NORTHWESTERN
UNIVERSITY

MSiA 430: Software Tools for BDA

Spring 2023

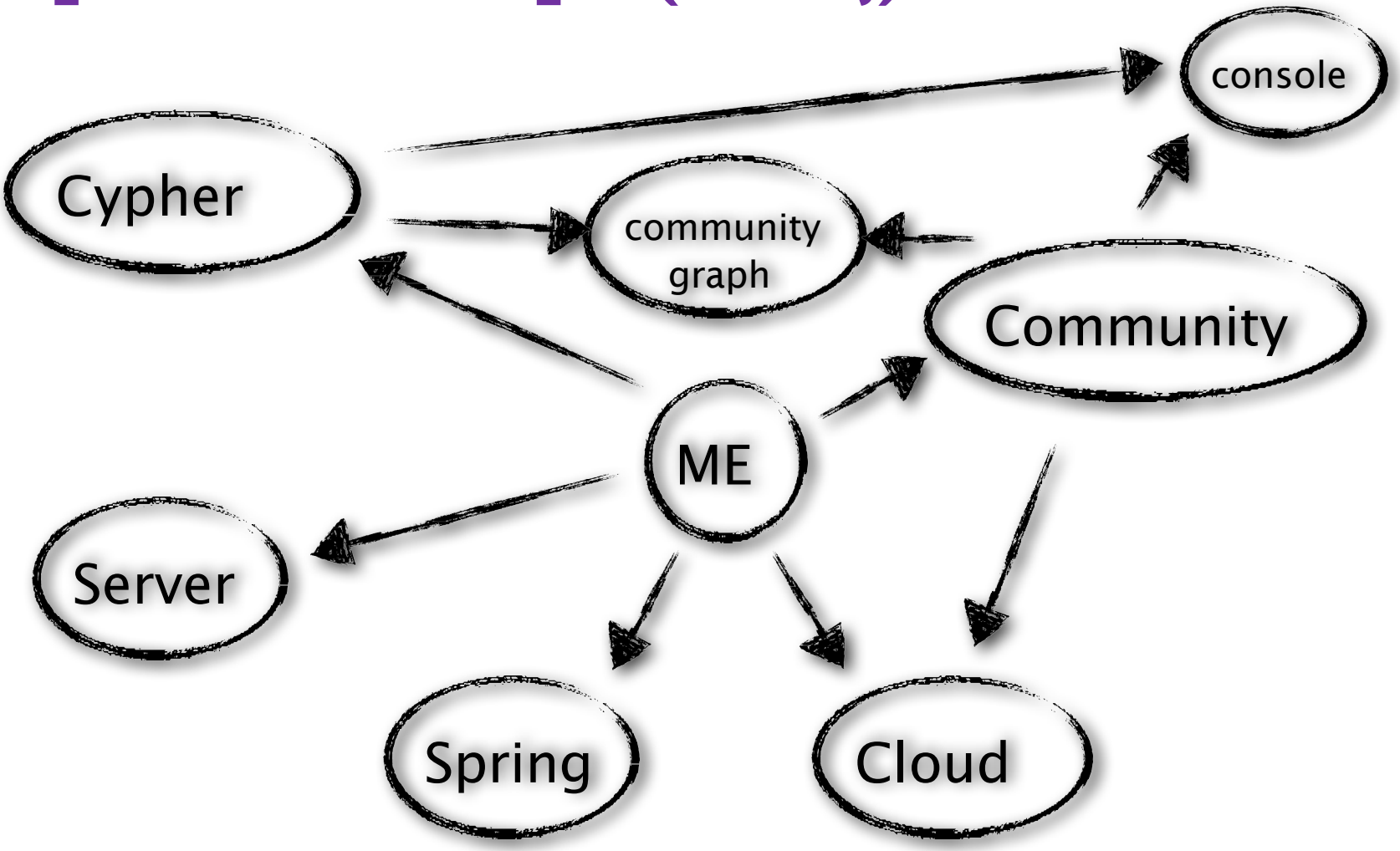
GraphDB – Part 1

Instructor: Goce Trajcevski

TA: Elliot Gardner

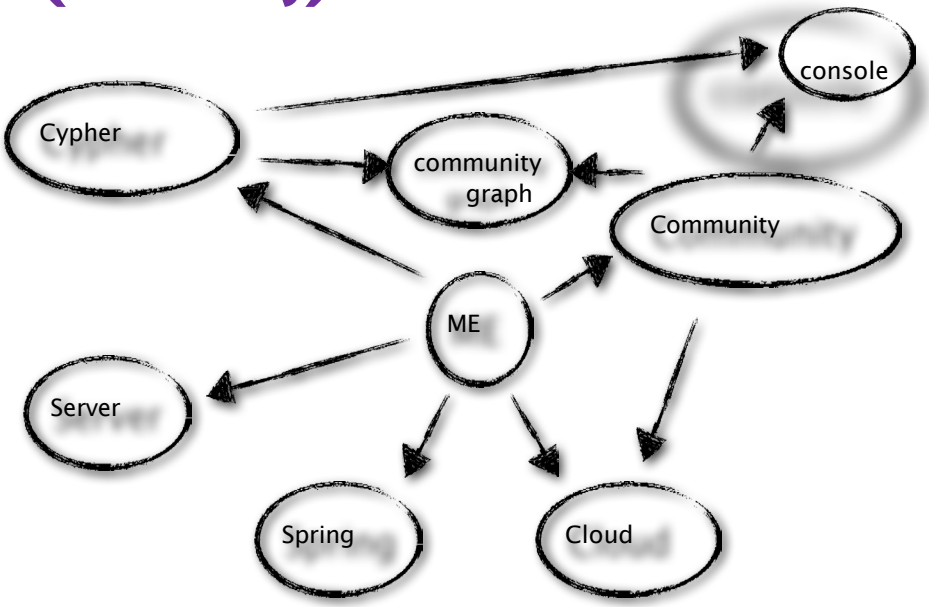
- Intro/Motivation
- Basics of Graph Theory
  - Representations
  - Algorithms
- Relational vs. GraphDB
- Basics of GraphDB
- Design (case studies)

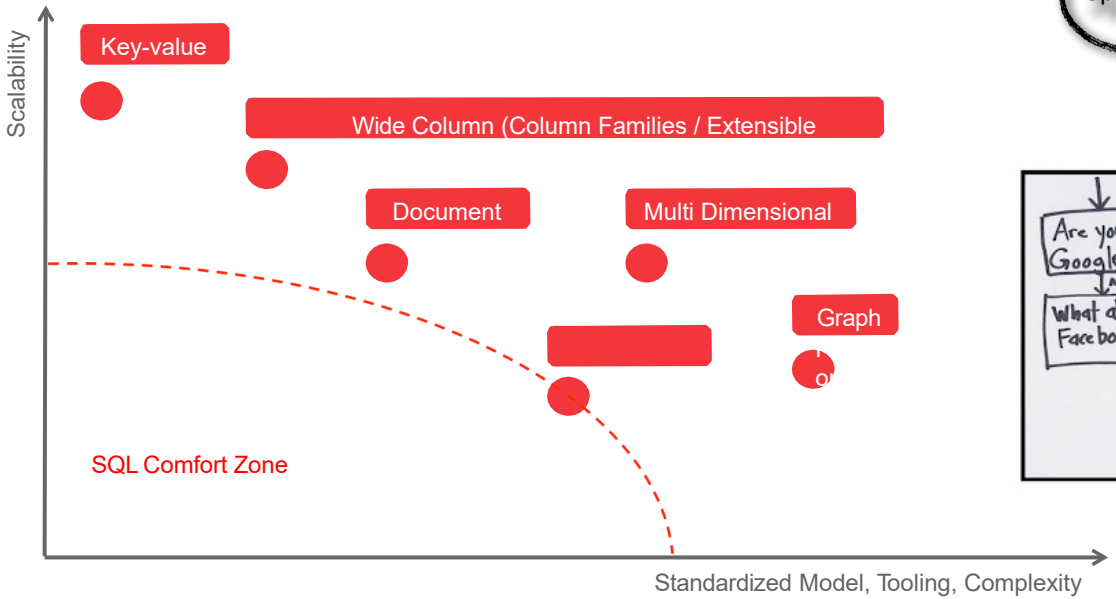
(Michael) - [:WORKS_ON]-> (Neo4j)



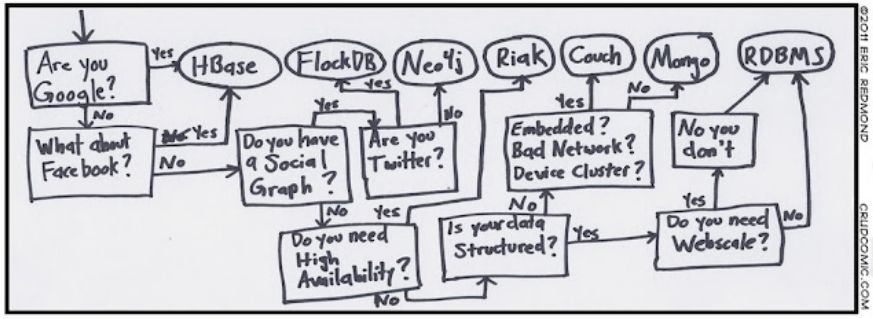
3

(Michael) - [:WORKS_ON]-> (Neo4j)





3



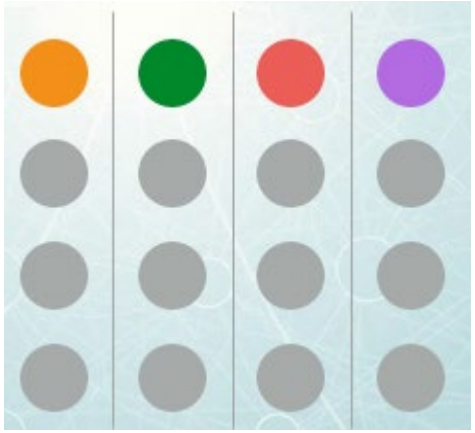
SYSTEMS OF RECORD

Relational Database Model

Structured

Pre-computed

Based on rigid rules



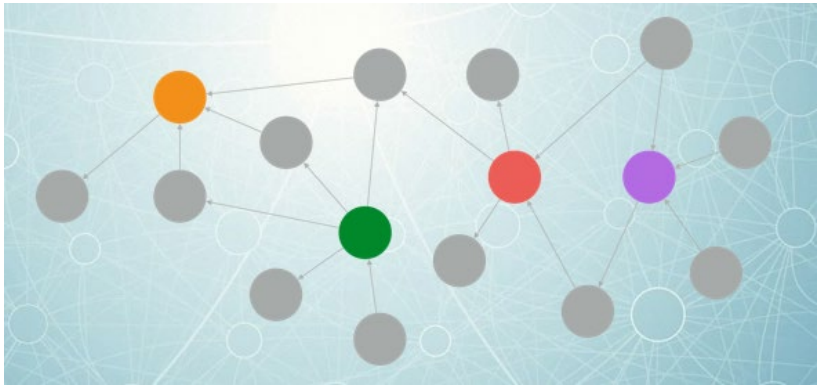
SYSTEMS OF ENGAGEMENT

NoSQL Database Model

Highly Flexible

Real-Time

Queries Highly Contextual



Think of Relational Tables and a DB Query

NORTHWESTERN UNIVERSITY

EmpID	Name	PictureRef
4951870	John Doe	s3://acme-pics/4951870.png
9765207	Jane Smith	s3://acme-pics/9765207.png
4150915	Shyam Bhatt	s3://acme-pics/
7566243	Kathryn Bate	

EmpID	Manager ID	StartDate	EndDate
4951870	9765207	20170101	null
9765207	7566243	20150130	null
4150915	8795882	20141215	20150312
7566243	8509238	20120605	20140124

EmpID	Building	Office
4951870	1200	124A
9765207	1300	187D
4150915	45	432C

Retrieve (the pictures of) all the employees who have offices in the same buildings as their managers

Think of a Relational DB Query

NORTHWESTERN UNIVERSITY

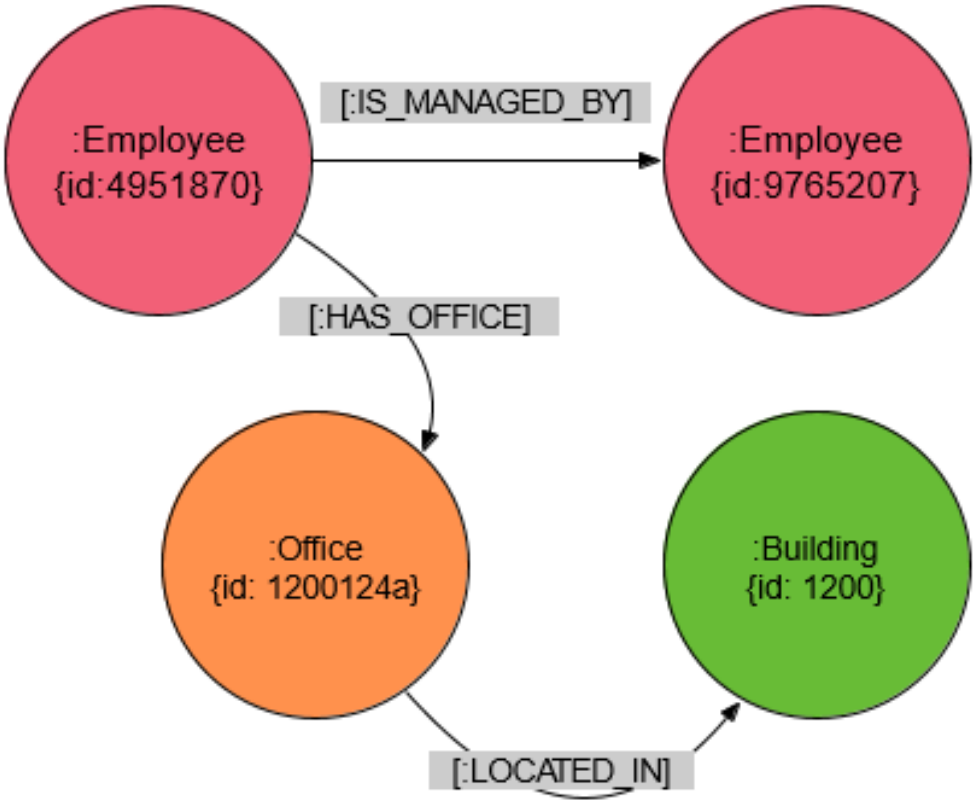
Retrieve (the pictures of) all the employees who have offices in the same buildings as their managers

EmpID	Name	PictureRef	EmpID	Manager ID	StartDate	EndDate
4951870	John Doe	s3://acme-pics/4951870.png	51870	9765207	20170101	null
9765207	Jane Smith	s3://acme-pics/9765207.png	65207	7566243	20150130	null
Shyam Bhatt				8795882	20141215	20150312
Kathryn Bates				8509238	20120605	20140124

EmpID	Building	Office
4951870	1200	124A
9765207	1300	187D
4150915	45	432C

Too many Index Lookups; Very Expensive!

… vs. Graph DB settings



1 Index Lookup
(find :Employee nodes)

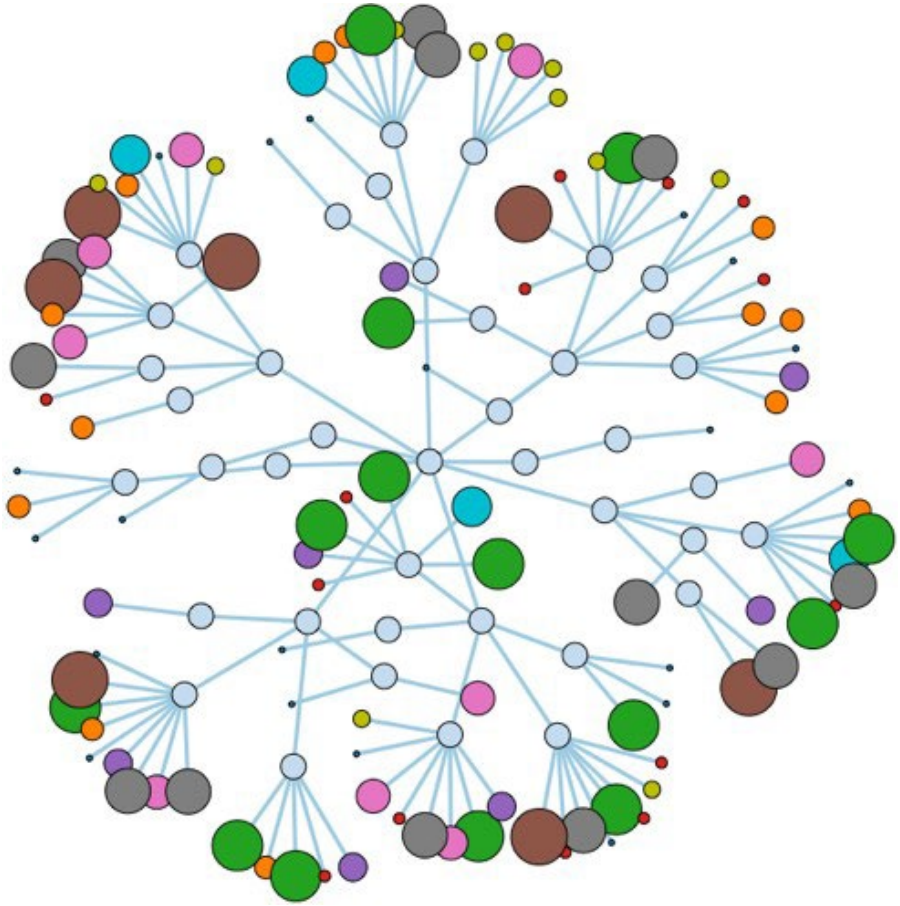
Then Index-Free Adjacency

NORTHWESTERN
UNIVERSITY

Addressing Data Complexity With Graphs

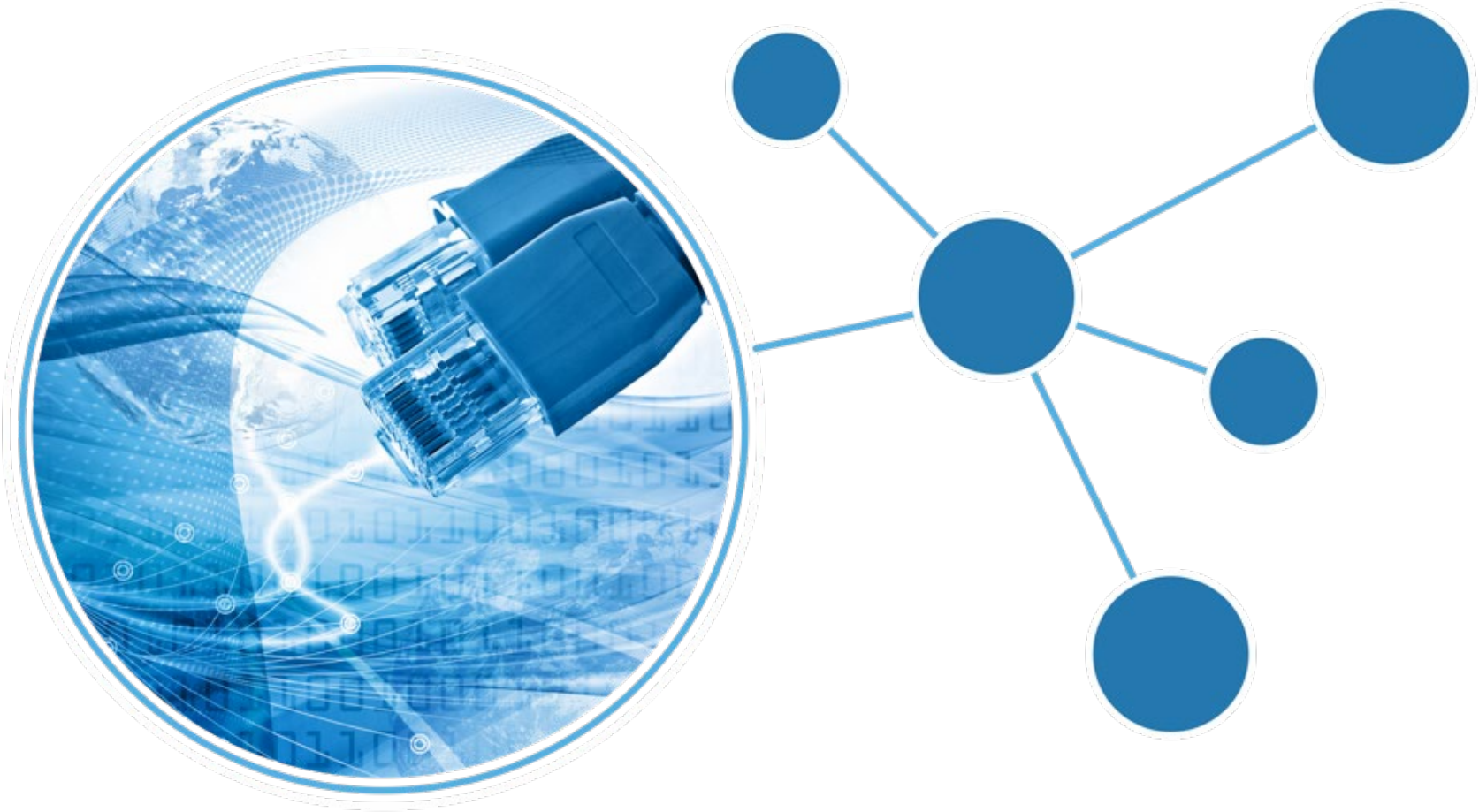
complexity = f(size, semi-structure, **connectedness**)

Are Graphs Everywhere ?

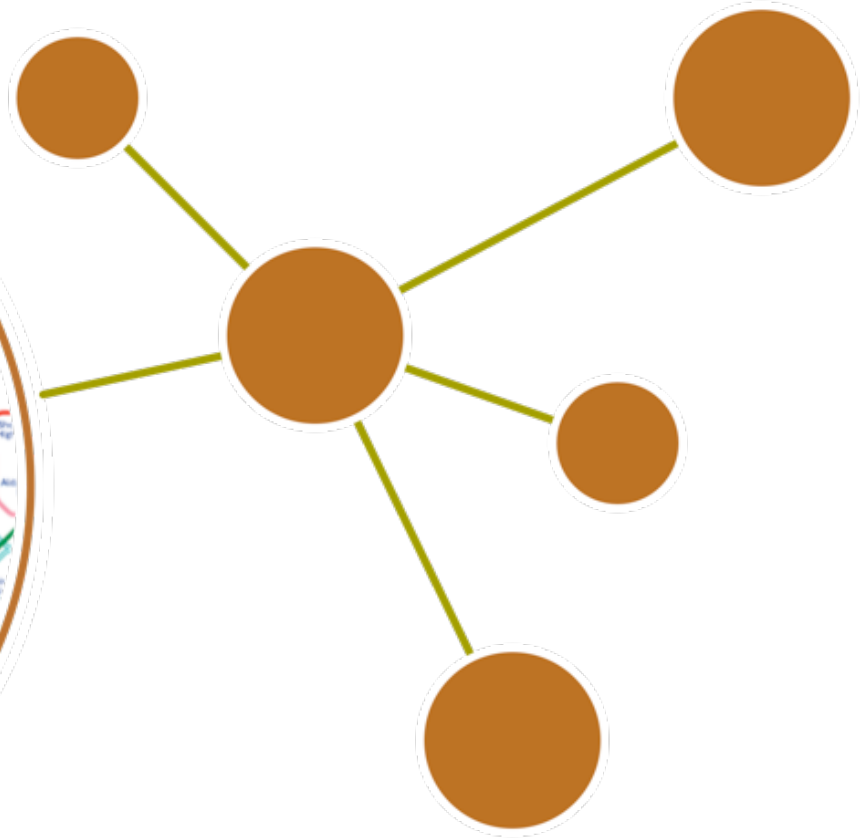


Social Network

(Network) Impact Analysis



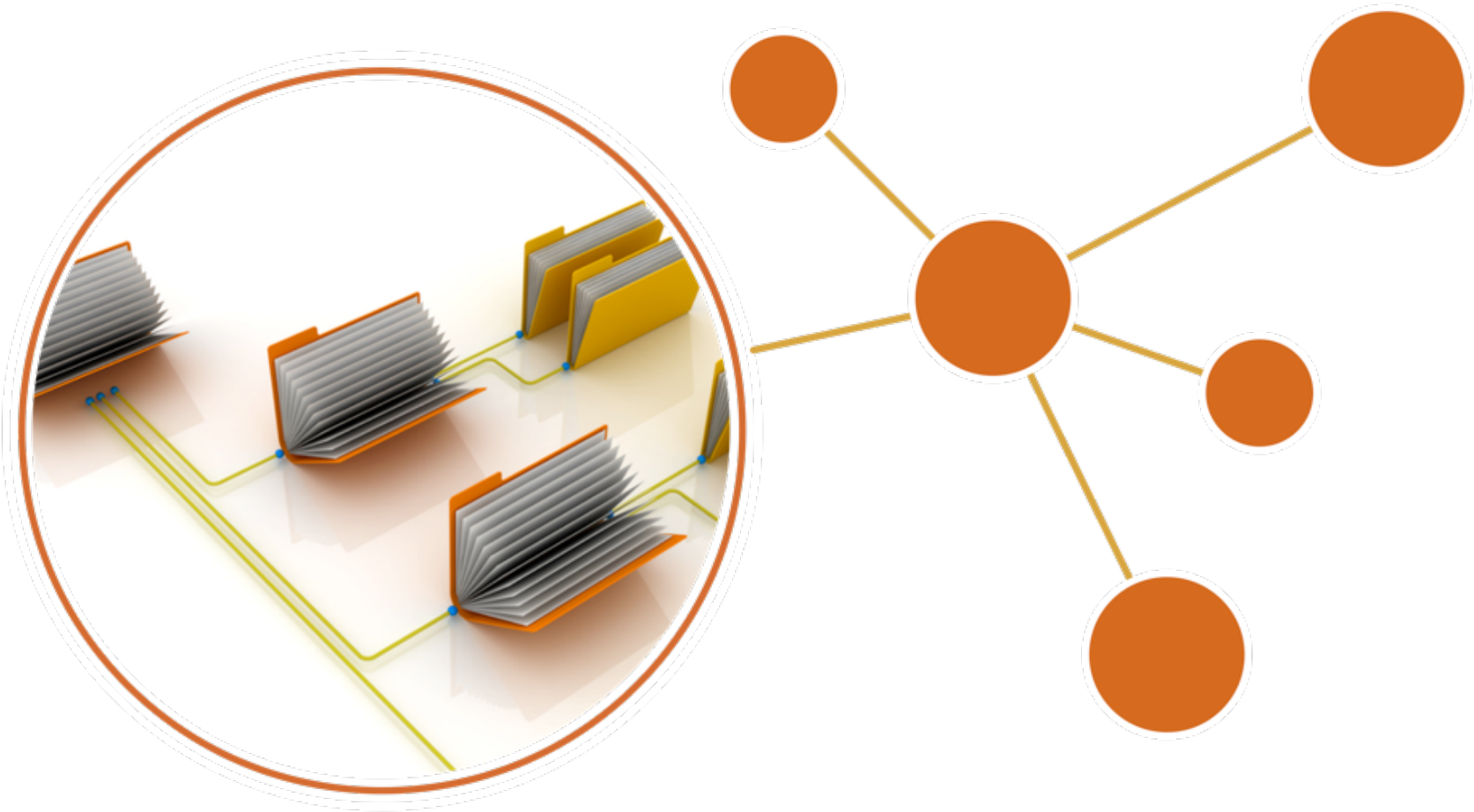
Route Finding



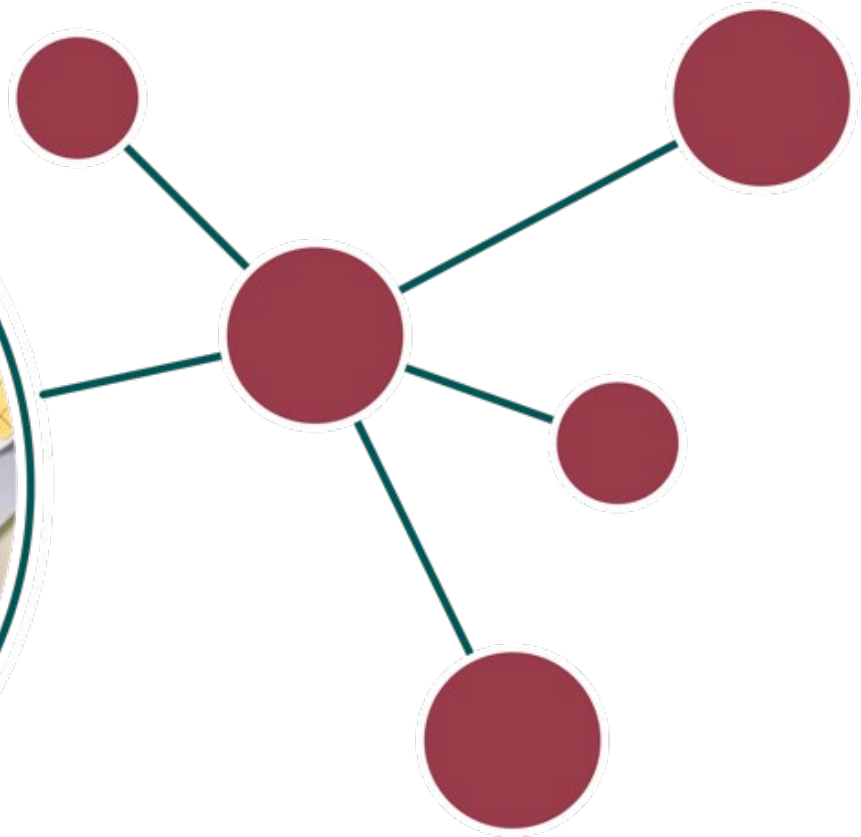
Recommendations

Logistics

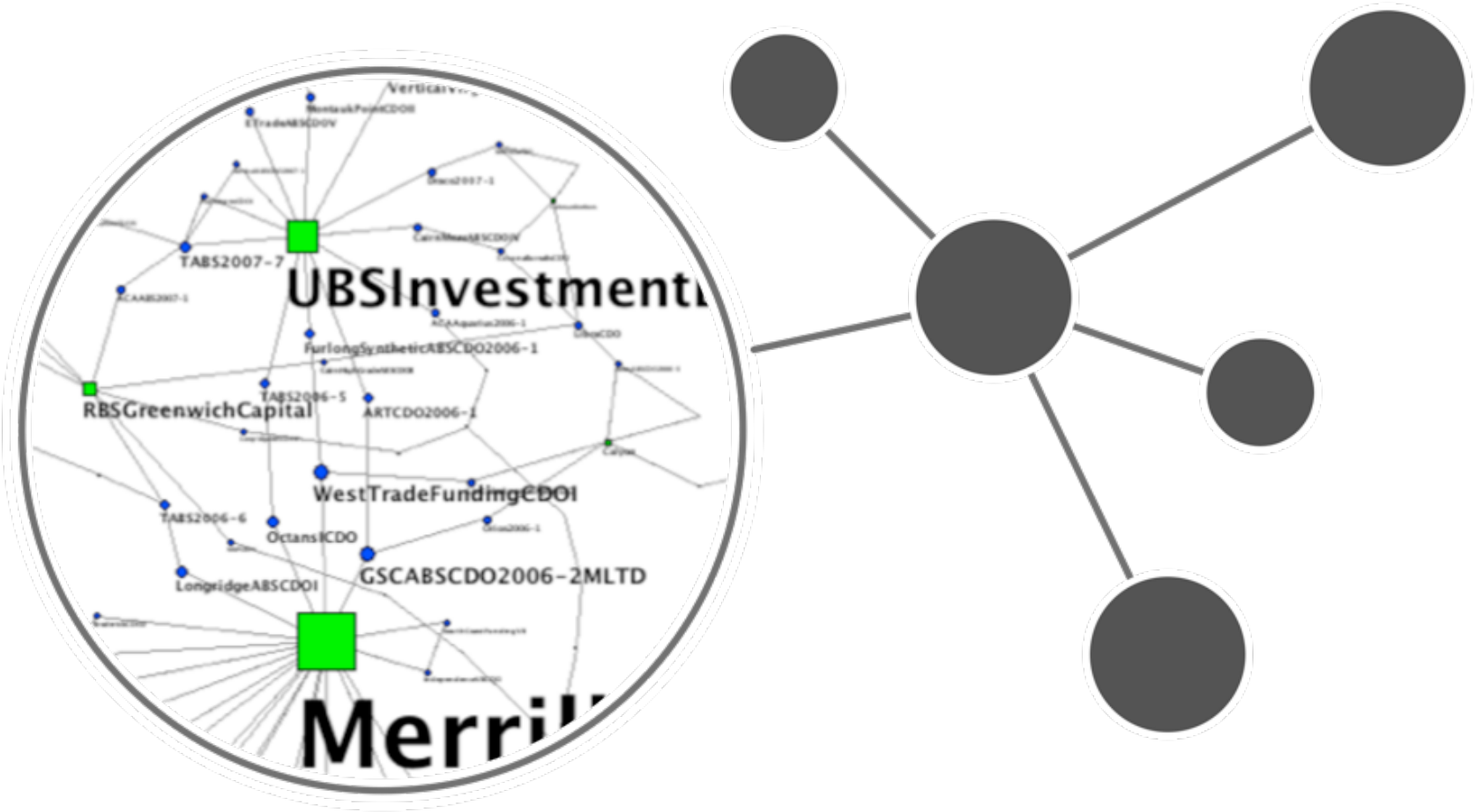
Access Control



Fraud Analysis



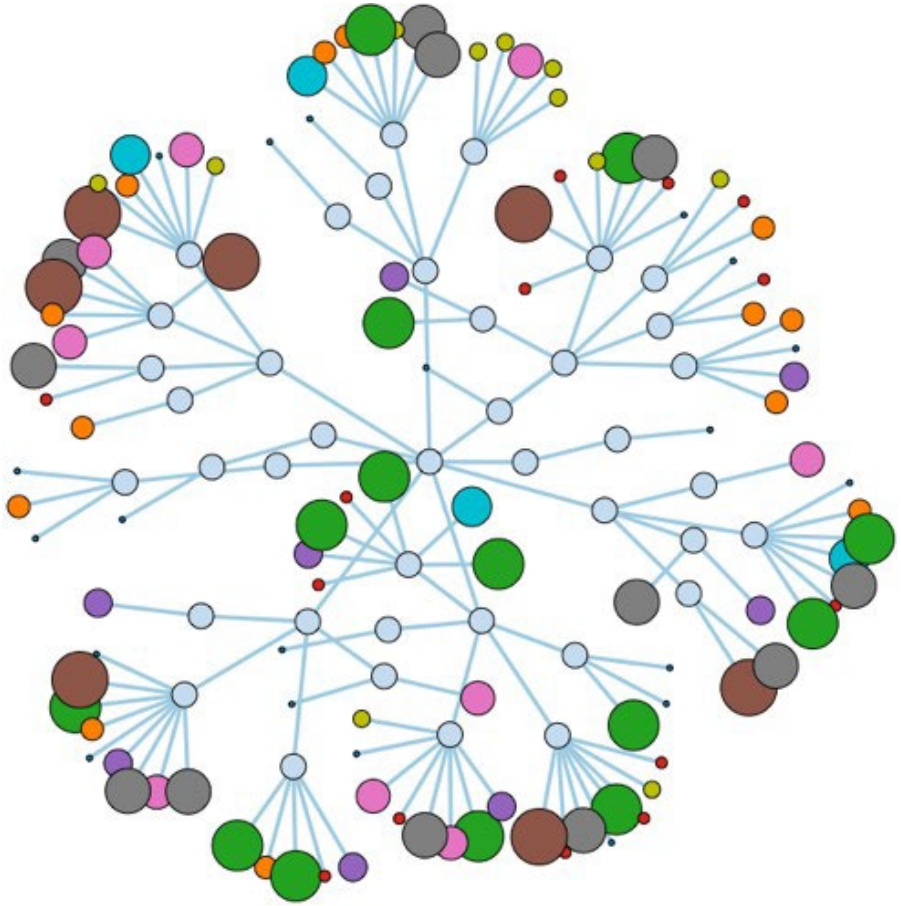
Securities and Debt



Market Analysis (the top-10 companies and collaborative connections)

NORTHWESTERN UNIVERSITY

Graphs Are (almost) Everywhere…



On degrees of separation…

NORTHWESTERN UNIVERSITY

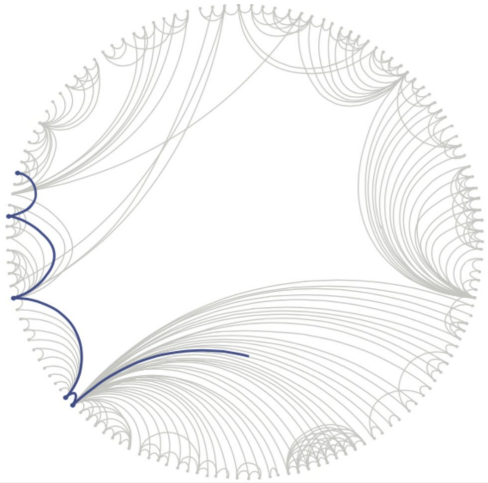
- *I read somewhere that everybody on this planet is separated by only six other people.*
- *Six degrees of separation. Between us and everybody else on this planet. The president of the United States. A gondolier in Venice. Fill in the names. . . .*
- *How every person is a new door, opening up into other worlds. Six degrees of separation between me and everyone else on this planet. But to find the right six people . . . –*

John Guare, Six Degrees of Separation (1990)

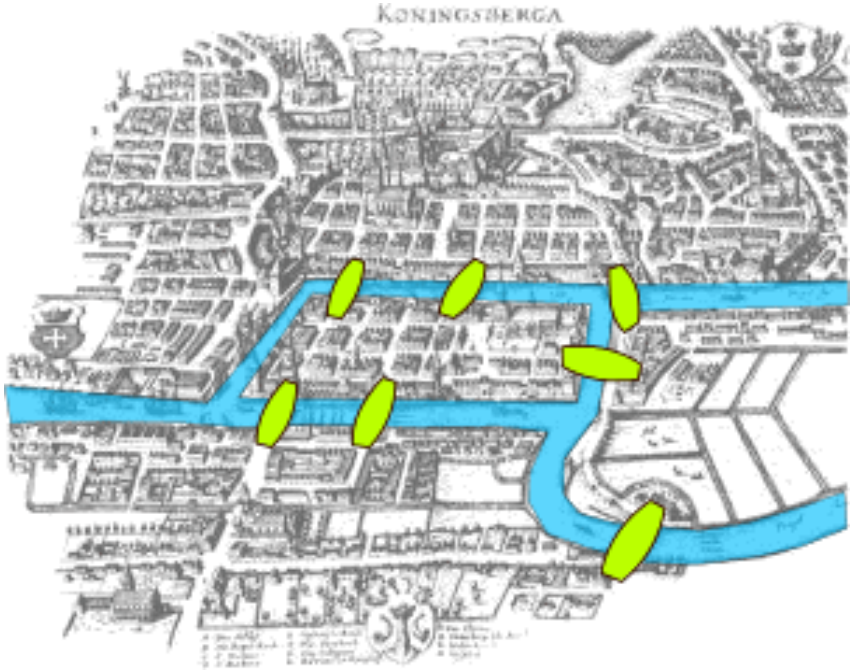
On degrees of separation…

NORTHWESTERN UNIVERSITY

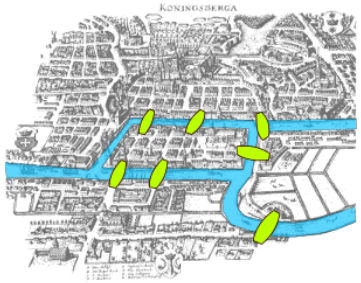
- **Each person in the world (at least among the 1.59 billion people active on Facebook) is connected to every other person by an average of three and a half other people.**
- The average distance we observe is 4.57, corresponding to 3.57 intermediaries or “degrees of separation.”
- Within the US, people are connected to each other by an average of 3.46 degrees.



GRAPH THEORY



GRAPH THEORY – Beginning…

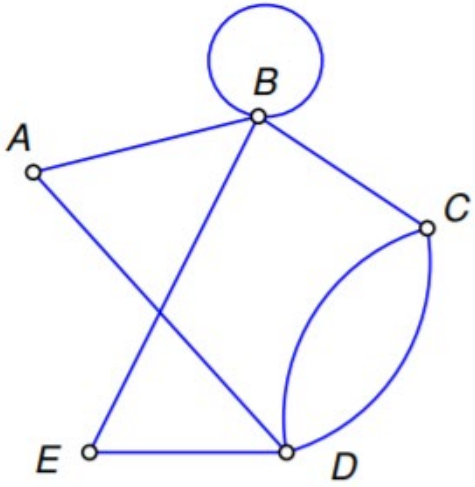


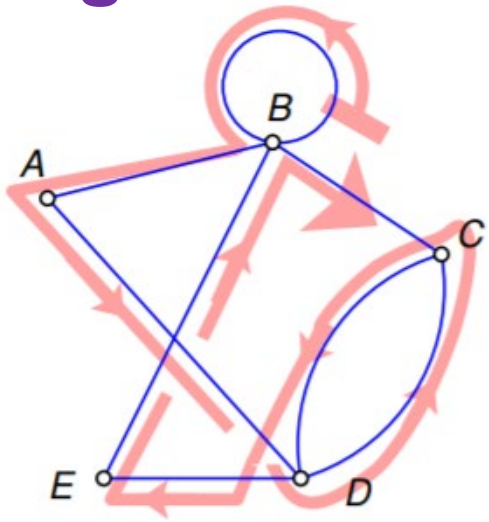
An **Euler path** is a path that uses every edge of a graph exactly once.

An **Euler circuit** is a circuit that uses every edge of a graph exactly once.

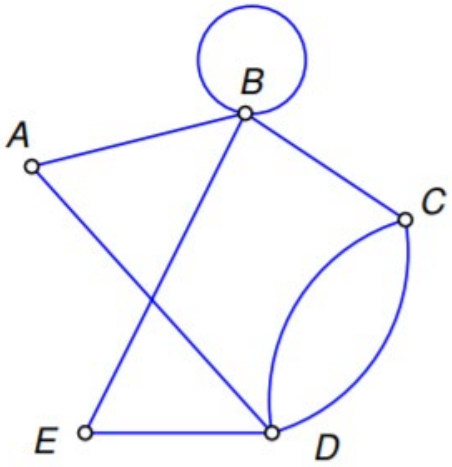
- An Euler path starts and ends at different vertices.
- An Euler circuit starts and ends at the same vertex.

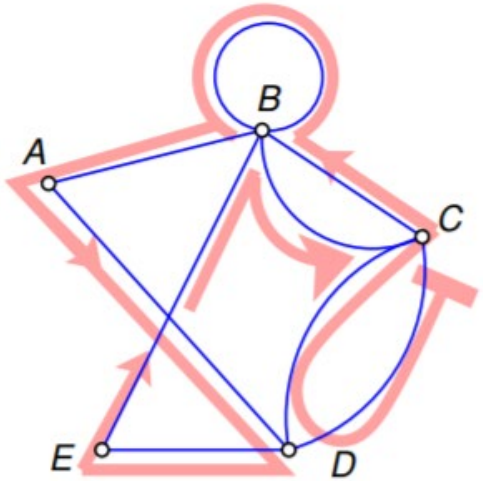
GRAPH THEORY – Beginning…





An Euler path: BBADCDEBC





An Euler circuit: CDCBBADEBC

The History of Graph Theory

NORTHWESTERN UNIVERSITY

- 1736: Leonard Euler writes a paper on the “Seven Bridges of Konisberg”
- 1845: Gustav Kirchoff publishes his electrical circuit laws
- 1852: Francis Guthrie poses the “Four Color Problem”
- 1878: Sylvester publishes an article in Nature magazine that describes graphs
- 1936: Dénes Kőnig publishes a textbook on Graph Theory
- 1941: Ramsey and Turán define Extremal Graph Theory
- 1959: De Bruijn publishes a paper summarizing Enumerative Graph Theory
- 1959: Erdos, Renyi and Gilbert define Random Graph Theory
- 1969: Heinrich Heesch solves the “Four Color” problem
- 2003: Commercial Graph Database products start appearing on the market

Graph Theory Terminology...

NORTHWESTERN UNIVERSITY

- VERTEX: A single node in a graph data structure
- EDGE: A connection between a pair of VERTICES
- PROPERTIES: Data items that belong to a particular Vertex or Edge
- WEIGHT: A quantity associated with a particular Edge
- GRAPH: a network of linked vertex/edge objects

...Graph Theory Terminology...

NORTHWESTERN UNIVERSITY

- SIMPLE/UNDIRECTED GRAPH: A Graph where each VERTEX may be linked to one or more Vertex objects via Edge objects and each Edge object is connected to exactly two Vertex objects. Furthermore, neither Vertex connected to an Edge is more significant than the other.

- DIRECTED GRAPH: A Simple/Undirected Graph where one Vertex in a Vertex + Edge + Vertex group (an “Arc” or “Path”) can be considered the “Head” of the Path and the other can be considered the “Tail”.

- MIXED GRAPH: A Graph in which some paths are Undirected and others are Directed.

...Graph Theory Terminology

- LOOP: An Edge that is doubly-linked to the same Vertex
- MULTIGRAPH: A Graph that allows multiple Edges and Loops
- QUIVER: A Graph where Vertices are allowed to be connected by multiple Arcs. A Quiver may include Loops.
- WEIGHTED GRAPH: A Graph where a quantity is assigned to an Edge, e.g. a Length assigned to an Edge representing a road between two Vertices representing cities.
- HALF EDGE: An Edge that is only connected to a single Vertex
- LOOSE EDGE: An Edge that isn't connected to any Vertices.
- CONNECTIVITY: Two Vertices are Connected if it is possible to find a path between them.

Graph Representation

NORTHWESTERN
UNIVERSITY

Two basic approaches for in-memory representation:

- Sequential:
  - Essentially, a 2D array (for a matrix…)
- Linked:
  - Via list with dynamic memory allocation

Graphs – Sequential Representation

- Adjacency Matrix Representation
- Incidence Matrix Representation
- Circuit Matrix Representation
- Cut Set Matrix Representation
- Path Matrix Representation

Graphs – Adjacency Matrix

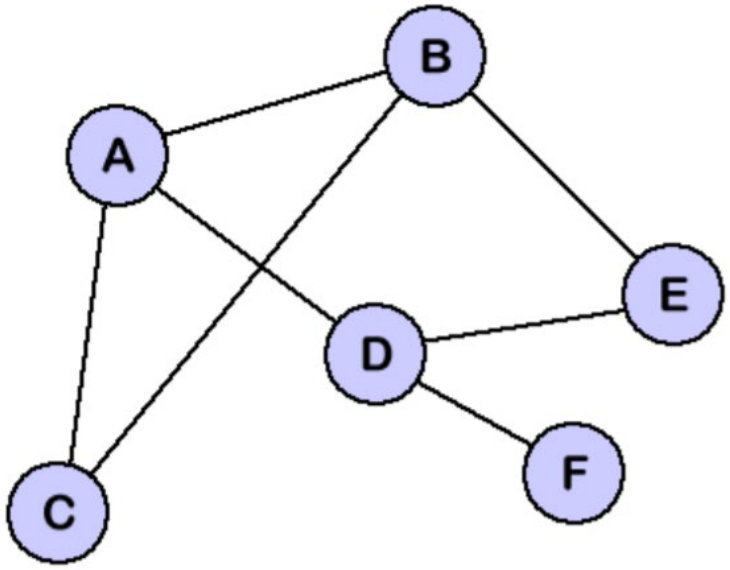
NORTHWESTERN UNIVERSITY

The Adjacency matrix of a graph G with n vertices is N x N. It is given by $A=[a_{ij}]$.

a_{ij} =1 if i^{th} and j^{th} vertices are adjacent.

=0 if i^{th} and j^{th} vertices are not adjacent.

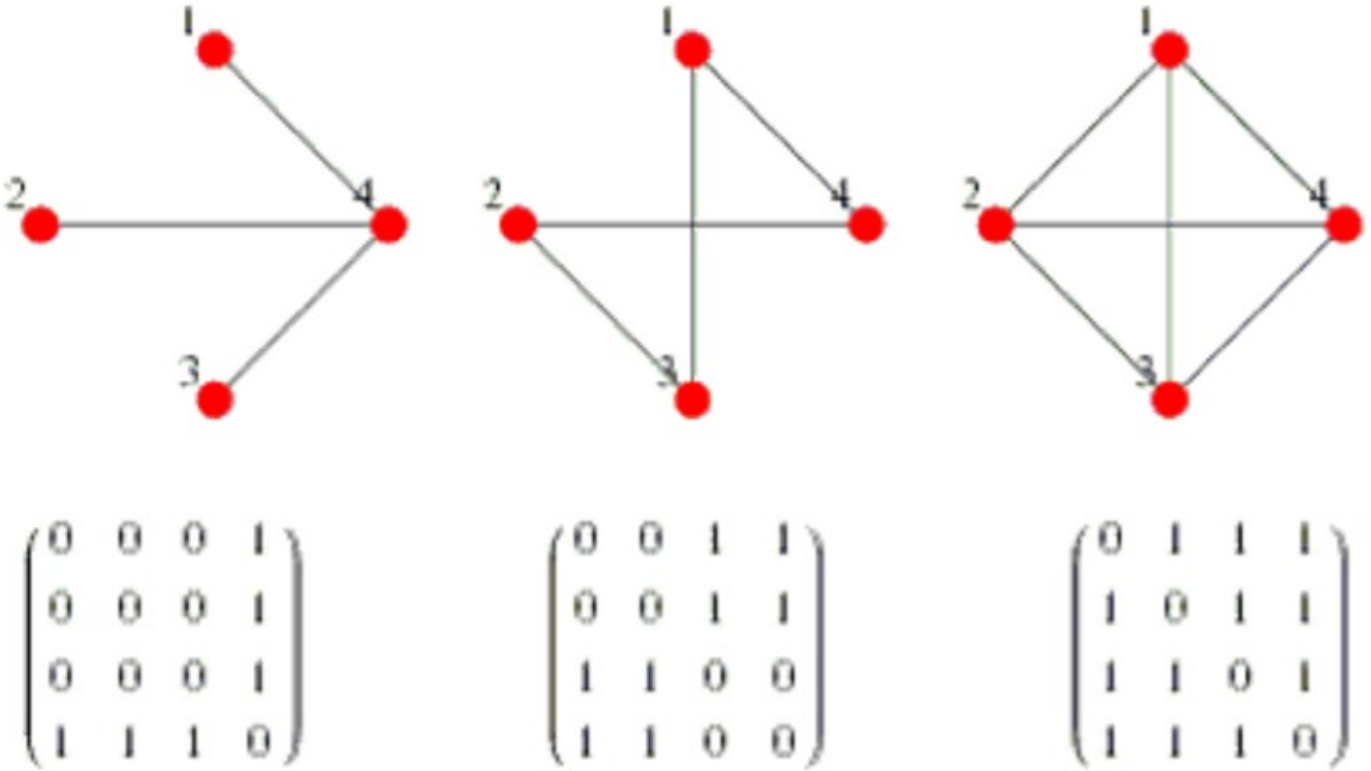
Graphs – Adjacency Matrix



	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	1	0	1	0
C	1	1	0	0	0	0
D	1	0	0	0	1	1
E	0	1	0	1	0	0
F	0	0	0	1	0	0

Note: for undirected graphs, the matrix is symmetric along the main diagonal.

Graphs – Adjacency Matrix



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Note++: if edges have weights, than those values are the entries of the matrix…

Graphs – Adjacency Matrix

NORTHWESTERN UNIVERSITY

NOTE:
To determine whether two nodes i and j are connected (and to determine the length of the connection):

Simply check whether there is a non-zero value in the (i,j) entry of: A, or A^2, or A^3, or…

Graphs – Incidence Matrix

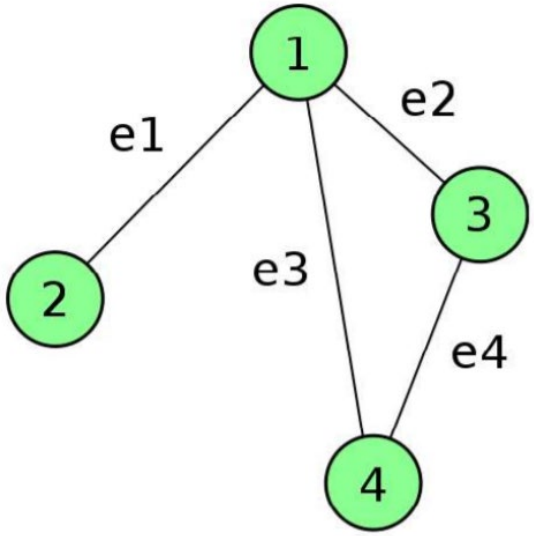
NORTHWESTERN UNIVERSITY

Unlike the adjacency matrix, the incidence matrix need not be of a "square" dimension

For a graph G with N vertices and E edges, the incidence matrix is N X E

$$m_{ij} = 1 \text{ if } e_j \text{ is incident on } v_i$$
$$= 0 \text{ otherwise}$$

Graphs – Incidence Matrix



$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c}
\begin{array}{cccc} e1 & e2 & e3 & e4 \end{array} \\
\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}
\end{array}
$$

Graphs – Incidence Matrix

NORTHWESTERN UNIVERSITY

NOTE: Incidence matrix can be used to check for connectivity, but is also used in *circuit theory (i.e., putting voltages on nodes and current on edges, the product* $A^T y = 0$ *gives a system of equations corresponding to Kirchhoff's currents law…).*

Graphs – Other Sequential (Matrix) Representations…

In each of them rows/columns correspond to a particular feature, and the columns/rows correspond to vertices (or edges) having that feature…

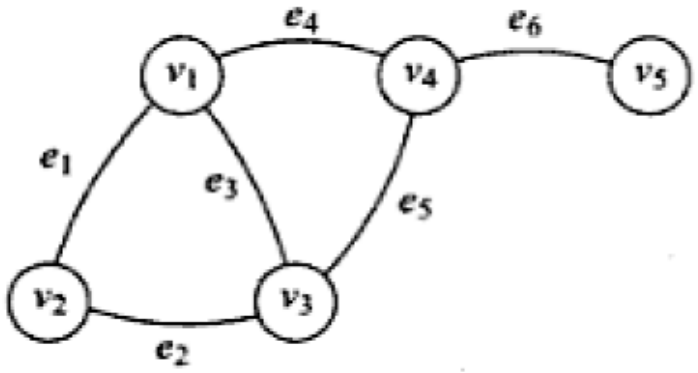
- Circuit matrix representation
- Path matrix representation
- Cutset matrix representation

Graphs – Other Sequential (Matrix) Representations… Circuits…

NORTHWESTERN UNIVERSITY

Circuit Matrix is represented by the number of different circuits T and the number of edges E is T x E in the graph. The Circuit Matrix C=Cij.

Cij=1 if ith circuit includes jth edge.
= 0 otherwise



$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Circuit 1 : $\{e_1, e_2, e_3\}$
Circuit 2 : $\{e_3, e_4, e_5\}$
Circuit 3 : $\{e_1, e_2, e_5, e_4\}$

Graphs – Other Sequential (Matrix) Representations… Circuits…

- Problems:

- Hamiltonian cycle exists in a given graph (whether directed or undirected), if there is a Hamiltonian path starting and ending at the same vertex.

- Hamiltonian path (a path in an undirected or directed graph that visits each vertex exactly once).

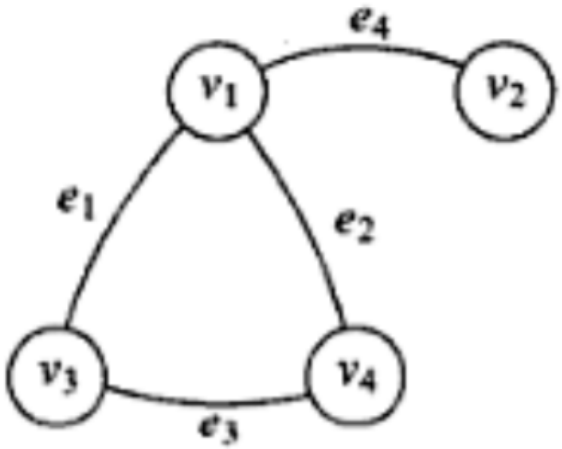
Both problems are NP-complete.

Graphs – Cutset Matrix

NORTHWESTERN UNIVERSITY

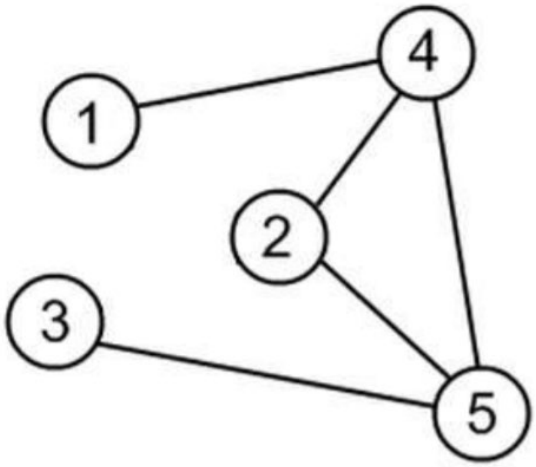
Rows = cutsets (i.e., collection of edges whose removal makes graph disconnected)

Columns = edges participating in a given cutset



$$S = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{array}\right] \end{array} \begin{array}{l} \\ 1 : \{e_4\} \\ 2 : \{e_1, e_2\} \\ 3 : \{e_2, e_3\} \\ 4 : \{e_1, e_3\} \end{array}$$

Graphs – Adjacency Representations

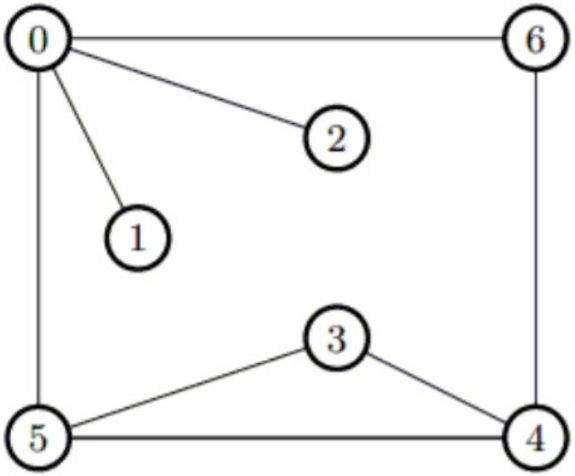


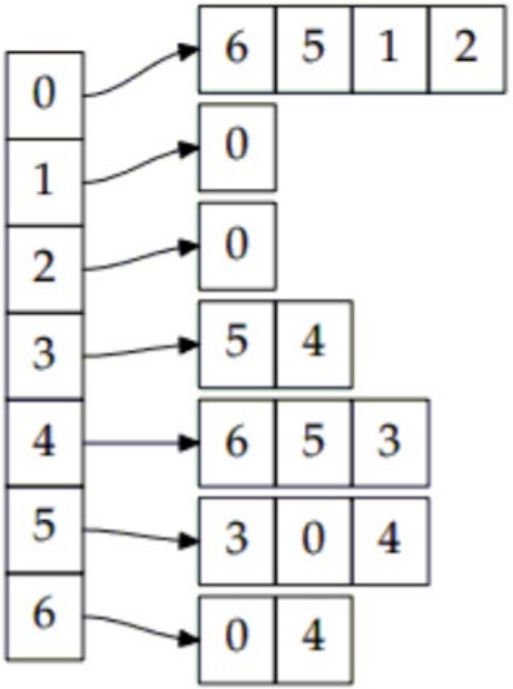
Graph

1	4
2	4 5
3	5
4	1 2 5
5	2 3 4

Adjacency list

Graphs – Adjacency Representations





Graphs – Storing

Although there are multiple representations (from mathematical perspective), in practice one either uses

adjacency matrix

or

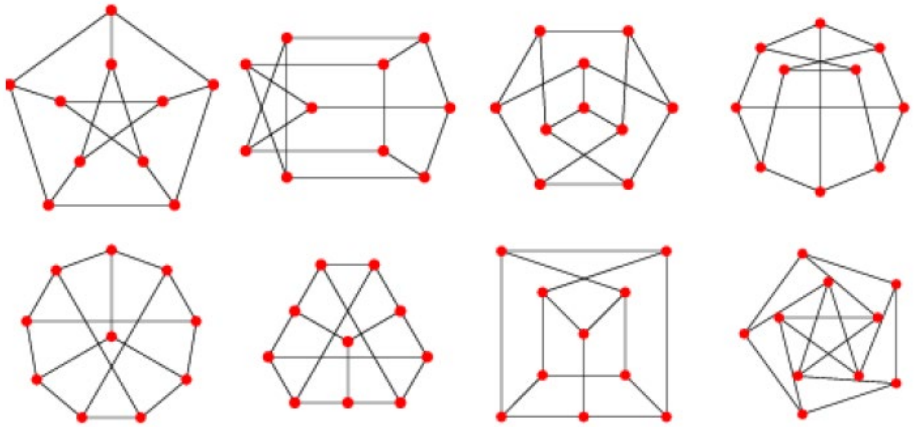
adjacency lists

as basic methods for storing graphs in computers…

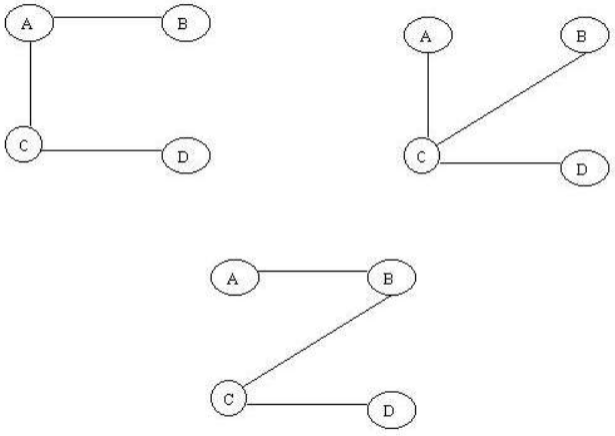
Graphs – Representations

- Adjacency list: an alternative to matrix-based representation.
- Uses less memory
- For every vertex it stores a list of all and only those vertices connected to that particular head-vertex
- (NOTE: think linked lists…)

Sparsity…



VS



COMMONLY USED GRAPH PREDICATES (Properties/Algorithms…)

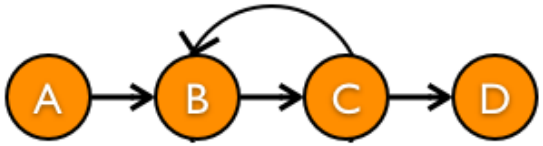
Mac Evans

Commonly Used Graph Algorithms...

NORTHWESTERN UNIVERSITY

- CONNECTEDNESS: Check whether or not a set of nodes in a Graph are connected.

  All of the nodes in the graph below are connected, e.g. A to B, A to C via B etc.

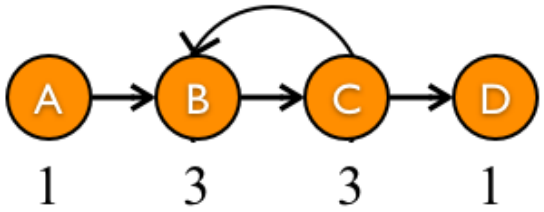


- SHORTEST PATH: The path between two nodes that visits the fewest intermediate nodes.

  In the graph above, A->B->C->D is shorter than A->B->C->B->D (disallowing loops)

- NODE DEGREE: The degree of a node in a network is a count of the number of connections it has to other nodes. The degree distribution is the probability distribution of these degrees in the whole network.

  In the graph below, A and D have a node degree of 1. B and C have a node degree of 3.
- NOTE: in-degree vs. out-degree…



Common Graph Algorithms: Shortest Path

Given a weighted directed graph G,
find the minimum-weight path from a given source vertex s to another vertex v

- "Shortest-path" = minimum weight
- Weight of path is sum of weights of edges
- E.g., a road map: what is the shortest path from Ames to Naperville?

Shortest Path – Properties

NORTHWESTERN UNIVERSITY

Shortest path consist of shortest sub-paths…

Proof: suppose some subpath is not a shortest path

- There must then exist a shorter subpath
- Could substitute the shorter subpath for a shorter path
- But then overall path is not shortest path. Contradiction

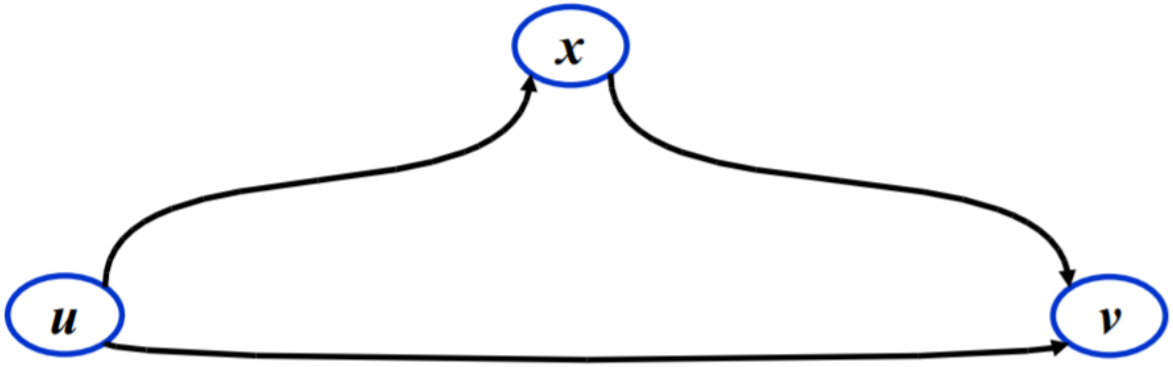
Shortest Path – Properties

NORTHWESTERN UNIVERSITY

Let $\delta(u,v)$ denote the weight of the shortest path from u to v

Shortest path satisfies the triangular inequality:

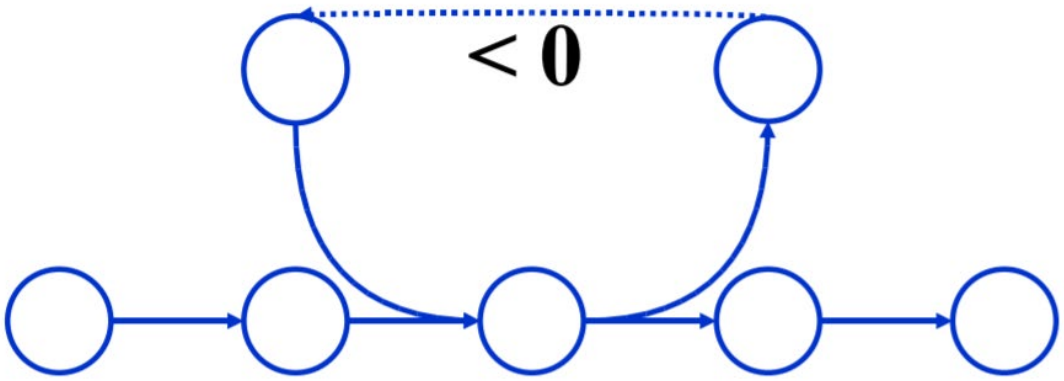
$$\delta(u,v) \leq \delta(u,x) + \delta(x,v)$$



Shortest Path – Properties

NORTHWESTERN UNIVERSITY

ASIDE: things are a bit weird when negative-weights are allowed (i.e., some shortest paths will simply not exist)

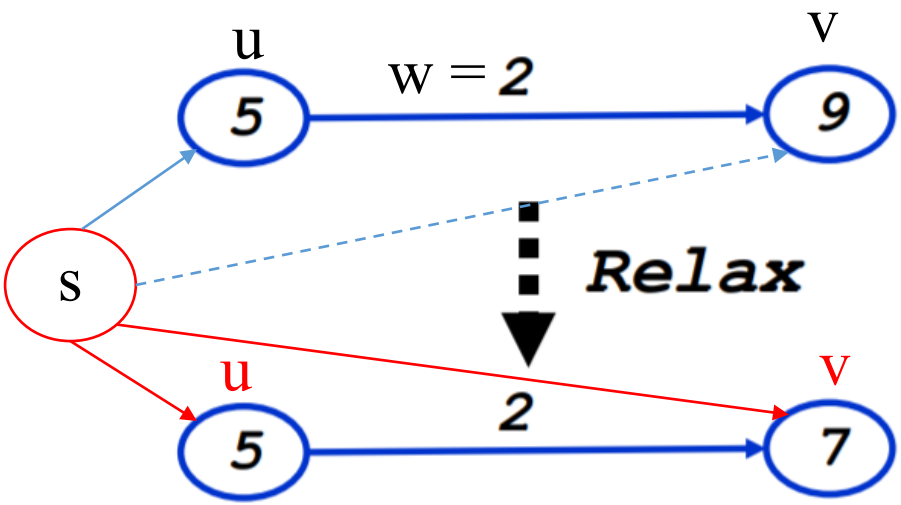


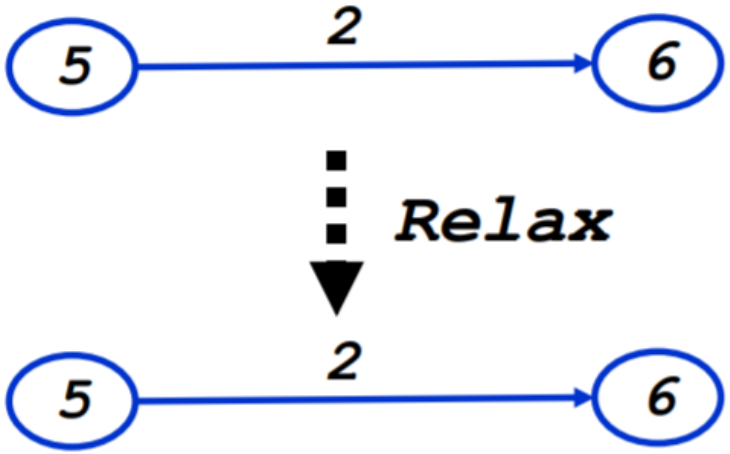
Shortest Path – Properties

NORTHWESTERN UNIVERSITY

- A key technique in shortest path algorithms is *relaxation*
  - Idea: for all v, maintain upper bound d[v] on δ(s,v)

```
Relax(u,v,w) {
     if (d[v] > d[u]+w) then d[v]=d[u]+w;
}
```





Shortest Path – Bellman-Ford Algorithm

NORTHWESTERN UNIVERSITY

```
BellmanFord()
    for each v ∈ V
        d[v] = ∞;
    d[s] = 0;
    for i=1 to |V|-1
        for each edge (u,v) ∈ E
            Relax(u,v, w(u,v));
    for each edge (u,v) ∈ E
        if (d[v] > d[u] + w(u,v))
            return "no solution";
```

Initialize d[], which will converge to shortest-path value δ

Relaxation: Make |V|-1 passes, relaxing each edge

Test for solution
Under what condition do we get a solution?

```
Relax(u,v,w): if (d[v] > d[u]+w) then d[v]=d[u]+w
```

NORTHWESTERN
UNIVERSITY

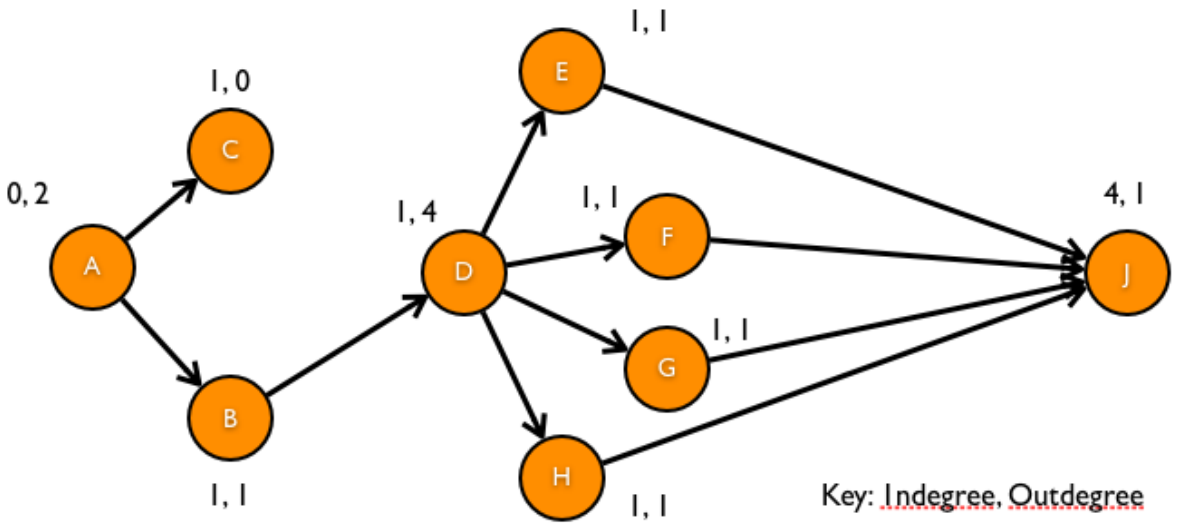
DO TUKA

...Commonly Used Graph Predicates (cont...)

- CENTRALITY: An assessment of the importance of a node within a network.

Degree Centrality is the simplest, being a count of the number of connections that a node has.

It may be expressed as “Indegree” (# of incoming connections) and “Outdegre” (# of outgoing connections).

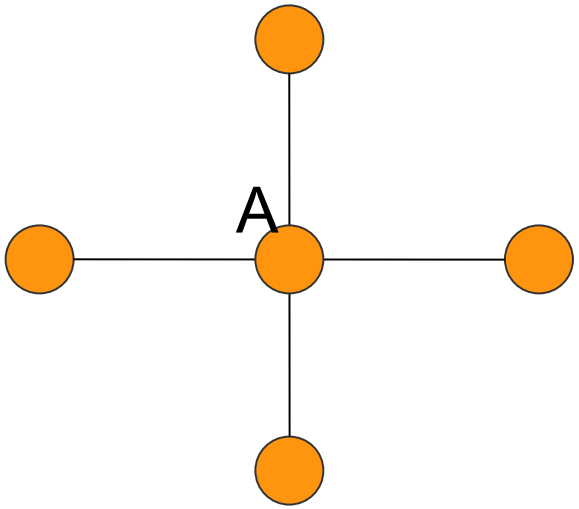


...Commonly Used Graph Algorithms...

NORTHWESTERN UNIVERSITY

- CLOSENESS CENTRALITY: Closeness considers the shortest paths between nodes and assigns a higher value to nodes that can be used to reach most other nodes most quickly.

  In the graph below, node A has the greatest centrality as all other nodes can be reached in one "hop", whereas others require 1 hop to A or 2 hops to any other node.

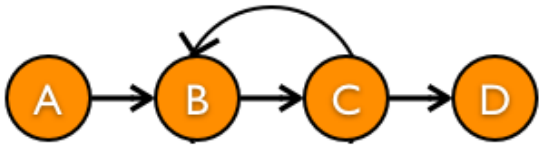


...Commonly Used Graph Algorithms...

NORTHWESTERN UNIVERSITY

- Recall: SHORTEST PATH: The path between two nodes that visits the fewest intermediate nodes.

  In the graph below, A->B->C->D is shorter than A->B->C->B->D (disallowing loops)



- AVERAGE PATH LENGTH: The average of all path lengths between all pairs of nodes in a graph.

- TRANSITIVE CLOSURE: The process of exploring a graph by traversing relationships until all nodes have been visited, but without revisiting nodes that are joined together in loops.

  In the graph above, A->B->C->D is a transitive closure.

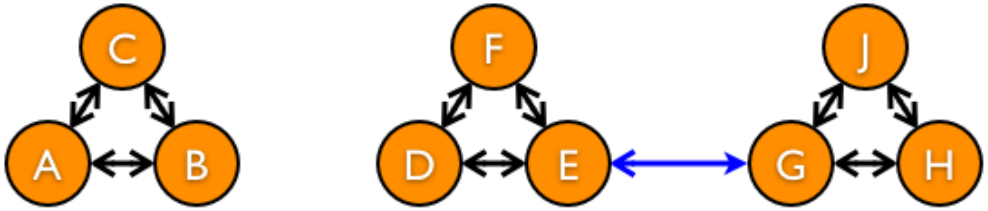
...Commonly Used Graph Algorithms...

NORTHWESTERN UNIVERSITY

- GRAPH DIAMETER (or SPAN): The greatest distance between any pair of nodes in a graph.

  It is computed by finding the shortest path between each pair of nodes. The maximum of these path lengths is a measure of the diameter of the graph.

  The diameters of the two graphs below are 2 and 5.



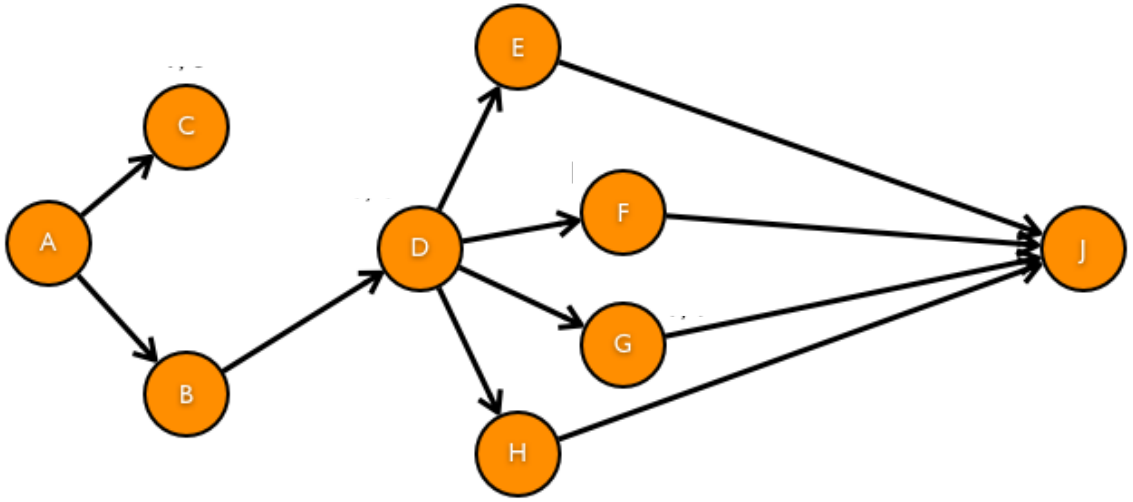
...Commonly Used Graph Algorithms...

NORTHWESTERN UNIVERSITY

- BETWEENESS CENTRALITY: A centrality measure of a node within a graph.

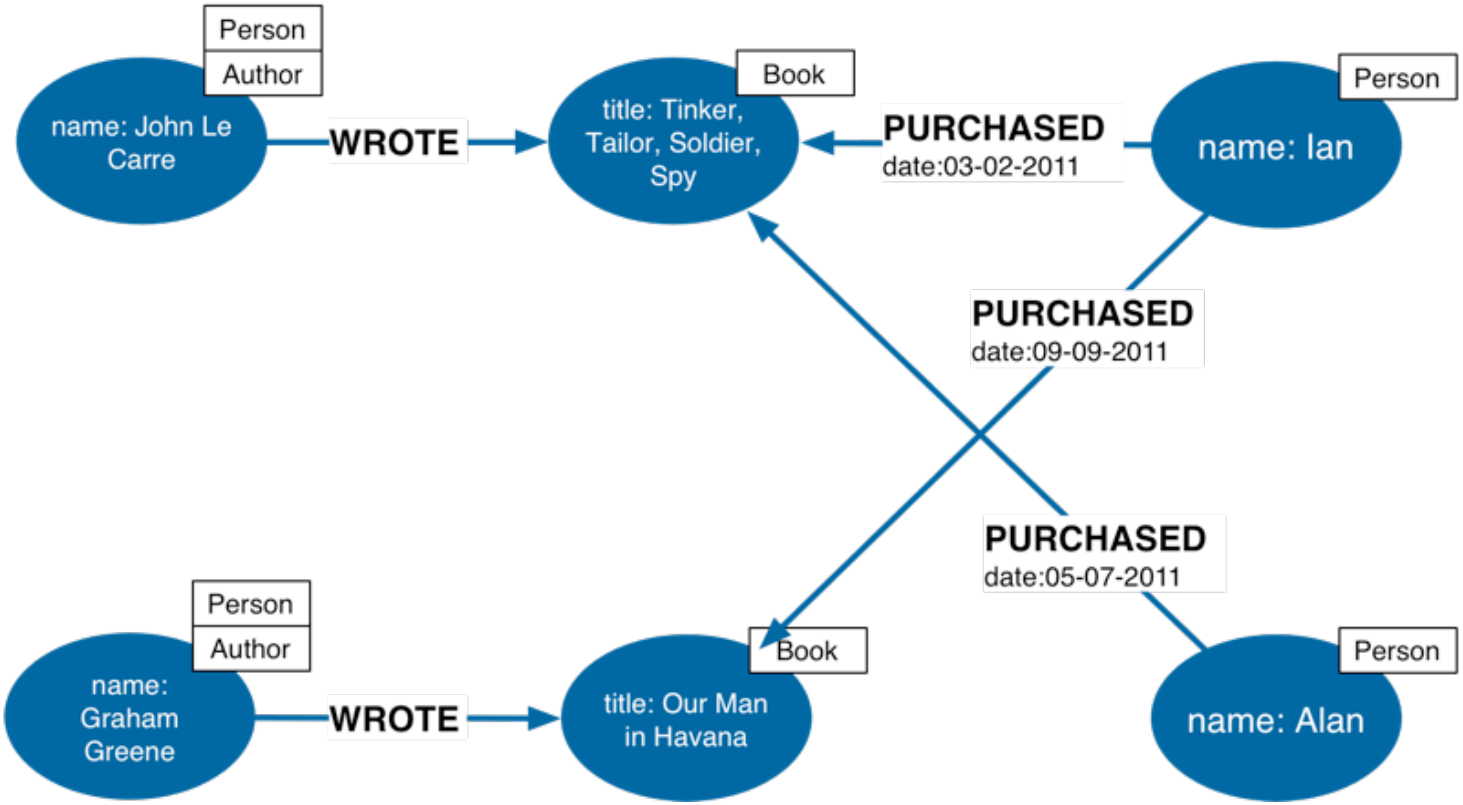
Nodes that have a high probability of being visited on a randomly chosen short path between two randomly chosen nodes have a high “betweeness”

In the graph below, node D has the highest betweeness centrality.



Graph Data/Model Basic Building Blocks

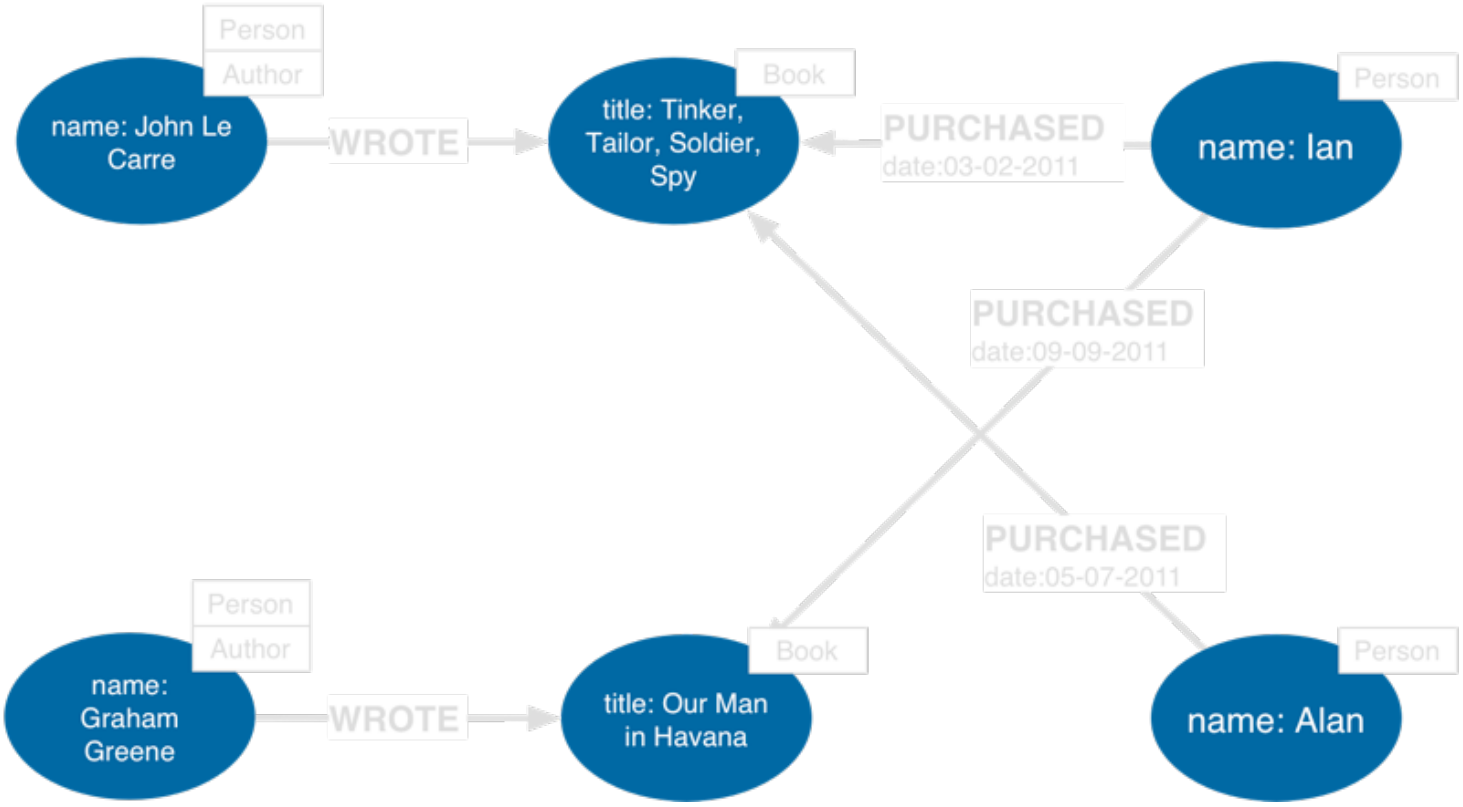
Property Graph Data Model



Four Building Blocks

1. Nodes
2. Relationships
3. Properties
4. Labels

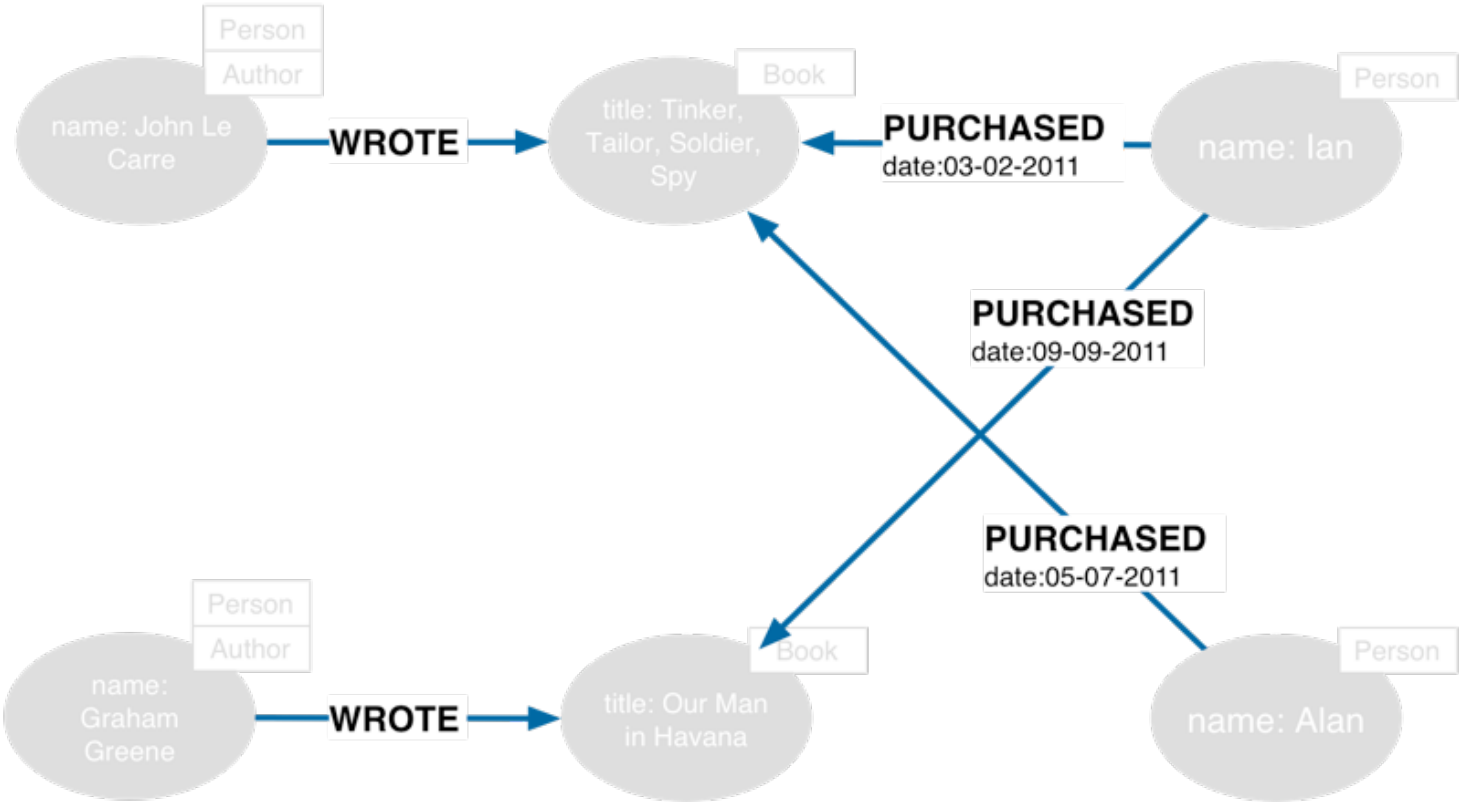
Nodes


Person
Author
name: John Le Carre
WROTE
Book
title: Tinker, Tailor, Soldier, Spy
PURCHASED
date:03-02-2011
Person
name: Ian
PURCHASED
date:09-09-2011
PURCHASED
date:05-07-2011
Person
Author
name: Graham Greene
WROTE
Book
title: Our Man in Havana
Person
name: Alan

Nodes

- Used to represent entities in your domain
- Can contain properties
  - Used to represent entity attributes and/or metadata (e.g. timestamps, version)
  - Key-value pairs
    - Java primitives
    - Arrays
    - `null` is not a valid value
  - Every node can have different properties

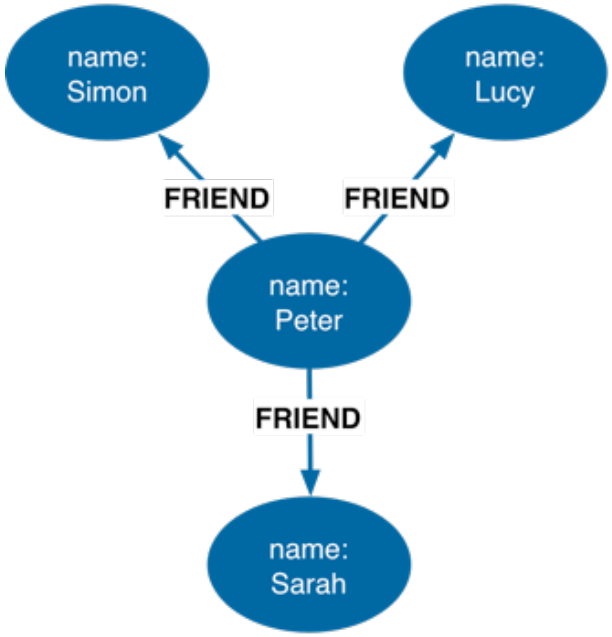
Relationships



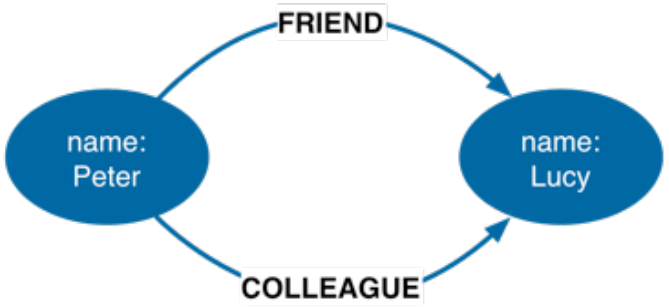
Relationships

- Every relationship has a name and a direction
  - Add structure to the graph
  - Provide semantic context for nodes
- Can contain properties
  - Used to represent quality or weight of relationship, or metadata
- Every relationship must have a start node and end node
  - No dangling relationships

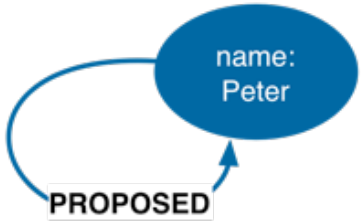
Relationships (continued)



Nodes can have more than one relationship



Nodes can be connected by more than one relationship



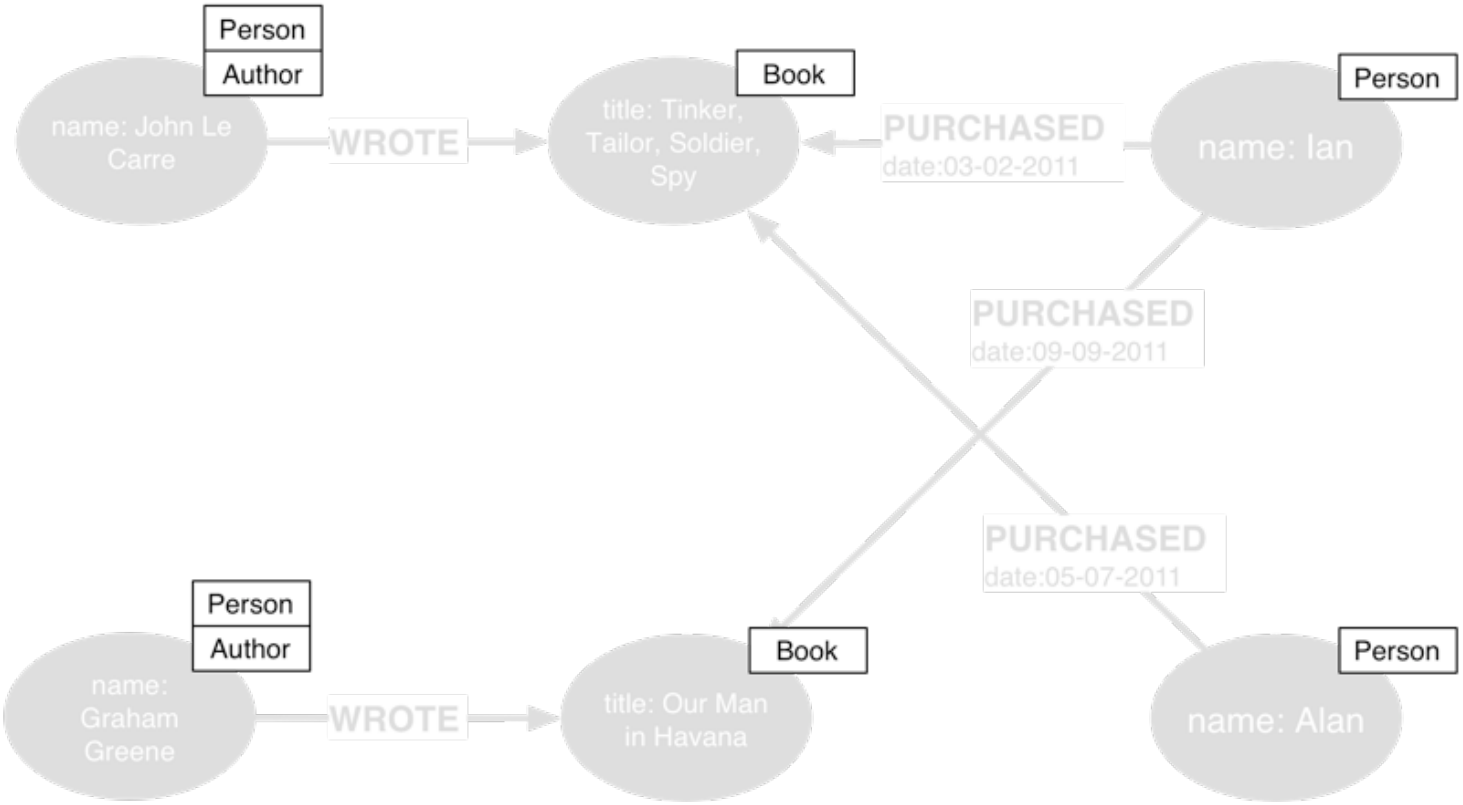
Self relationships are allowed

Variable Structure

NORTHWESTERN UNIVERSITY

- Relationships are defined with regard to node instances, not classes of nodes
  - Different nodes can be connected in different ways
  - Allows for structural variation in the domain
  - Contrast with relational schemas, where foreign key relationships apply to all rows in a table

Labels


Person
Author
name: John Le Carre
WROTE
Book
title: Tinker, Tailor, Soldier, Spy
PURCHASED
date:03-02-2011
Person
name: Ian
PURCHASED
date:09-09-2011
PURCHASED
date:05-07-2011
Person
Author
name: Graham Greene
WROTE
Book
title: Our Man in Havana
Person
name: Alan

Labels

- Every node can have zero or more labels attached
- Used to represent roles (e.g. user, product, company)
  - Group nodes
  - Allow us to associate indexes and constraints with groups of nodes

Four Building Blocks

- Nodes
  - Entities
- Relationships
  - Connect entities and structure domain
- Properties
  - Attributes and metadata
- Labels
  - Group nodes by role