DS 4300

Introduction to the Graph Data Model

Mark Fontenot, PhD
Northeastern University

What is a Graph Database

- Data model based on the graph data structure
- Composed of nodes and edges
  - edges connect nodes
  - each is uniquely identified
  - each can contain properties (e.g. name, occupation, etc)
  - supports queries based on graph-oriented operations
    - traversals
    - shortest path
    - *lots of others*

Where do Graphs Show up?

- Social Networks
  - yes… things like Instagram,
  - but also… modeling social interactions in fields like psychology and sociology
- The Web
  - it is just a big graph of “pages” (nodes) connected by hyperlinks (edges)
- Chemical and biological data
  - systems biology, genetics, etc.
  - interaction relationships in chemistry

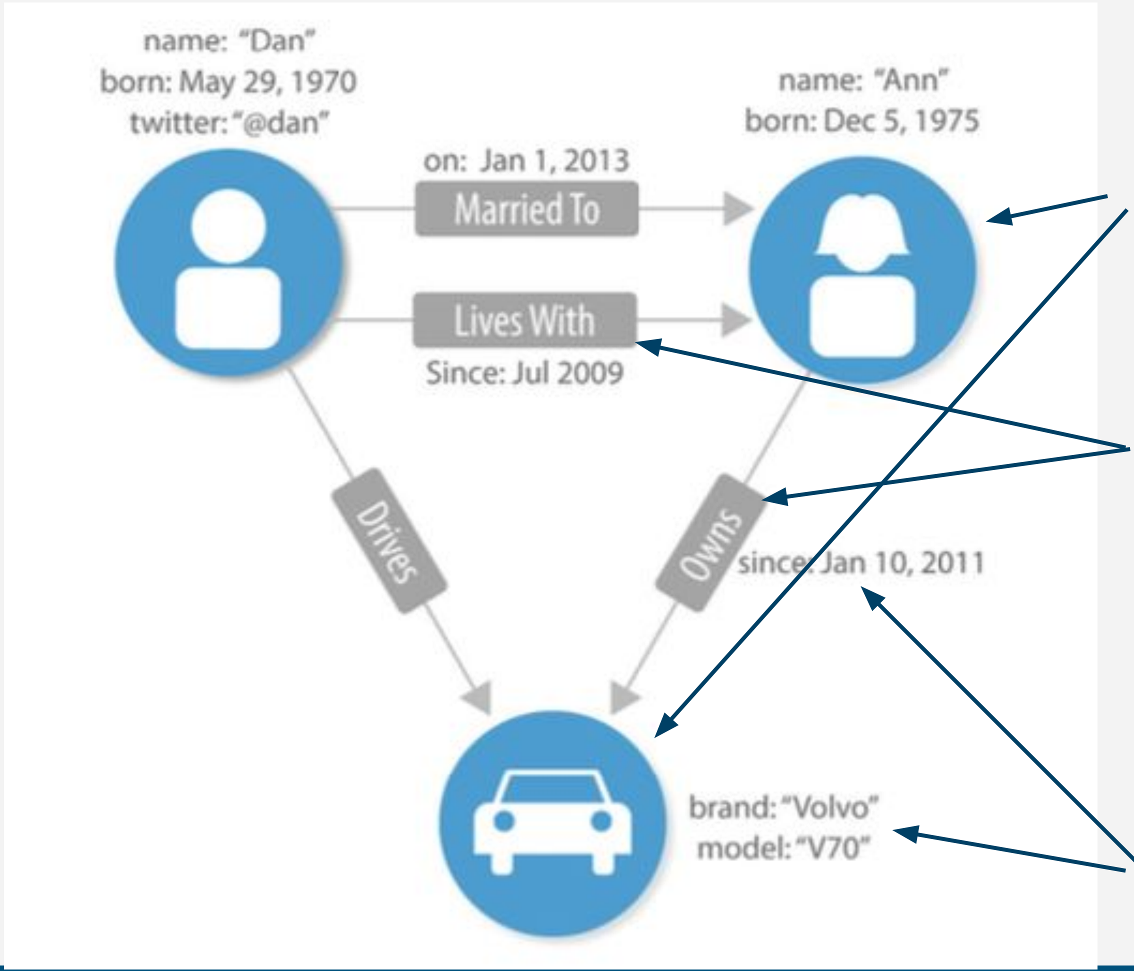
Basics of Graphs and Graph Theory

What is a graph?

Labeled Property Graph

- Composed of a set of node (vertex) objects and relationship (edge) objects
- Labels are used to mark a node as part of a group
- Properties are attributes (think KV pairs) and can exist on nodes and relationships
- Nodes with no associated relationships are OK. Edges not connected to nodes are <u>not</u> permitted.

Example



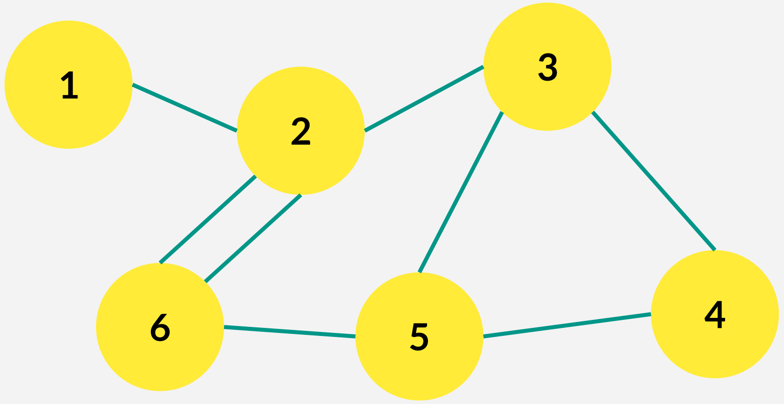
2 Labels:
- person
- car

4 relationship types:
- Drives
- Owns
- Lives_with
- Married_to

Properties

Paths

A ***path*** is an ordered sequence of nodes connected by edges in which no nodes or edges are repeated.



Ex: $1 \rightarrow 2 \rightarrow 6 \rightarrow 5$

Not a path:
$1 \rightarrow 2 \rightarrow 6 \rightarrow 2 \rightarrow 3$

Flavors of Graphs

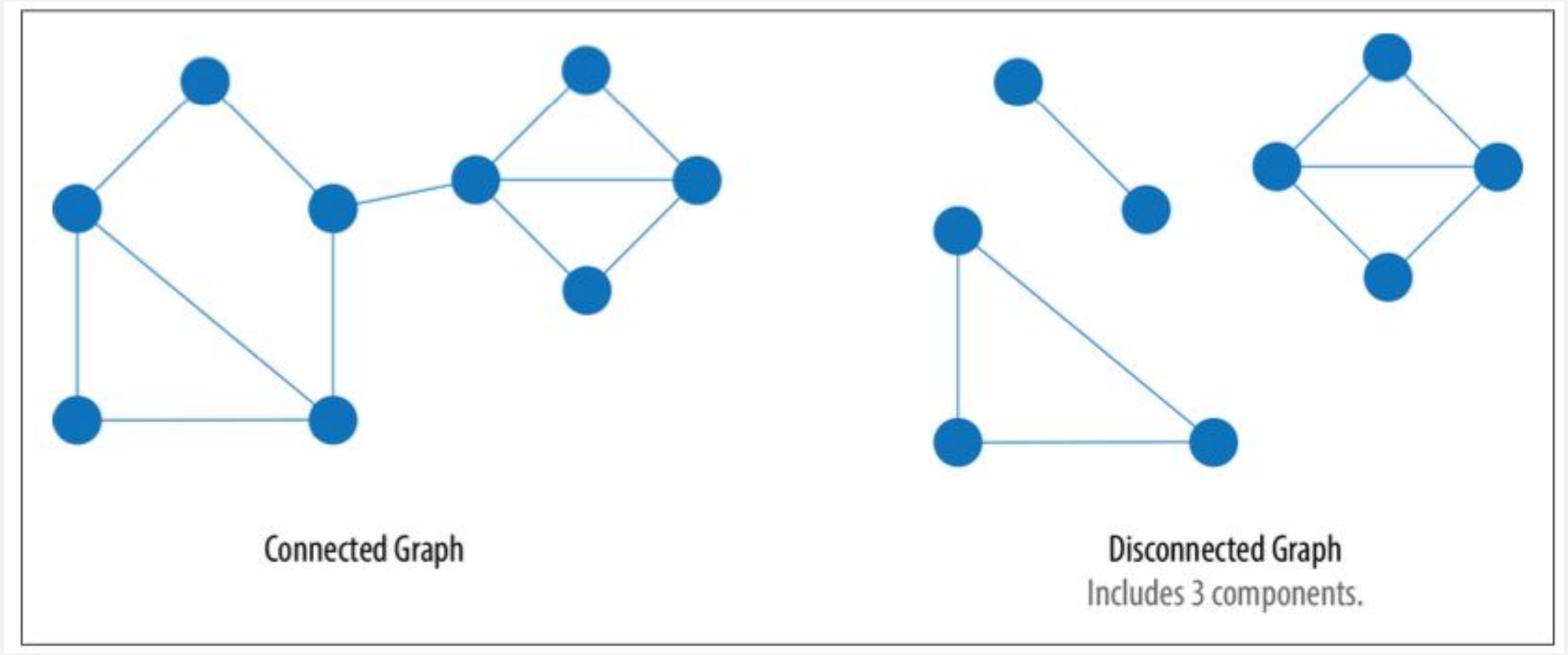
Connected (vs. Disconnected) – there is a path between any two nodes in the graph

Weighted (vs. Unweighted) – edge has a weight property (important for some algorithms)

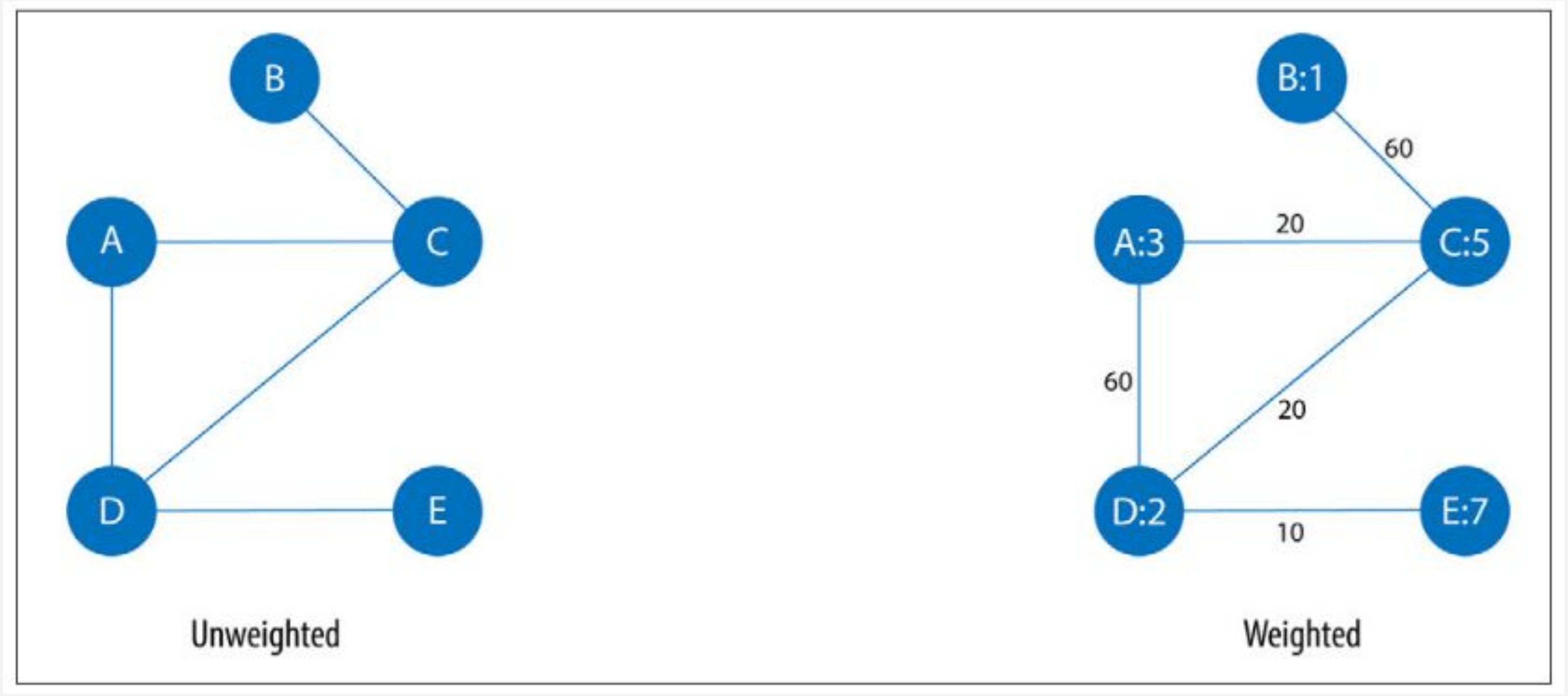
Directed (vs. Undirected) – relationships (edges) define a start and end node

Acyclic (vs. Cyclic) – Graph contains no cycles

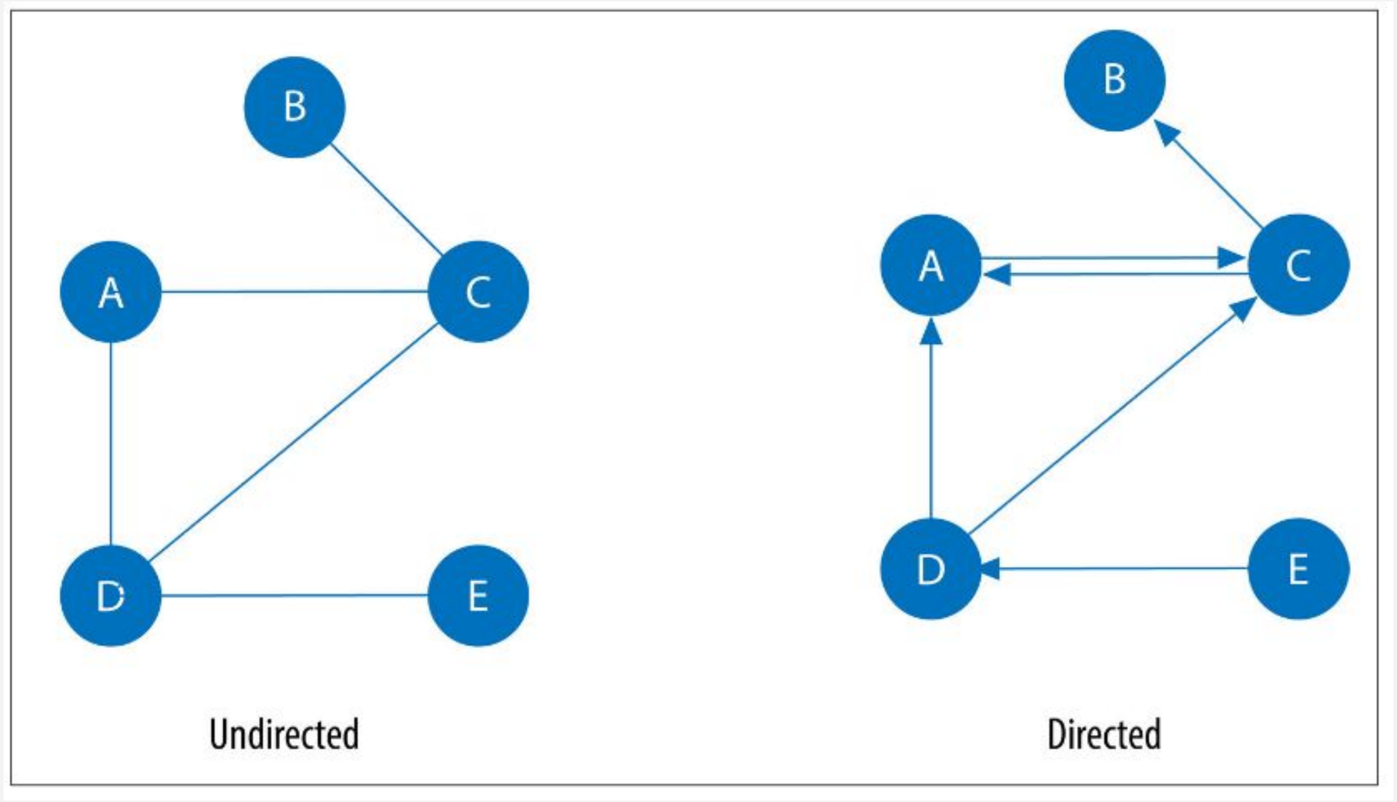
Connected vs. Disconnected



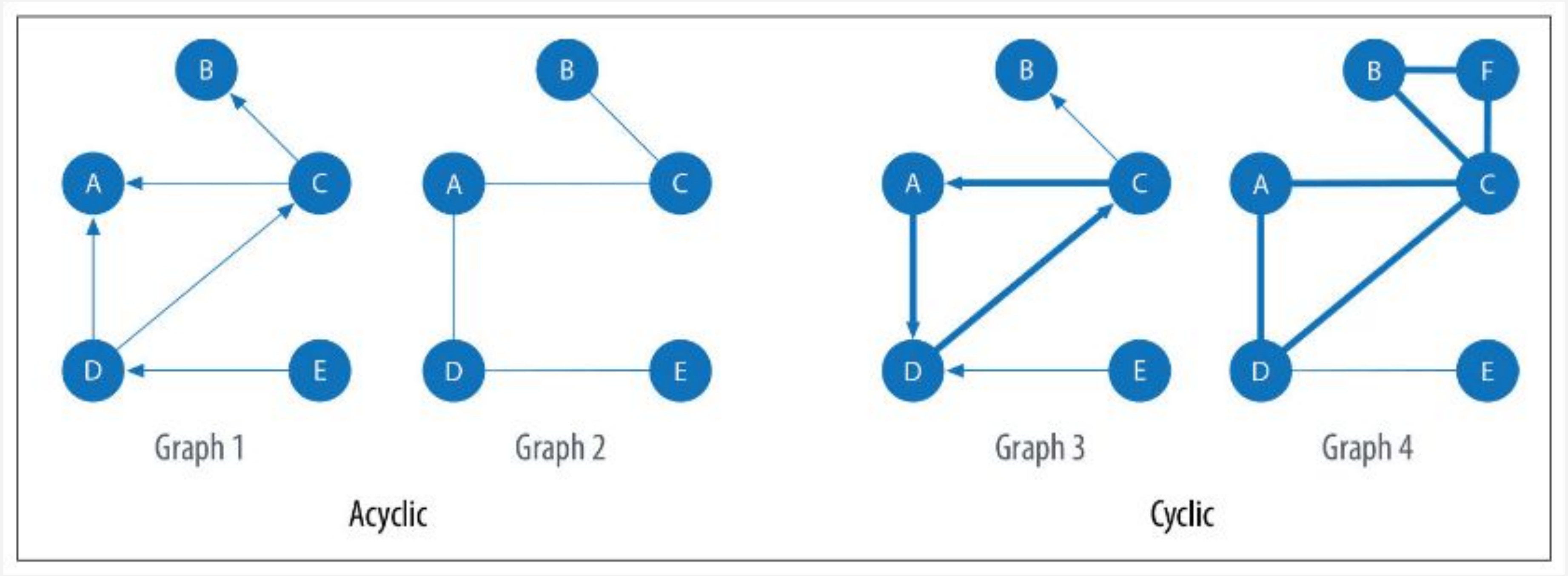
Weighted vs. Unweighted



Directed vs. Undirected

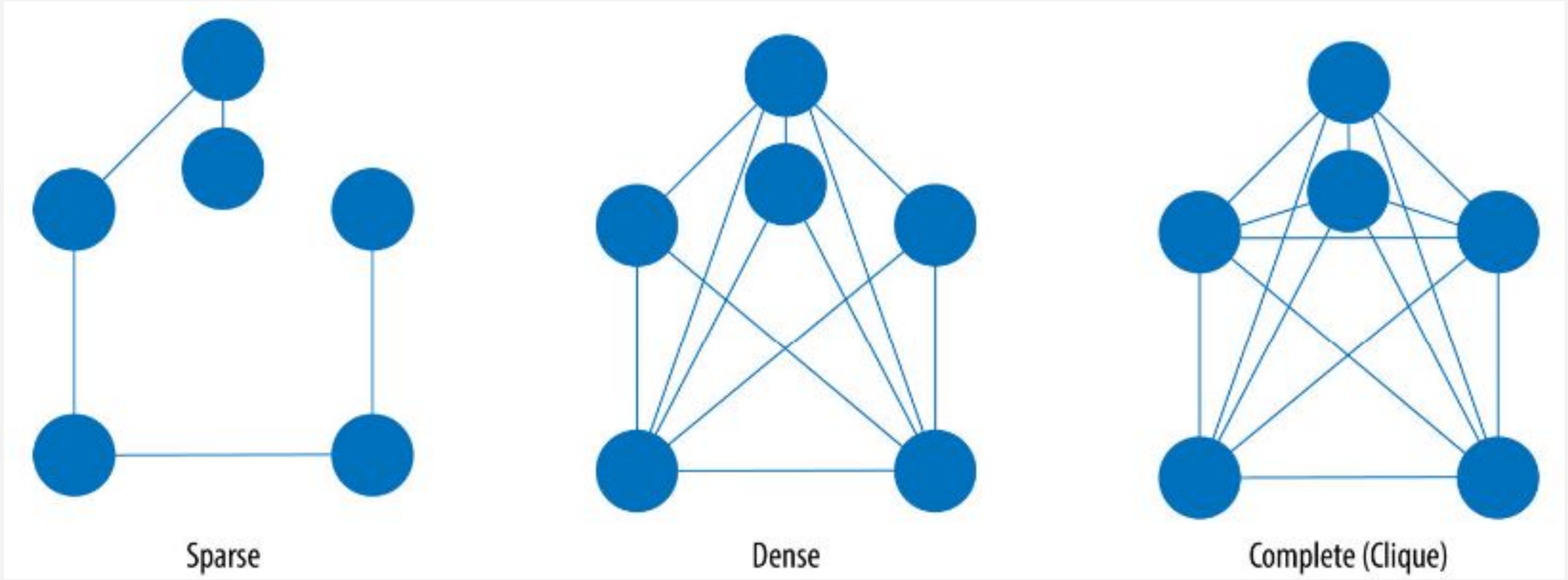


Cyclic vs Acyclic

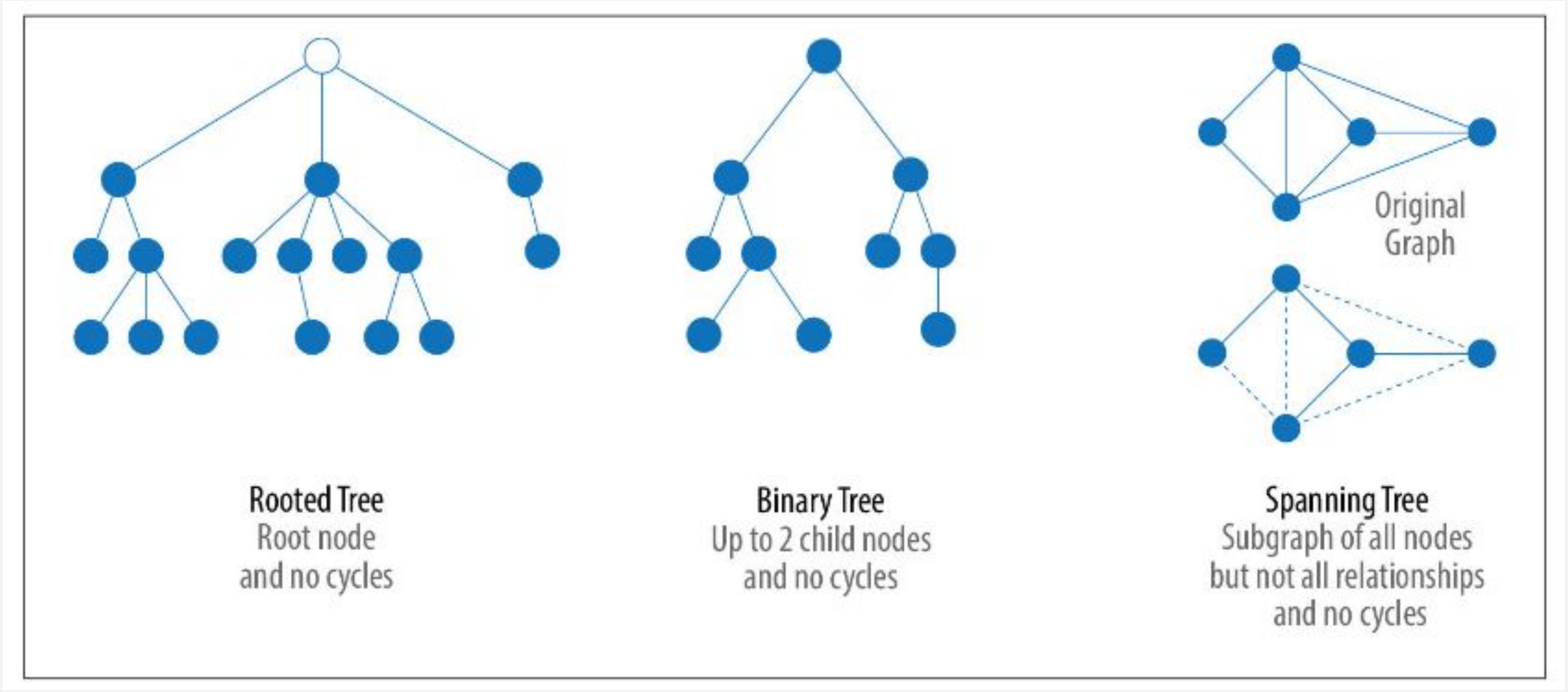


* even without directions, a graph is acyclic if paths do not exist that allow you to get back to where you started

Sparse vs. Dense



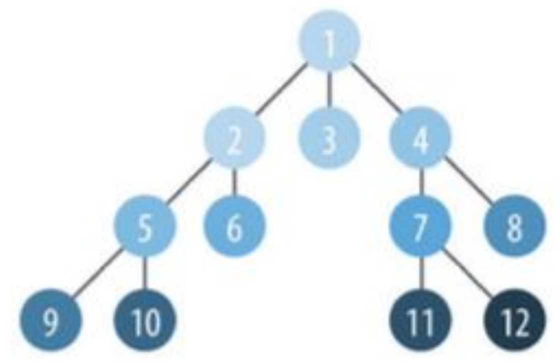
Trees


Original Graph
Rooted Tree
Root node and no cycles
Binary Tree
Up to 2 child nodes and no cycles
Spanning Tree
Subgraph of all nodes but not all relationships and no cycles

Types of Graph Algorithms - Pathfinding

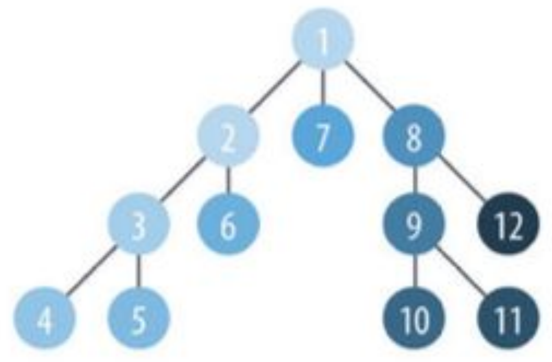
- **Pathfinding**
  - finding the shortest path between two nodes, if one exists, is probably the most common operation
  - “shortest” means fewest edges or lowest weight
  - Average Shortest Path can be used to monitor efficiency and resiliency of networks.
  - Minimum spanning tree *(keep edges of tree with lowest total weight), cycle detection, max/min flow… are other types of pathfinding

BFS vs DFS



Breadth First Search

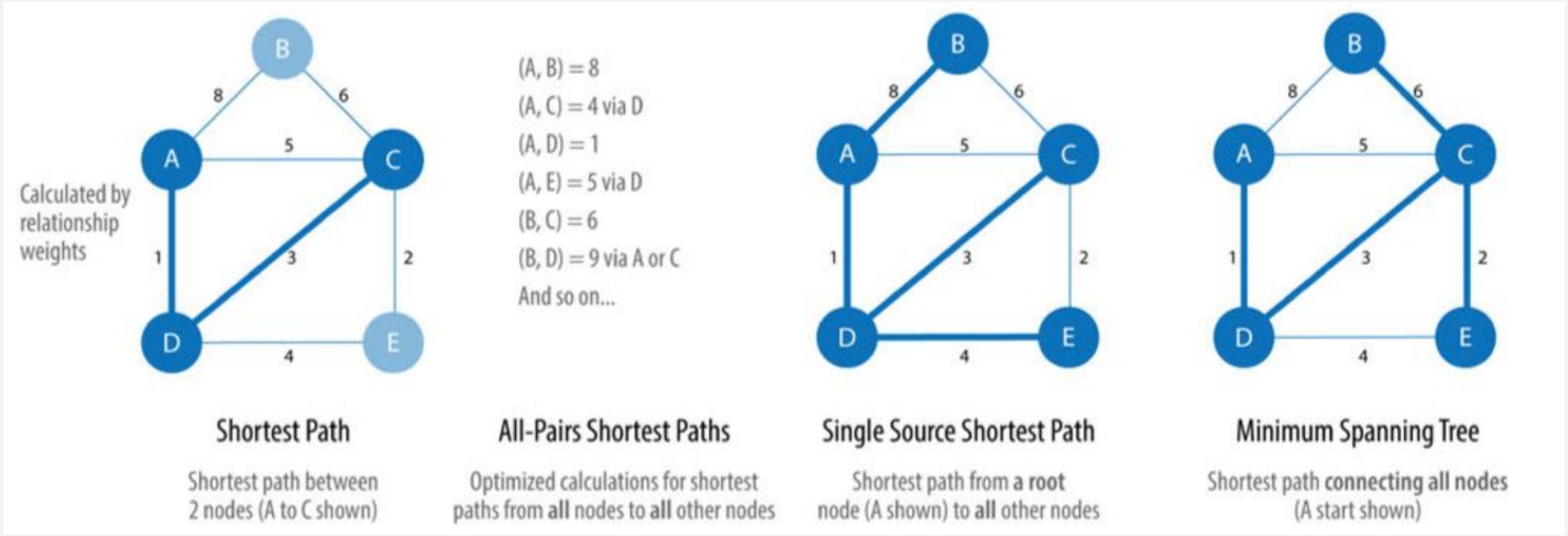
Visits nearest neighbors first



Depth First Search

Walks down each branch first

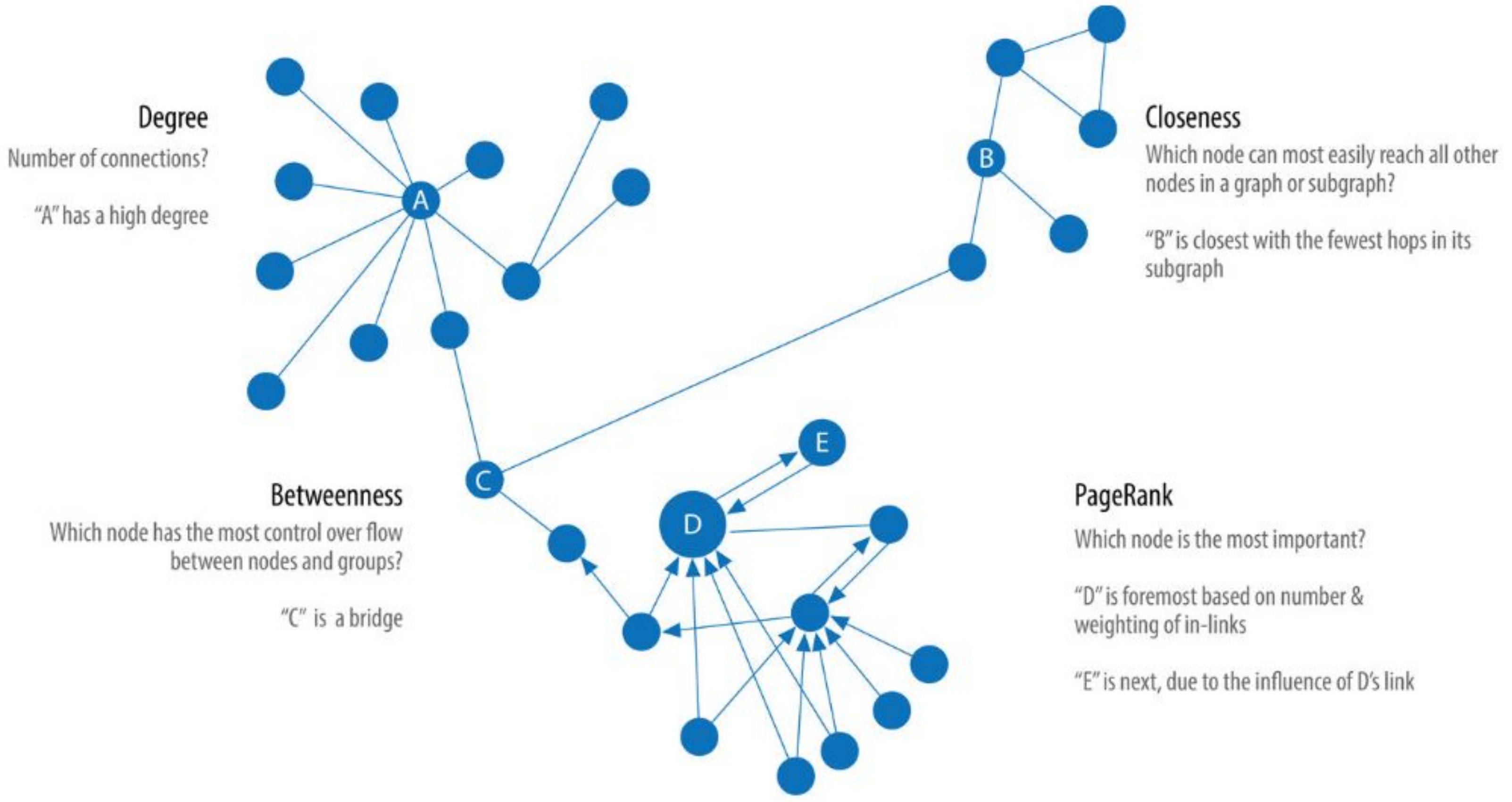
Shortest Path


B
8
6
A
5
C
Calculated by relationship weights
1
3
2
D
4
E
(A, B) = 8
(A, C) = 4 via D
(A, D) = 1
(A, E) = 5 via D
(B, C) = 6
(B, D) = 9 via A or C
And so on...
Shortest Path
Shortest path between 2 nodes (A to C shown)
All-Pairs Shortest Paths
Optimized calculations for shortest paths from all nodes to all other nodes
Single Source Shortest Path
Shortest path from a root node (A shown) to all other nodes
Minimum Spanning Tree
Shortest path connecting all nodes (A start shown)

Types of Graph Algorithms - Centrality & Community Detection

- **Centrality**
  - determining which nodes are “more important” in a network compared to other nodes
  - EX: Social Network Influencers?
- **Community Detection**
  - evaluate clustering or partitioning of nodes of a graph and tendency to strengthen or break apart

Centrality



Degree
Number of connections?
"A" has a high degree
A
Closeness
Which node can most easily reach all other nodes in a graph or subgraph?
"B" is closest with the fewest hops in its subgraph
B
Betweenness
Which node has the most control over flow between nodes and groups?
"C" is a bridge
C
PageRank
Which node is the most important?
"D" is foremost based on number & weighting of in-links
"E" is next, due to the influence of D's link
D
E

Some Famous Graph Algorithms

- **Dijkstra's Algorithm** - single-source shortest path algo for positively weighted graphs
- **A* Algorithm** - Similar to Dijkstra's with added feature of using a heuristic to guide traversal
- **PageRank** - measures the importance of each node within a graph based on the number of incoming relationships and the importance of the nodes from those incoming relationships

Neo4j

- A Graph Database System that supports both transactional and analytical processing of graph-based data
- Relatively new class of no-sql DBs
- Considered schema optional (one can be imposed)
- Supports various types of indexing
- ACID compliant
- Supports distributed computing
- Similar: Microsoft CosmoDB, Amazon Neptune

??