Graphs, Trees, and How to Visualize Them

November 16th, 2021

Ashley Suh – PhD Candidate – Tufts University – Ashley.Suh@Tufts.edu

slides credits:

Paul Rosen (U of South Florida), Alex Bigelow (U of Utah), Miriah Meyer (Linköping U), Hanspeter Pfister (Harvard), Jeff Heer (U of Washington)

Outline for today

Graphs? Graphs!

- Clarification & motivation

Establishing common graph vocabulary

- Definitions, formalisms

A tour through the Tree & Graph Visualization Zoo

- Examples

Effective graph drawing

- Common graph visualization problems
- Common graph visualization solutions

In-class activity & goodbyes

etiquette

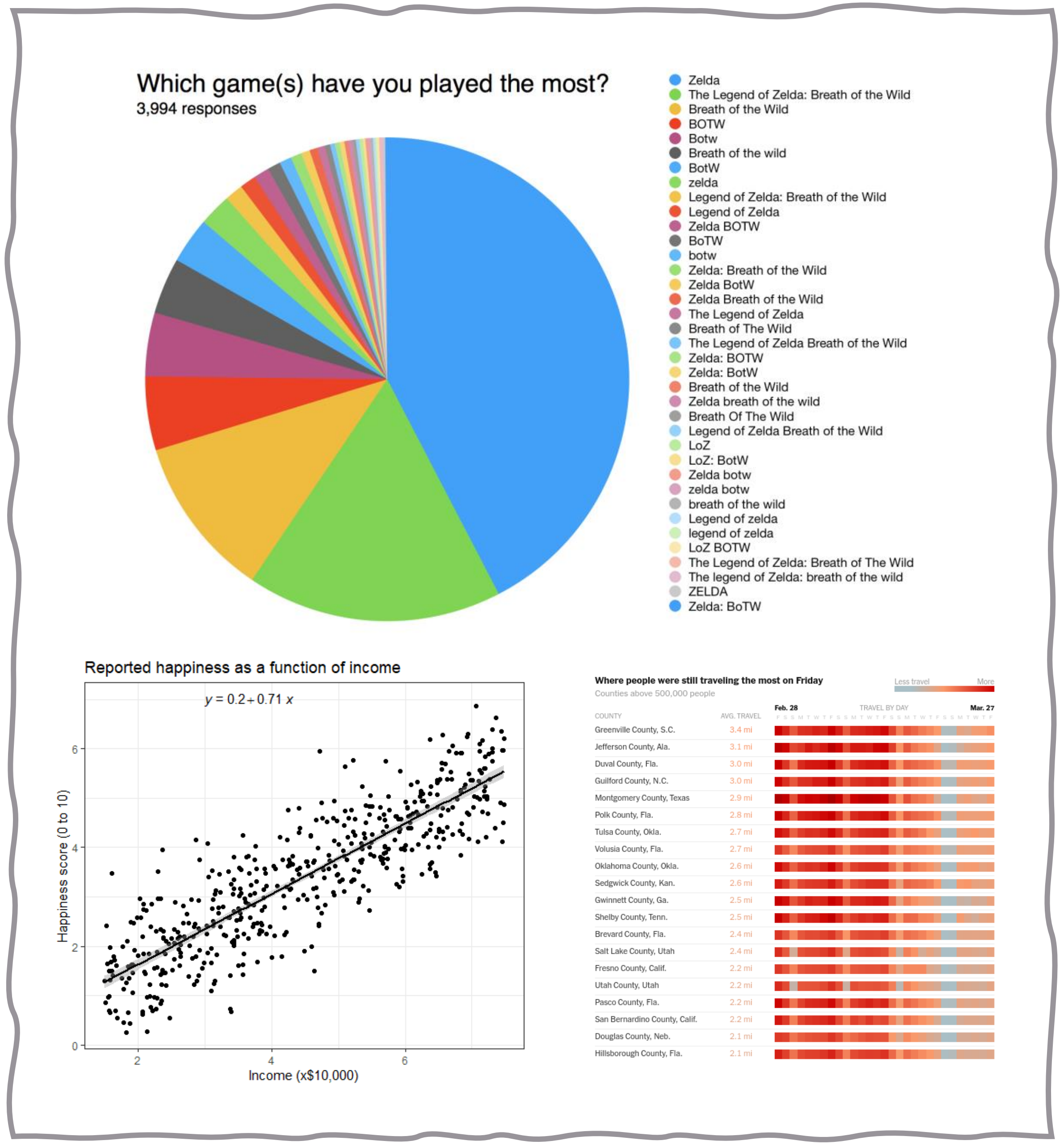
- Feel free to walk around during lecture, we'll break to stretch + drink water
- Interrupt me or use 'raise hand' to ask questions and/or answer questions
- Use chat for discussion and asking each other questions

Who I am

- CS undergrad at USF, MS at Tufts
  - Started working on graph research as an undergrad
- PhD candidate in the visual analytics lab at Tufts
- Lover of graphs & visualization

Me giving a talk about one of my favorite things: graphs

Graphs:
what are they, what aren't they?



Graphs

Common charts that represent data are often referred to as "graphs", or "graph visualizations"

- Bar charts
- Line charts
- Pie charts
- Etc.

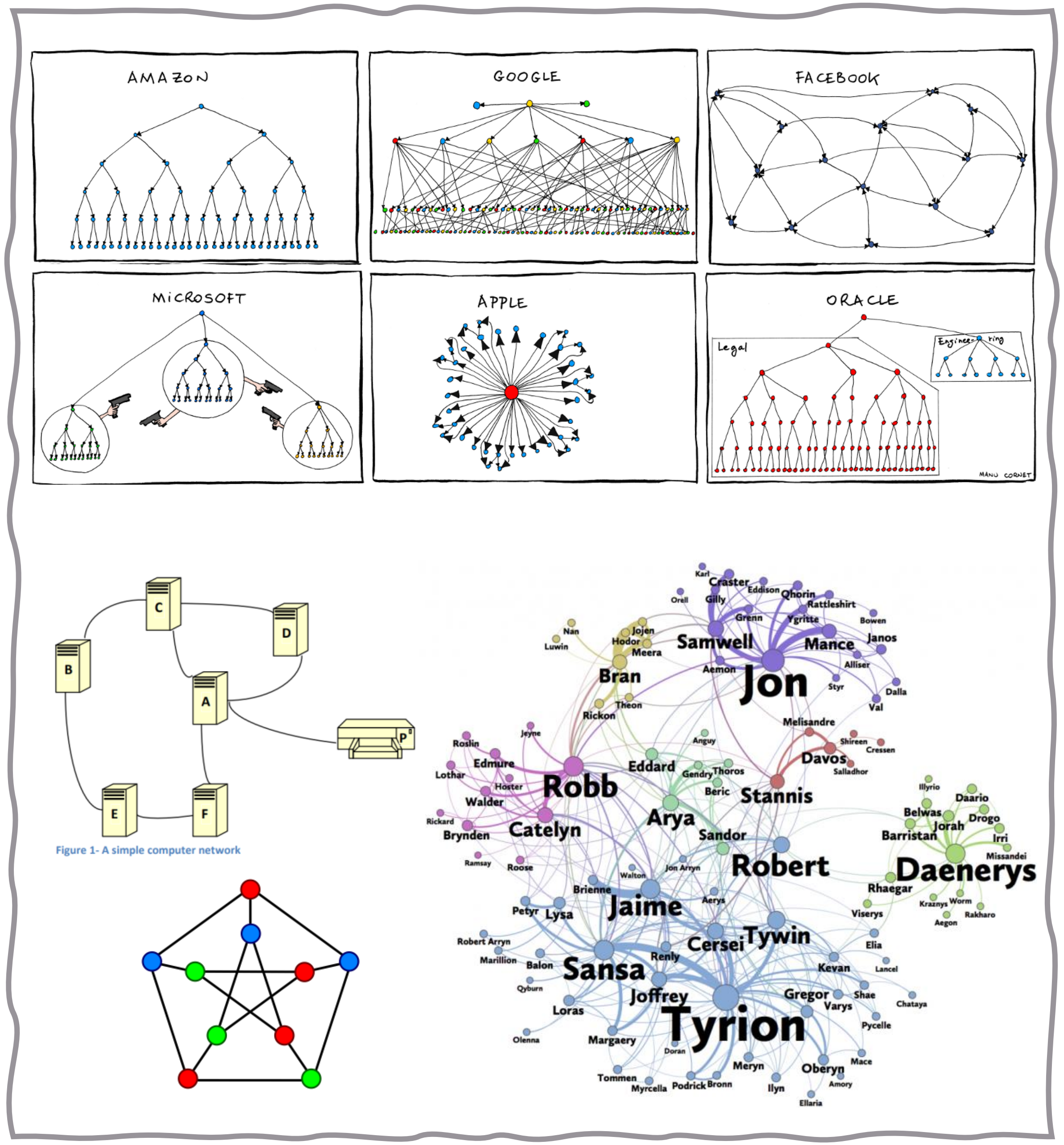
Clarification: for this lecture, when I refer to **graphs**, I do *not* mean the type of charts shown on the left.

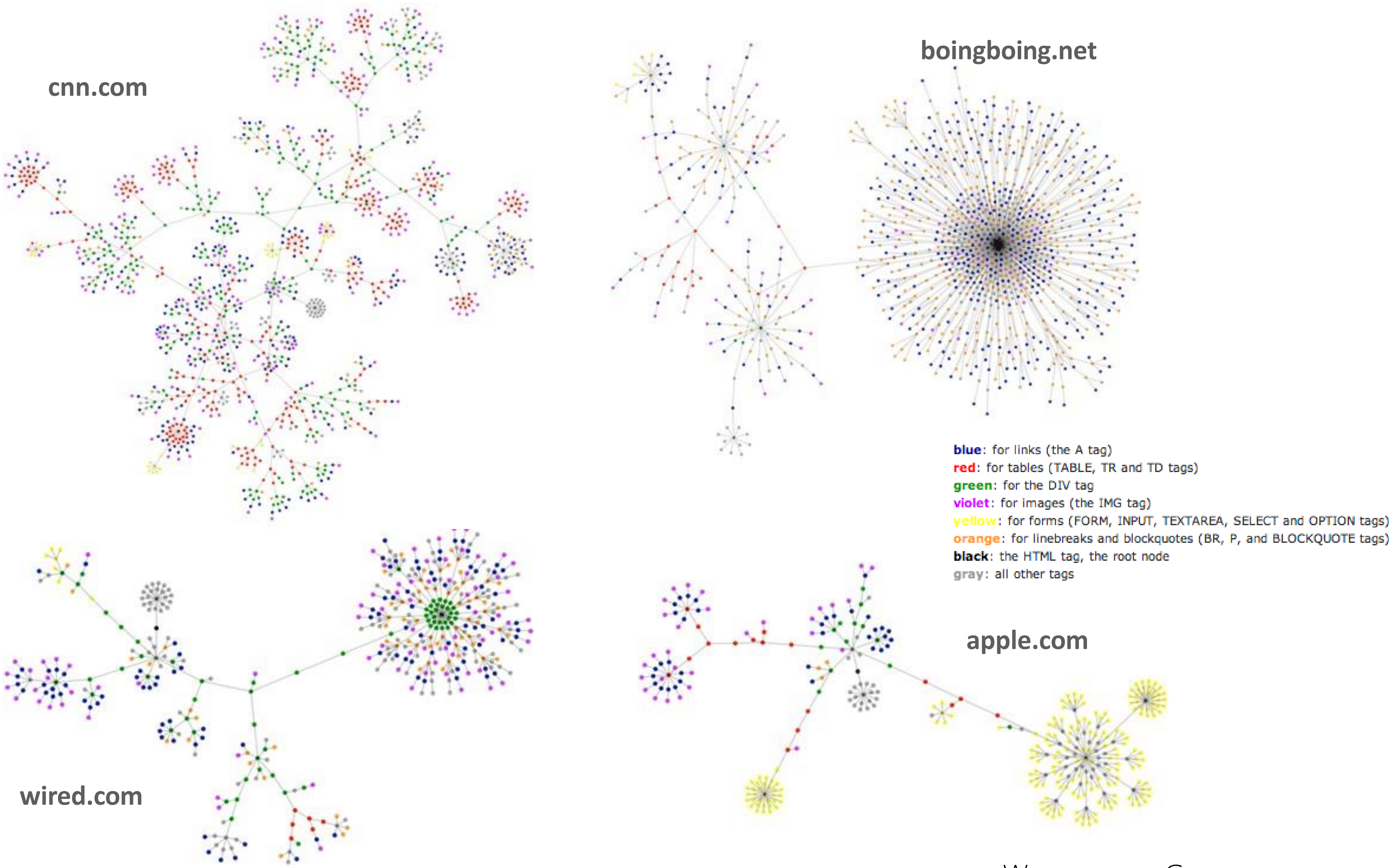
Graphs

Let’s instead talk about **graphs**, **networks**, & **trees** in the mathematical sense: a model for representing items and the relationships between those items

- Social / friendship networks
- Computer networks
- Energy or transportation grids
- Organizational structures
- Etc.

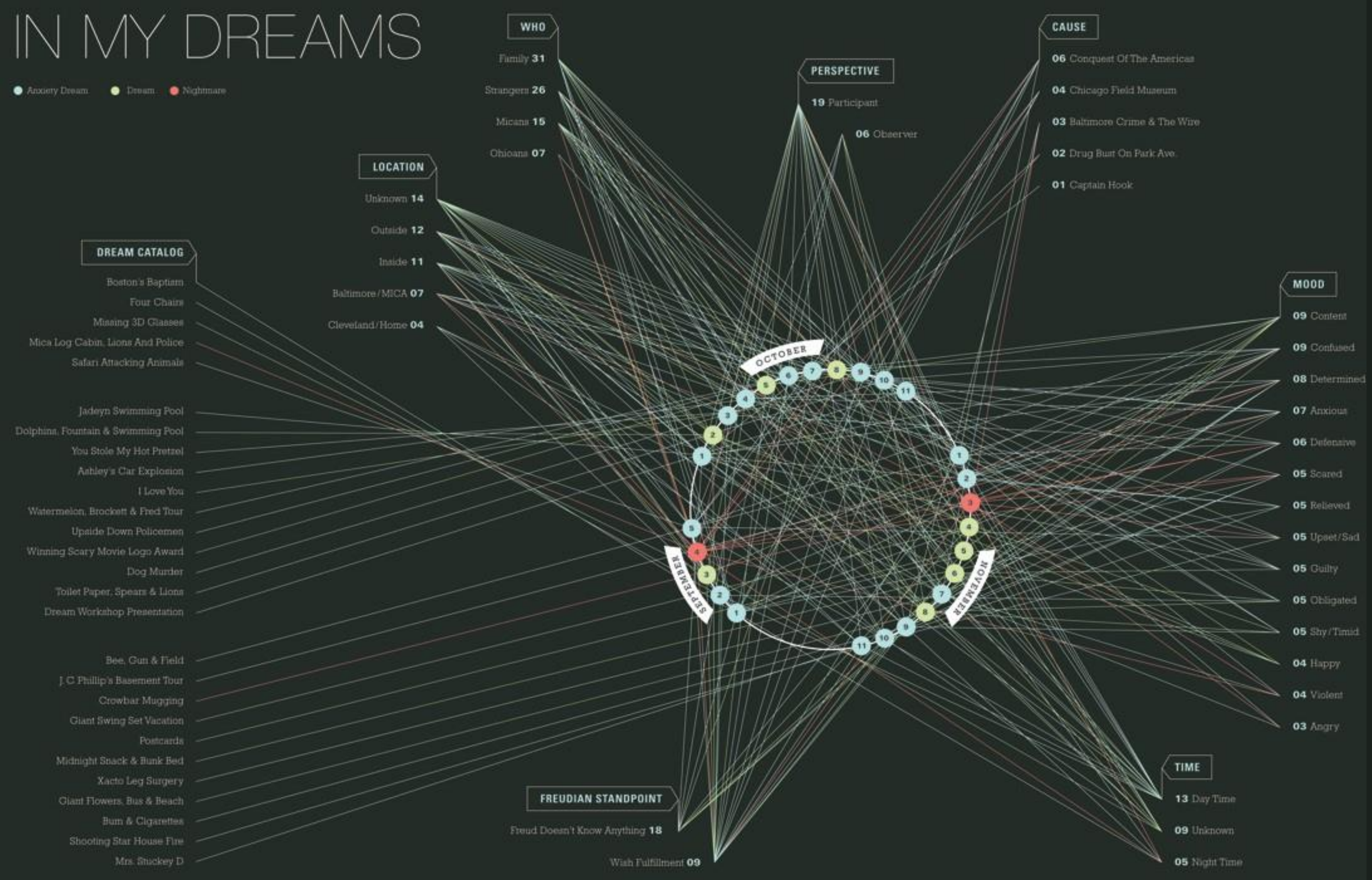
Why do we care about visualizing graphs?





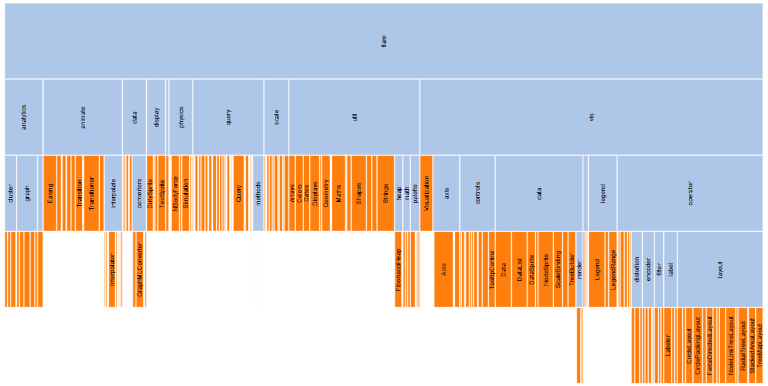
Webpages as Graphs

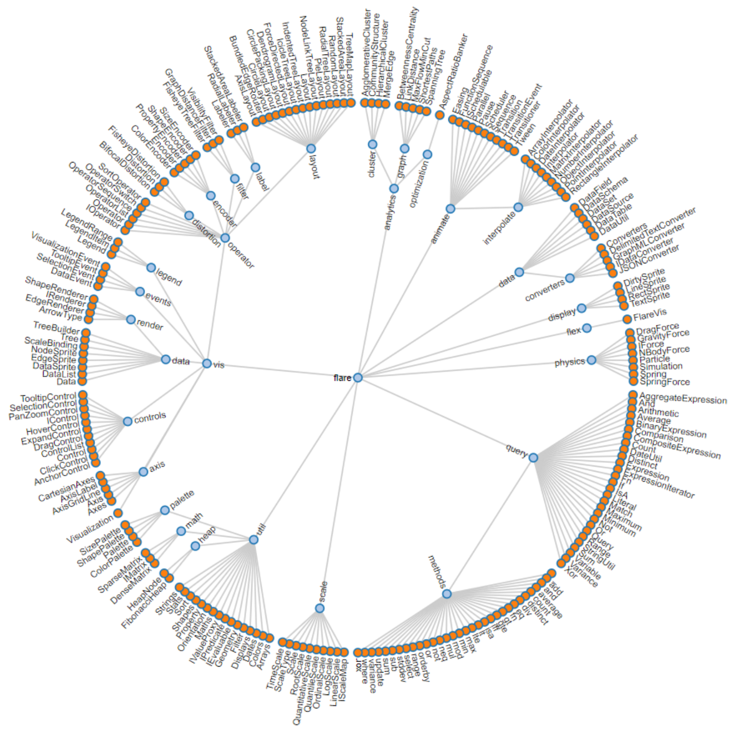
[Paul Butler]

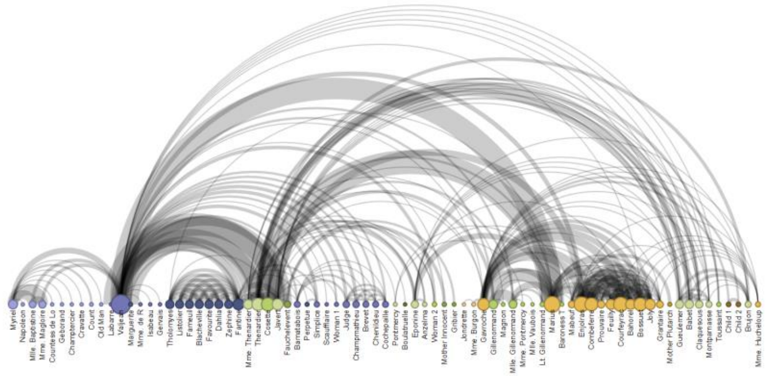


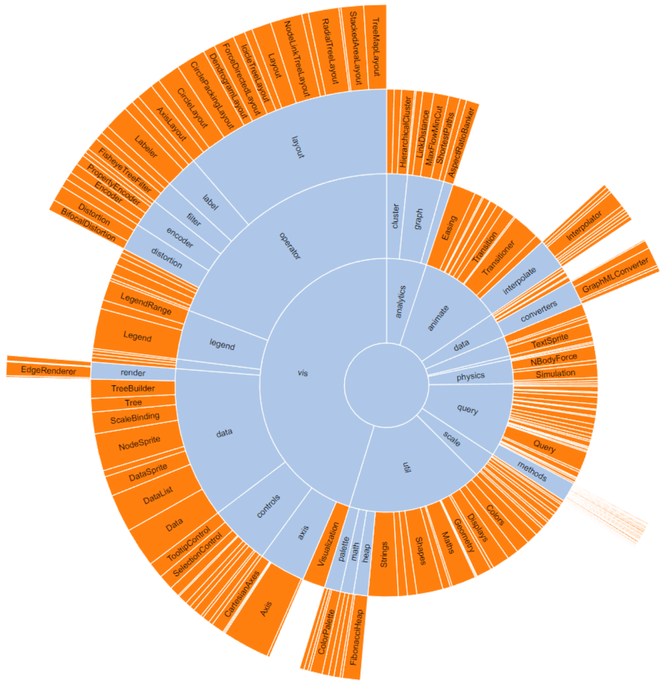
Kailie Parrish

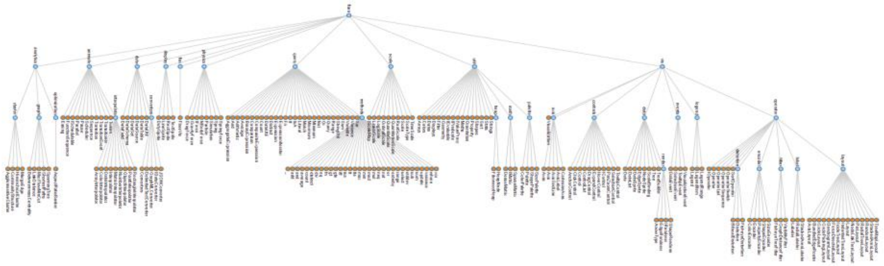
There are many graph visualization techniques

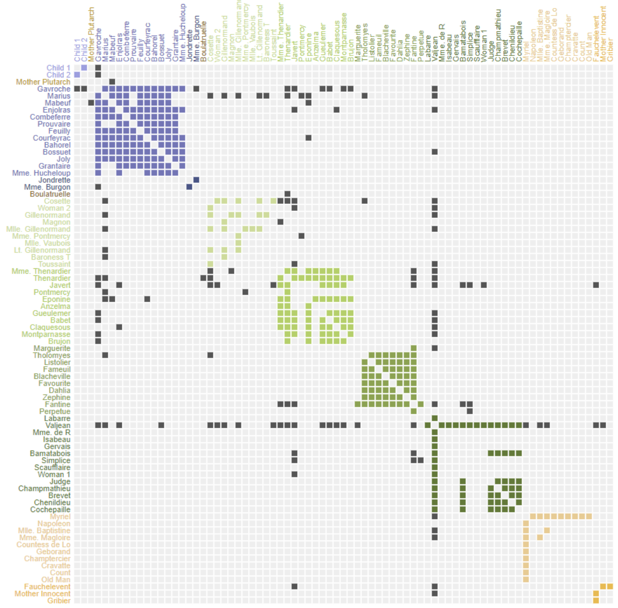






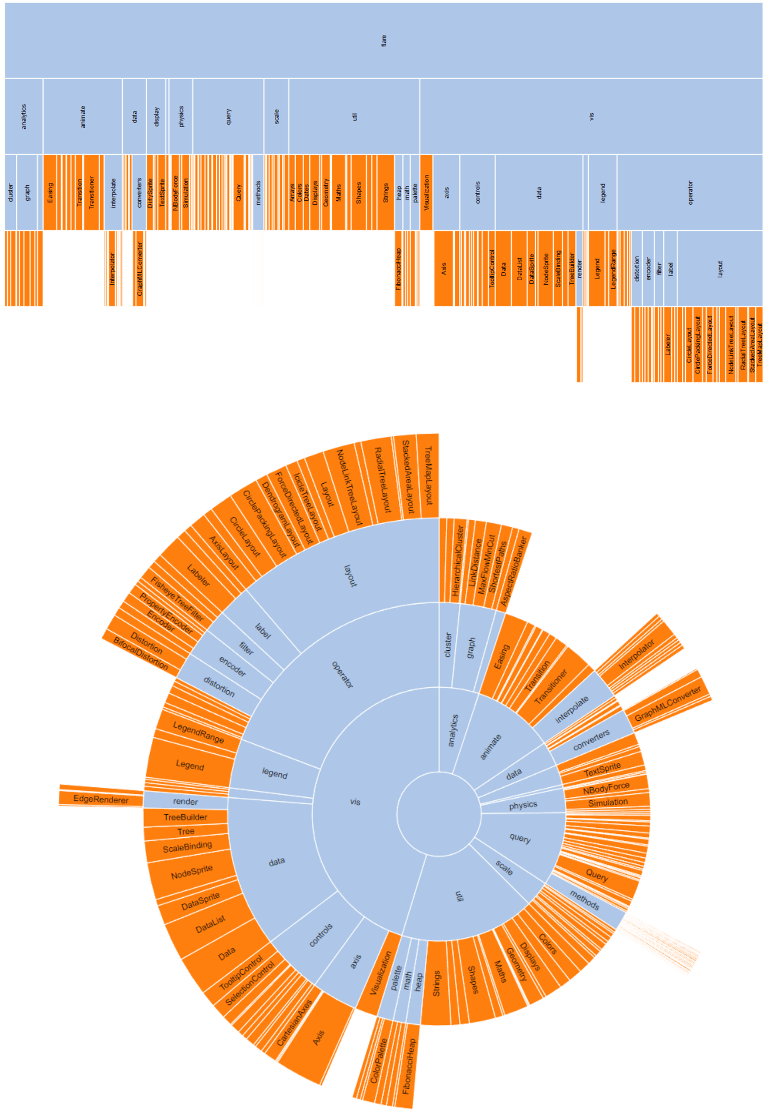




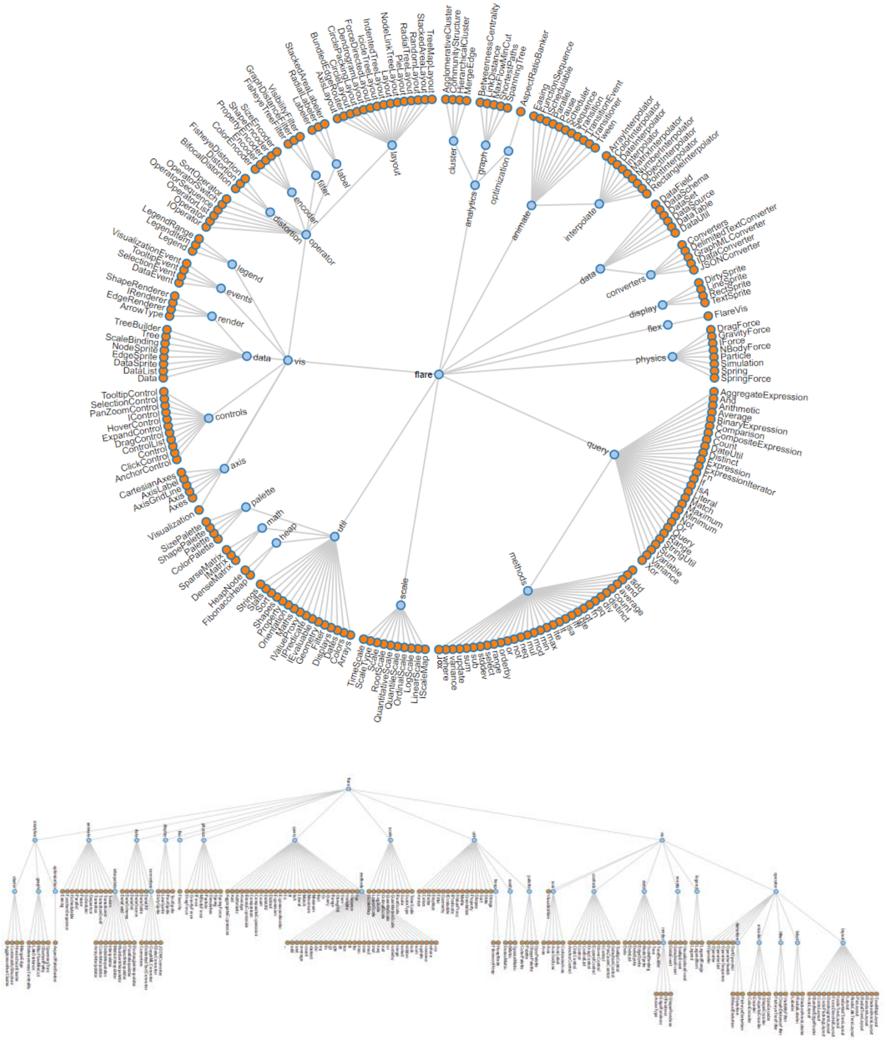


+ many, many more that can't fit in this slide

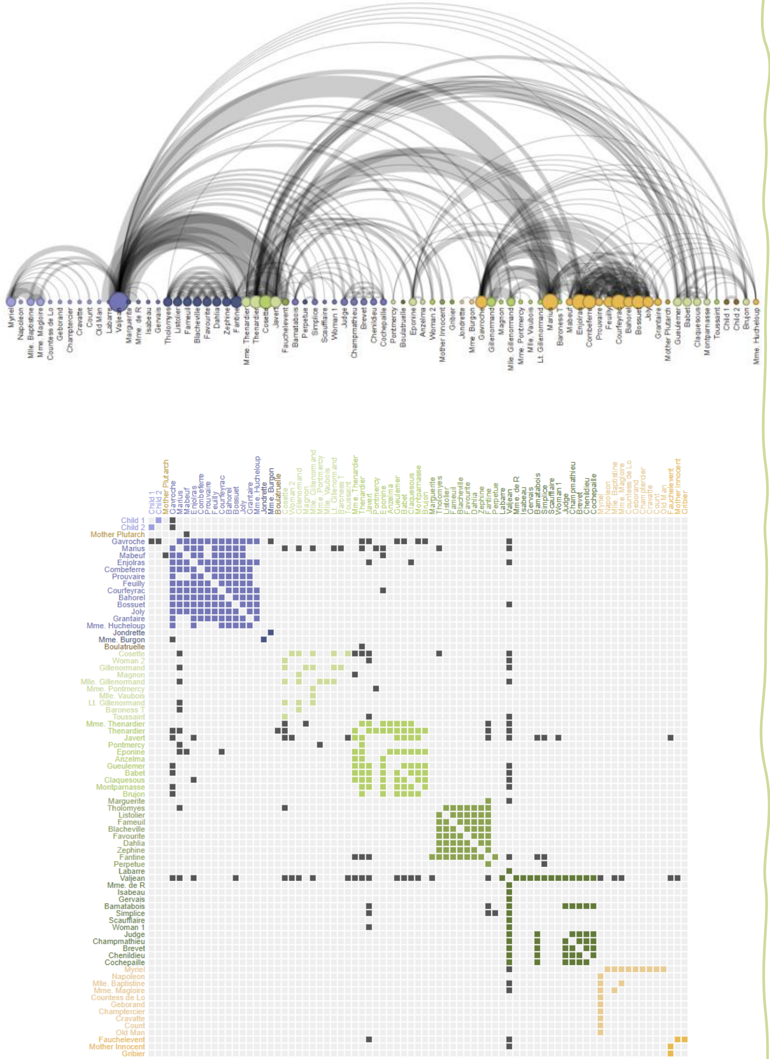
Which graph visualization is best?



same data being visualized on top / bot

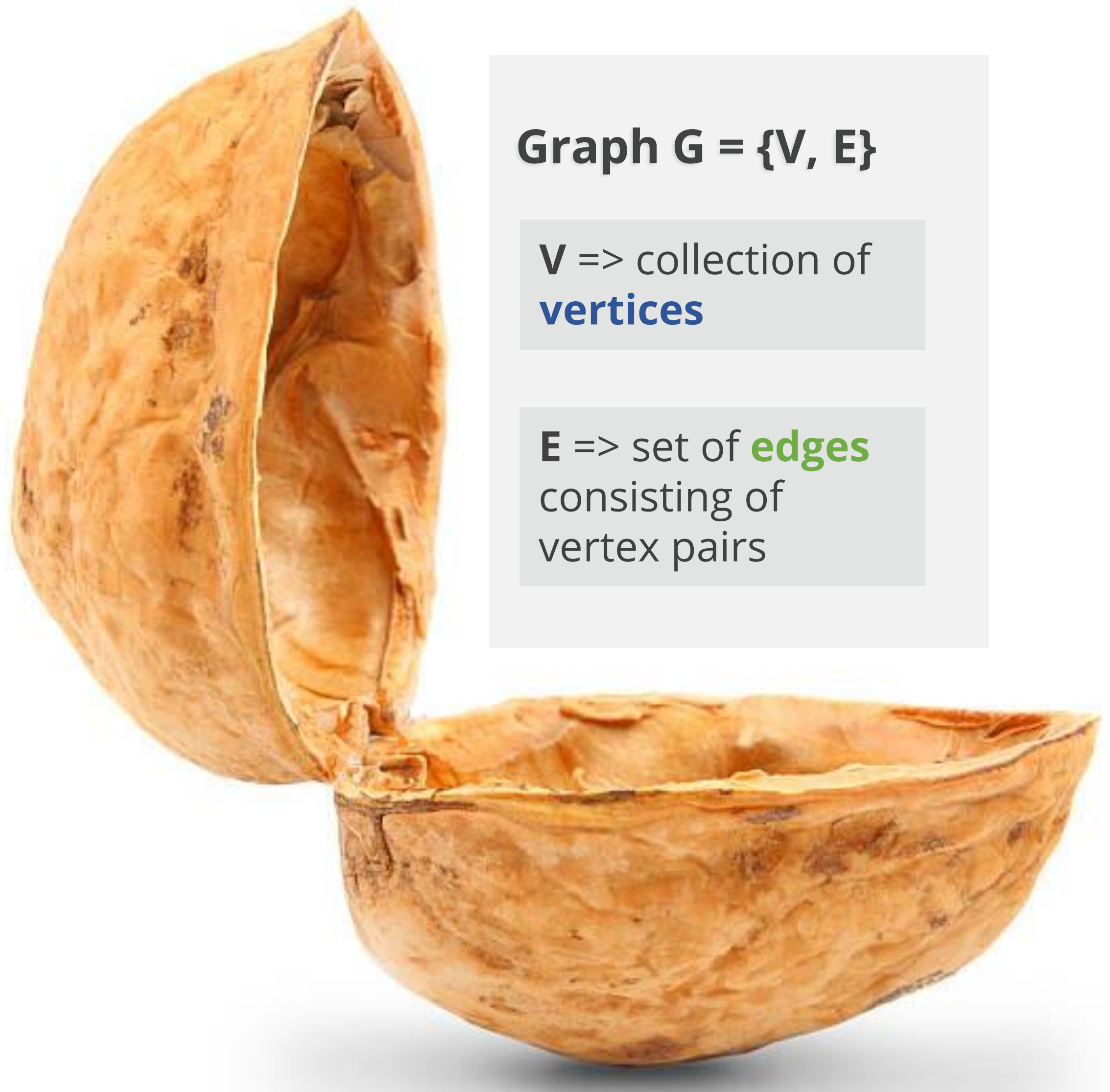


same data being visualized on top / bot



same data being visualized on top / bot

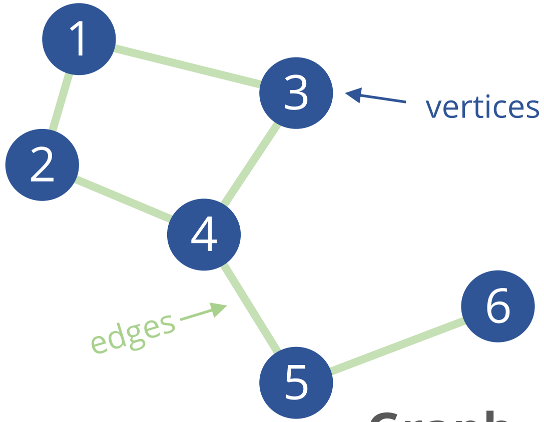
Establishing common graph vocabulary



Graph G = {V, E}

V => collection of **vertices**

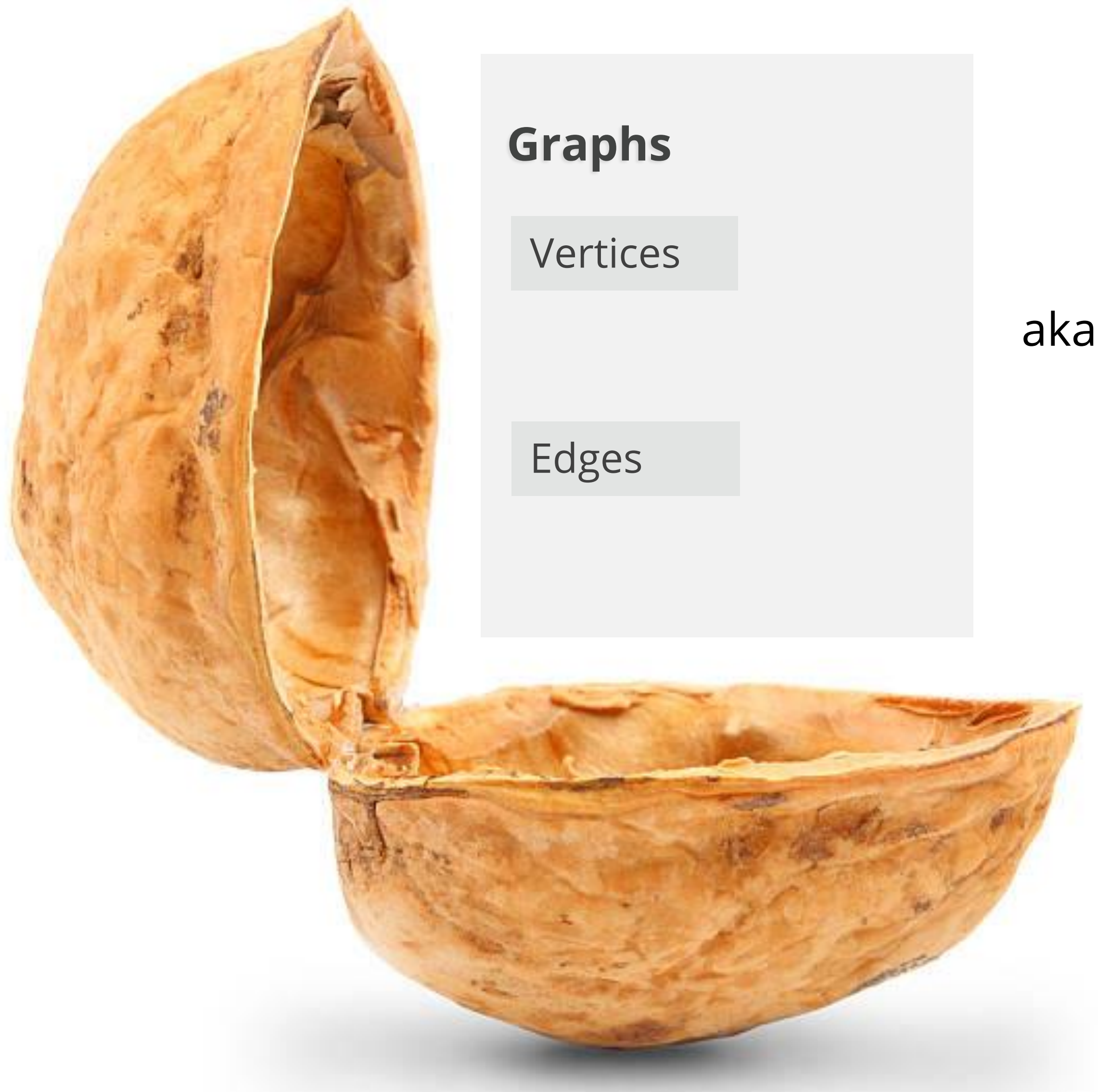
E => set of **edges** consisting of vertex pairs



Graph
(represented as a node-link diagram)

G = {V, E}

V = {1, 2, 3, 4, 5, 6}
E = {(1,2), (1,3), (2,4), (3,4), (4,5), (5,6)}



Graphs

Vertices

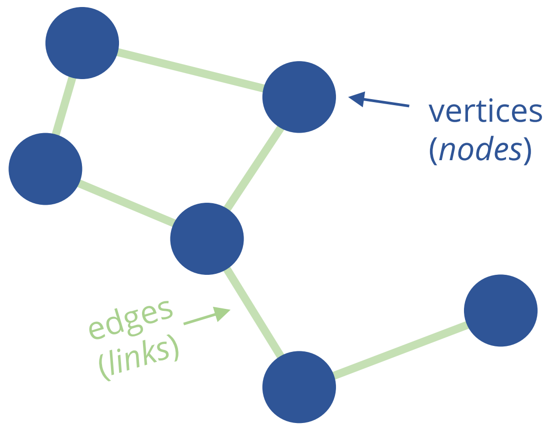
Edges

aka

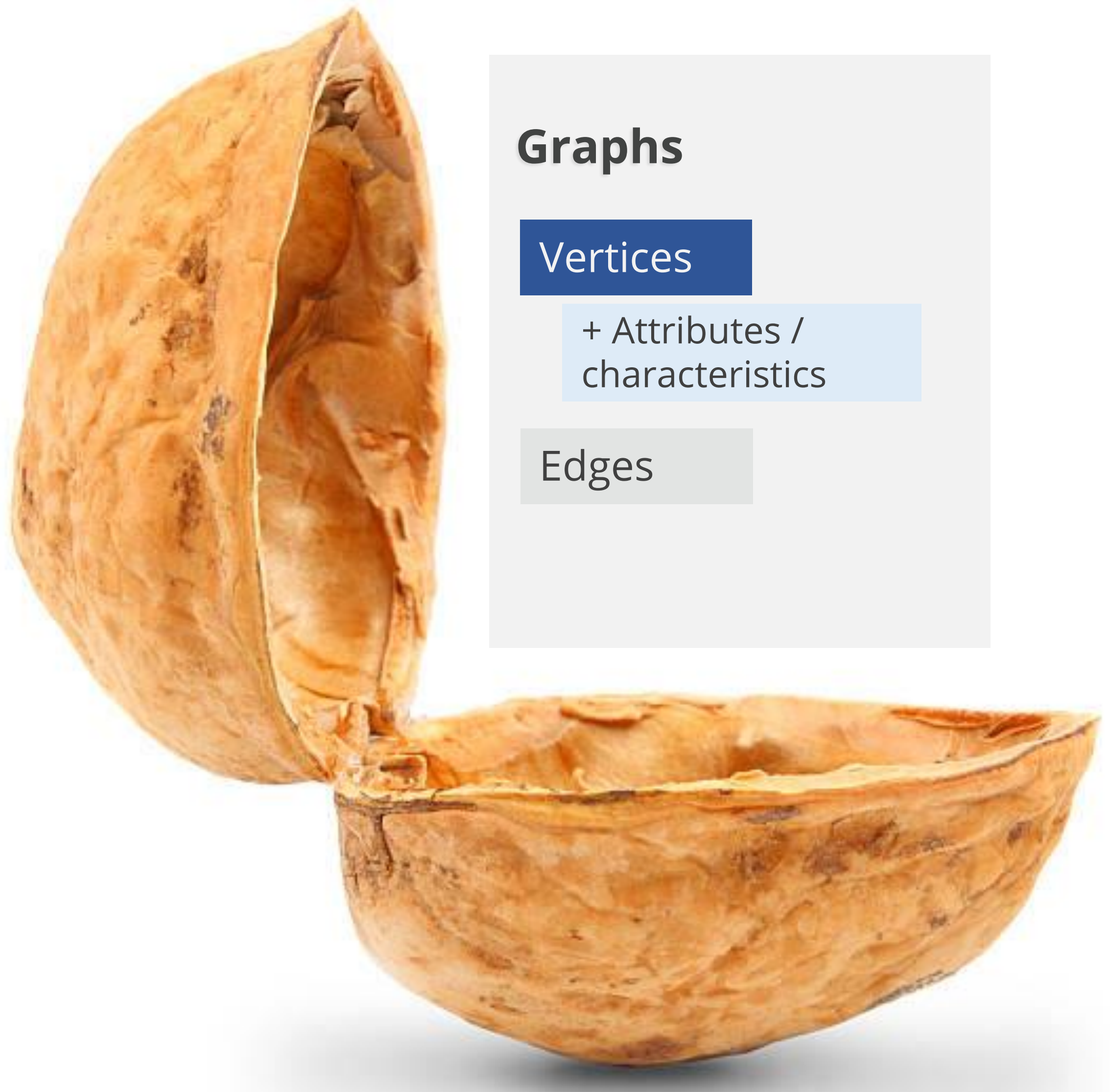
Networks

Nodes

Links



Graph (*network*)

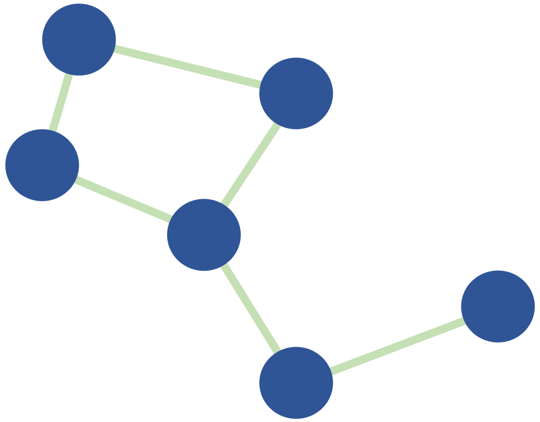


Graphs

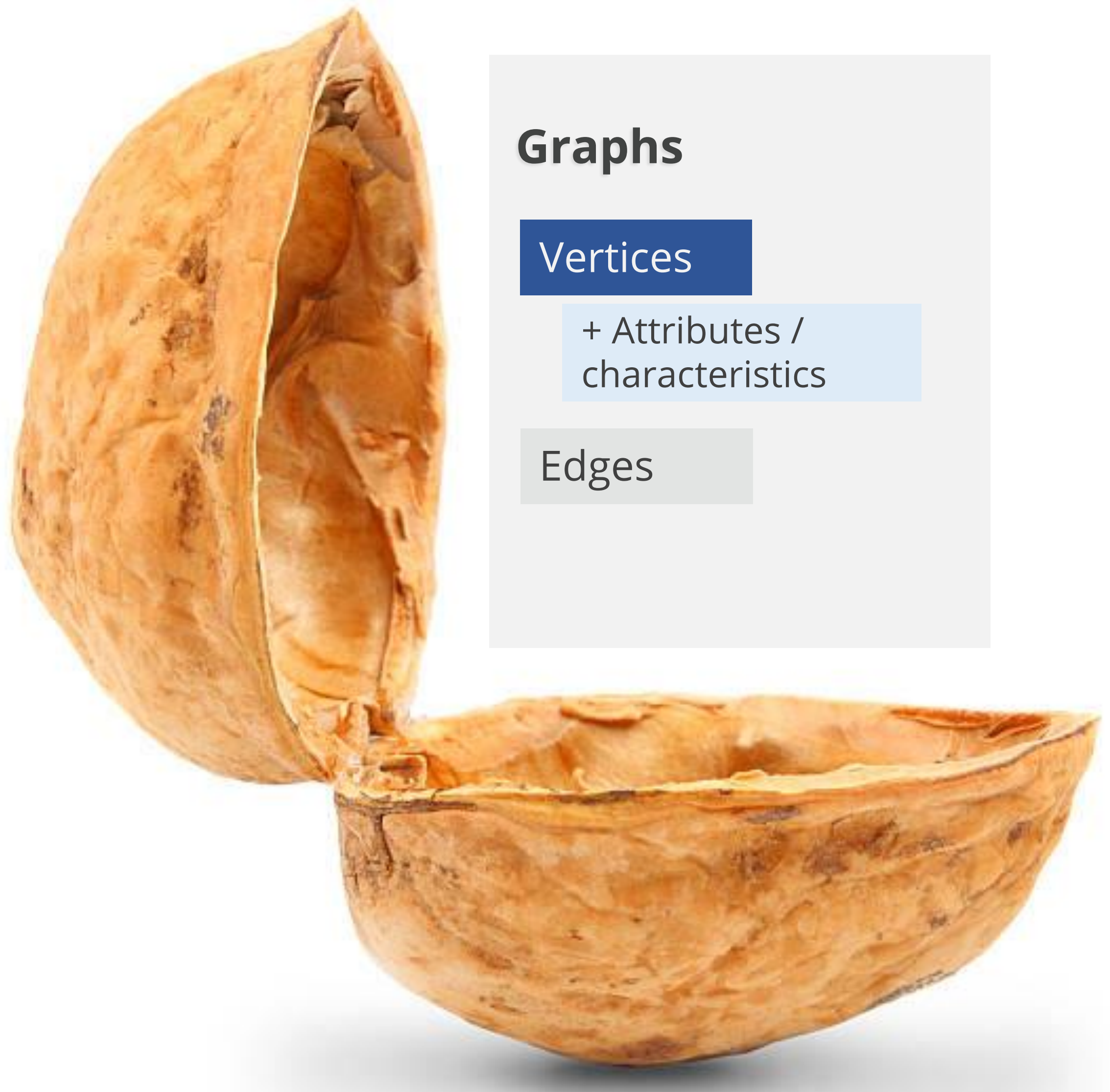
Vertices

+ Attributes / characteristics

Edges



Vertex id	Name	Favorite color	Popularity
1	Sam	Blue	6
2	Sebastian	Green	7
3	Abigail	Purple	8
4	Haley	Pink	2
5	Shane	Orange	4
6	Leah	Purple	7

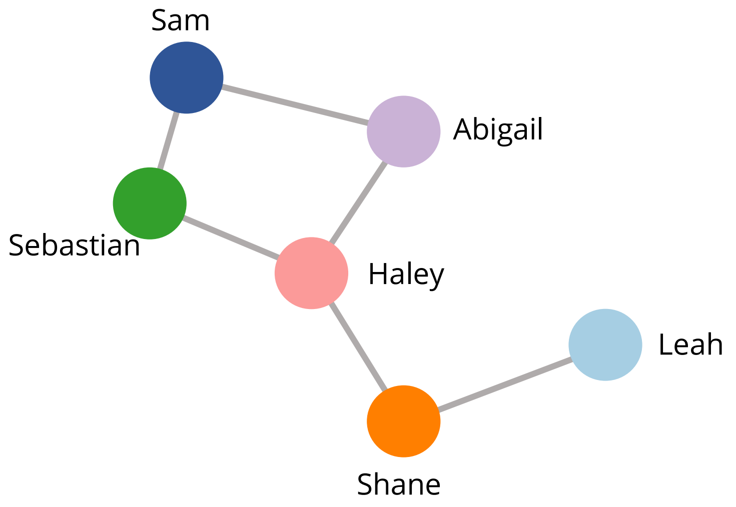


Graphs

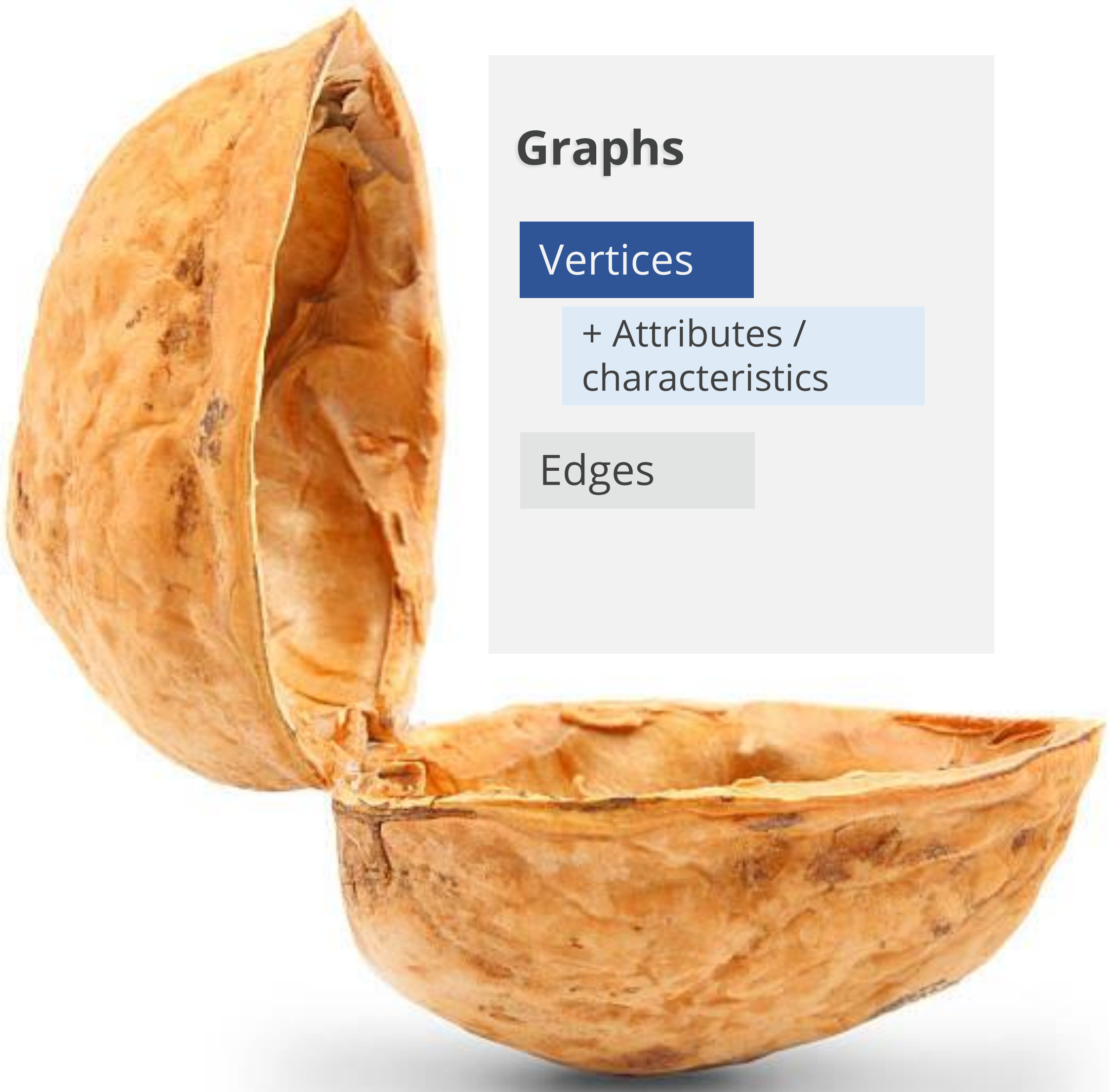
Vertices

+ Attributes / characteristics

Edges



Vertex id	Name	Favorite color	Popularity
1	Sam	Blue	6
2	Sebastian	Green	7
3	Abigail	Purple	9
4	Haley	Pink	2
5	Shane	Orange	4
6	Leah	Purple	7

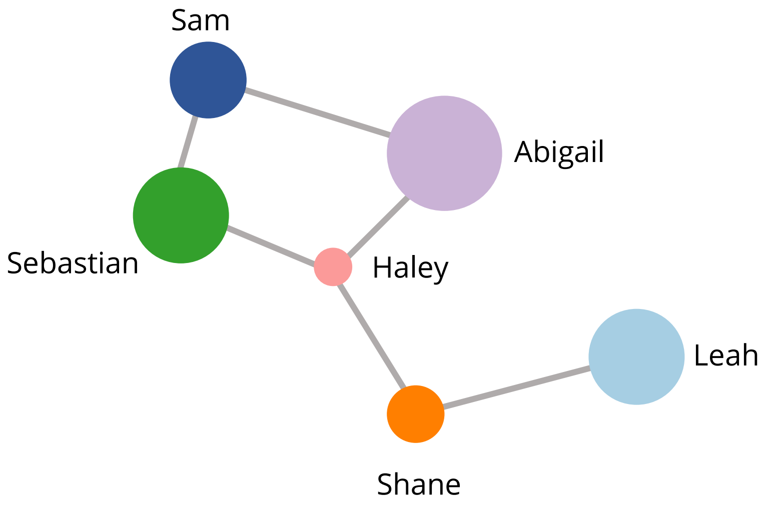


Graphs

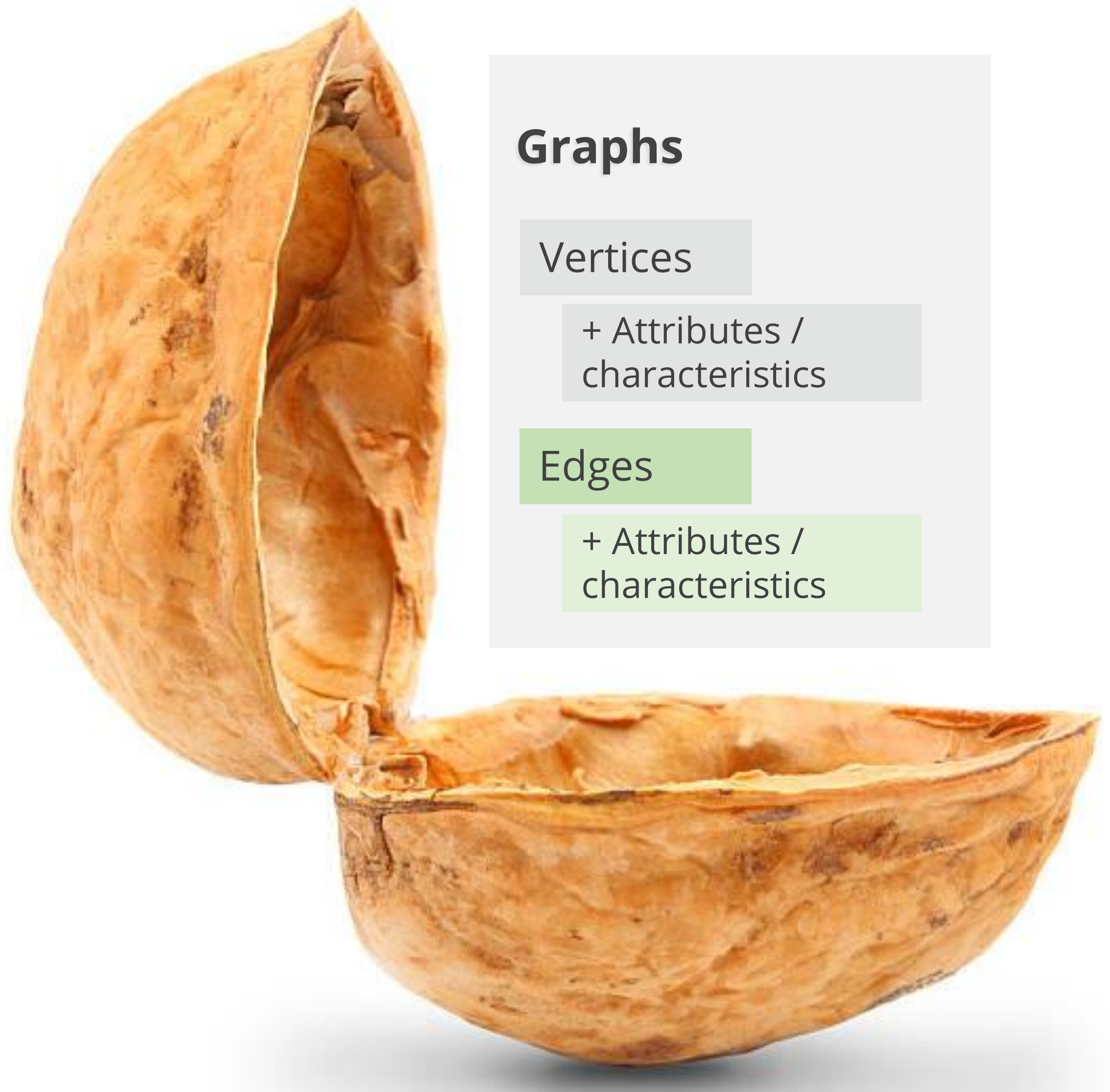
Vertices

+ Attributes / characteristics

Edges



Vertex id	Name	Favorite color	Popularity
1	Sam	Blue	6
2	Sebastian	Green	7
3	Abigail	Purple	9
4	Haley	Pink	2
5	Shane	Orange	4
6	Leah	Purple	7



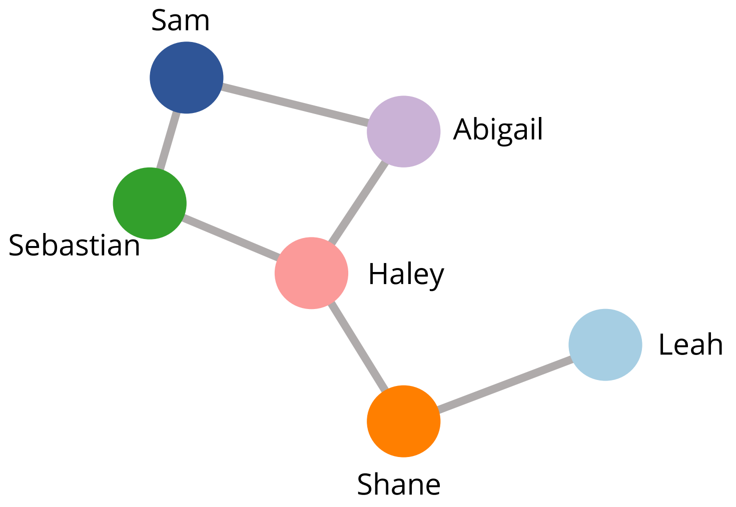
Graphs

Vertices

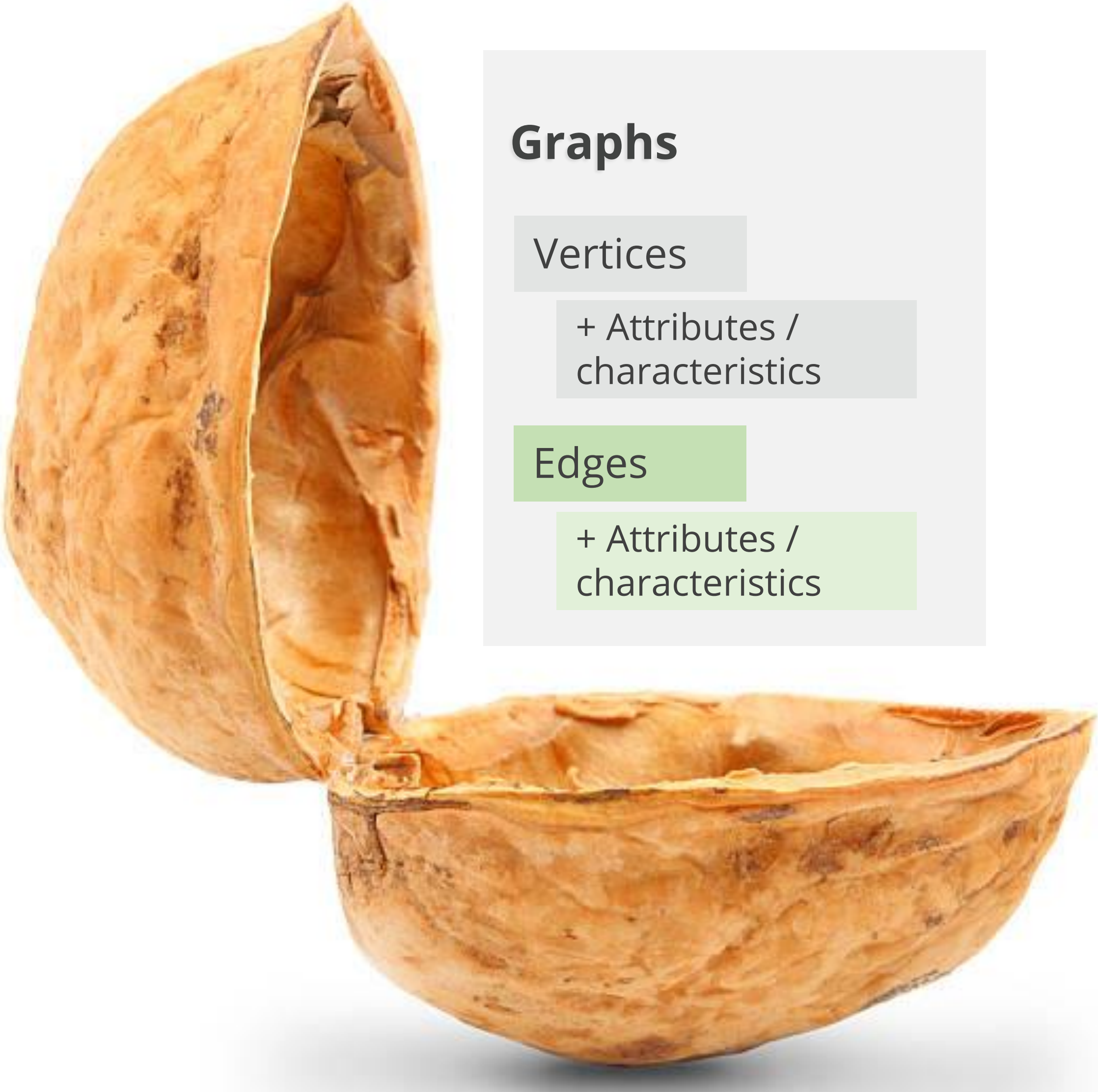
+ Attributes / characteristics

Edges

+ Attributes / characteristics



Edge id	Source	Target	Friend value
1	Sam	Sebastian	10
2	Sam	Abigail	6
3	Sebastian	Haley	1
4	Abigail	Haley	2
5	Haley	Shane	1
6	Shane	Leah	2



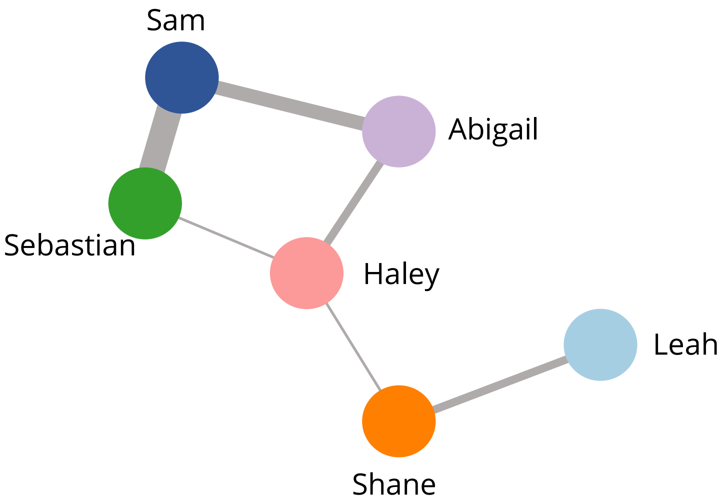
Graphs

Vertices

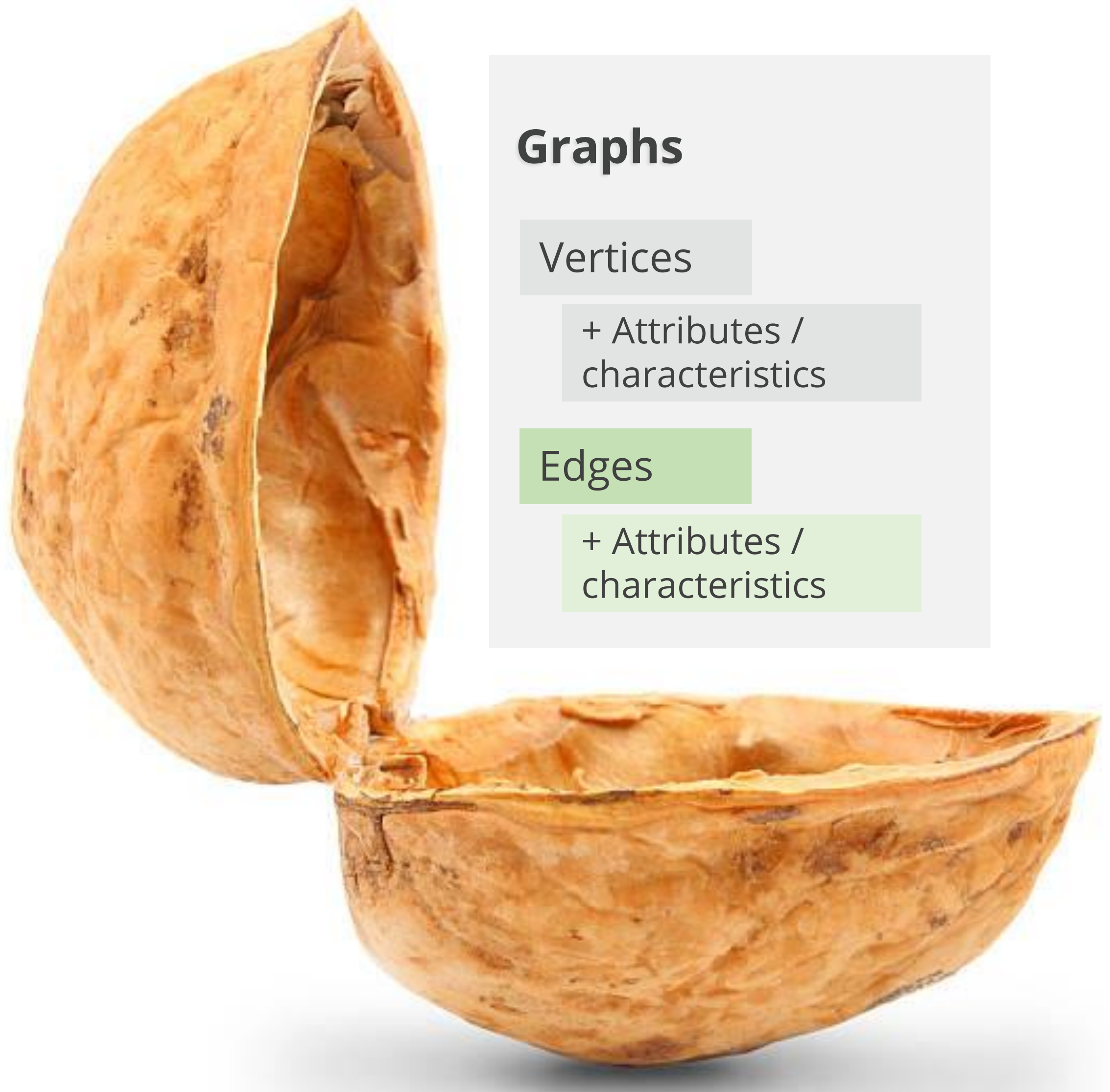
+ Attributes / characteristics

Edges

+ Attributes / characteristics



Edge id	Source	Target	Friend value
1	Sam	Sebastian	10
2	Sam	Abigail	6
3	Sebastian	Haley	1
4	Abigail	Haley	2
5	Haley	Shane	1
6	Shane	Leah	2



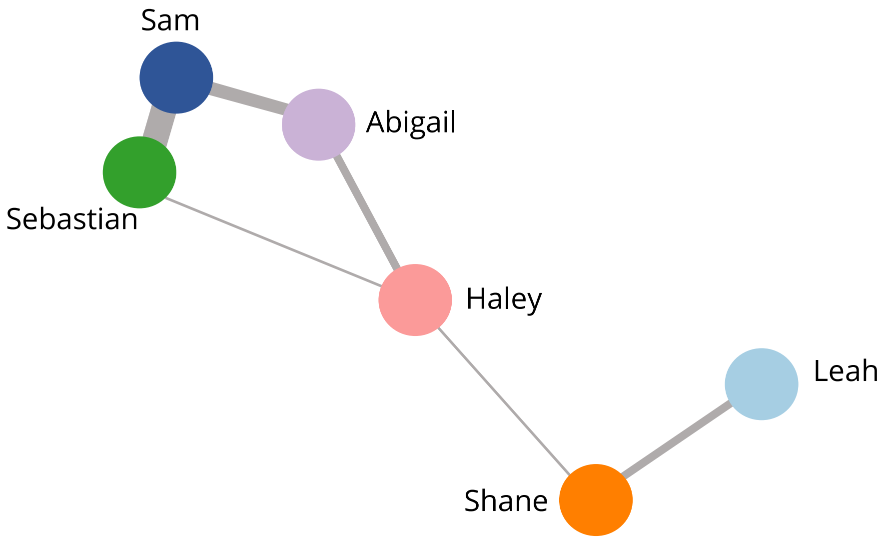
Graphs

Vertices

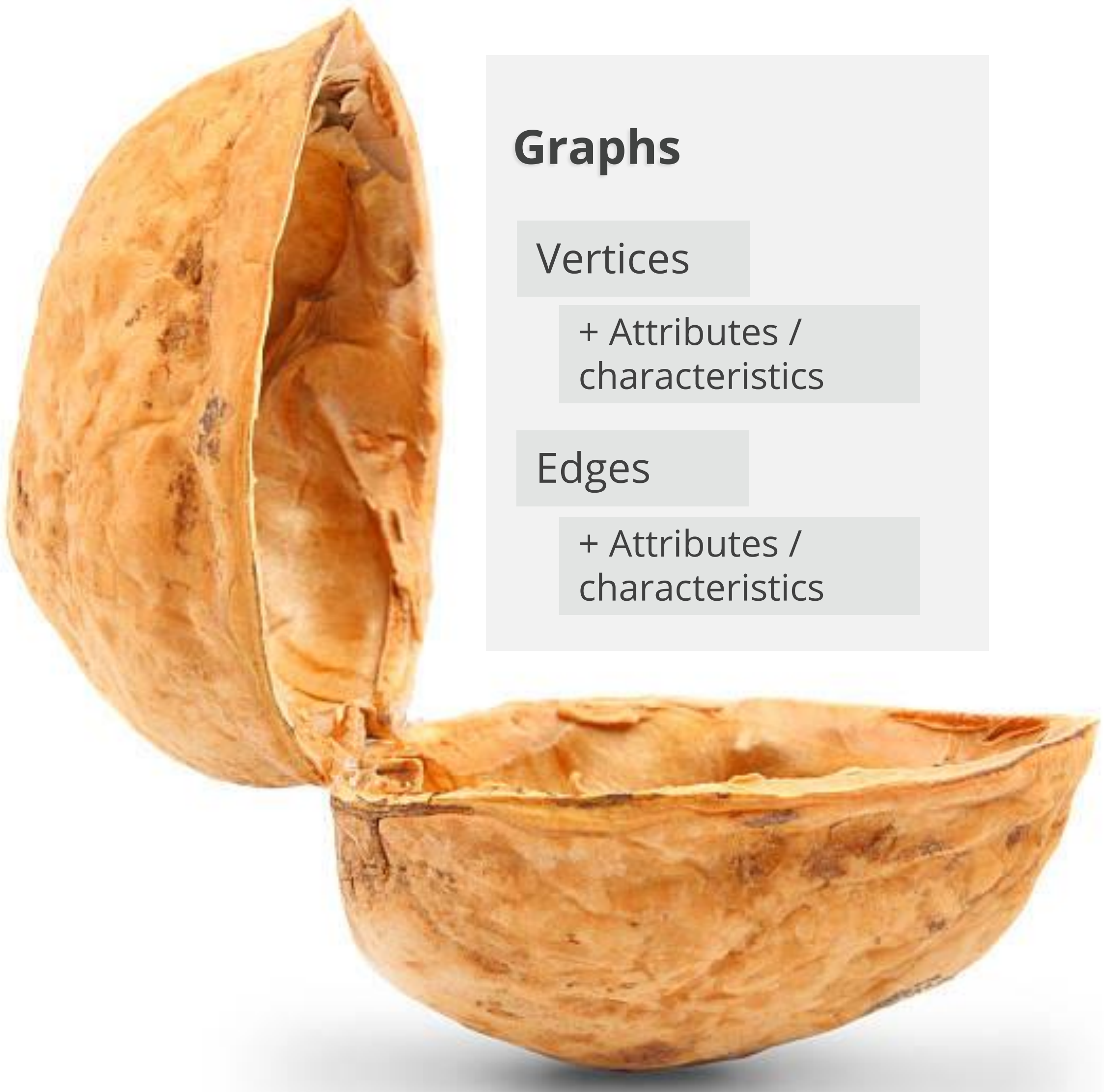
+ Attributes / characteristics

Edges

+ Attributes / characteristics

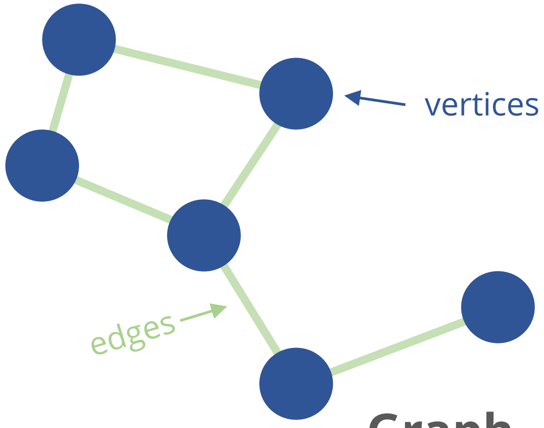


Edge id	Source	Target	Friend value
1	Sam	Sebastian	10
2	Sam	Abigail	6
3	Sebastian	Haley	1
4	Abigail	Haley	2
5	Haley	Shane	1
6	Shane	Leah	2



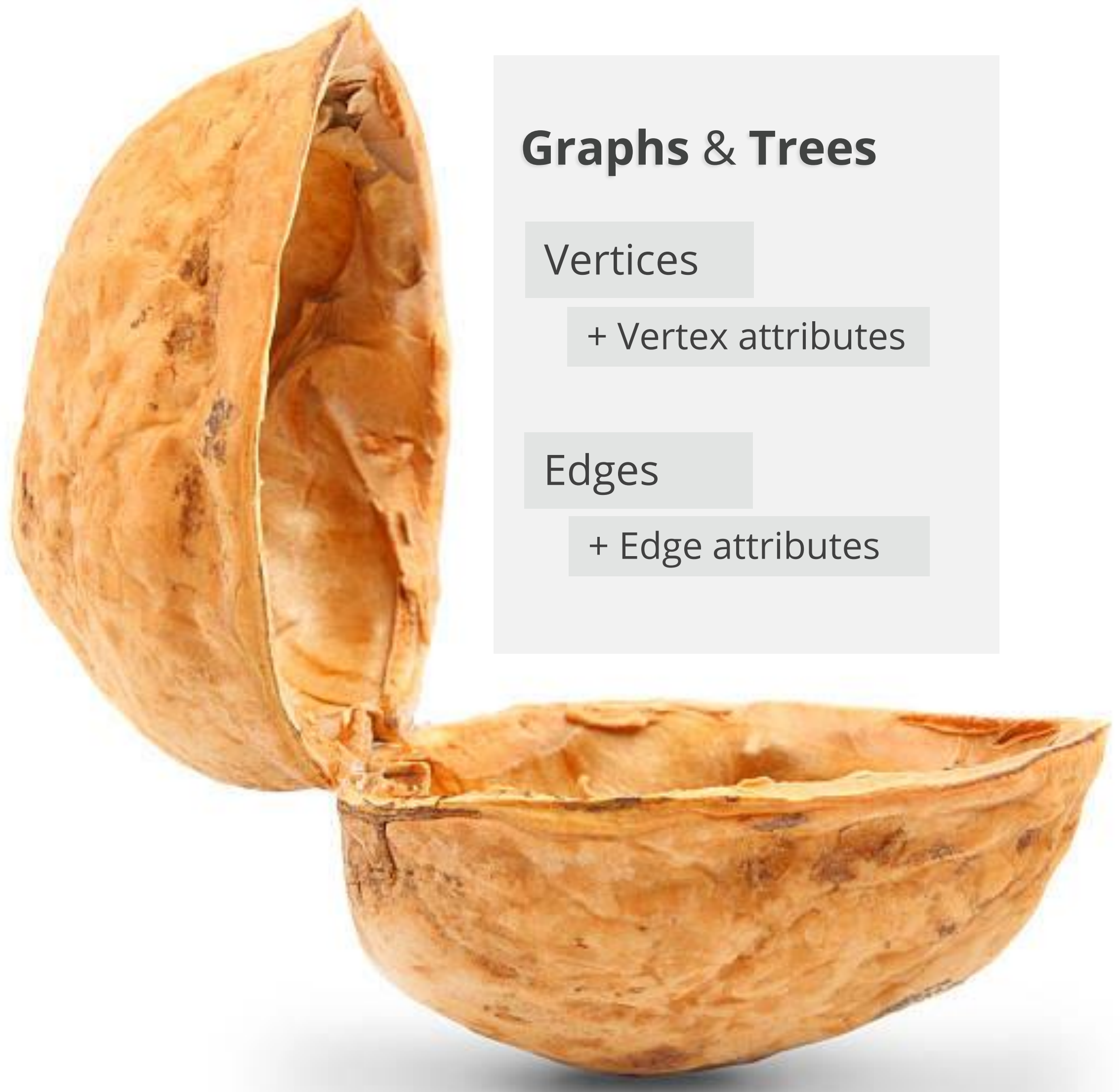
Graphs

- Vertices
  - + Attributes / characteristics
- Edges
  - + Attributes / characteristics



Graph

Questions?



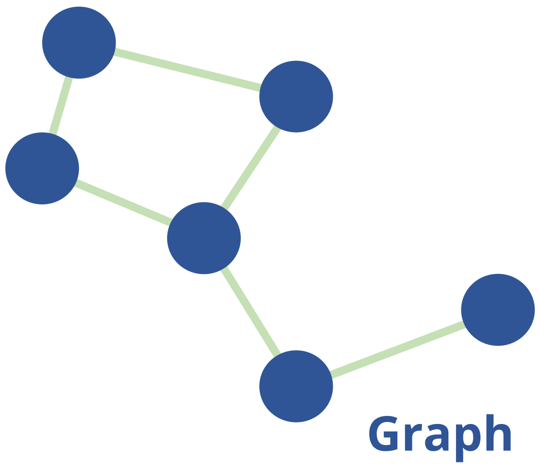
Graphs & **Trees**

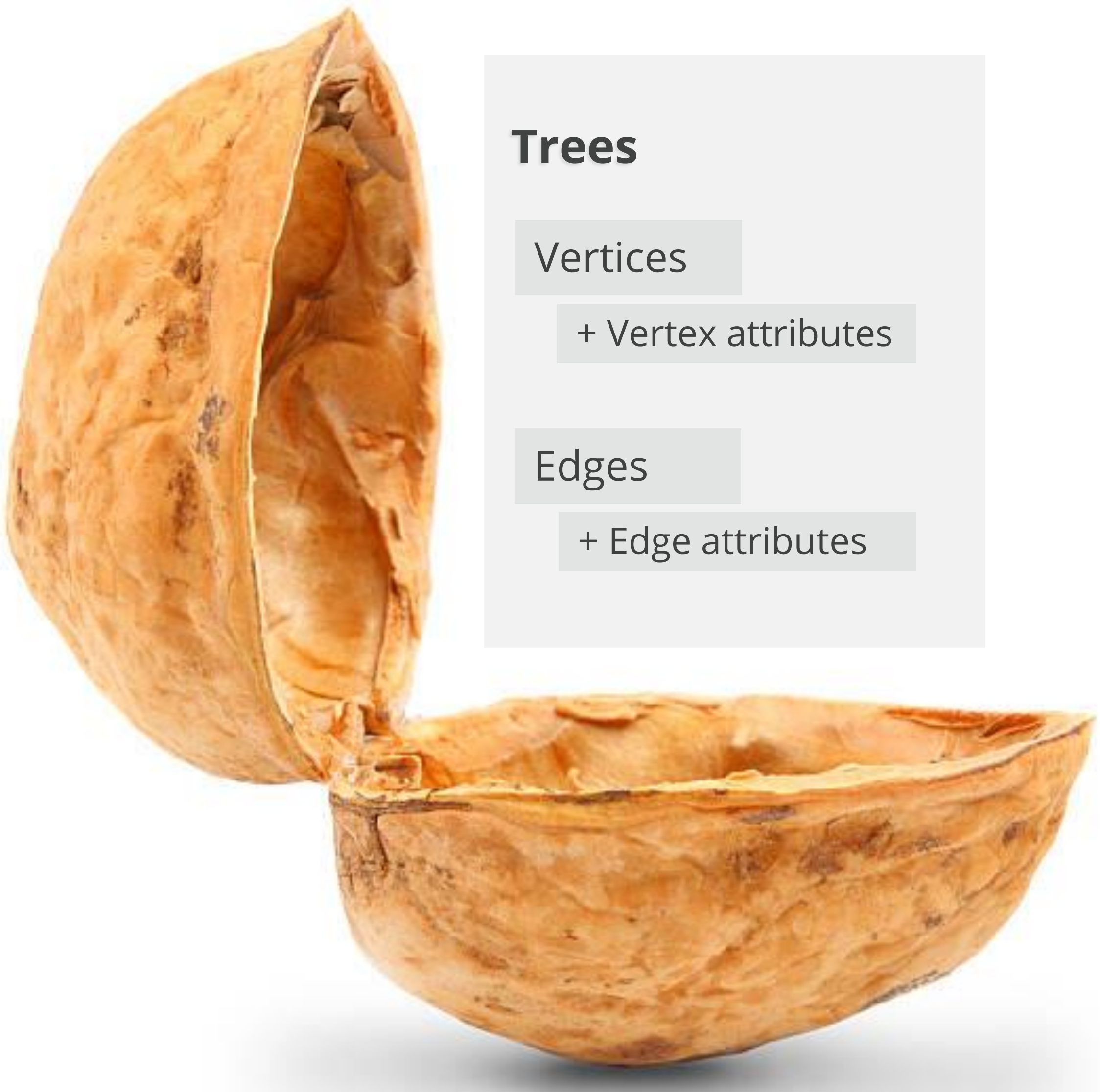
Vertices

+ Vertex attributes

Edges

+ Edge attributes





Trees

Vertices

+ Vertex attributes

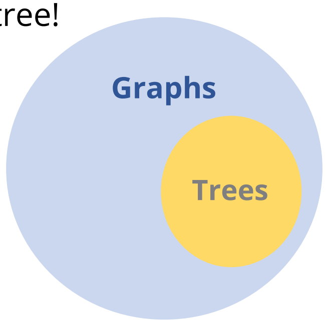
Edges

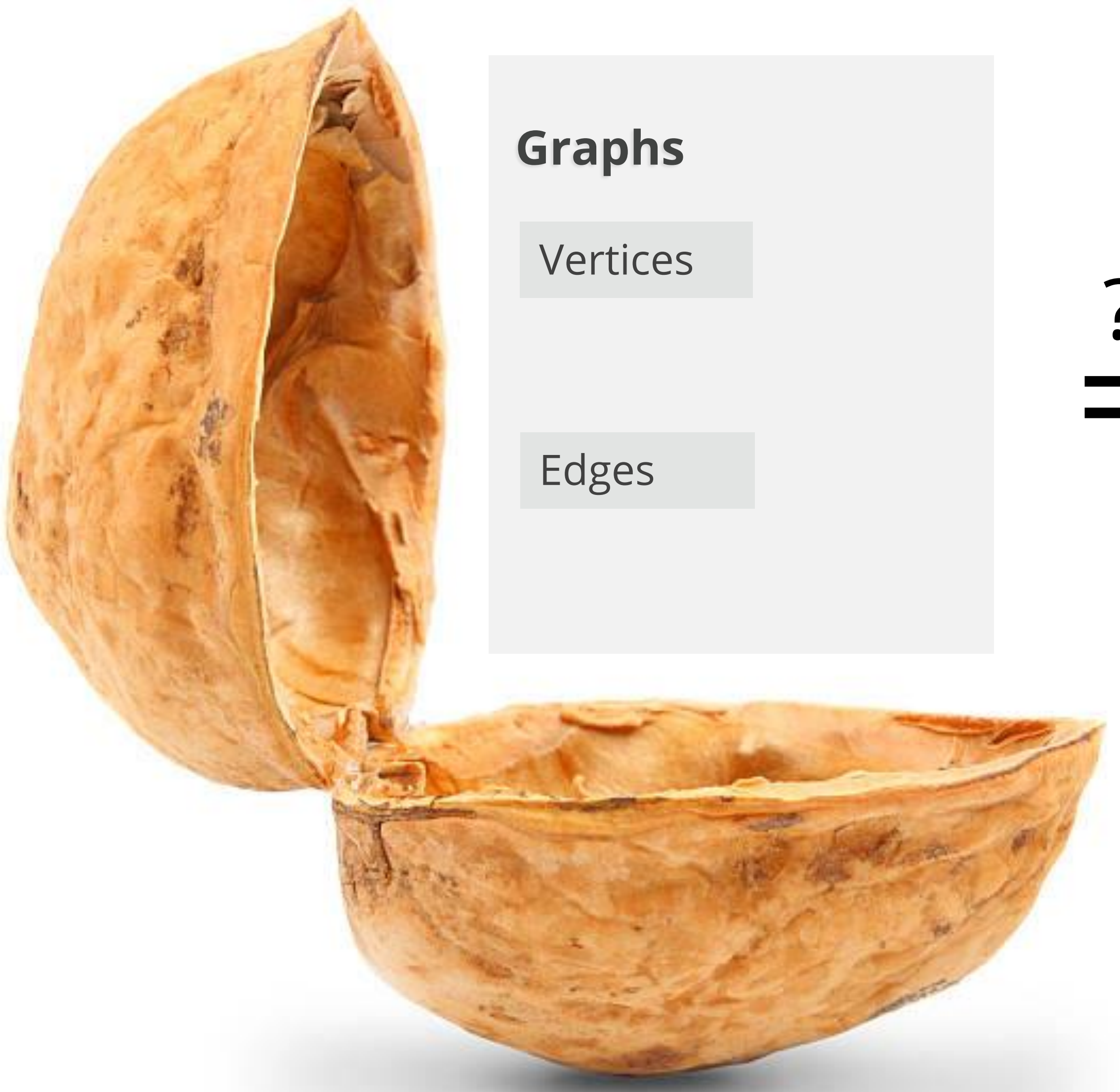
+ Edge attributes

Properties of trees

- Connected
- Hierarchical structure
- One path between any pair of vertices
- No cycles in the graph
- Removing an edge would create a disconnected graph

Every tree is a graph but not every graph is a tree!





Graphs

Vertices

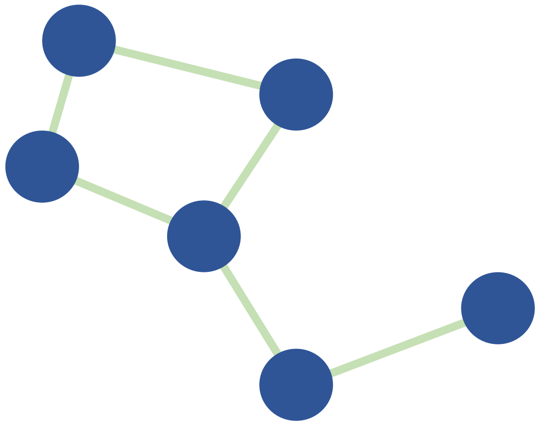
Edges

$\stackrel{?}{=}$

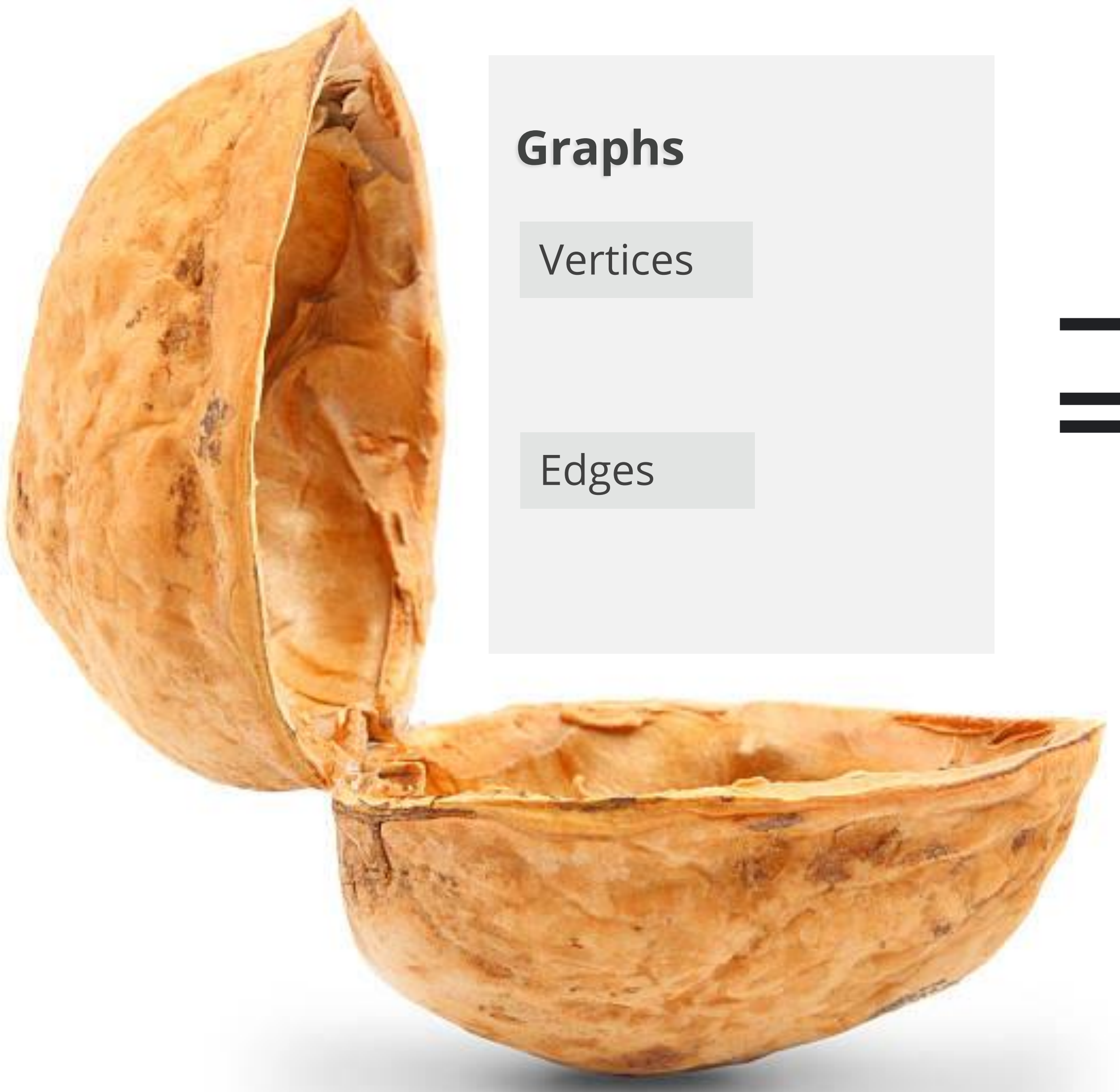
Trees

Vertices

Edges



Is this graph a tree?



Graphs

Vertices

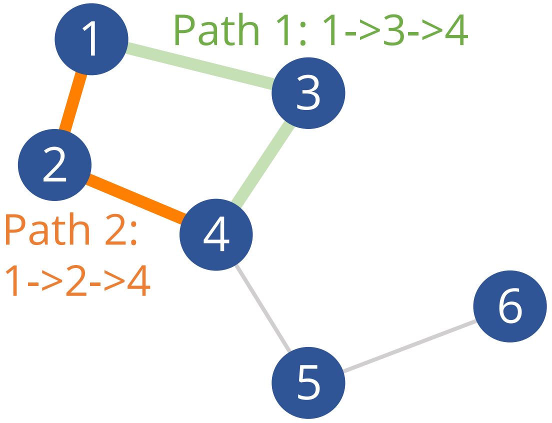
Edges

$\supseteq$

Trees

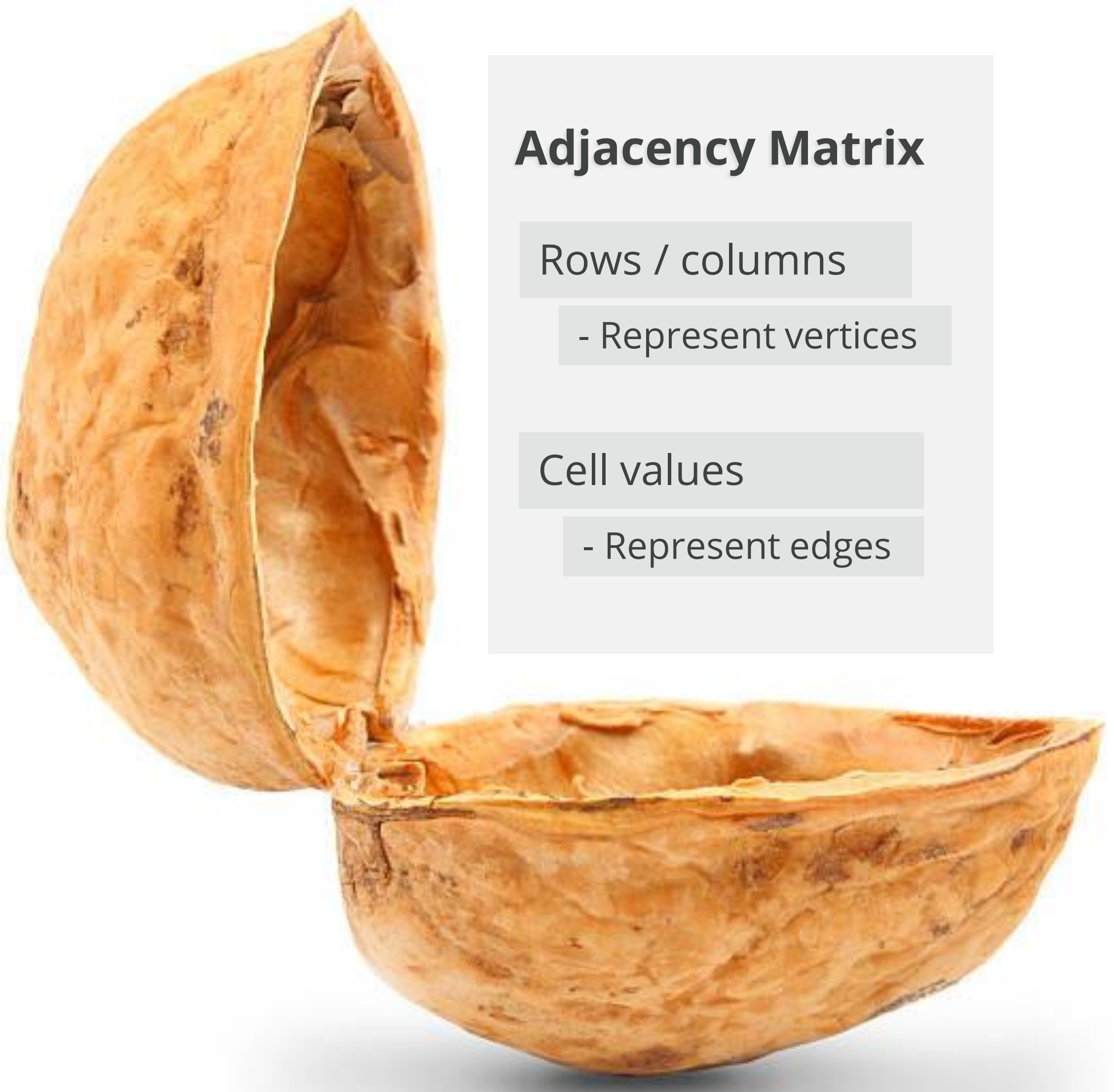
Vertices

Edges



Is this graph a tree?

No



Adjacency Matrix

Rows / columns

- Represent vertices

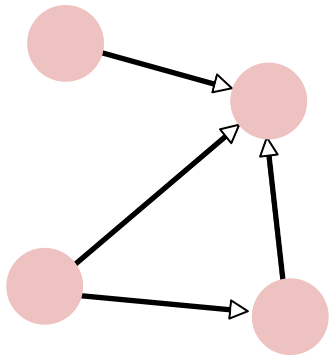
Cell values

- Represent edges

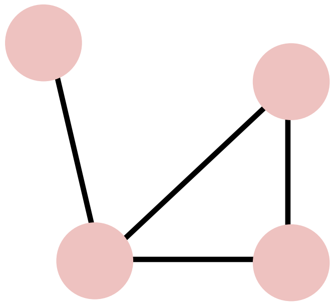
	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1
4	1	0	1	0

Properties of adjacency matrices

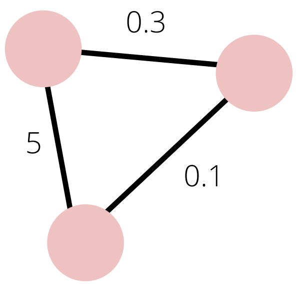
- Another graph representation
- Symmetrical along the diagonal
    - Can read from top or bottom half
    - Typically, all 0's on the diagonal (unless self-loops)
- Non-zero cell value means an edge exists between that pair
    - Zero cell value means no edge exists
    - Cell values can also be edge 'weights' (so not just 0/1)



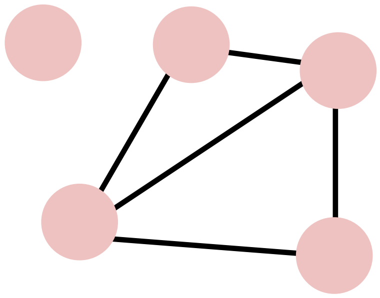
Directed graph



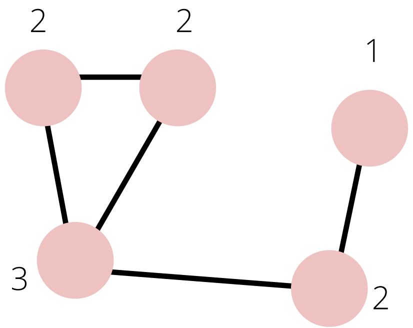
Undirected graph



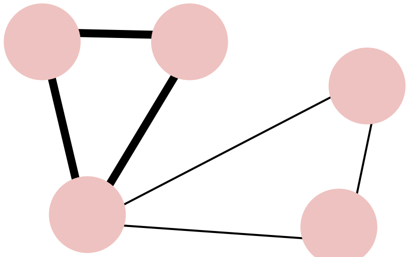
Weighted graph



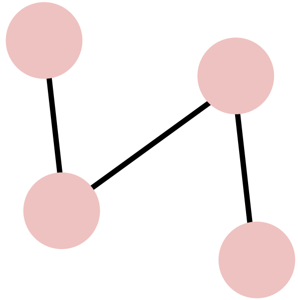
Unconnected graph



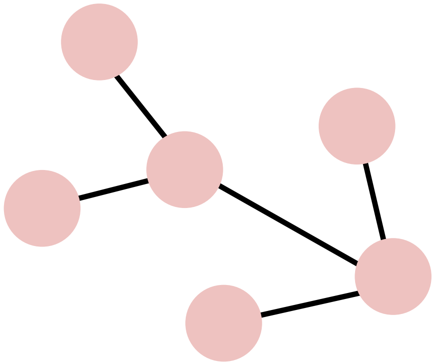
Node degrees



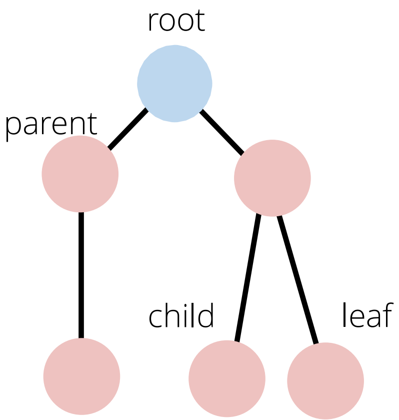
A cycle in a graph



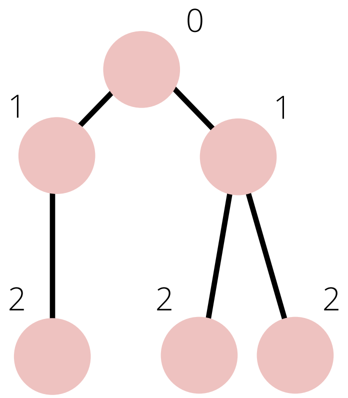
Acyclic graph



Connected acyclic graph, a.k.a. **tree**



Rooted tree or hierarchy



Node depths

Sanity / Attention check!

GRAPHS VS. **NETWORKS**?

NODES VS. **VERTICES**?

LINKS VS. **EDGES**?

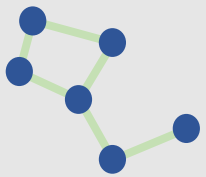
GRAPHS VS. **TREES**?

ADJACENCY MATRICES?

Sanity check

Graphs vs. **Networks**

→ it's just semantics

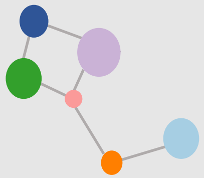


G = {v, e}

N = {n, l}

Nodes vs. **Vertices**

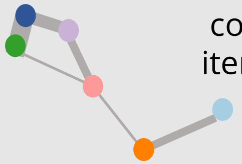
→ it's just semantics



items in G

Links vs. **Edges**

→ it's just semantics



connects items in G

Graphs vs. **Trees**

→ trees are a subset of graphs

graphs with no cycles

Adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

compact graph representation

→ rows/columns are vertices; cell values are edges

Visualizing graphs

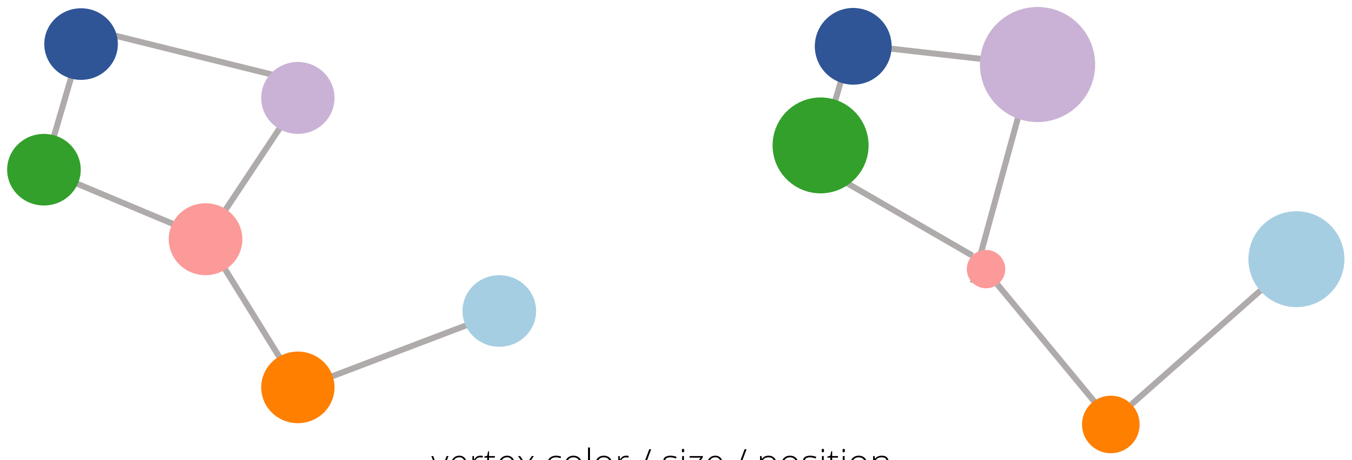
Many ways to visualize, encode, and *lay out* the same graph data

Vertex id	Name	Favorite color	Popularity
1	Sam	Blue	6
2	Sebastian	Green	7
3	Abigail	Purple	9
4	Haley	Pink	2
5	Shane	Orange	4
6	Leah	Purple	7

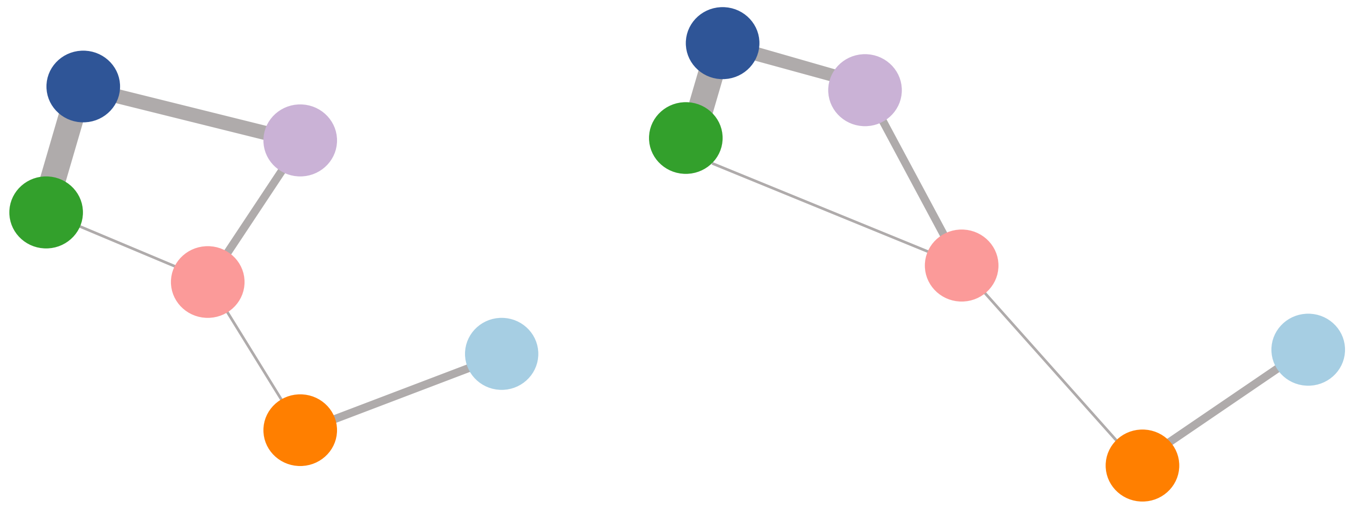
Edge id	Source	Target	Friend value
1	Sam	Sebastian	10
2	Sam	Abigail	6
3	Sebastian	Haley	1
4	Abigail	Haley	2
5	Haley	Shane	1
6	Shane	Leah	2

Question: what if our graph has no innate attributes?

Node-link diagrams

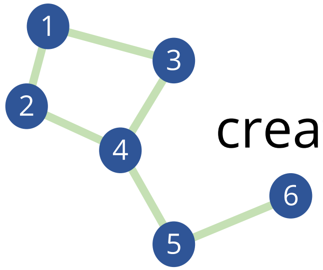


vertex color / size / position



edge opaqueness / weight (thickness) / distance (length)

Graph drawing exercise



Graph
(represented as a node-link diagram)

create an *aesthetically-pleasing* **node-link diagram** from this simple adjacency matrix

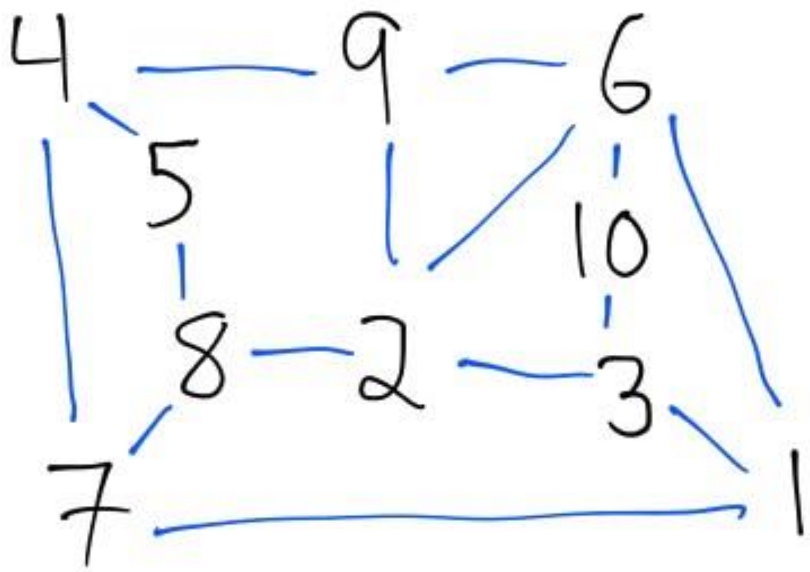
	1	2	3	4	5	6	7	8	9	10
1	0	0	**1**	0	0	**1**	**1**	0	0	0
2	0	0	**1**	0	0	**1**	0	**1**	**1**	0
3	**1**	**1**	0	0	0	0	0	0	0	**1**
4	0	0	0	0	**1**	0	**1**	0	**1**	0
5	0	0	0	1	0	0	0	**1**	0	0
6	**1**	**1**	0	0	0	0	0	0	**1**	**1**
7	**1**	0	0	**1**	0	0	0	**1**	0	0
8	0	**1**	0	0	**1**	0	**1**	0	0	0
9	0	**1**	0	1	0	**1**	0	0	0	0
10	0	0	**1**	0	0	**1**	0	0	0	0

"Nice-looking" ~ *aesthetically-pleasing*: what does that mean to you?

Graph drawing exercise

create an *aesthetically-pleasing* **node-link diagram** from this simple adjacency matrix

	1	2	3	4	5	6	7	8	9	10
1	0	0	**1**	0	0	**1**	**1**	0	0	0
2	0	0	**1**	0	0	**1**	0	**1**	**1**	0
3	**1**	**1**	0	0	0	0	0	0	0	**1**
4	0	0	0	0	**1**	0	**1**	0	**1**	0
5	0	0	0	1	0	0	0	**1**	0	0
6	**1**	**1**	0	0	0	0	0	0	**1**	**1**
7	**1**	0	0	**1**	0	0	0	**1**	0	0
8	0	**1**	0	0	**1**	0	**1**	0	0	0
9	0	**1**	0	1	0	**1**	0	0	0	0
10	0	0	**1**	0	0	**1**	0	0	0	0



Bonus: is this a tree or a graph?

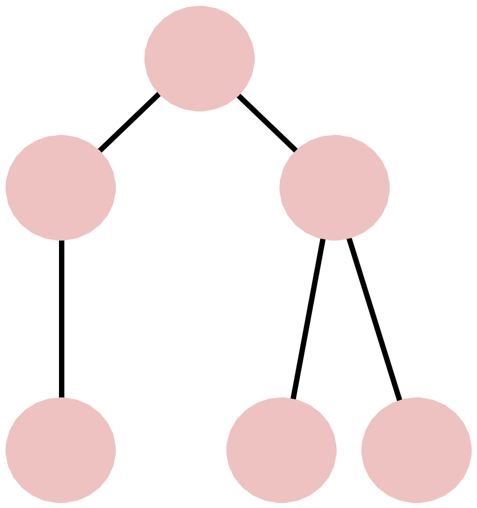
A Tour through the **Tree Visualization** Zoo

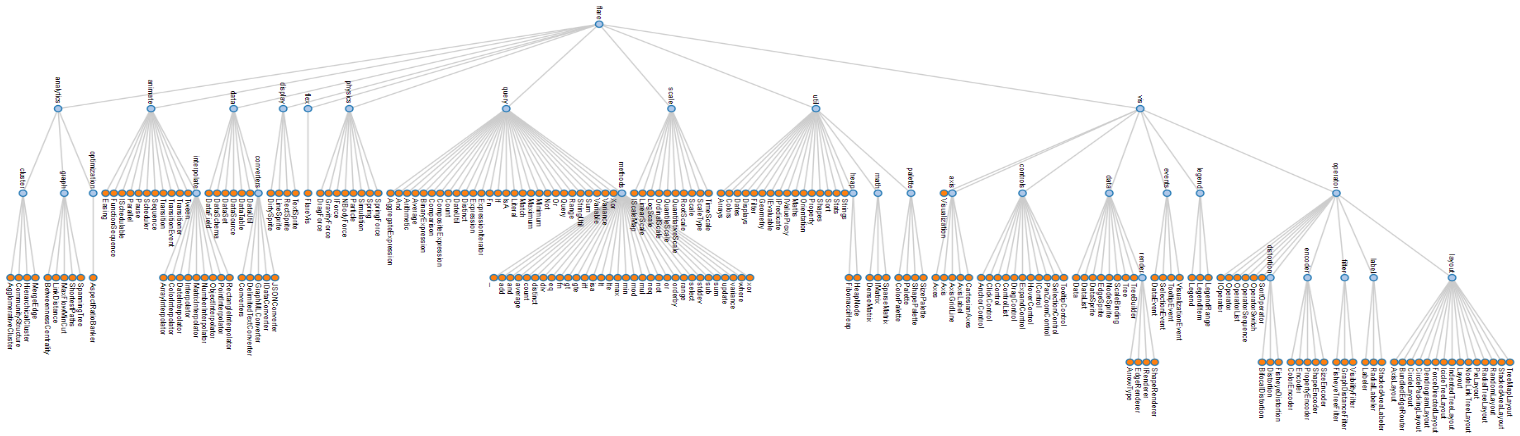
+ node-link diagram
+ layered
+ indentation
+ enclosure

https://homes.cs.washington.edu/~jheer//files/zoo/

Node-link tree diagrams

- Nodes are distributed in space, connected by straight or curved lines
- Typical approach is to use 2D space to break apart breadth and depth
- Often, space is used to communicate hierarchical orientation



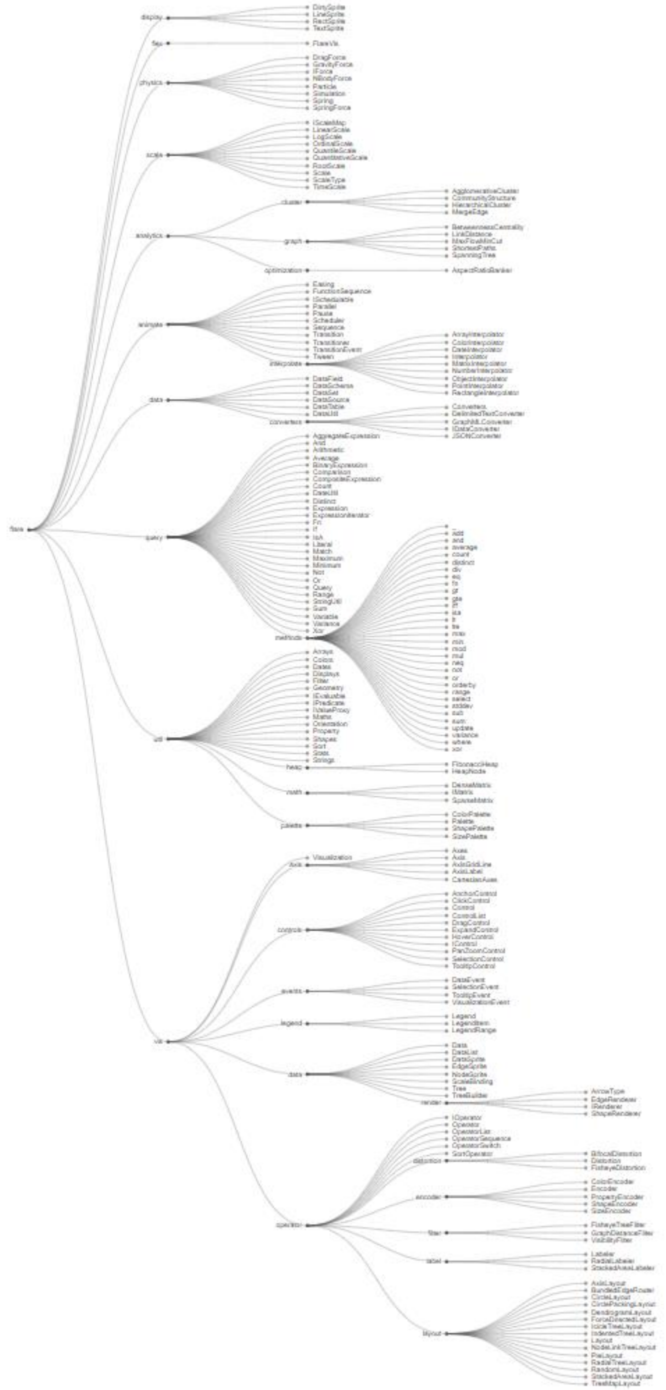


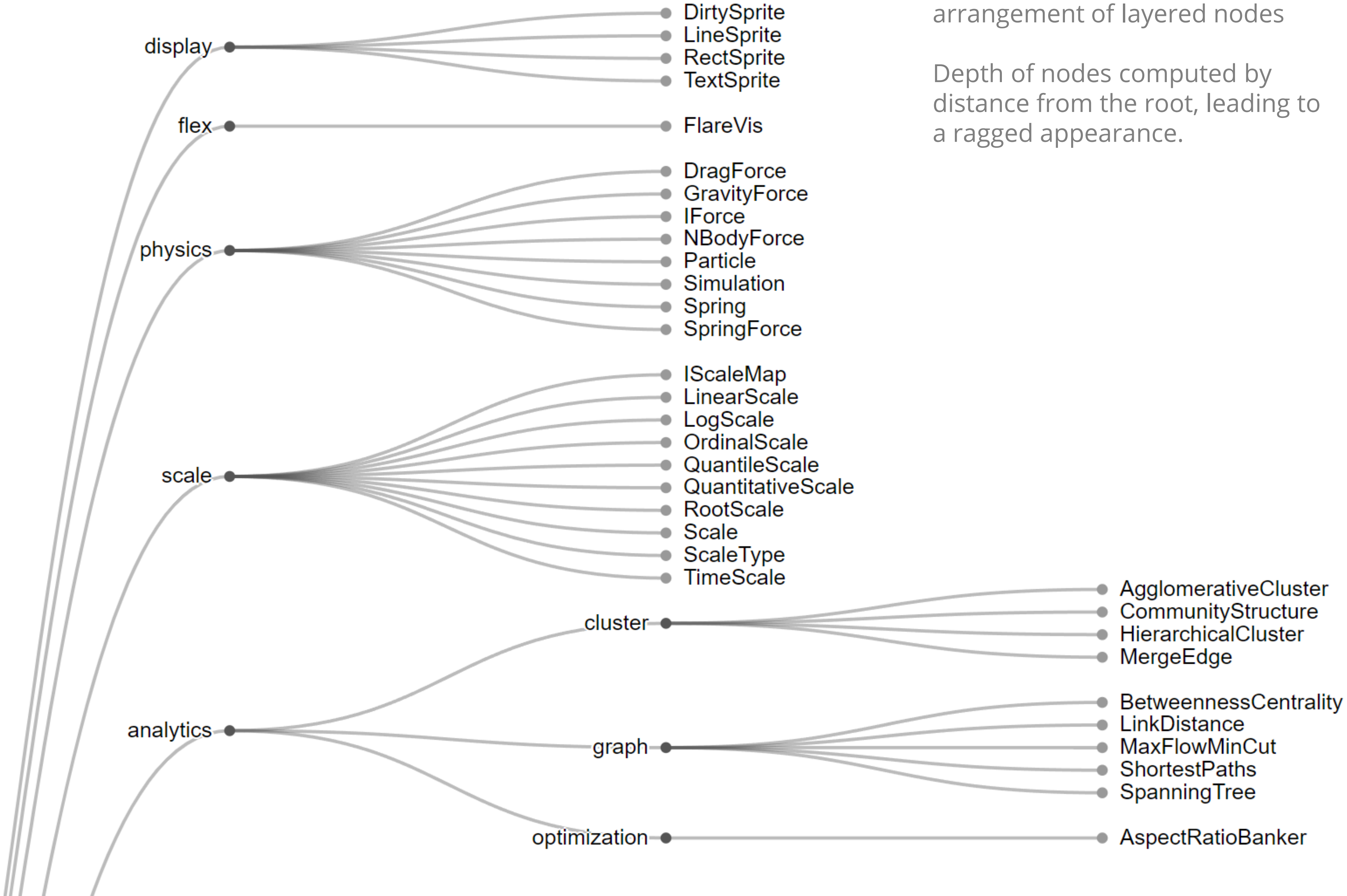
Tidy Tree

https://bl.ocks.org/mbostock/4339184

Implements the Reingold-Tilford algorithm for efficient, tidy arrangement of layered nodes

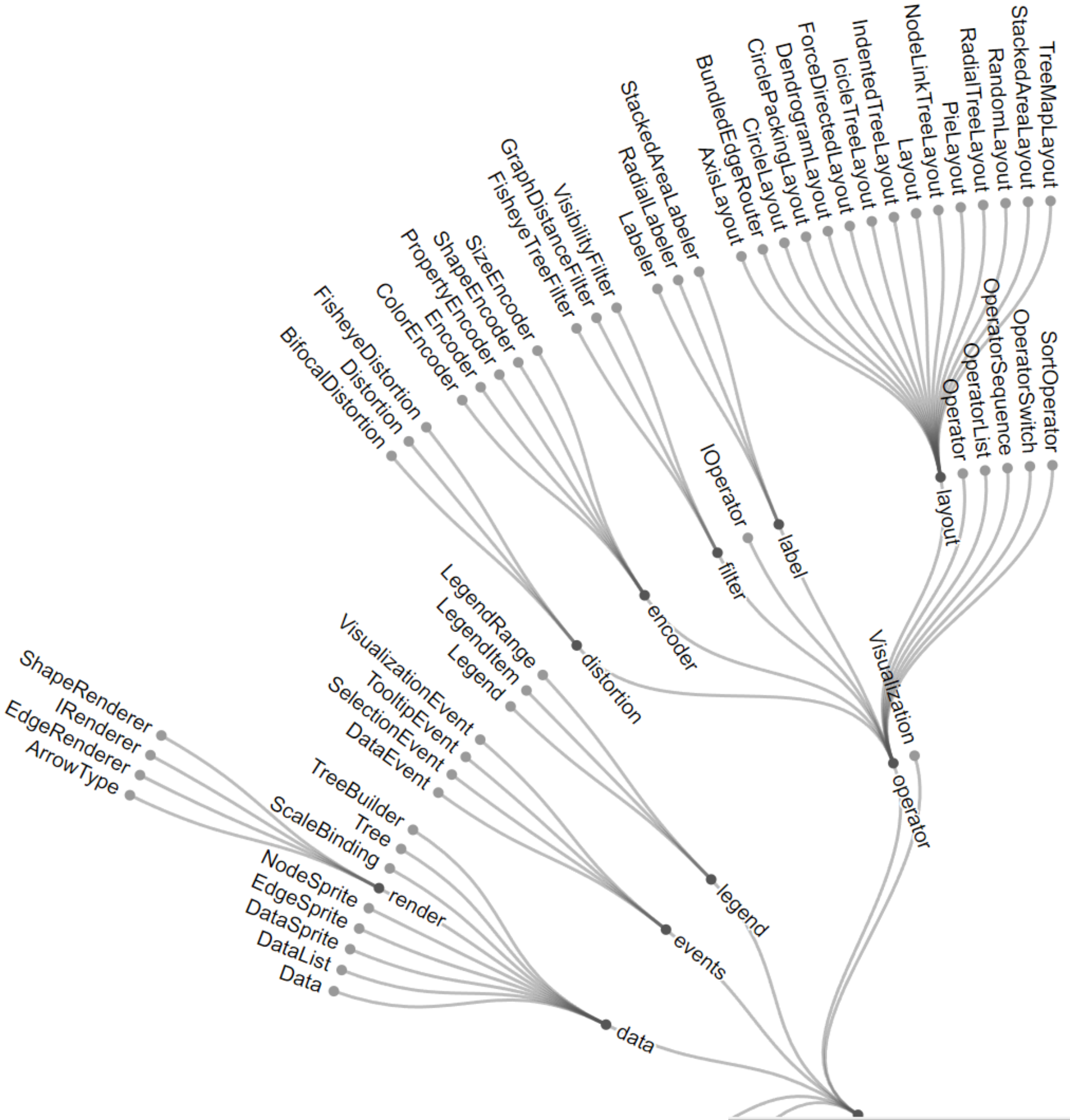
Depth of nodes computed by distance from the root, leading to a ragged appearance.





Radial Tidy Tree

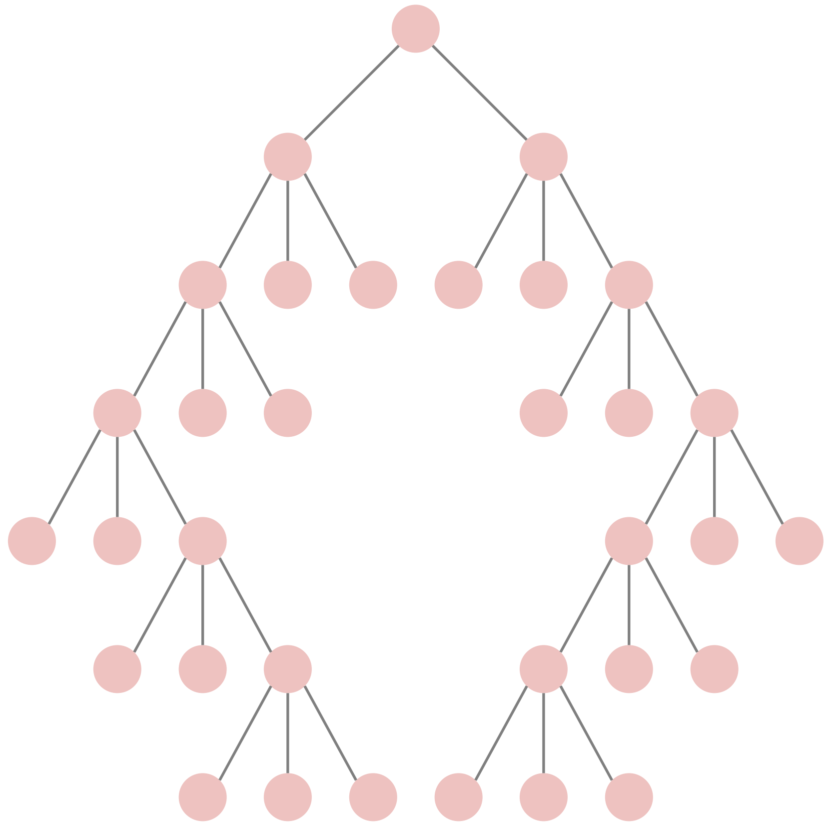
https://bl.ocks.org/mbostock/4063550



Reingold-Tilford algorithm for drawing node-link diagrams

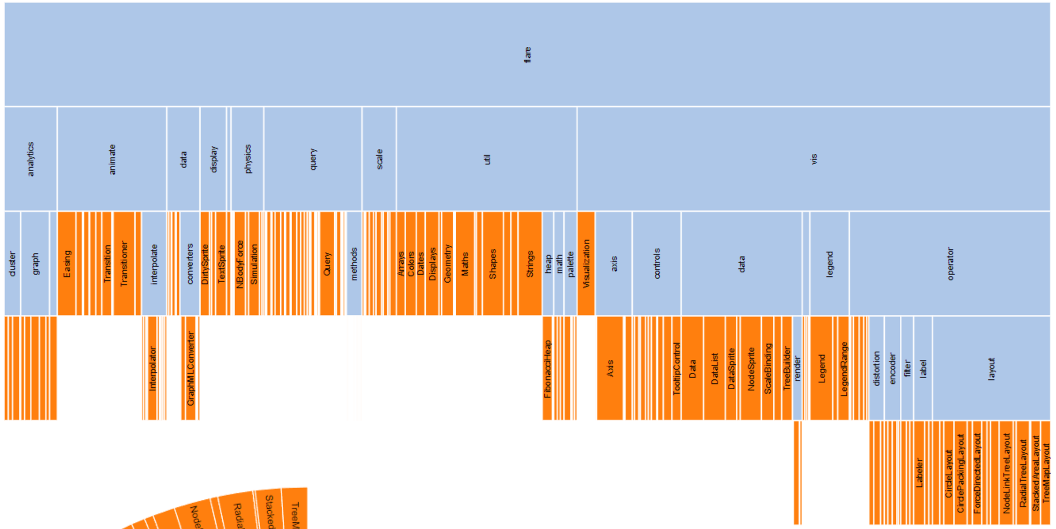
- Bottom-up recursive approach
  - Repeatedly divide space by leaf count
- For each parent, make sure subtrees are drawn
- Make smarter use of space

  + Maximize density and symmetry
  + Clearly encode depth level
  + No edge crossings
  + Pack subtrees as closely as possible
  + Centers parent over subtrees

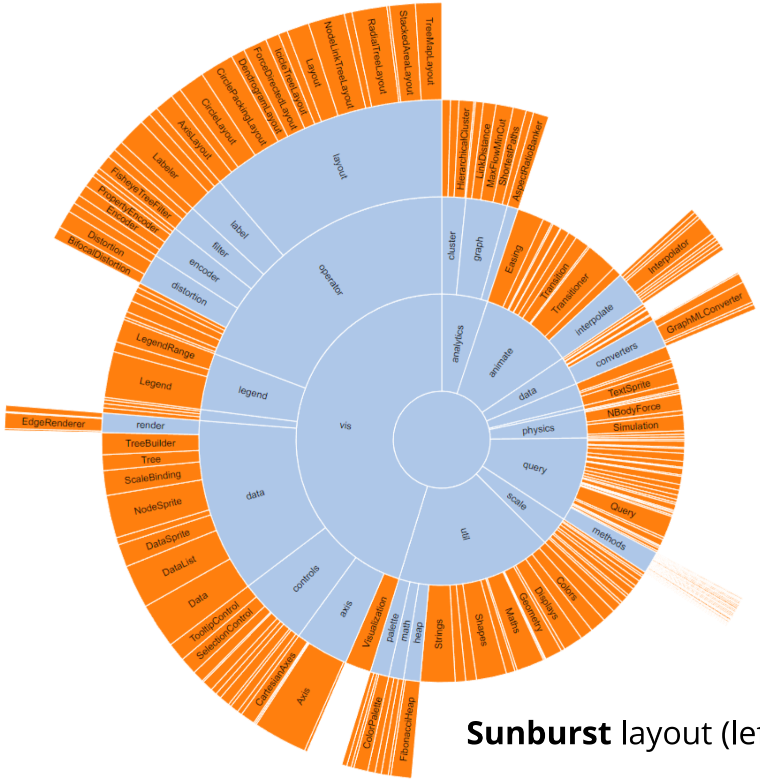


Layered (adjacency) diagrams

- Space-filling variant of node-link diagrams
- Nodes drawn as solid areas (arcs or bars)
- Placement relative to adjacent nodes reveals place in hierarchy
  - Root node at top / center
  - Leaf nodes at bottom



Icicle layout (above)

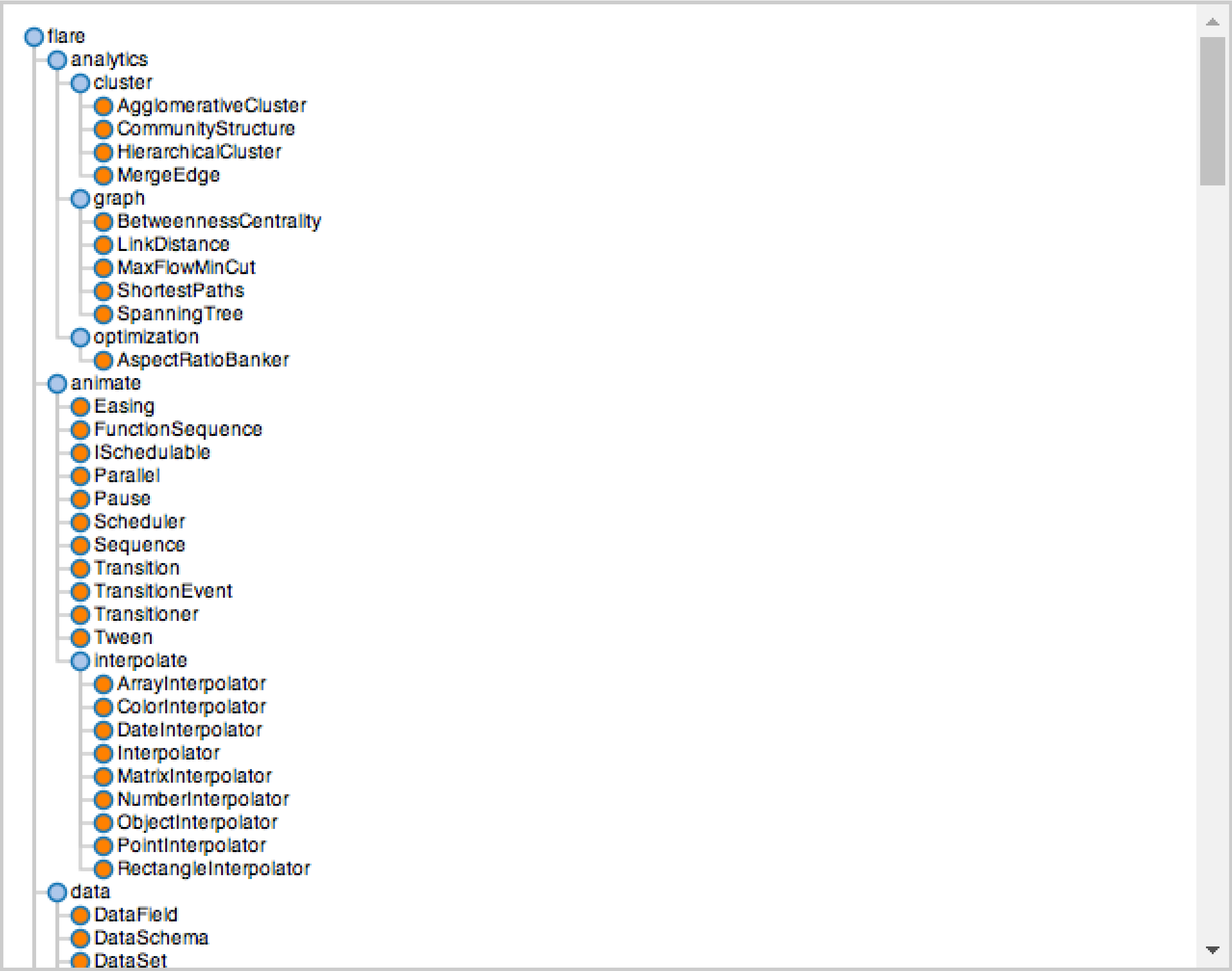


Sunburst layout (left)

Potential problem: can we run out of (screen) space?

Indentation

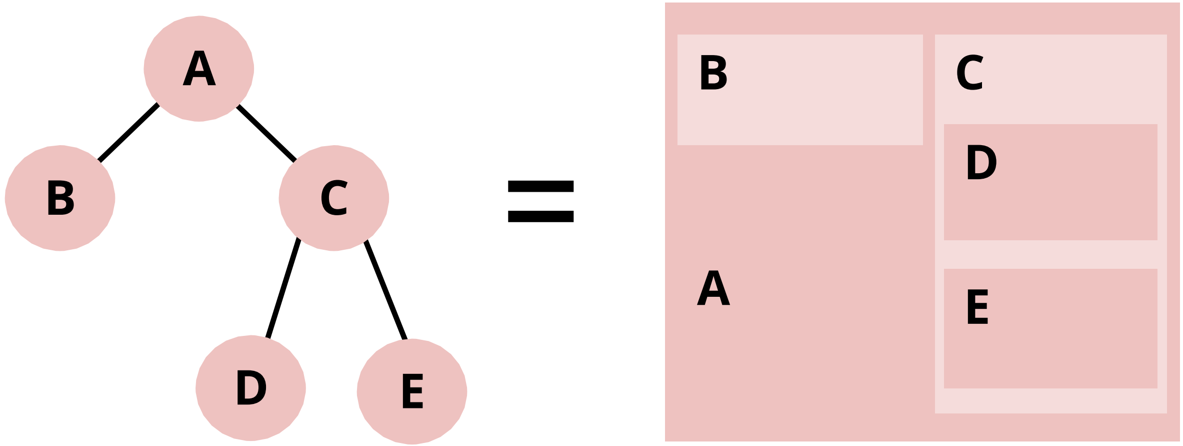
- Used to show parent / child relationships
- Potentially a lot of scrolling!



Question: where does this indented tree representation appear often?

Enclosure (treemap) diagrams

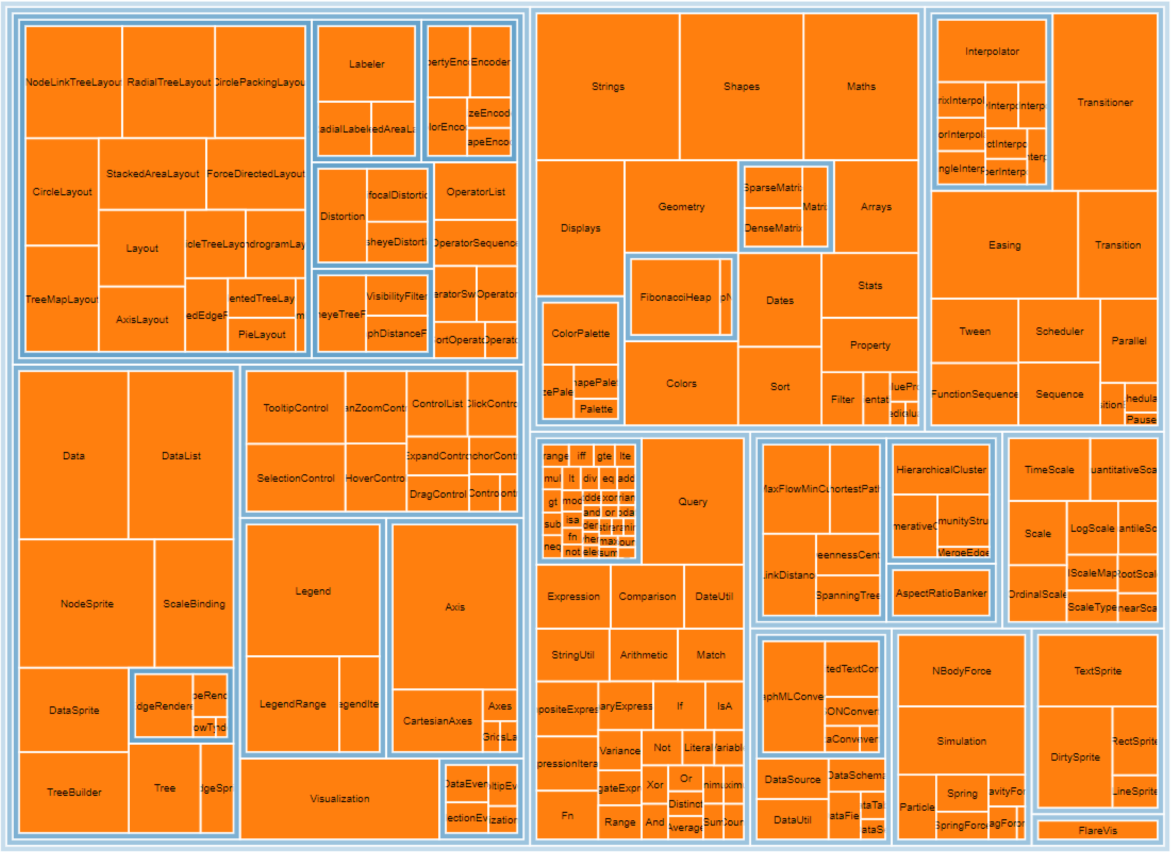
- Encodes tree structure using *spatial enclosure*
  - Enclosure indicates hierarchy
- Benefits:
  - Provides single view of entire tree
  - Easier to spot small / large nodes



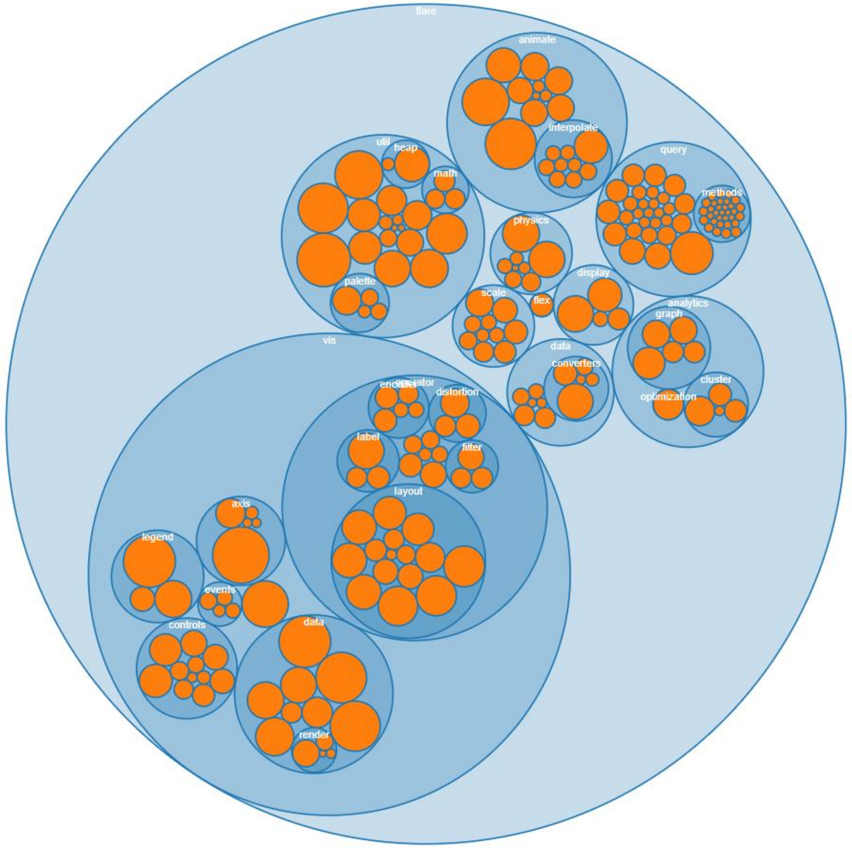
The *treemap* was introduced by Ben Shneiderman in 1991.

It uses containment, rather than adjacency, to represent the hierarchy.

Enclosure diagrams



Enclosure diagram (treemap)



Circle-packing layout

Potential problem: is it easy for you to visually discern the *depth* of the tree?

‘What Do You Think Is the Most Important Problem Facing This Country Today?’

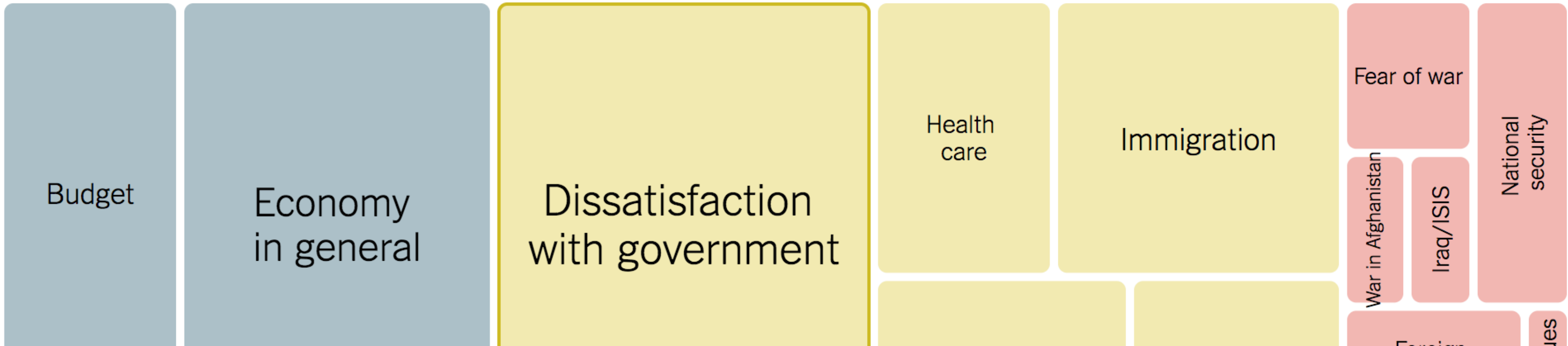
By GREGOR AISCH and ALICIA PARLAPIANO FEB. 27, 2017

Since the presidency of Franklin D. Roosevelt, the Gallup polling organization has asked Americans an open-ended question: “What do you think is the most important problem facing this country today?”

As Donald J. Trump prepares for his first major address to the nation on Tuesday, he has a unique set of issues to tackle. But there is not one singular issue that is dominating the American consciousness.

January 2015

The biggest problems cited by Americans this month:



Effective Graph Drawing
(return at 12:40p EST)

How do we deal with dense graphs?

How do we draw graphs in an aesthetically-pleasing way?

How do we deal with drawing big graphs?

+ node-link layouts
  + Reingold-Tilford (discussed previously)

+ force-directed layouts

+ adjacency diagrams

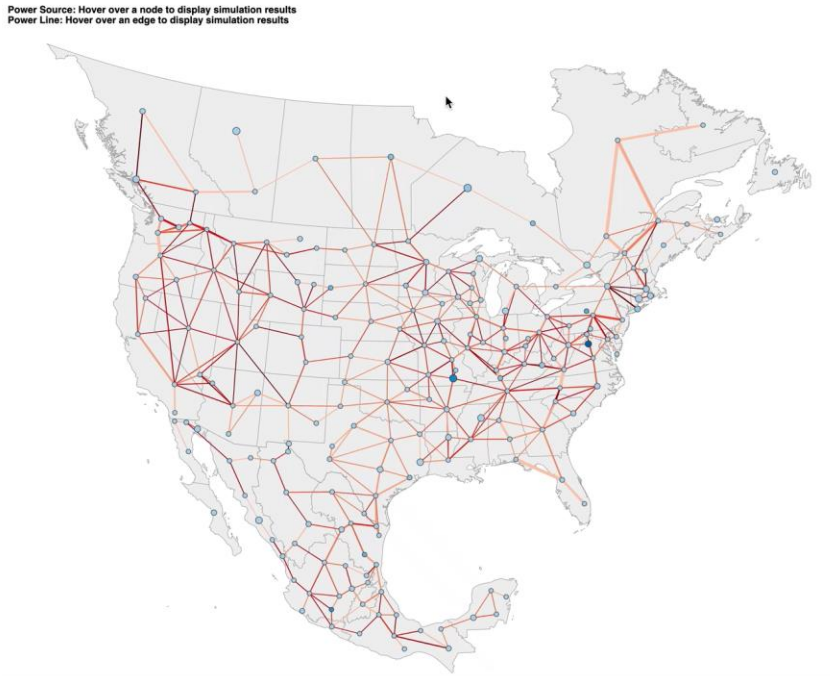
+ aggregate views
  + Motif glyphs
  + PivotGraph

How do we draw graphs *effectively?*

Primary concern: the **spatial layout** of vertices and edges

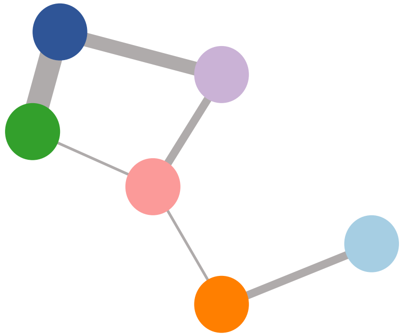
Often (but not always) the goal is to effectively depict the **graph structure**

- Connectivity, path-following
- Network distance
- Clustering
- Ordering (e.g., hierarchy level)



Visualizing the Reliability and Security of the North American Power Grid System in 2050
Work done for the National Renewable Energy Laboratory
code can be found @ https://github.com/ashleysuh/nerc-visualization

draw these graphs using https://gephi.org/

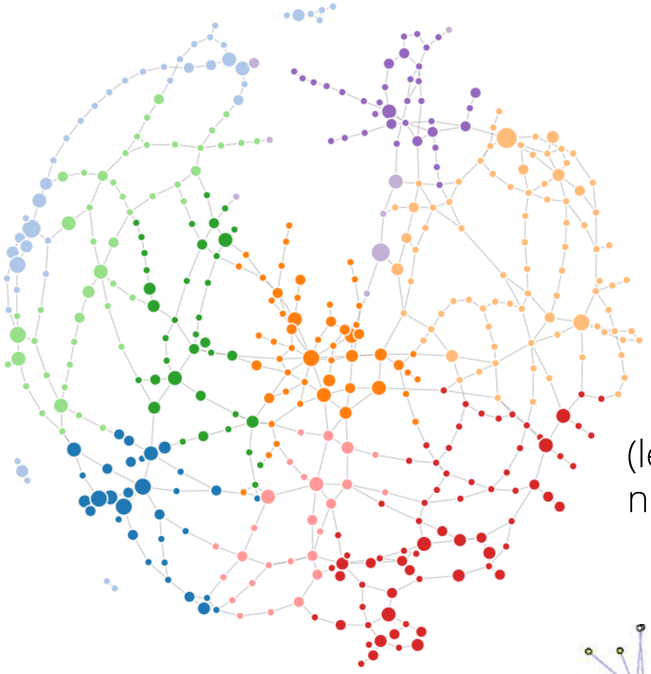


Node-link diagrams (again)

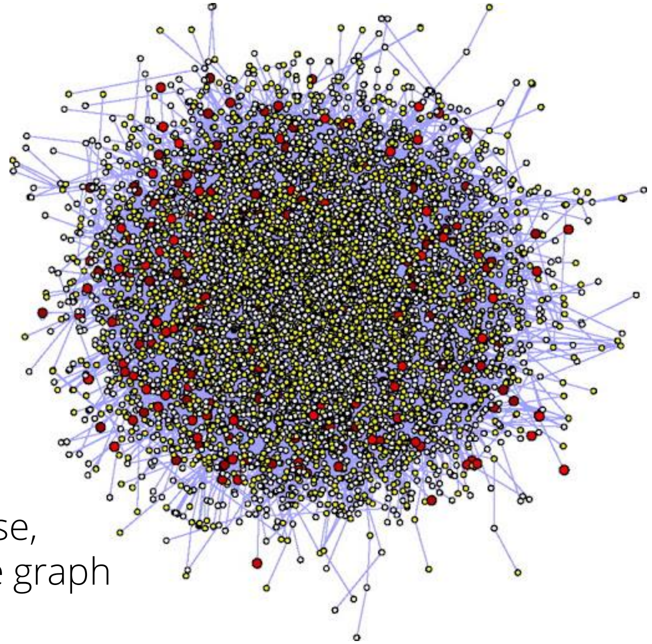
Reingold-Tilford algorithm

PROS:
- understandable visual mapping
- shows overall structure, clusters, paths
- flexible, many variations / layouts

CONS:
- most trivial algorithms are > $O(n^2)$
- not good for dense (very connected) graphs

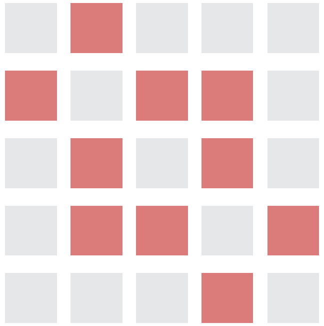


(left) a nice, decluttered node-link diagram



(right) a dense, "hairball"-like graph

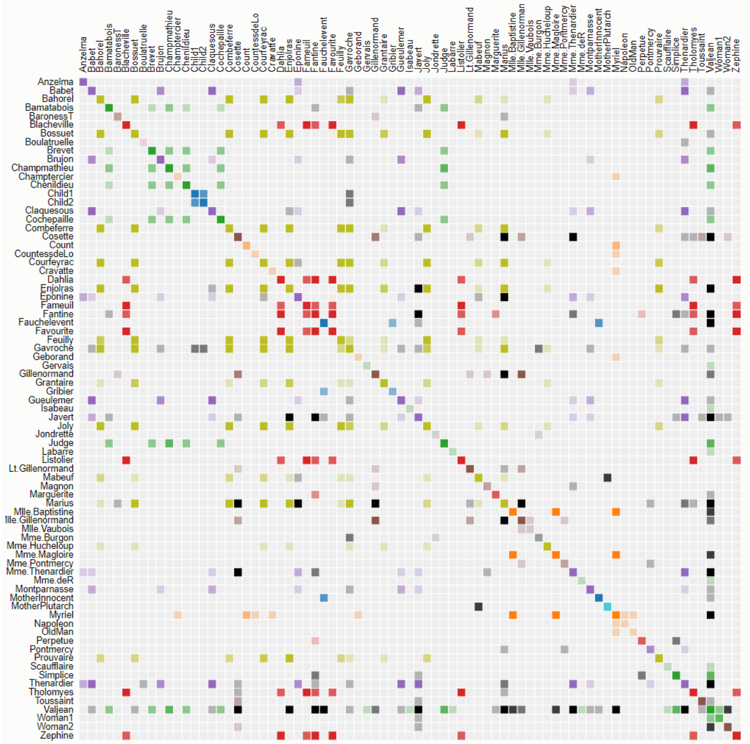
try this interactively using https://bost.ocks.org/mike/miserables/



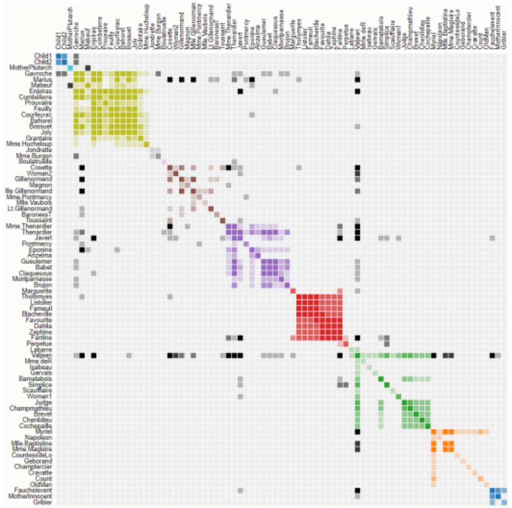
Adjacency Diagram

PROS:
- great for dense graphs
- visually scalable
- can spot clusters

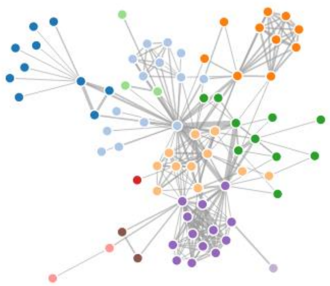
CONS:
- row order affects what you can see
- abstract visualization
- hard to follow paths



Why does ordering matter?
(above) layout ordered by Name
(left, top) layout ordered by Frequency
(left, bot) layout ordered by Cluster



try it interactively at https://observablehq.com/@d3/force-directed-graph



Force-directed graph drawing

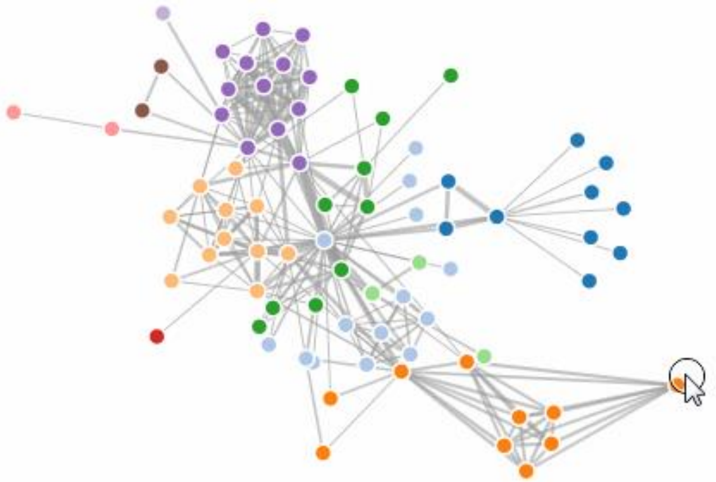
Physical-based model (attractive & repulsive forces)

PROS:

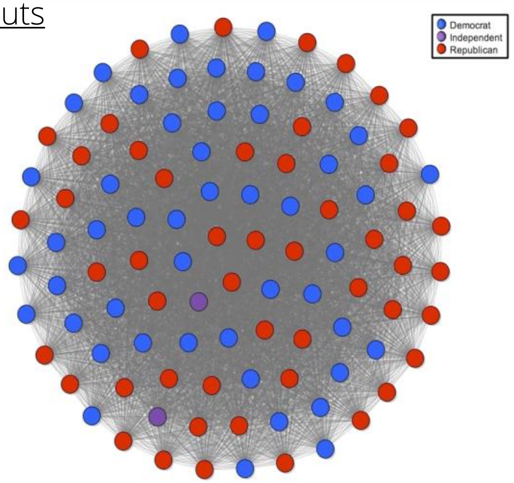
- aesthetically-pleasing layout
- interactive (pull & drag!)
- automatic & flexible layout

CONS:

- forces are computationally expensive ~ $O(n^2)$
- doesn't work well on dense graphs



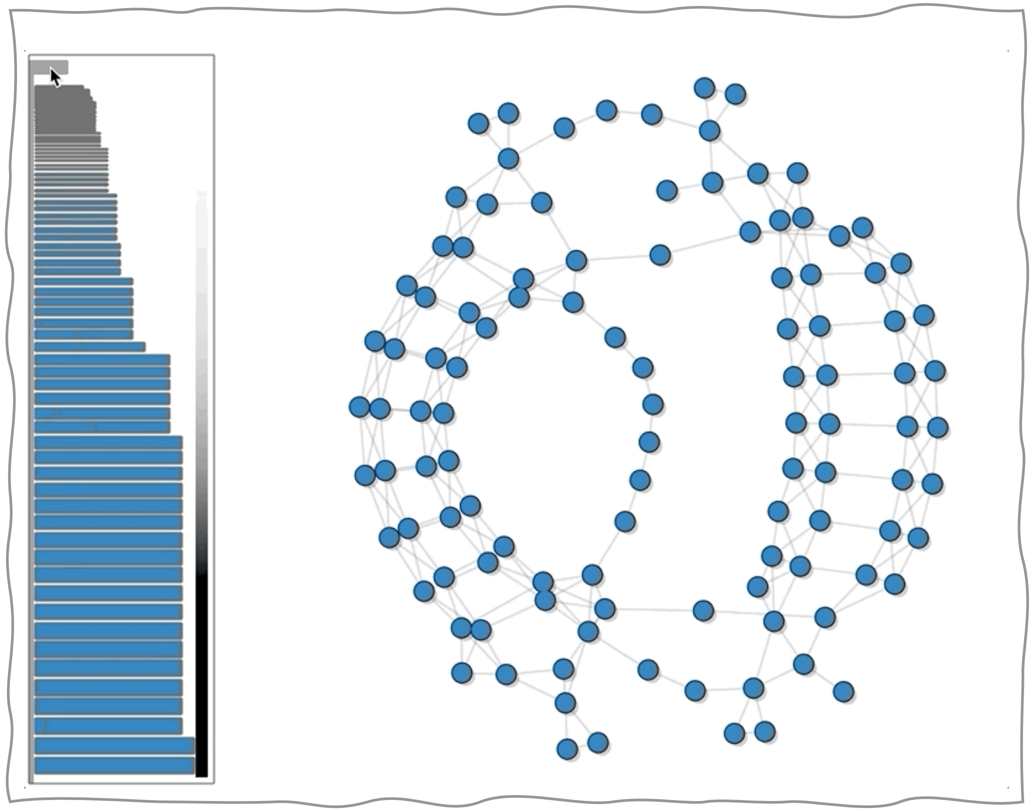
Interactive force-directed layouts
(above) Les Mis dataset
(below) Voting network



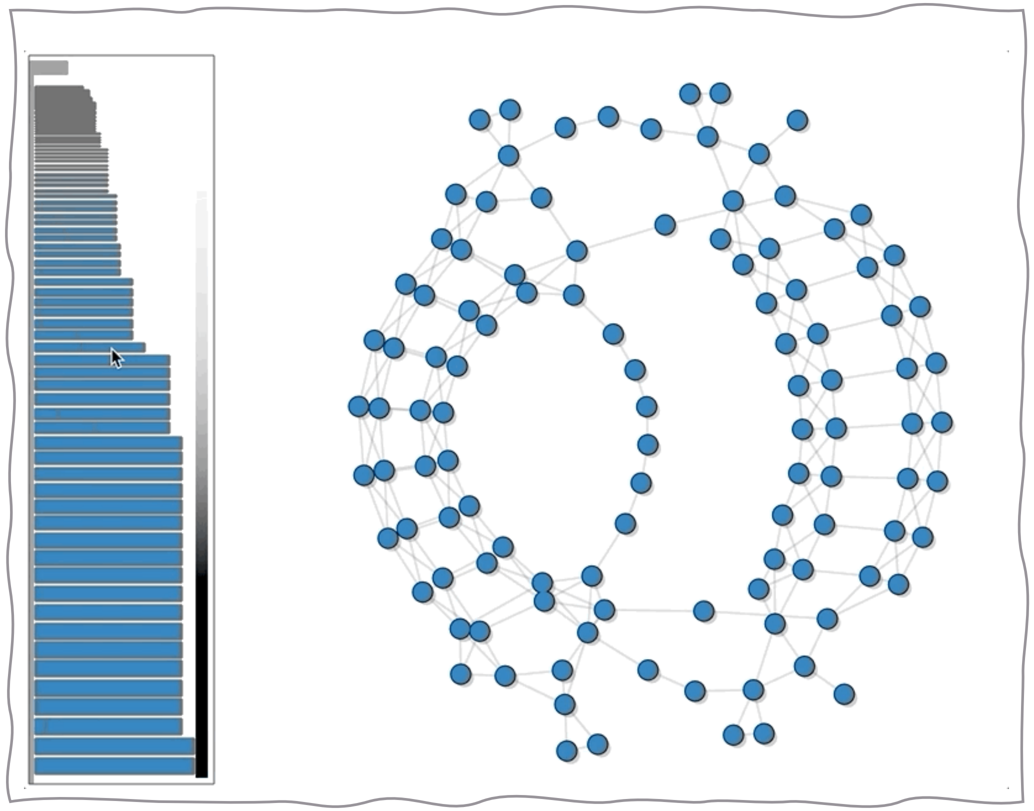
Potential problem: how can we interact with a force-directed graph if it's highly connected?

Better* interactive force-directed layouts

* I'm biased because this is my own research project ☺



Clustering (attracting) nodes

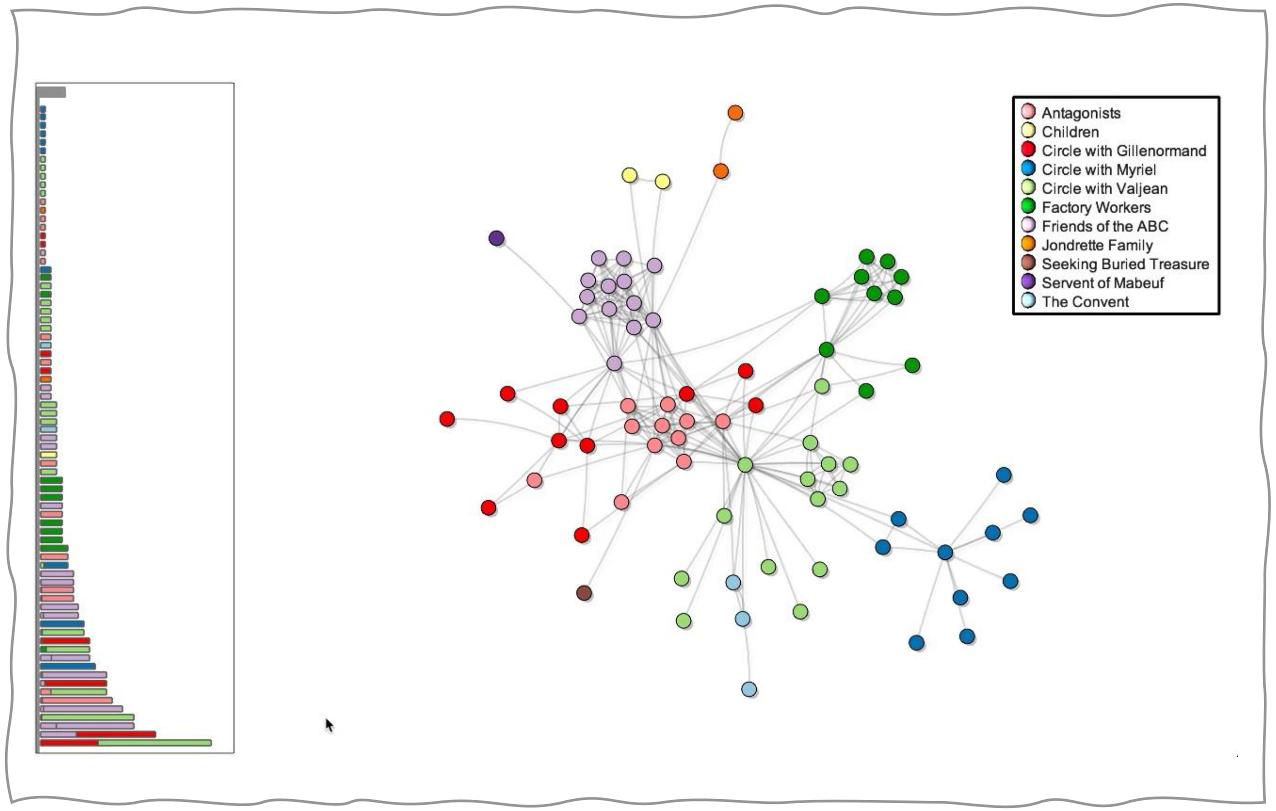


Detangling (repulsing) nodes

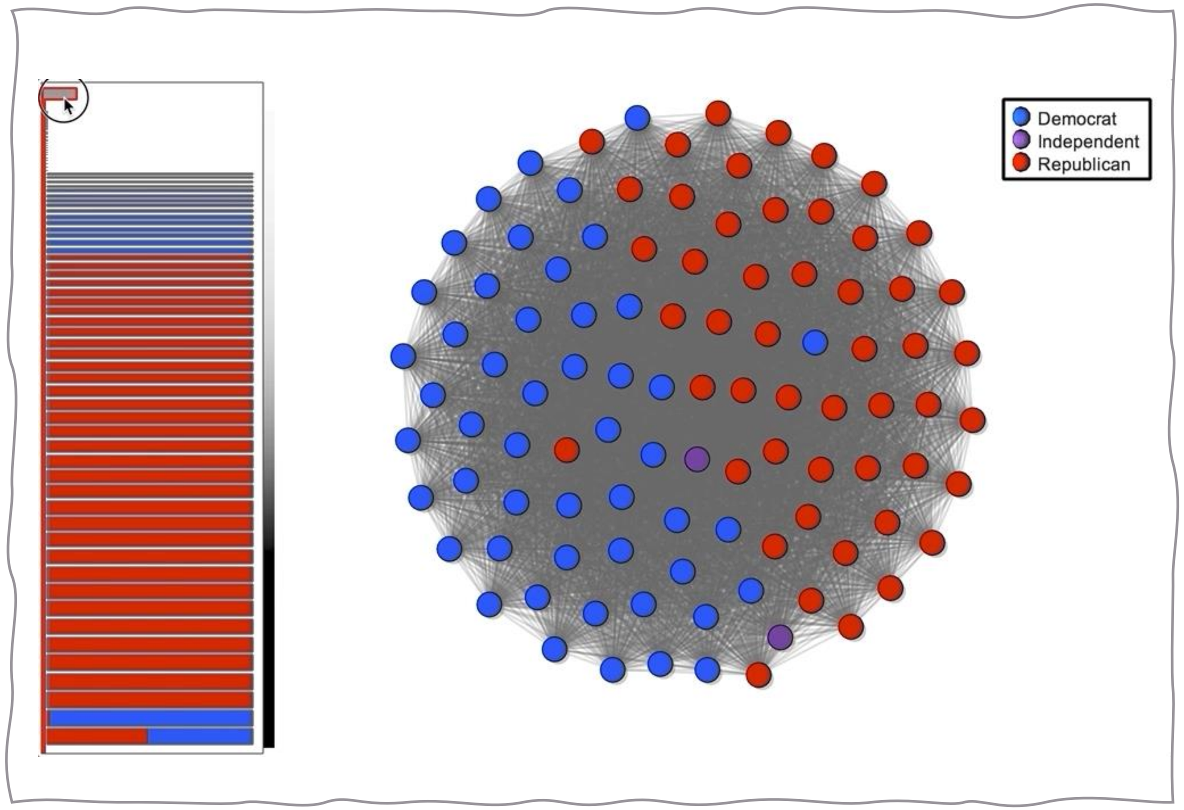
we add additional interactive forces, based on the underlying structure of the graph (e.g., vertex distance), to cluster and/or detangle the layout

Better* interactive force-directed layouts

* I'm biased because this is my own research project ☺



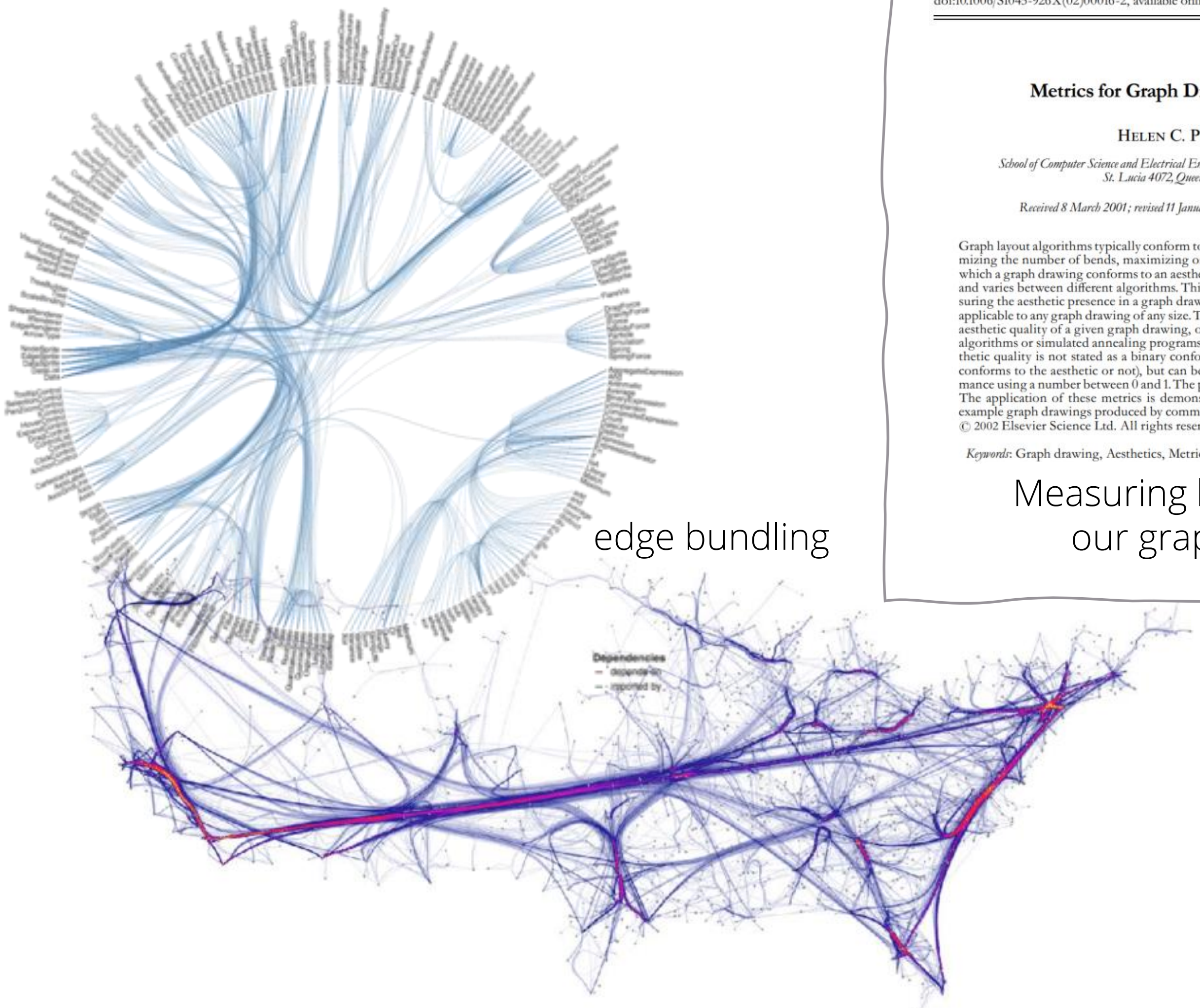
Les Mis dataset



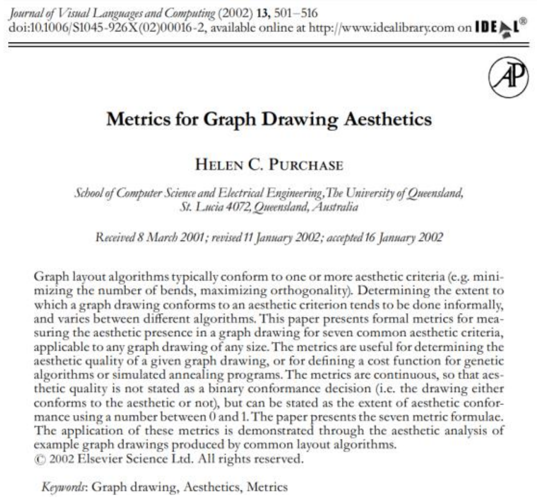
Voting network

Benefits seem clear
Any potential limitations or problems to this approach?

Some other graph drawing techniques



edge bundling



Journal of Visual Languages and Computing (2002) **13**, 501–516
doi:10.1006/S1045-926X(02)00016-2, available online at http://www.idealibrary.com on IDEAL®

Metrics for Graph Drawing Aesthetics

HELEN C. PURCHASE

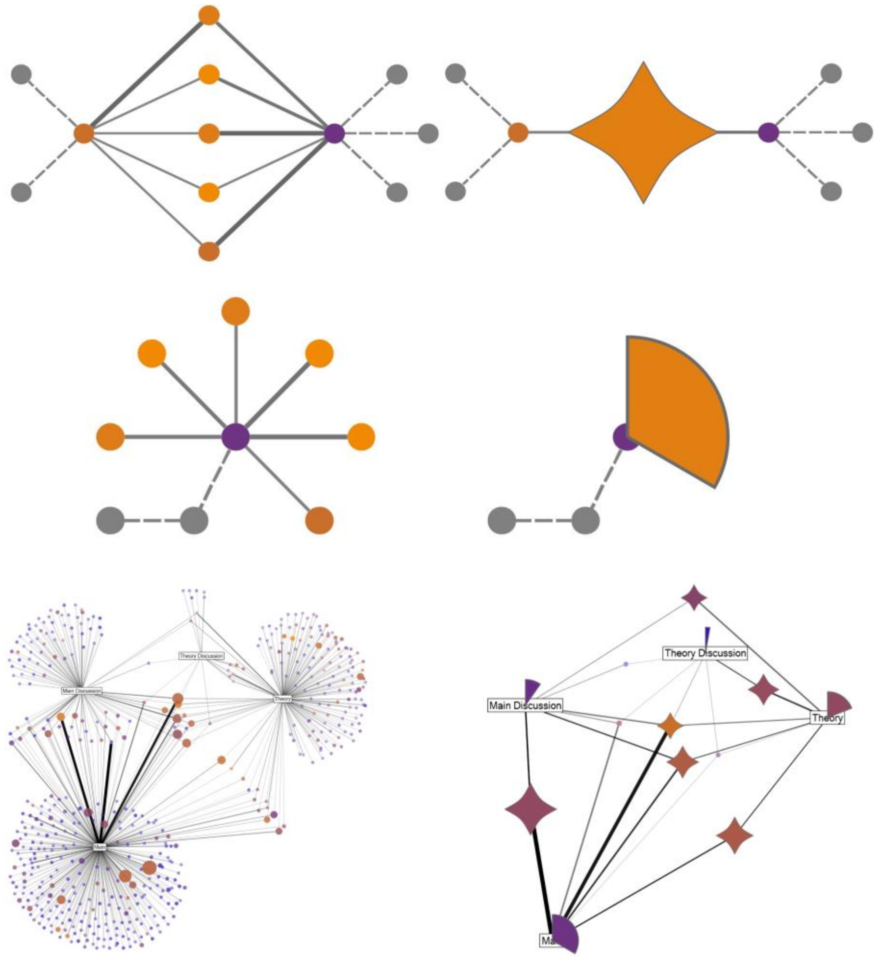
School of Computer Science and Electrical Engineering, The University of Queensland, St. Lucia 4072, Queensland, Australia

Received 8 March 2001; revised 11 January 2002; accepted 16 January 2002

Graph layout algorithms typically conform to one or more aesthetic criteria (e.g. minimizing the number of bends, maximizing orthogonality). Determining the extent to which a graph drawing conforms to an aesthetic criterion tends to be done informally, and varies between different algorithms. This paper presents formal metrics for measuring the aesthetic presence in a graph drawing for seven common aesthetic criteria, applicable to any graph drawing of any size. The metrics are useful for determining the aesthetic quality of a given graph drawing, or for defining a cost function for genetic algorithms or simulated annealing programs. The metrics are continuous, so that aesthetic quality is not stated as a binary conformance decision (i.e. the drawing either conforms to the aesthetic or not), but can be stated as the extent of aesthetic conformance using a number between 0 and 1. The paper presents the seven metric formulae. The application of these metrics is demonstrated through the aesthetic analysis of example graph drawings produced by common layout algorithms.

Keywords: Graph drawing, Aesthetics, Metrics

Measuring how pretty
our graphs are



Motif Glyphs (aka aggregate views)

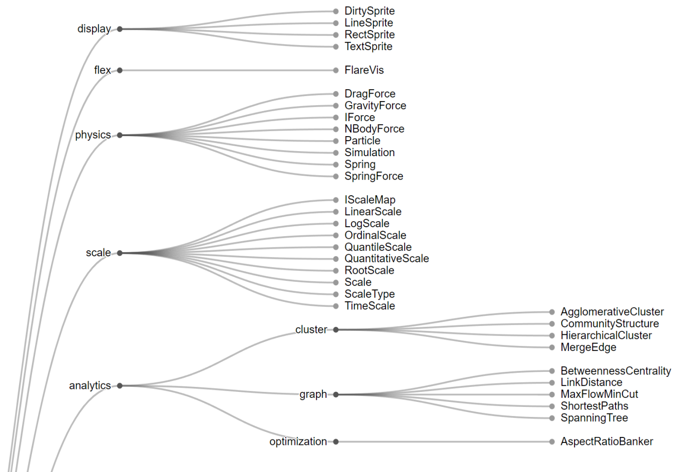
Recap: graph visualizations

Trees

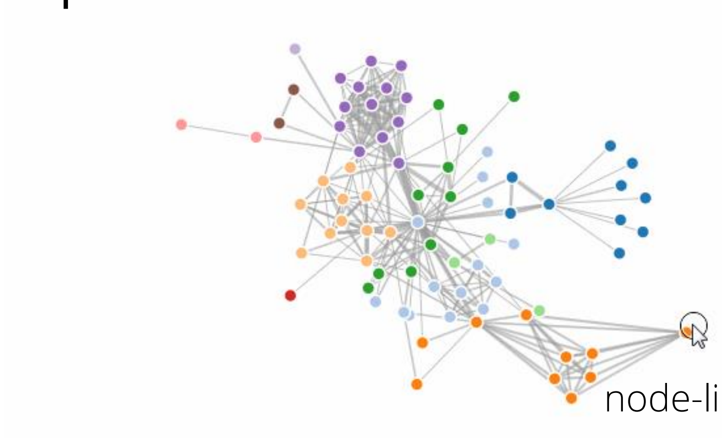
- indentation
  - simple, effective for small trees
- node link and layered
  - looks good but needs space
- enclosure (treemaps)
  - great for size related tasks but suffer in structure related tasks

Graphs / networks

- node link diagram
  - familiar, but problematic for large /dense graphs
- adjacency matrix / diagram
  - efficient but abstract
- aggregated views
  - not always possible, not always appropriate



node-link (tree)


node-link (graph)

Closing remarks

No best graph visualization technique

- Need a good spatial layout for vertices / edges
  - We like aesthetically-pleasing graphs!
- Maintain & highlight the structure of the graph
  - Good for analysis
- Reduce visual clutter (minimize overlapping edges)
- Computationally efficient / feasible

We didn’t even cover *analyzing* graphs / networks!

- We like visualizing graphs so we can analyze them ☺
- Lots of graph analysis techniques
- Complementary graph visualization + analysis systems / tools

Break

In-class activity

Exercise: draw the following graph data as an **adjacency matrix** & **node-link diagram**

V = {1, 2, 3, 4, 5}

E = {(1,2), (1,3), (2,3), (3,4), (4,1), (5,1)}

Edge weights = {(1), (7), (4), (2), (2), (1)}

Node Classes = {A, B, A, A, B}

- Draw an adjacency matrix with any appropriate visual encodings
- Draw a node-link diagram in two ways:
    1. Bad layout
    2. Aesthetically-pleasing layout
- Be creative with your attribute encodings!

Think about the following **questions**:

1. What are the advantages / disadvantages to these methods? Which do you prefer?
2. Why is your bad layout 'bad', and your good layout 'good'?

<u>Zoom instructions</u>

- Take a screenshot of this slide!
- In breakout rooms, individually and/or collaboratively draw your graphs
- Answer the questions together for each person in the group

Activity wrap-up

Volunteers? Thoughts?

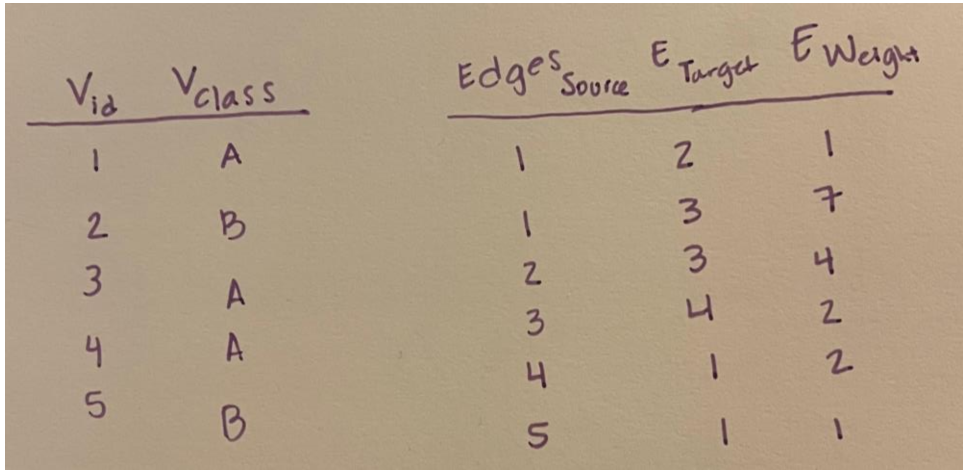
Share your adjacency matrices

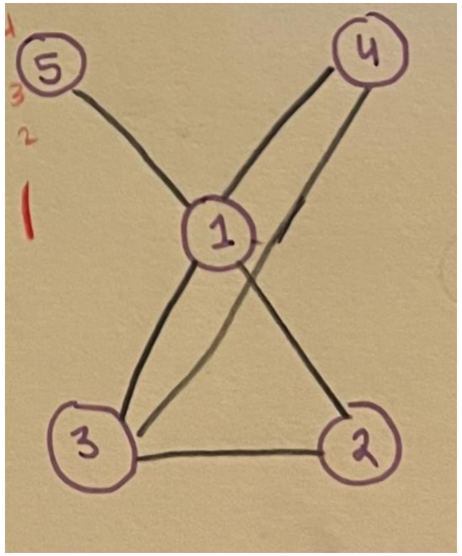
Share your node-link diagrams

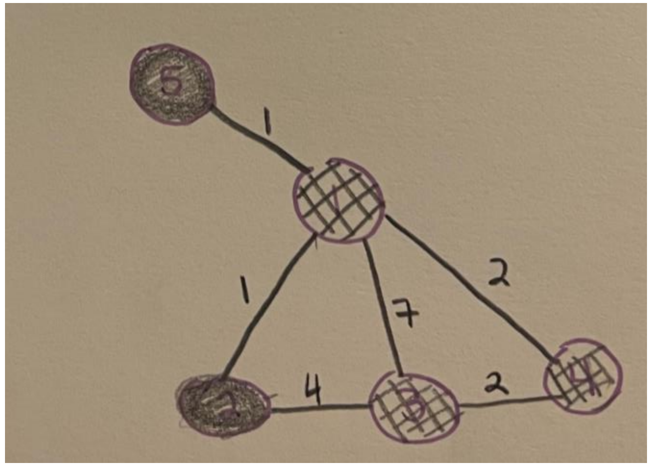
- Bad / good layouts

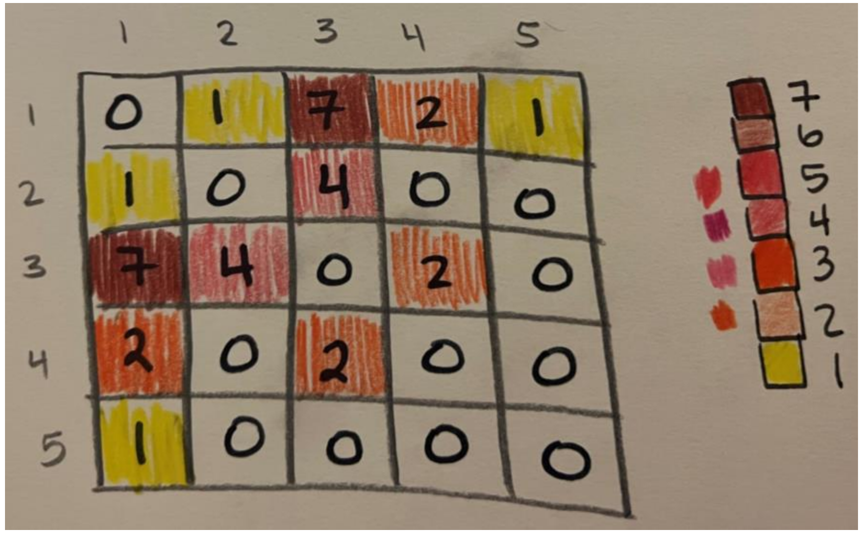
Share your responses to the Qs

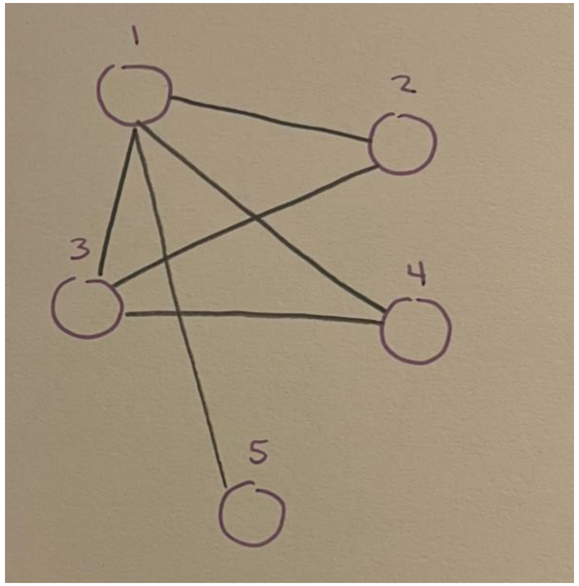
Activity wrap-up

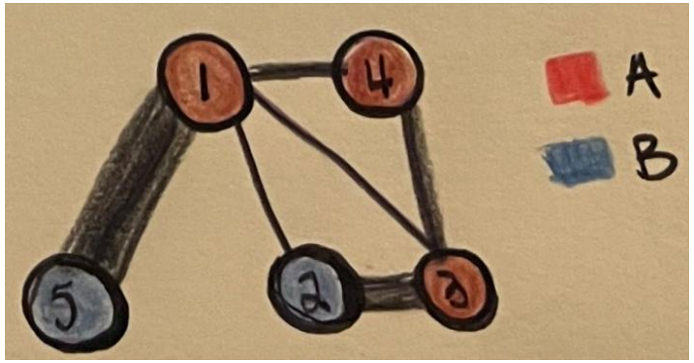












Tools for graph analysis

Network Analysis Tools

- Gephi - an interactive graph analysis application
- NodeXL - a graph analysis plug-in for Excel
- GUESS - a combined visual/scripting interface for graph analysis
- Pajek - another popular network analysis tool
- NetworkX - graph analysis library for Python
- SNAP - graph analysis library for C++

Thank you!

Questions?

Want to talk more about graphs, research, and/or graduate school?

Email me! **Ashley.Suh@Tufts.edu**