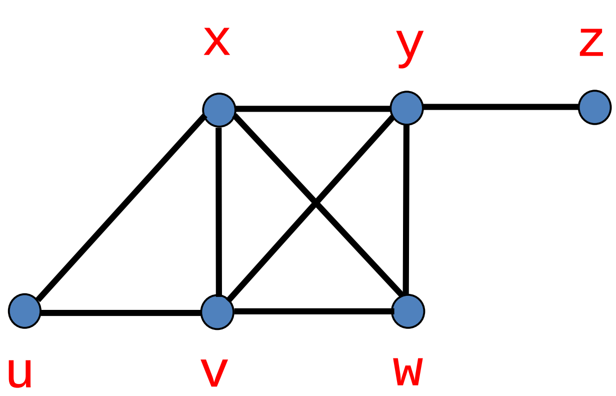
	u	v	w	x	y	z
u:	0	1	0	1	0	0
v:	1	0	1	1	1	0
w:	0	1	0	1	1	0
x:	1	1	1	0	1	0
y:	0	1	1	1	0	1
z:	0	0	0	0	1	0

|V| x |V| array

Adjacency lists for G

```
u: v-->x
v: u-->w-->x-->y
w: v-->x-->y
x: u-->v-->w-->y
y: v-->w-->x-->z
z: y
```

2×|E| entries in all



G

Representation: Directed graph D

Adjacency matrix for D

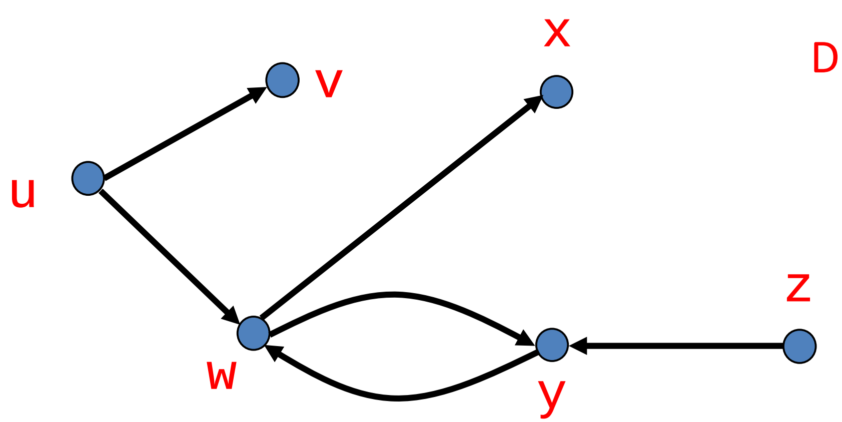
	u	v	w	x	y	z
u:	0	1	1	0	0	0
v:	0	0	0	0	0	0
w:	0	0	0	1	1	0
x:	0	0	0	0	0	0
y:	0	0	1	0	0	0
z:	0	0	0	0	1	0

|V| x |V| array

Adjacency lists for D

```
u: v-->w
v:
w: x-->y
x:
y: w
z: y
```

|E| entries in all



Graph representations

- *Adjacency matrix*
  – one row and one column for each vertex
  – the element at row i & column j contains a 1 if there is an edge between vertex i to vertex j, contains 0 otherwise

- *Adjacency lists*
  - one list for each vertex
  - list i is a linked list of all the neighbors of node i

For directed graphs:

- *Adjacency matrix*
  - one row and column for each vertex
  - the element at row i & column j contains a 1 if there is an edge *from* vertex i to vertex j, contains 0 otherwise

- *Adjacency lists*
  - one list for each vertex
  - list i is a linked list of all nodes j such that there is an edge *from* vertex i *to* vertex j