DATA VISUALIZATION

Visualizing Trees and Graphs

Tobias Isenberg
tobias.isenberg@inria.fr

RECAP

you have learned about

- perception and color
- simple plots
- multi-attribute data visualization

DATA AND ITS STRUCTURE

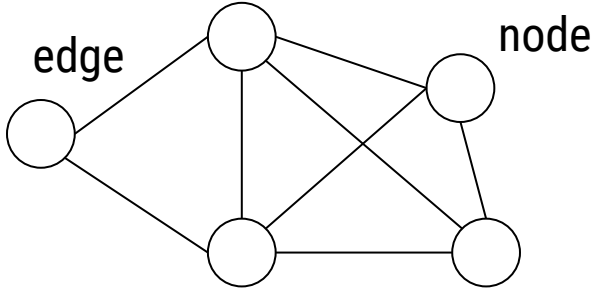
STRUCTURED DATA

0.103	0.176	0.387	0.300	0.379
0.333	0.384	0.564	0.587	0.857
0.421	0.309	0.654	0.729	0.228
0.266	0.750	1.056	0.936	0.911
0.225	0.326	0.643	0.337	0.721
0.187	0.586	0.529	0.340	0.829
0.153	0.485	0.560	0.428	0.628

UNSTRUCTURED DATA

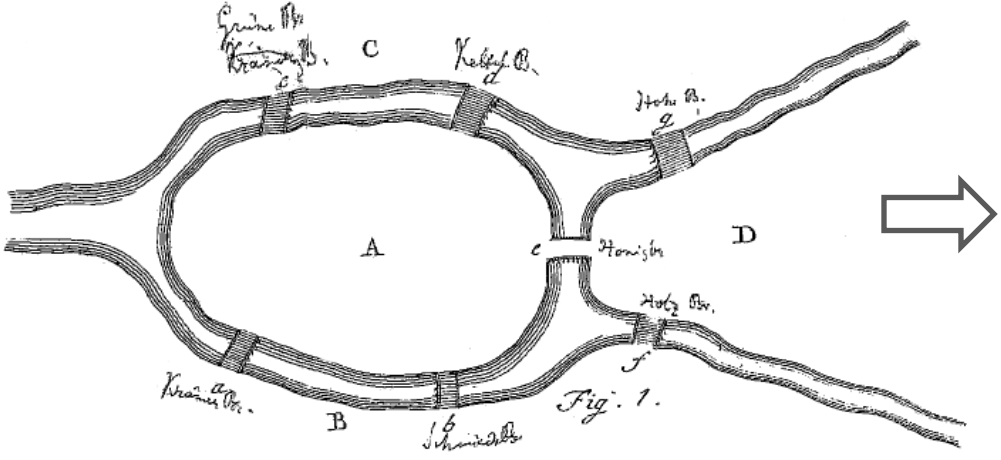
DATA RELATIONSHIPS

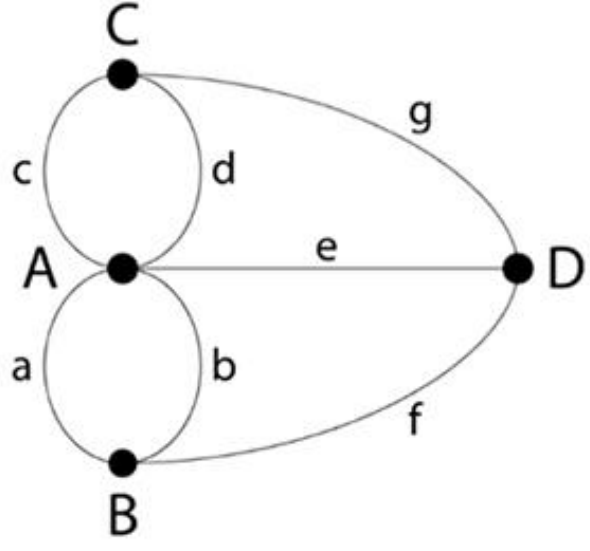
- there are relationships between the data items
- we can use a graph representation



almost anything can be a graph

The 7 bridges of Königsberg (now Kaliningrad, Russia), 1741

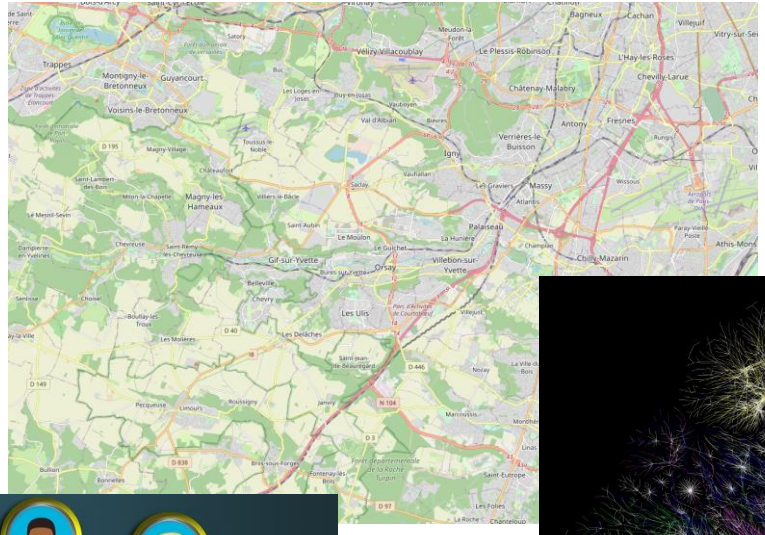


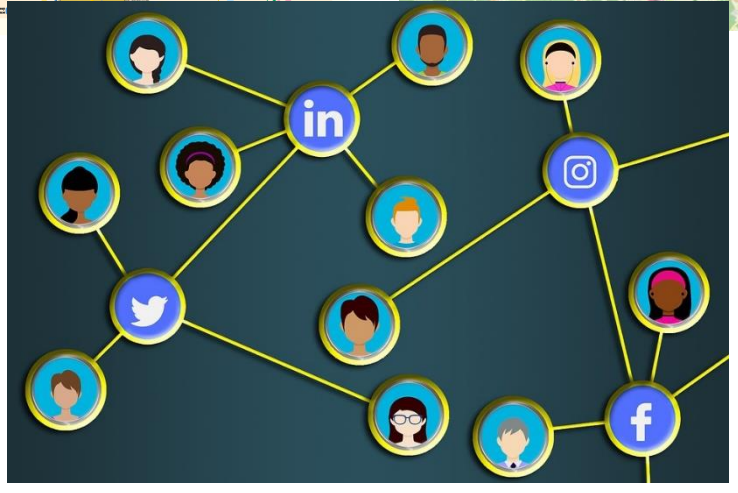


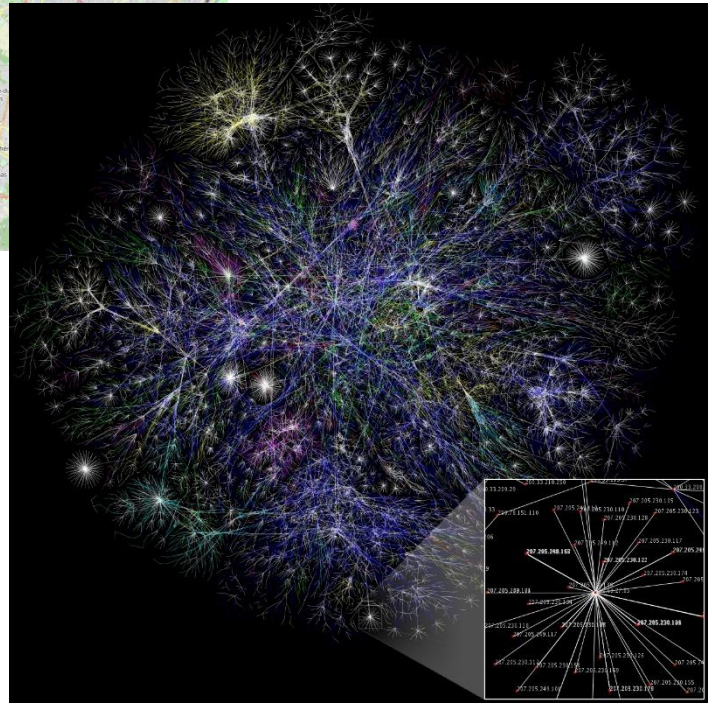
How can you cross all 7 bridges without crossing the same one twice?

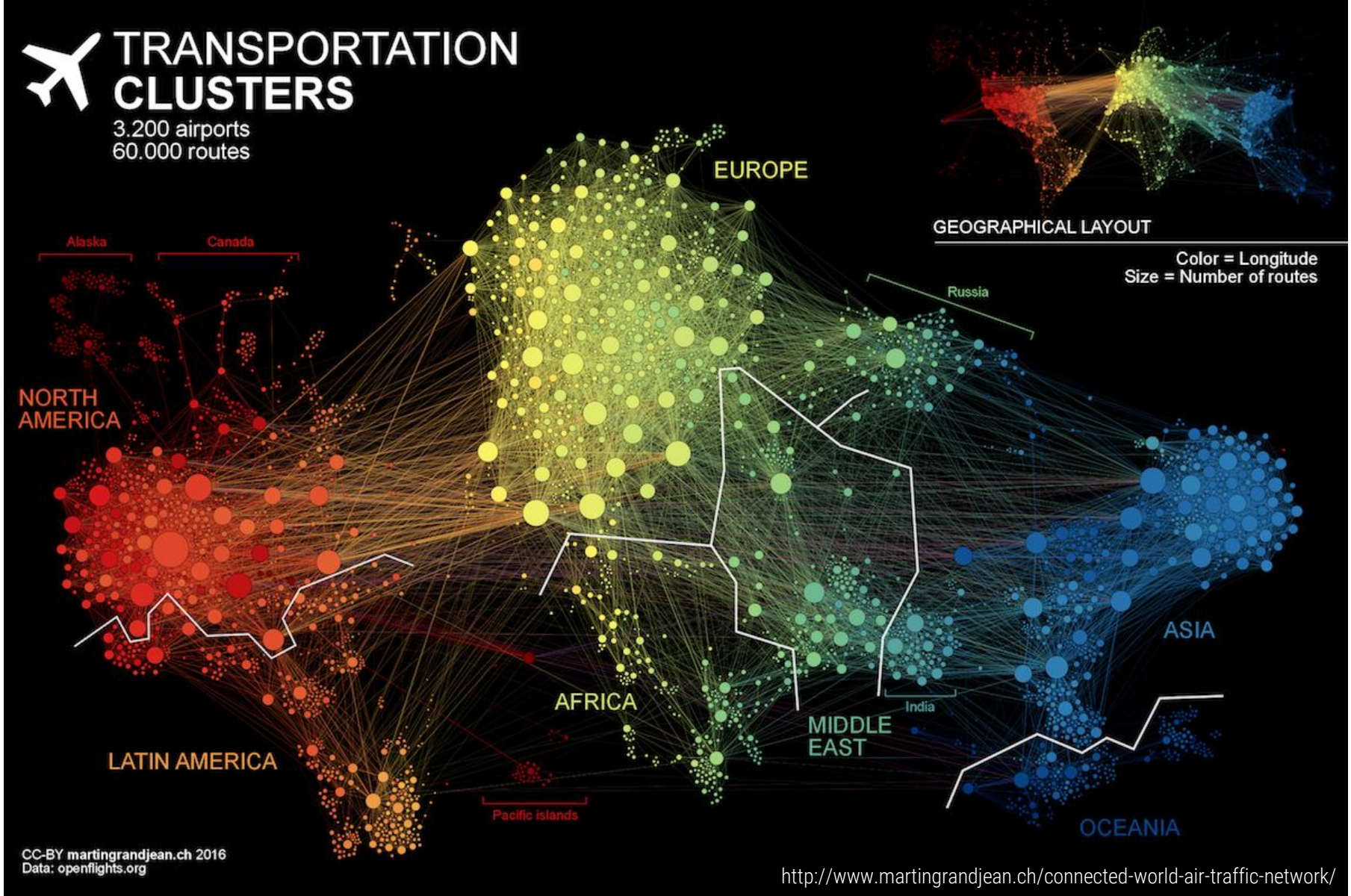
Euler's conceptualization of the same (topological) problem

What other graphs do you know and use?







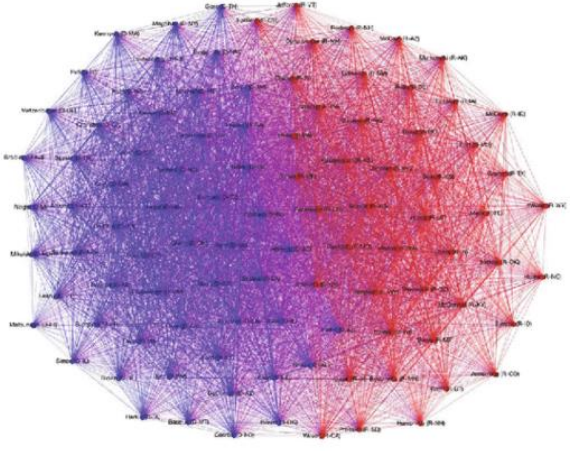


http://www.martingrandjean.ch/connected-world-air-traffic-network/

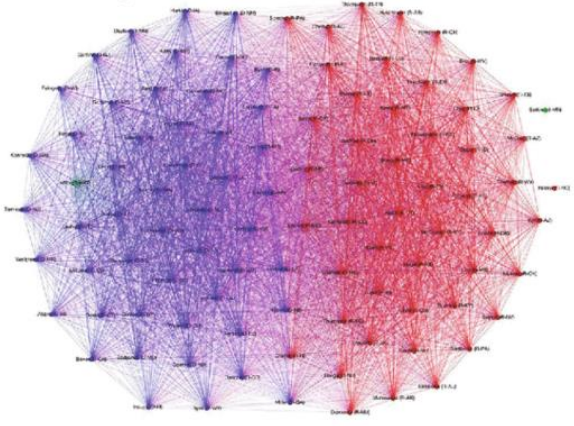
Senators casting the same votes

Democrat Republican

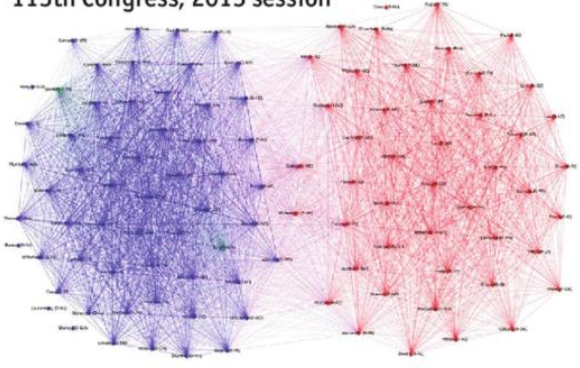
101st Congress, 1989 session



107th Congress, 2002 session

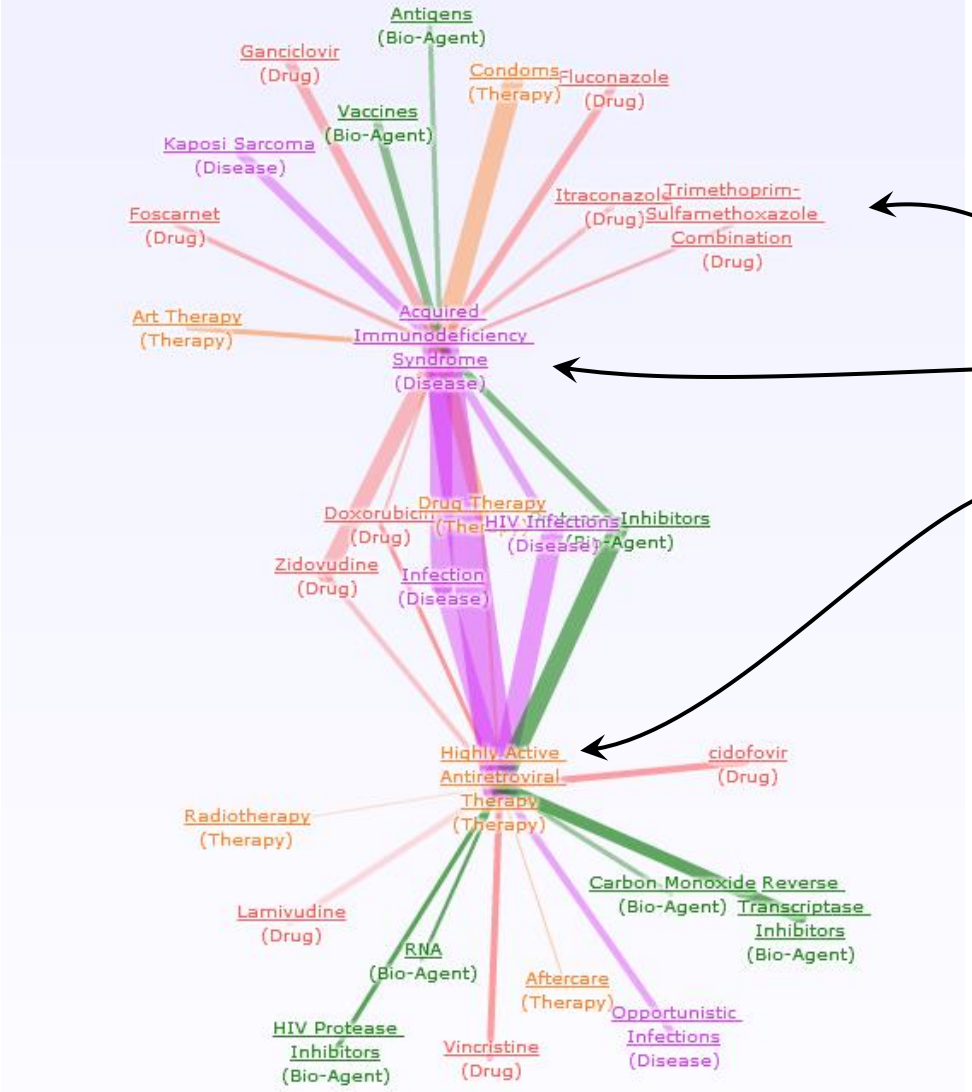


113th Congress, 2013 session



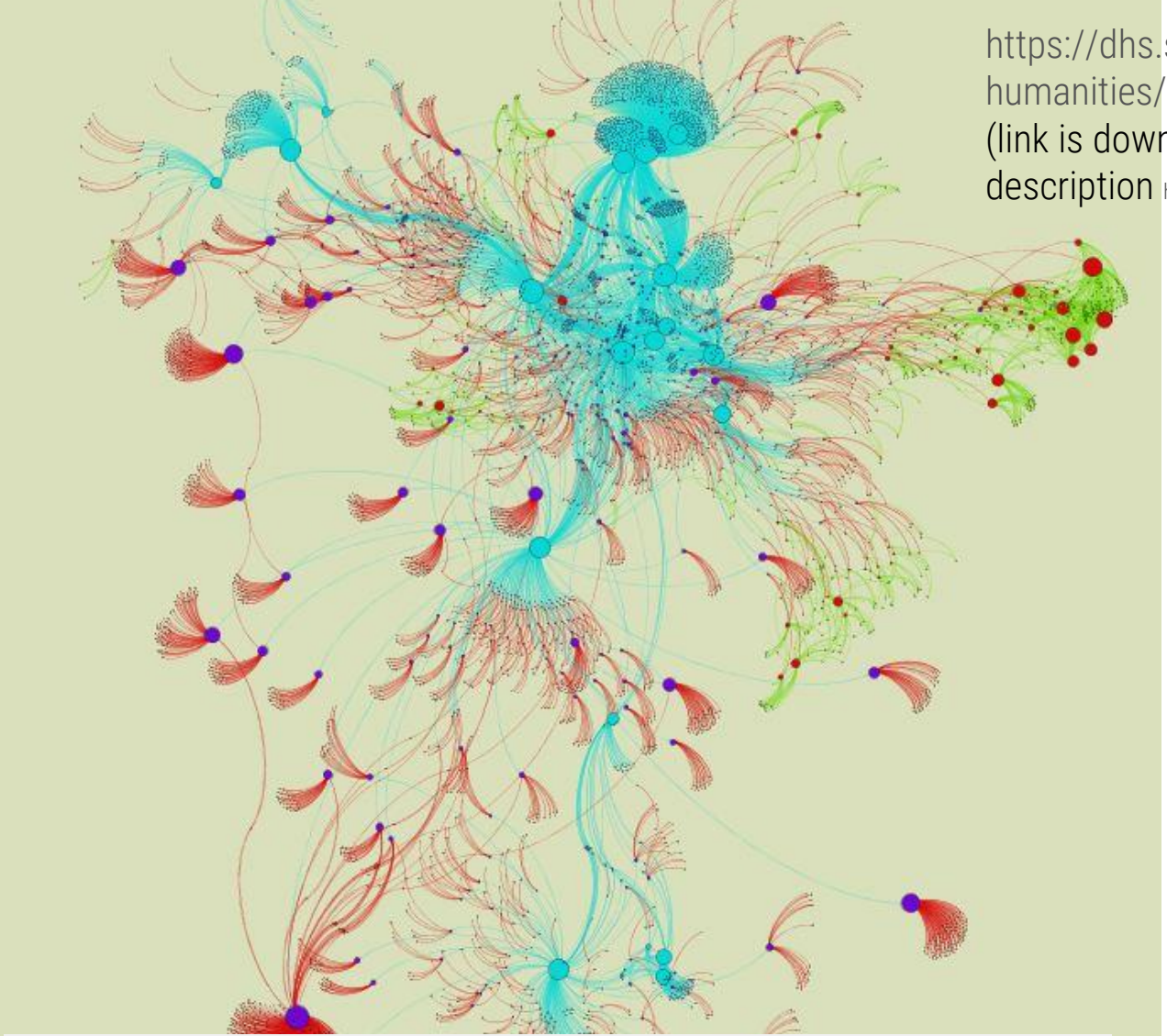
Sources: GovTrack.us, Renzo Lucioni

Independent



http://www.curehunter.com/

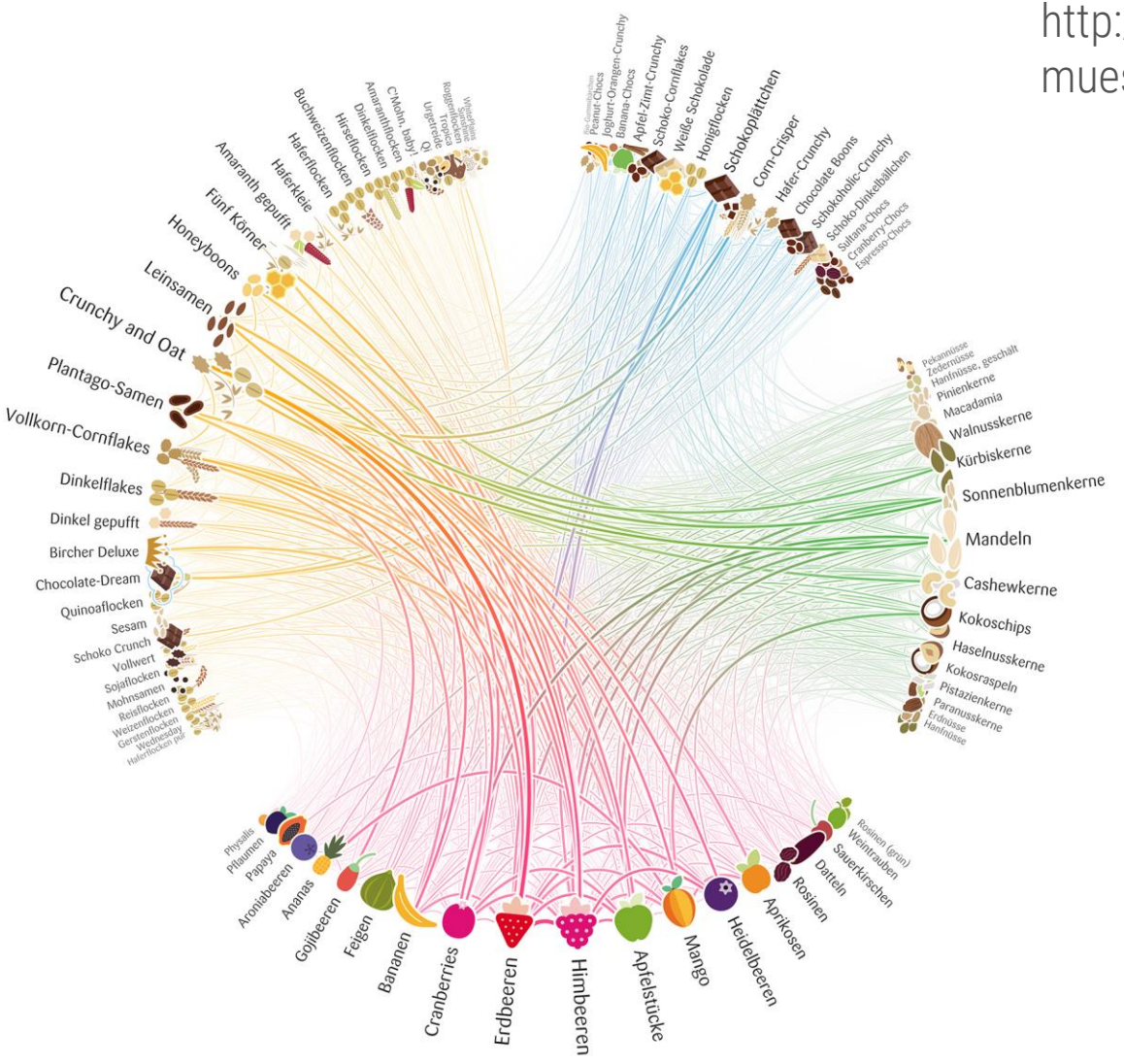
visual dictionary of drugs, diseases and therapies



Top Contributors to the Catalogue of Life and their associated species, references and databases

https://dhs.stanford.edu/spatial-humanities/visualizing-databases/
(link is down for now, but here is a simple description http://www.visualcomplexity.com/vc/project.cfm?id=759)

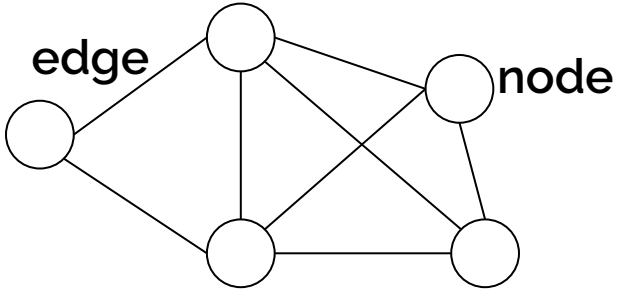
http://truth-and-beauty.net/projects/muesli-ingredient-network



GRAPHS

graphs

- describe relations among data items
- using nodes and edges
  (vocabulary: node/vertex, edge/link)

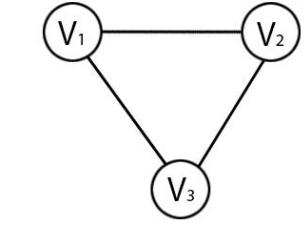


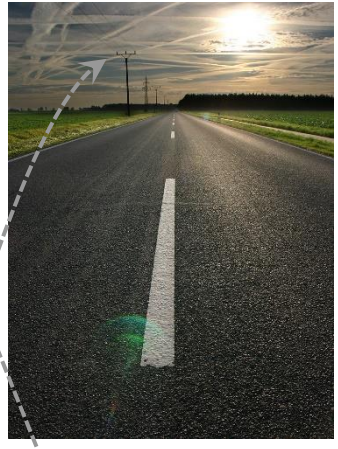
DEFINITIONS

undirected graph:
edges have no orientation

directed graph (digraph):
edges have orientation

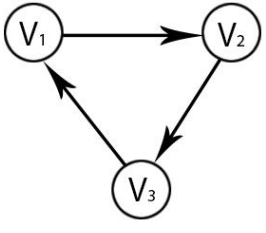
Undirected Graph





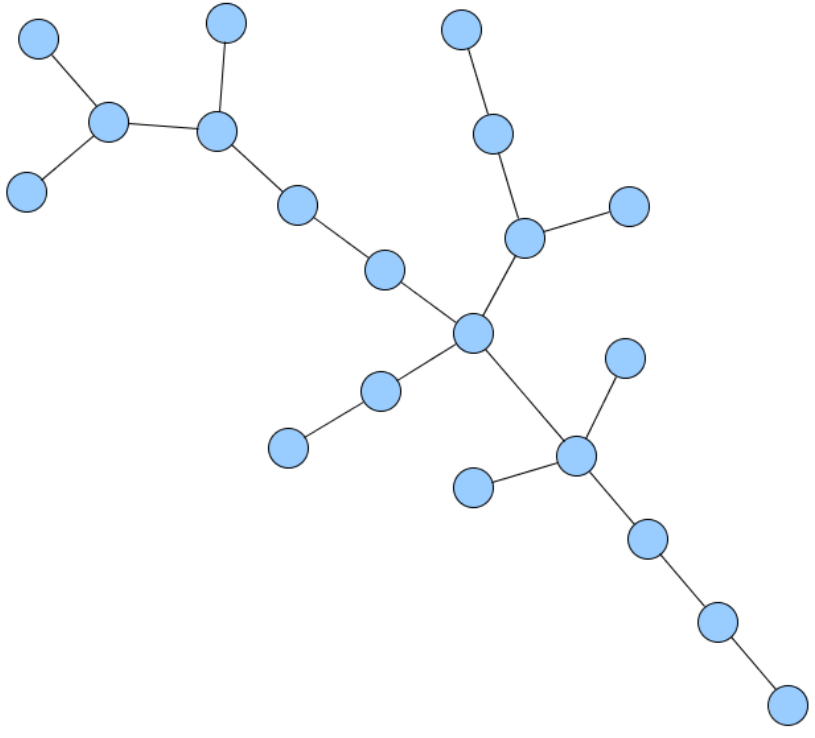
"The road to Offen 2" by Jasmic, CC BY-ND 2.0

Directed Graph



DEFINITIONS

a **tree** is a connected graph with no cycles

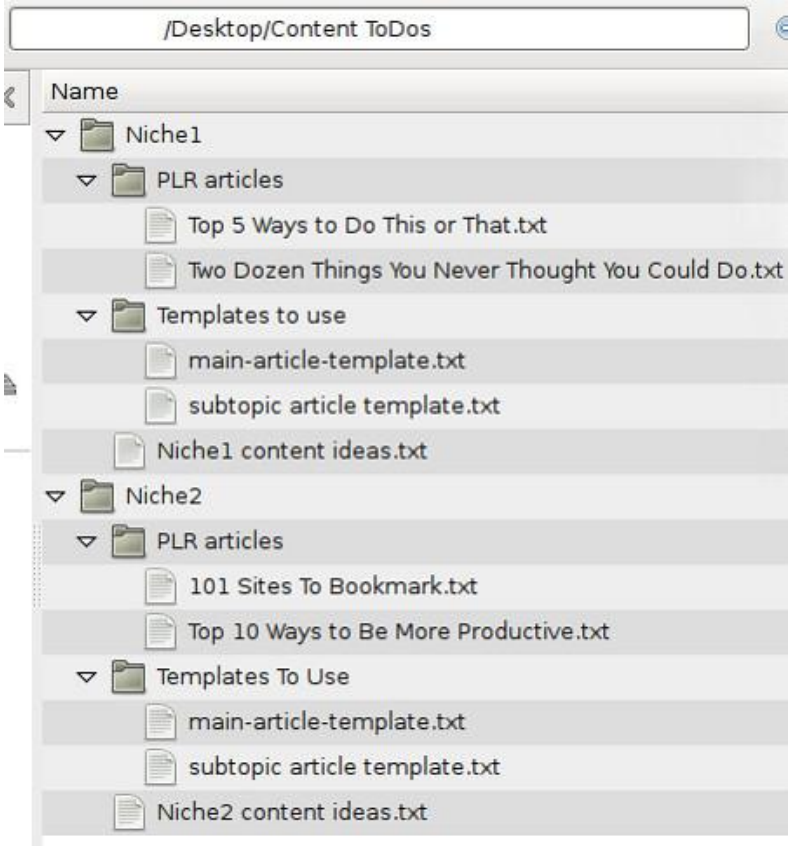


DEFINITIONS

The most important nonlinear data structure in computer science (Donald Knuth, 1997)

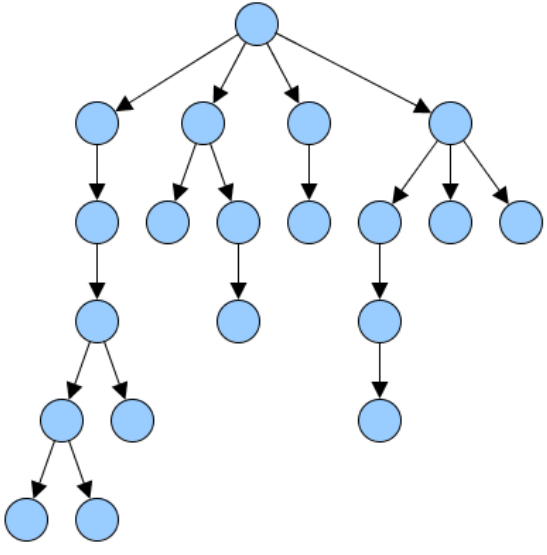
a **directed tree** is a digraph (directed graph) whose underlying graph is a tree

- a directed tree consists of a number of nodes and parent-child relationships
- every node has just one parent and any number of children
- *directed trees are the most common form in computer science*



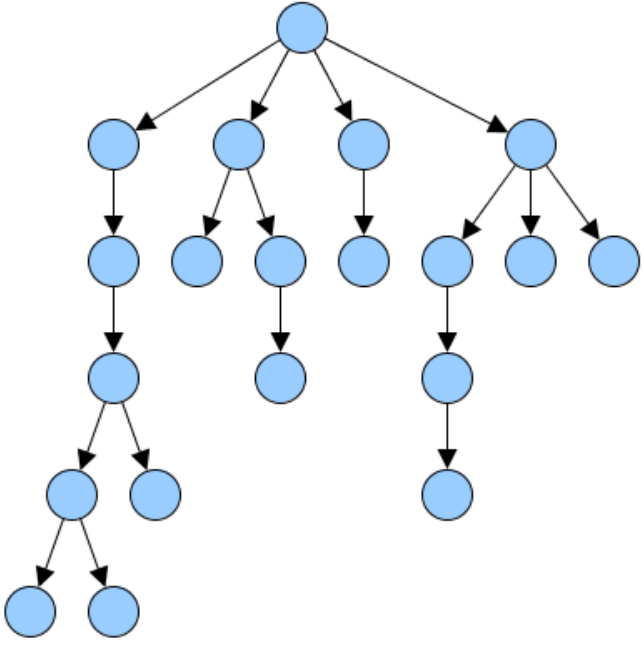
DEFINITIONS

- **degree** – the number of children of a node
- **leaf nodes** are nodes without children

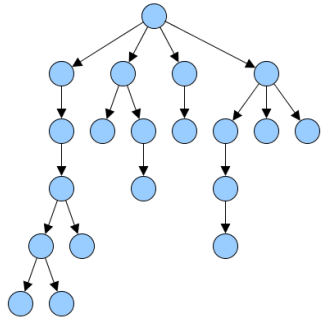


DEFINITIONS

edge – the connection between parent and child **nodes**



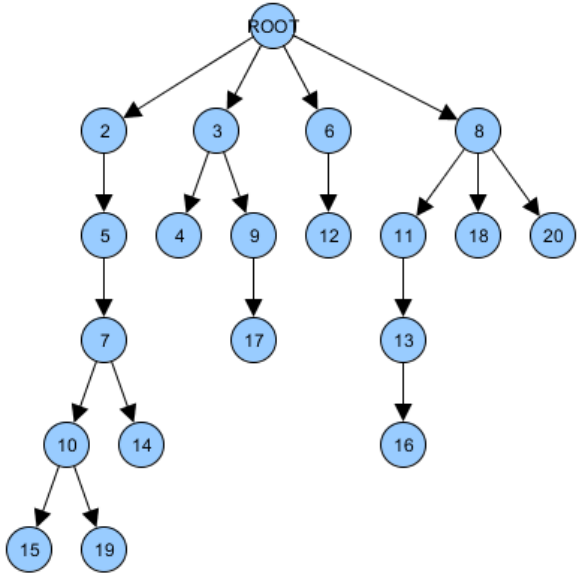
DEFINITIONS



- a **rooted tree** is a directed tree with a distinguished vertex r, called the **root**, such that for every other vertex v there is **directed path from r to v**
- the **root node** is the only node with **no parent**
- (any node may act as a root in undirected trees)

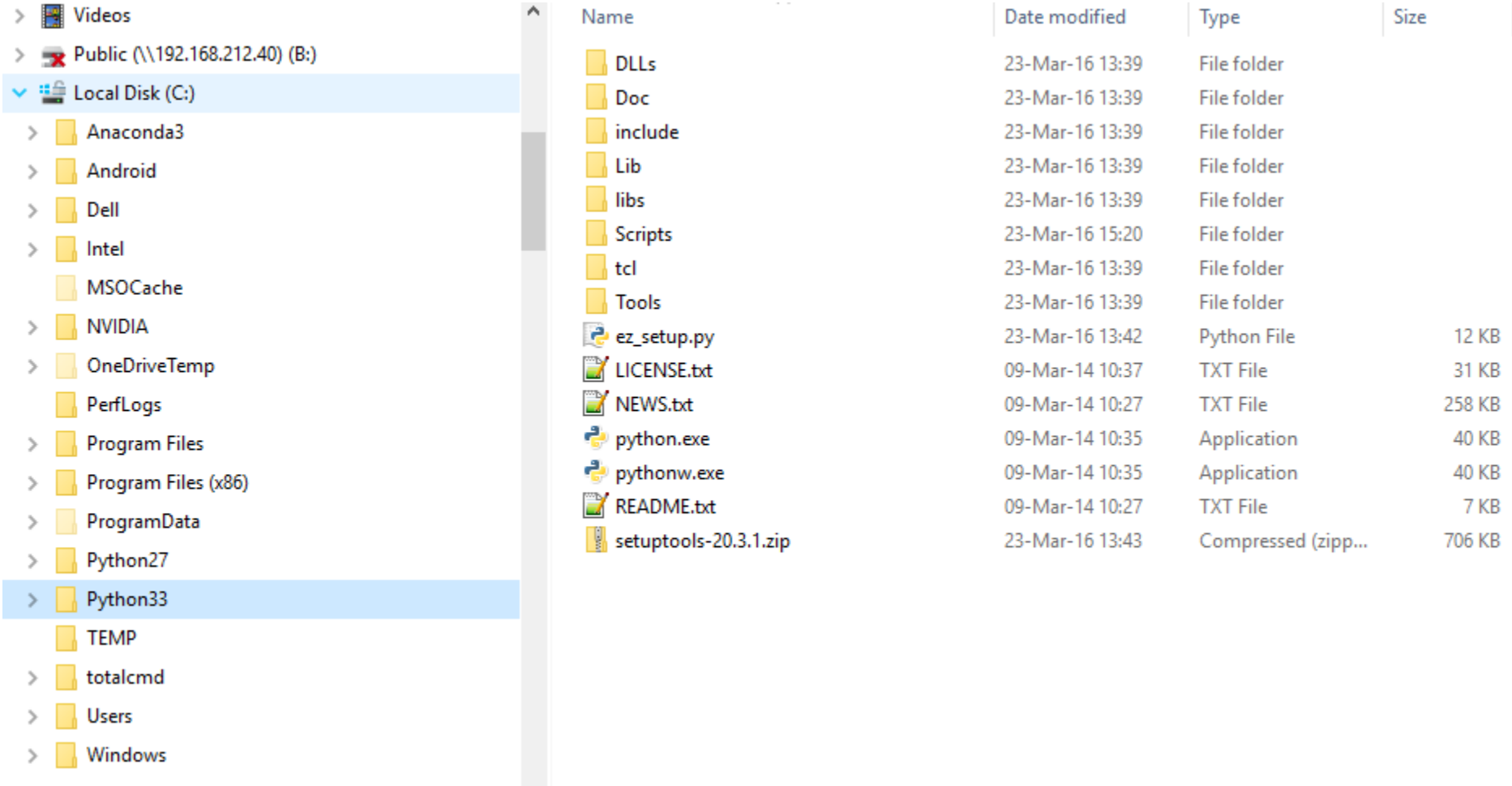
DEFINITIONS

an **ordered tree** is a rooted tree in which the children of each vertex are assigned a fixed ordering



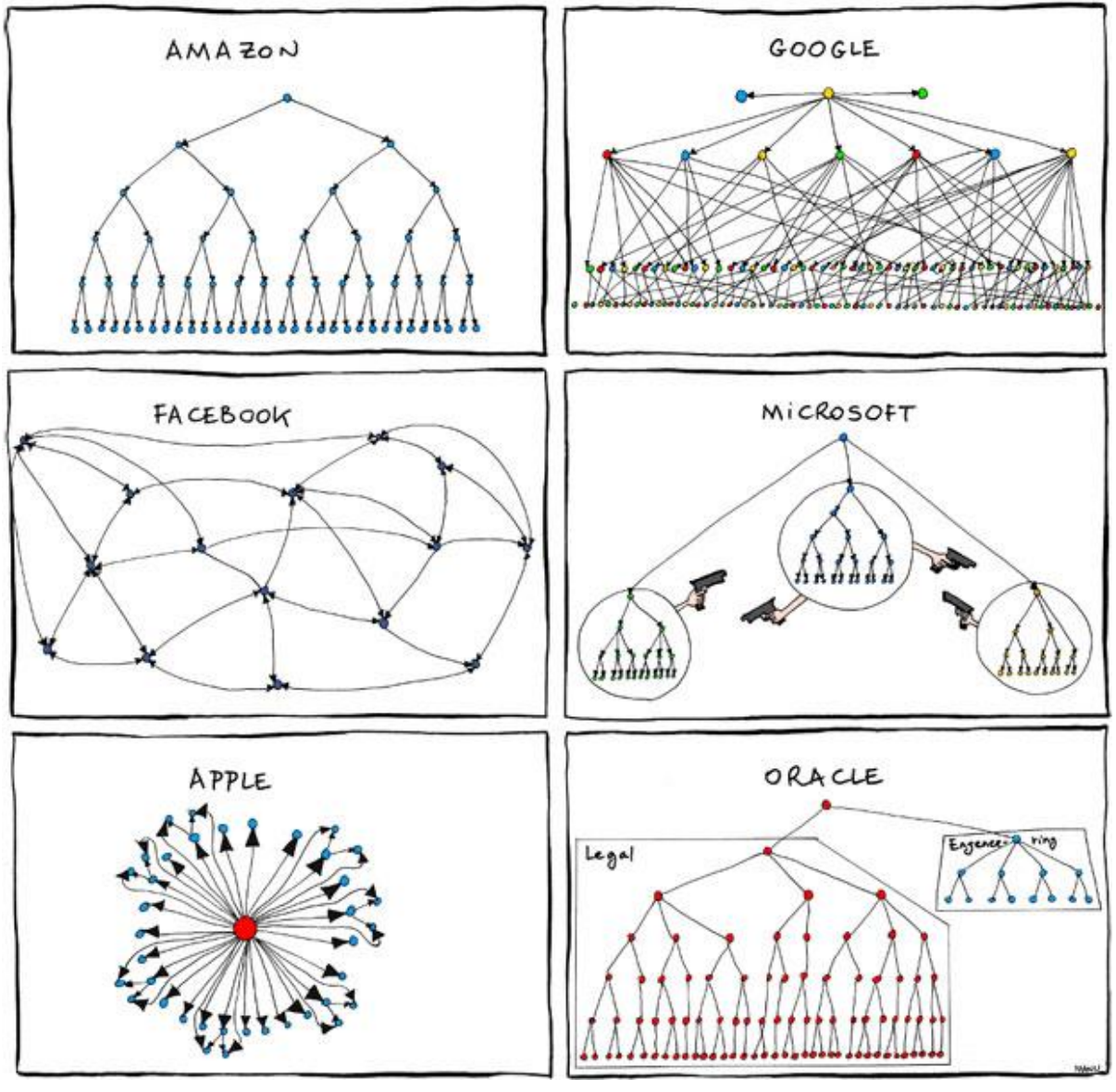
EXAMPLES OF TREES

HIERARCHIES



HIERARCHIES

https://www.youtube.com/watch?v=mkJ-Uy5dt5g



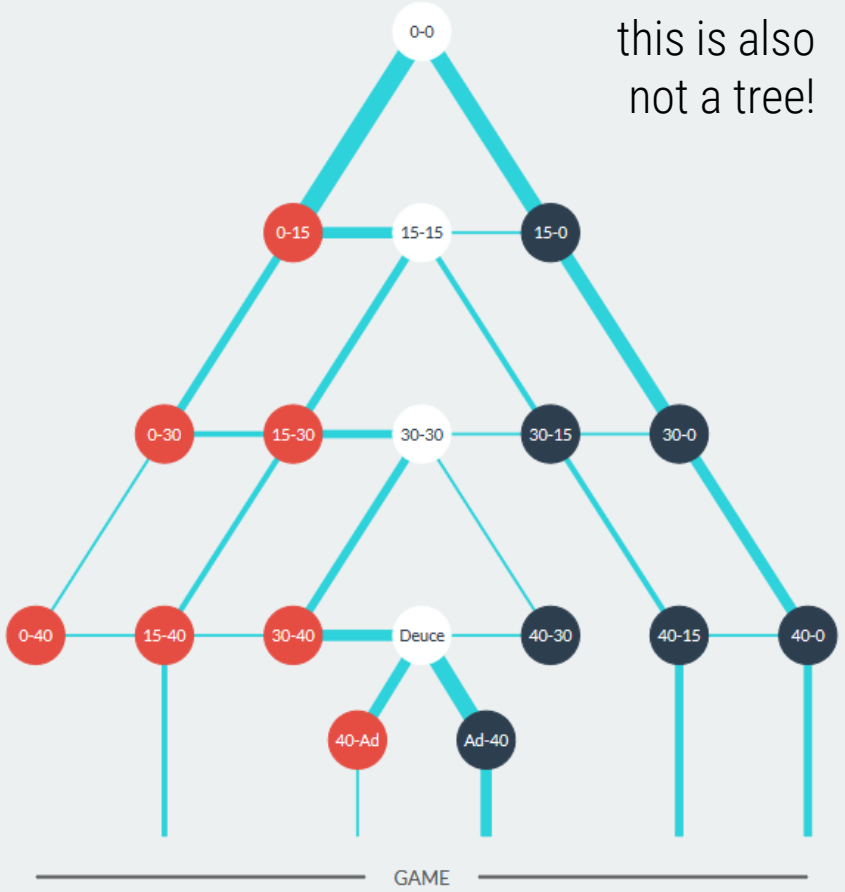
org charts aren't always trees, though

http://www.bonkersworld.net/organizational-charts/

DECISION PROCESS

NADAL

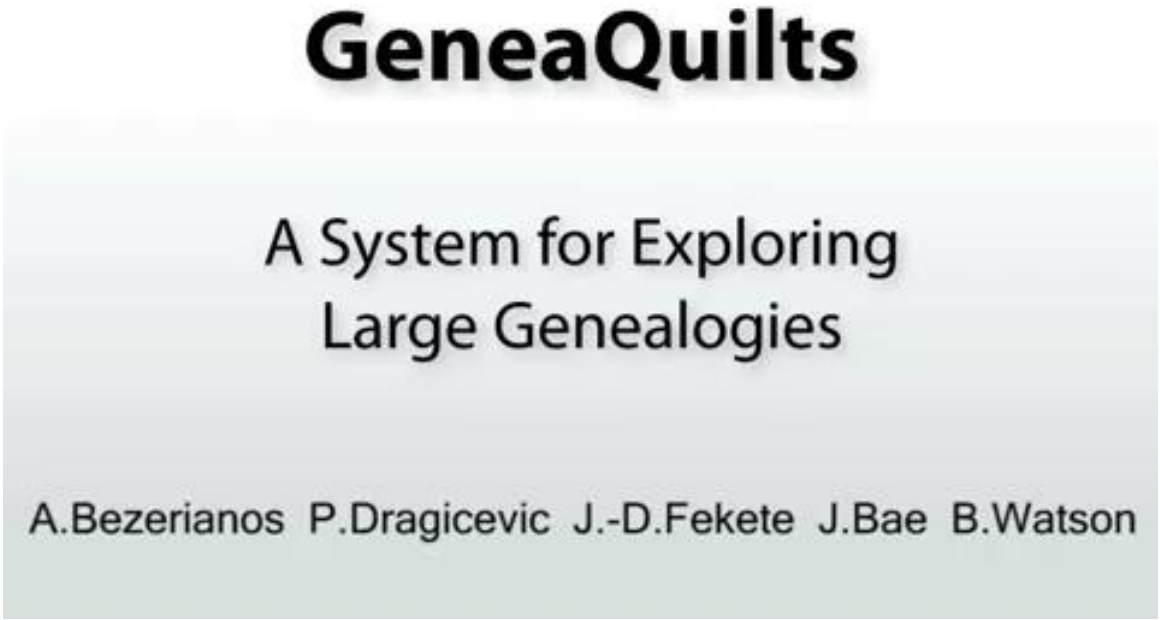
- Indian Wells ›
- Monte-Carlo ›
- Madrid ›
- Rome ›
- Roland Garros ›
  - Brands 4-6, 7-6(4), 6-4, 6-3
  - Klizan 4-6, 6-3, 6-3, 6-3
  - Fognini 7-6(5), 6-4, 6-4
  - Nishikori 6-4, 6-1, 6-3
  - Wawrinka 6-2, 6-3, 6-1
  - Djokovic 6-4, 3-6, 6-1, 6-7(3), 9-7
  - Ferrer 6-3, 6-2, 6-3
- Wimbledon ›
- Rogers Cup ›



http://gamesetmap.com/nadalgametree/

BRANCHING PROCESSES

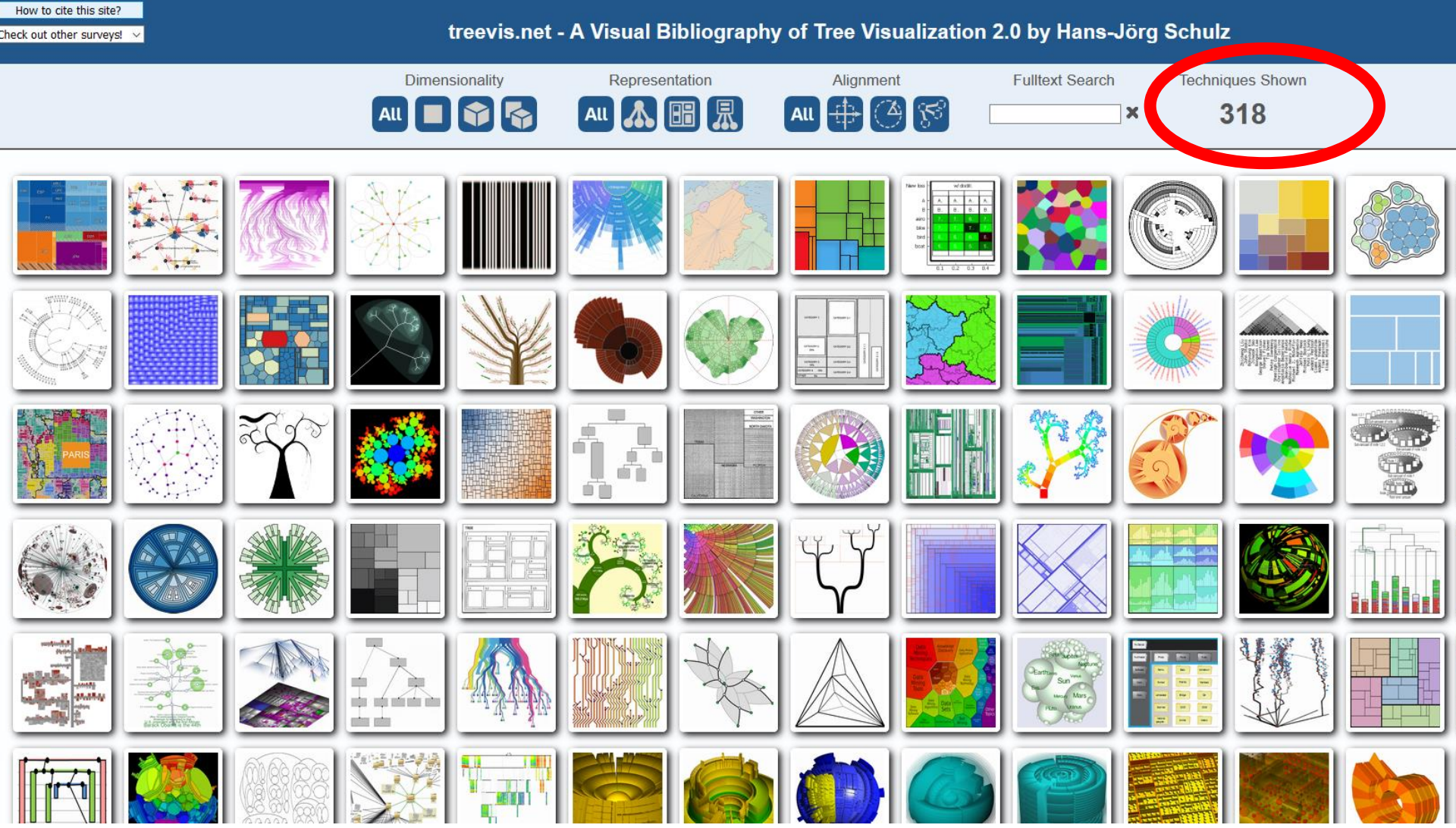
Think about it:
Is a family tree really a tree?



https://www.youtube.com/watch?v=gncBzqI7R-Q

TREE REPRESENTATION

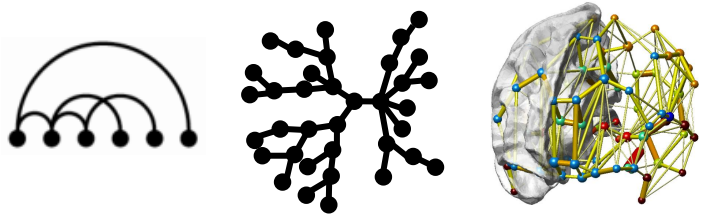
TECHNIQUES



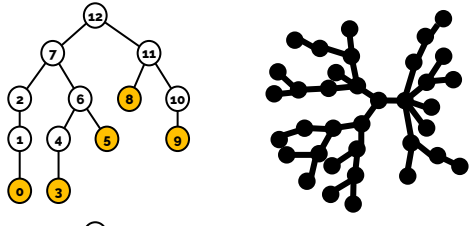
http://treevis.net/

CATEGORIZATIONS OF LAYOUTS

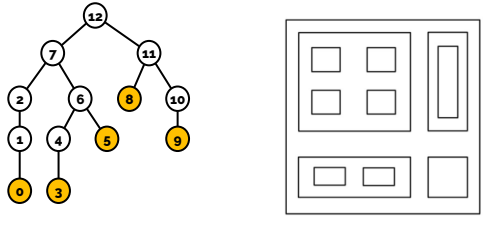
- many possible
- here we follow the categorization on treevis.net:
  - dimensionality of the layout



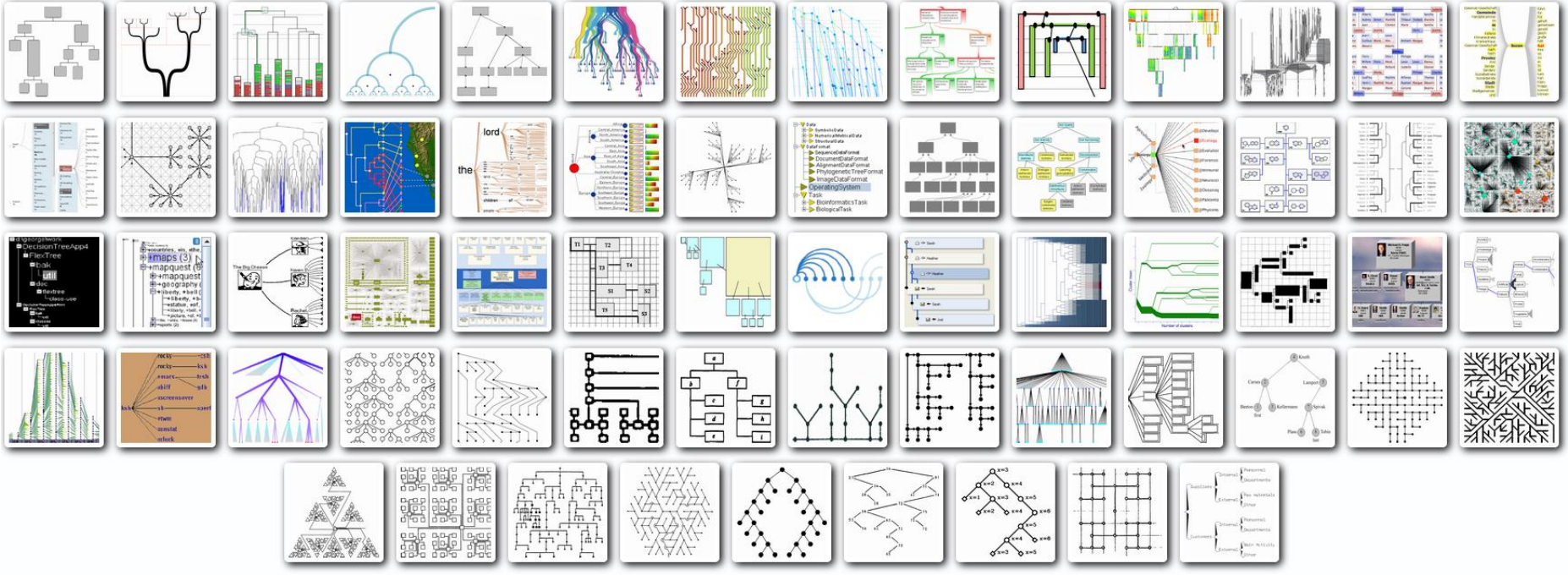
  - alignment of nodes in space



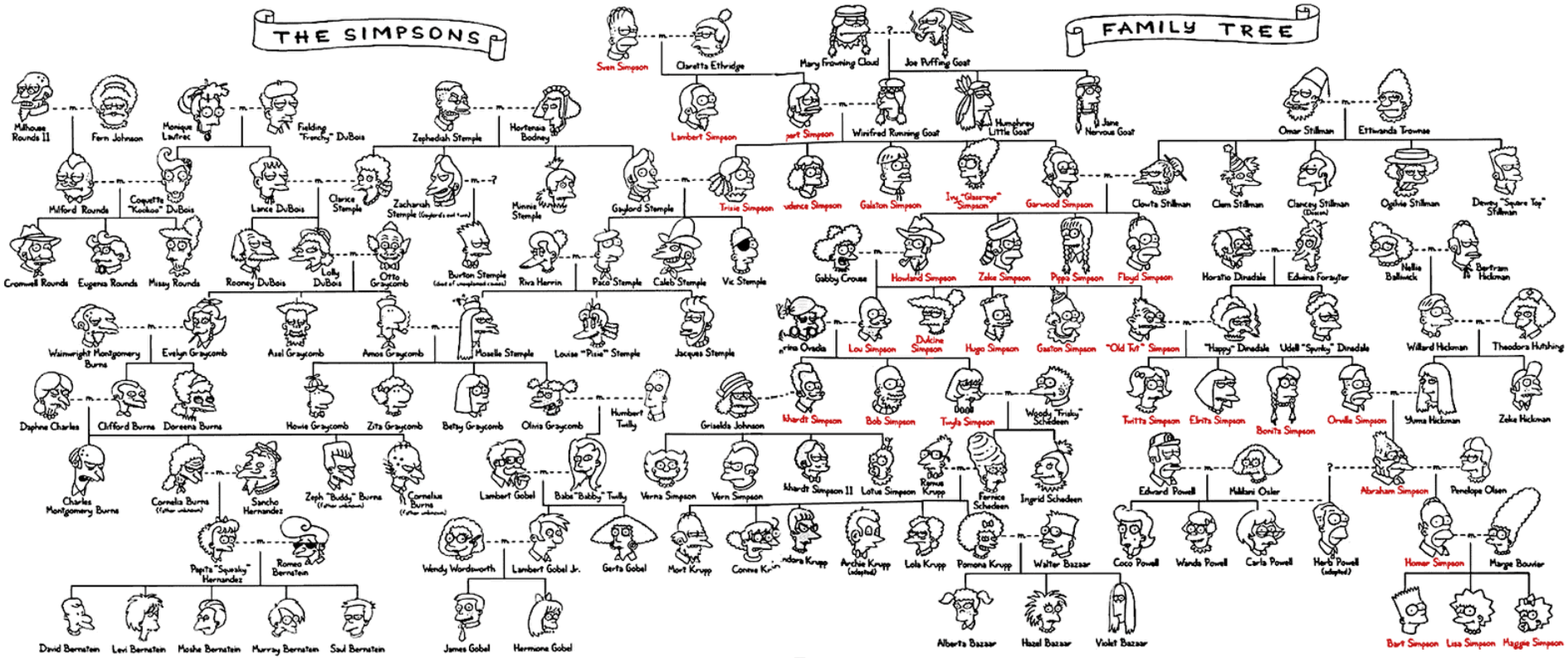
  - representation type



2D, AXIS-PARALLEL, EXPLICIT EDGES

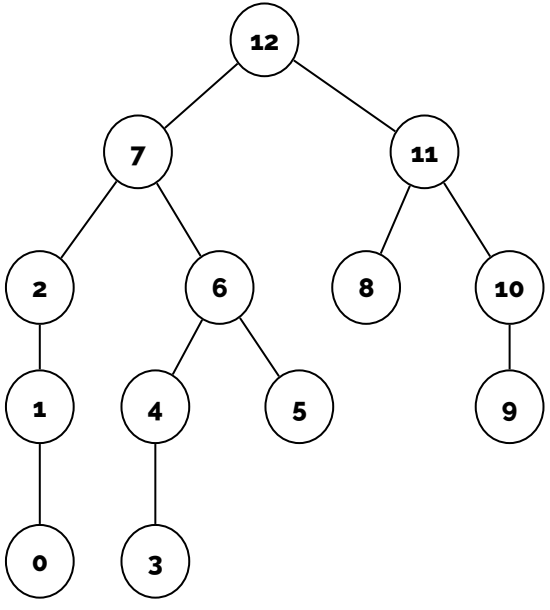


NODE-LINK



http://simpsons.wikia.com/wiki/

NODE-LINK ALGORITHM

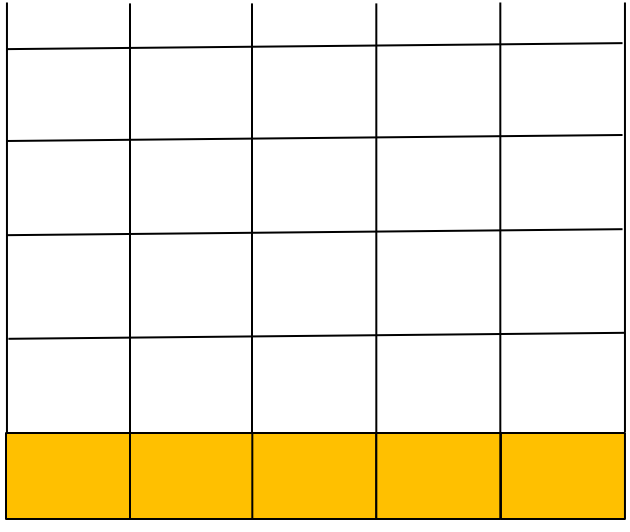


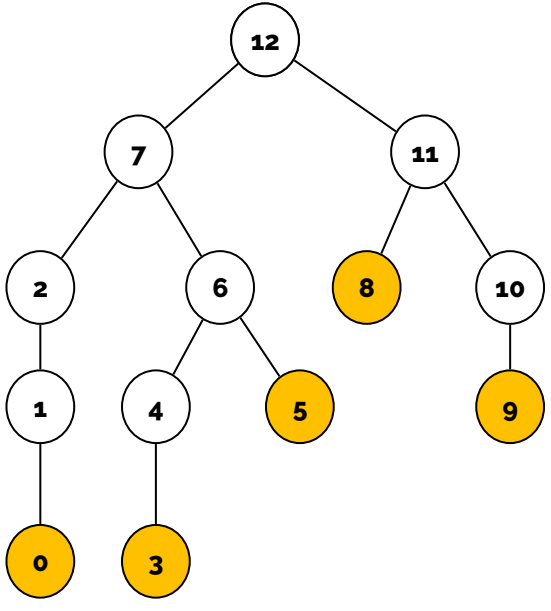
THE LAYOUT WE WANT – HOW DO WE GET THERE?

NODE-LINK ALGORITHM

SIMPLE APPROACH (Reingold-Tilford algorithm)

1) COUNT THE LEAVES IN THE SUBTREE
2) PLACE THE ROOT IN MIDDLE

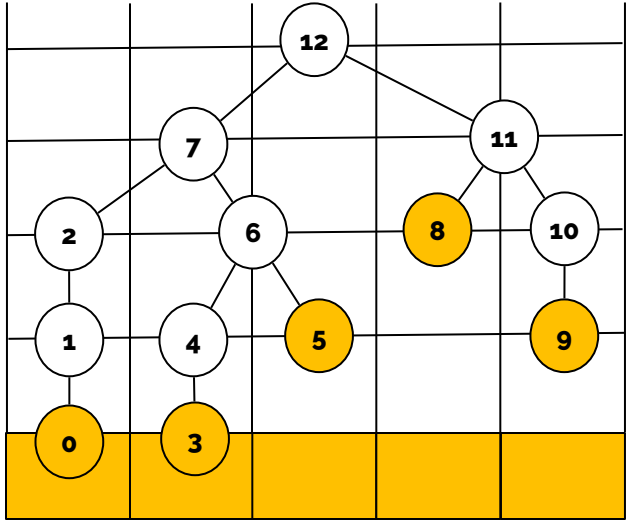


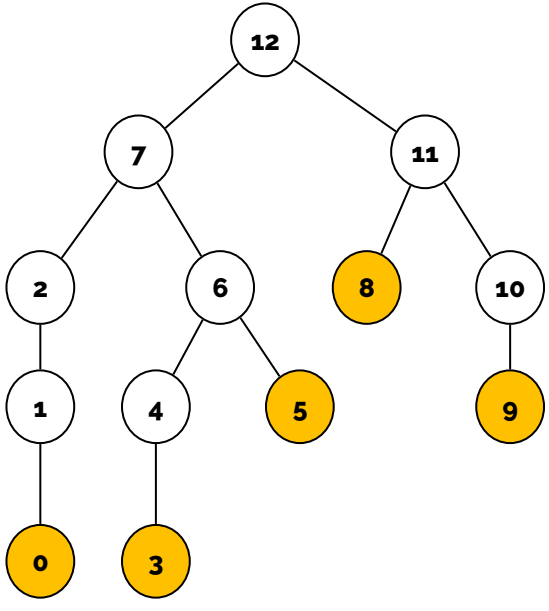


NODE-LINK ALGORITHM

SIMPLE APPROACH (Reingold-Tilford algorithm)

1) COUNT THE LEAVES IN THE SUBTREE
2) PLACE THE ROOT IN MIDDLE
3) RECURSIVELY DIVIDE AND REPEAT LEFT & RIGHT





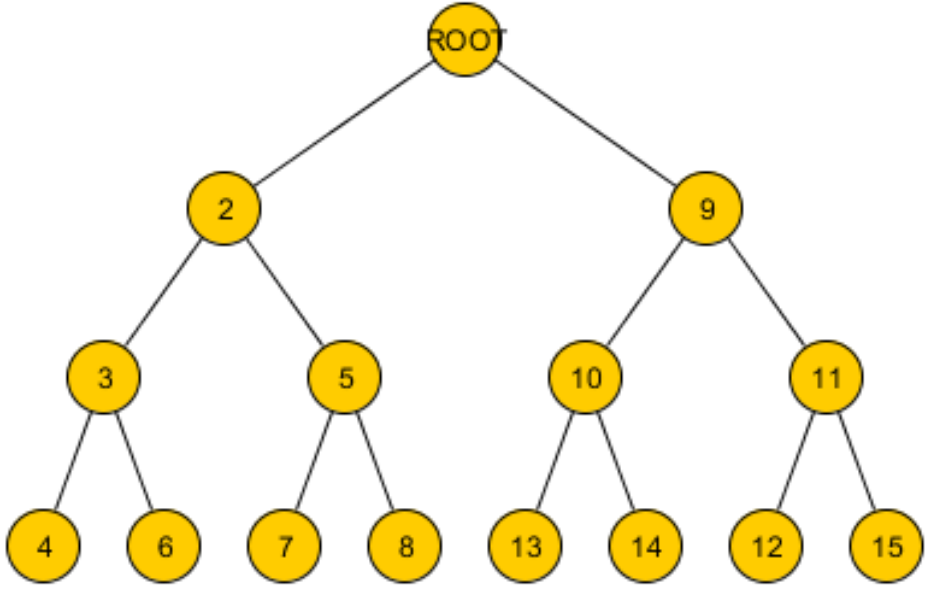
PROS/CONS

(of Reingold-Tilford algorithm)

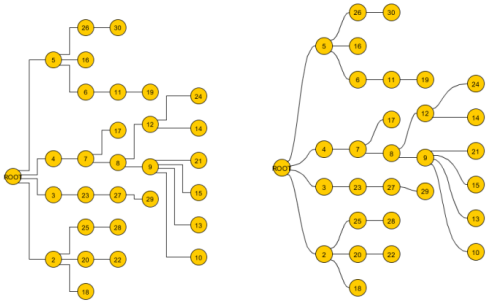
- nodes at the same distance from the root are horizontally aligned
- positive: simple to understand, clear symmetries
- negative: needs large area, often bad aspect ratio (much wider than tall)

WHAT CAN WE VARY IN THIS REPRESENTATION?

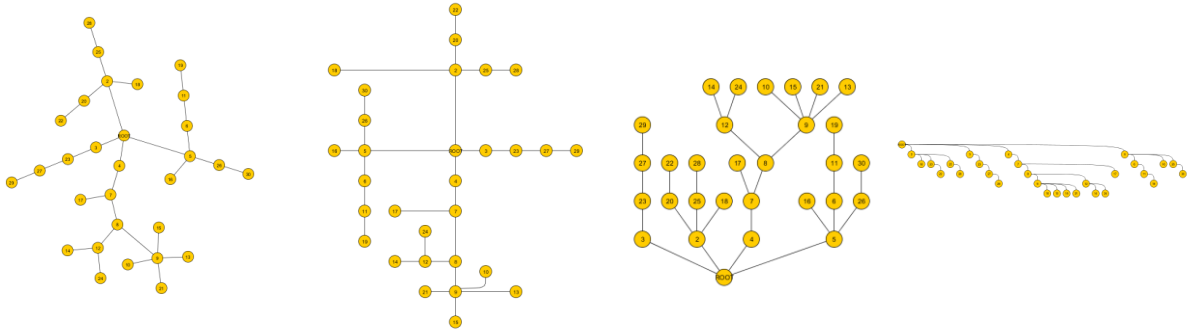
- marks that depict nodes
- visual variables used on marks to depict metadata
- type of links
- visual variables used on marks that depict the links
- placement of nodes



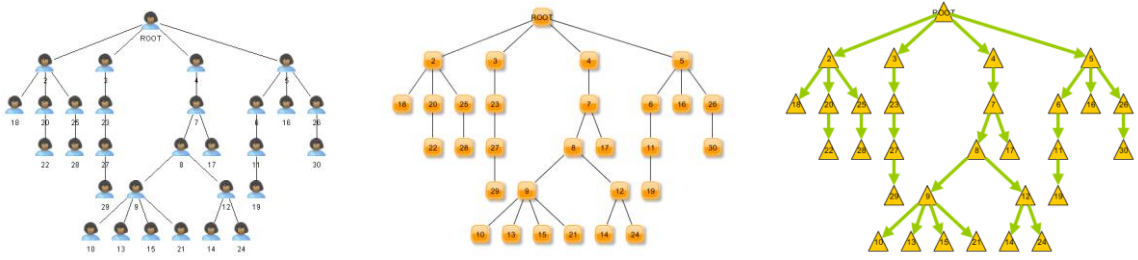
vary edges



node placement



marks



Images created with yEd: http://www.yworks.com

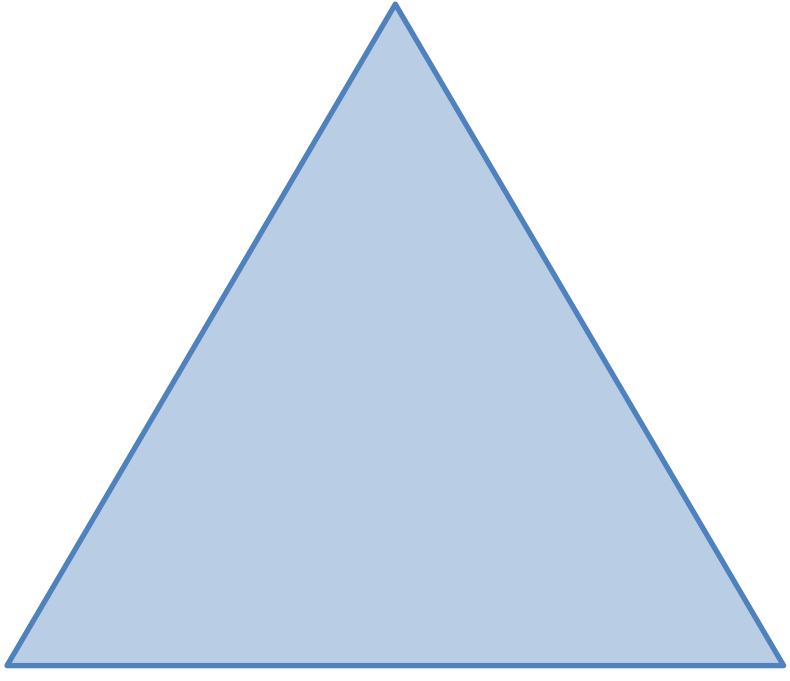
SPECIFIC ALGORITHMS

- usually described recursively
- most well known: Reingold-Tilford algorithm (what we just saw)
- lots of research in this direction:
  - Wetherell and Shannon 1978, Tidy Drawings of Trees
    https://doi.org/10.1109/TSE.1979.234212
  - Reingold and Tilford 1981, Tidier Drawing of Trees
    https://doi.org/10.1109/TSE.1981.234519
  - Walker 1990, A Node-positioning Algorithm for General Trees
    https://doi.org/10.1002/spe.4380200705
  - Buchheim et al. 2002, Improving Walker's Algorithm to Run in Linear Time
    https://doi.org/10.1007/3-540-36151-0_32

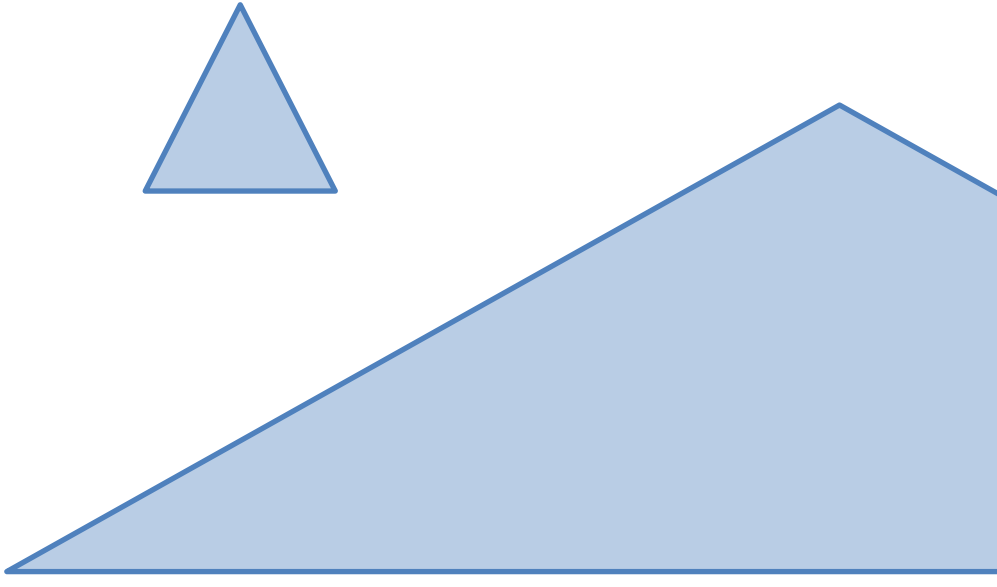
AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being laid out

- area: match area of your layout to the size of the display and data



✓

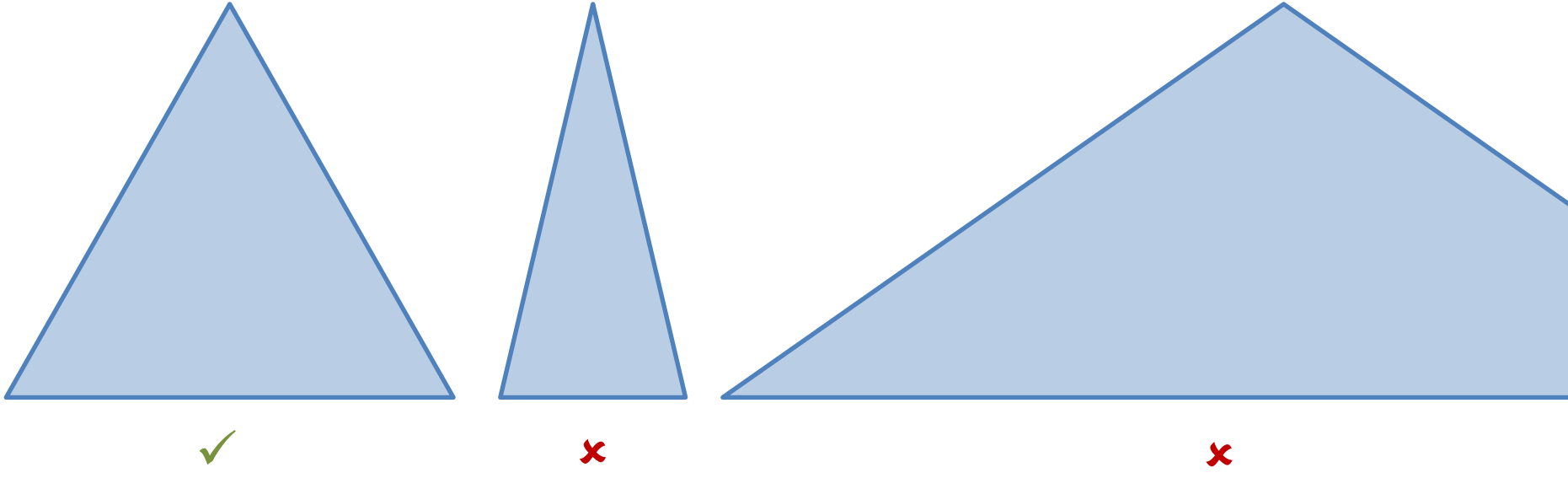


✗

AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being laid out

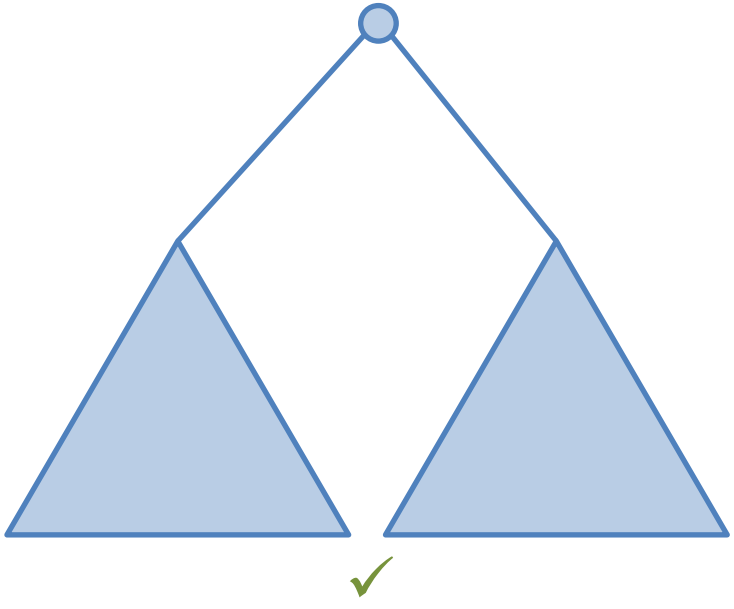
- aspect ratio: usually optimal if close to 1

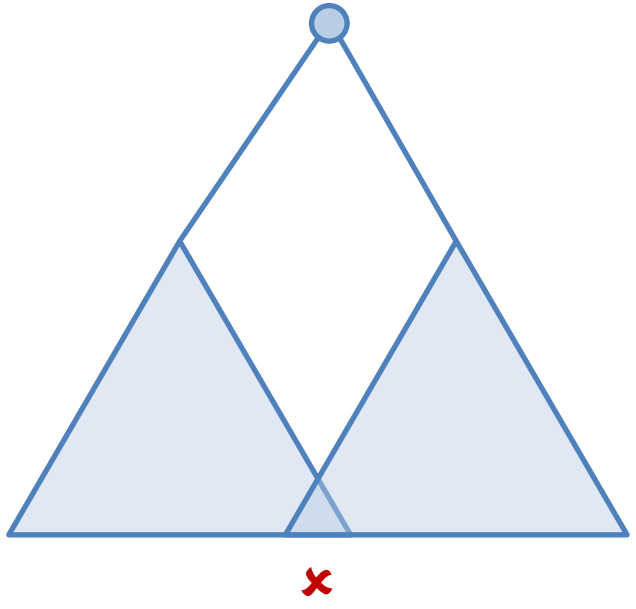


AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being laid out

- subtree separation: try not to overlap subtrees

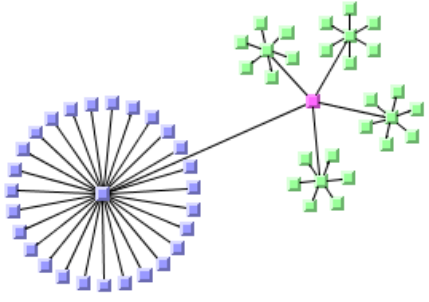




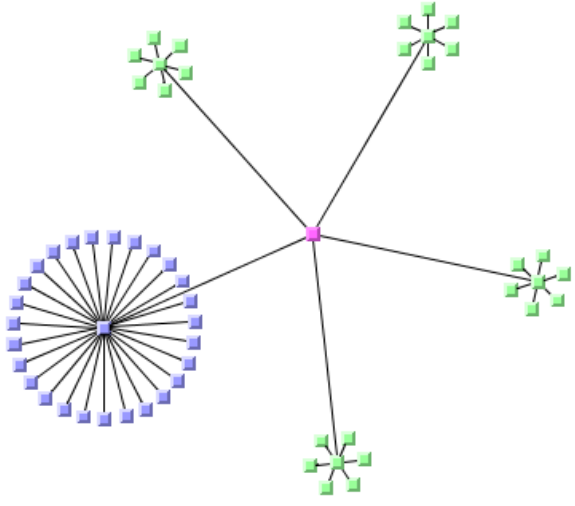
AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being laid out

- root-leaf distance: minimize distance from root to leaves



✓

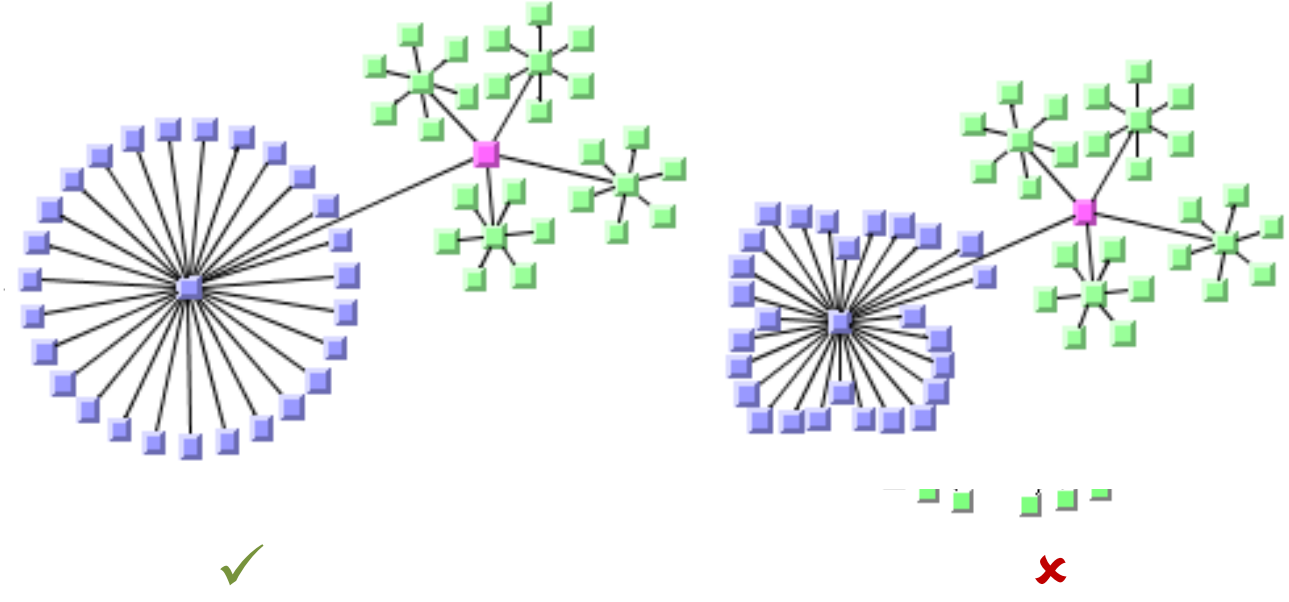


✘

AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being layed out

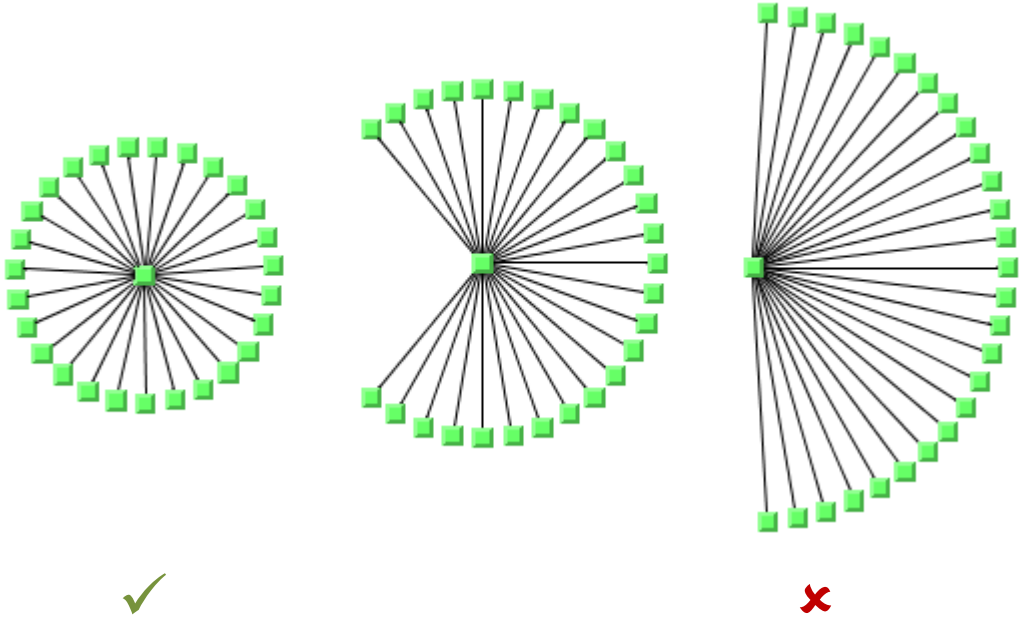
- edge lengths: minimize total, average, maximum, edge lengths & try to make edge lengths uniform



AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being layed out

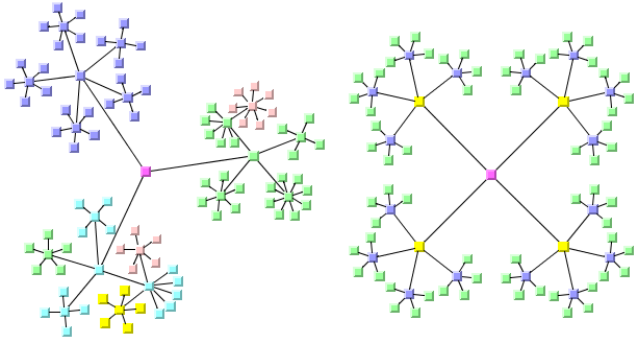
- angular resolution: increase angles formed by edges



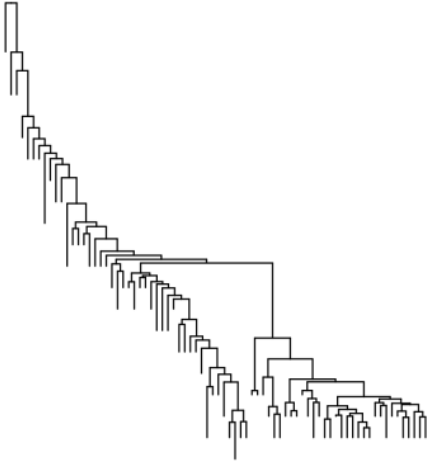
AESTHETICS

aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being laid out

- symmetry: symmetric layouts usually considered pleasing



✓



✘

AESTHETICS

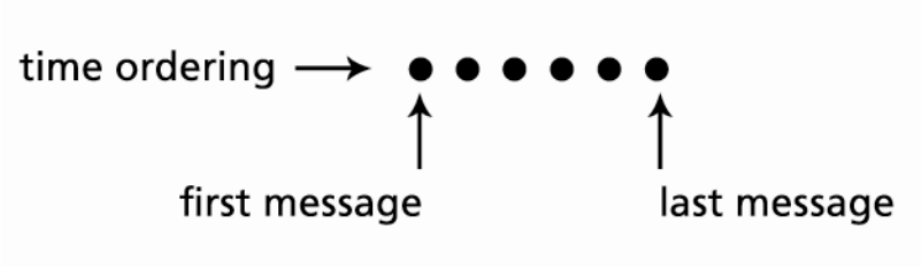
aesthetics of node-link tree algorithms describe properties that improve the perception of the data that is being laid out

- area: match area of your layout to the size of the display and data
- aspect ratio: usually optimal if close to 1
- subtree separation: try not to overlap subtrees
- root-leaf distance: minimize distance from root to leaves
- edge lengths: minimize total, average, maximum, edge lengths & try to make edge lengths uniform
- angular resolution: increase angles formed by edges
- symmetry: symmetric layouts usually considered pleasing

→ **these are guidelines, not laws: they can be broken, but we should know why**

LAYOUT DIMENSIONALITY: 2D – THREAD ARCS

e-mail visualization

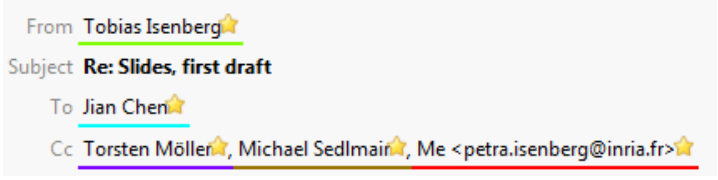


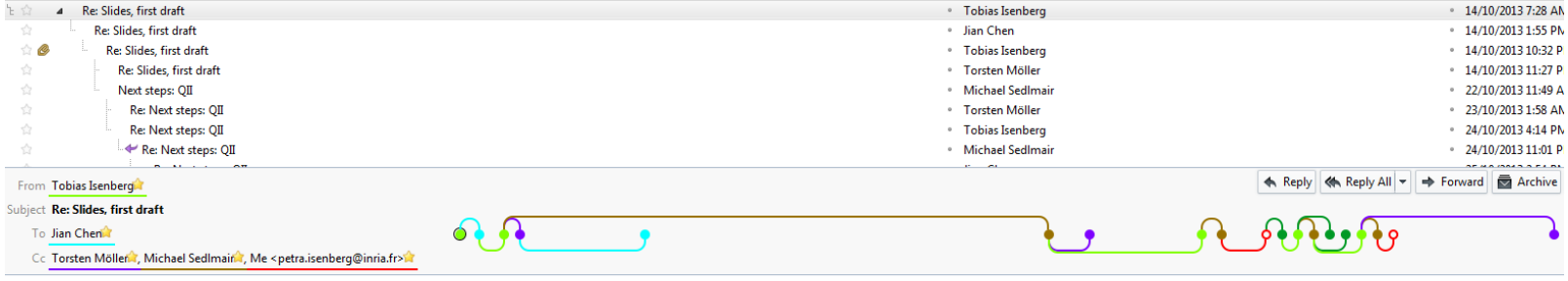
Thread Arcs: An email thread visualization
InfoVis'03: Proceedings of the IEEE Symposium on Information Visualization, pages 211-218.

THREADVIS

http://threadvis.github.io/

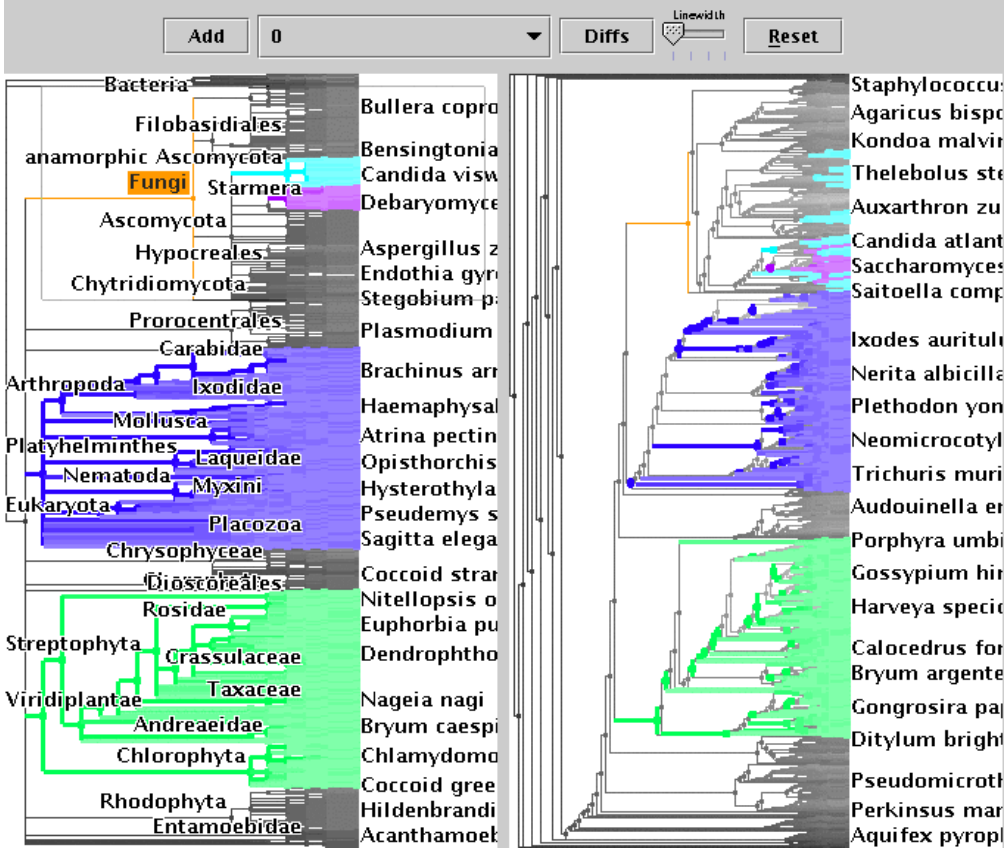
- time-scaling
- coloring people





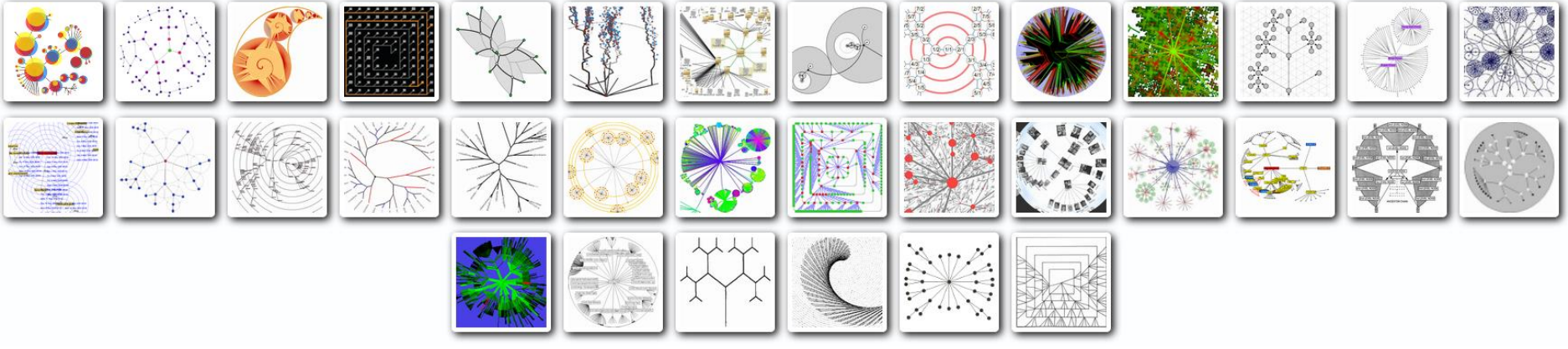
TREEJUXTAPOSER

rectilinear layout and interaction for comparison of very large trees

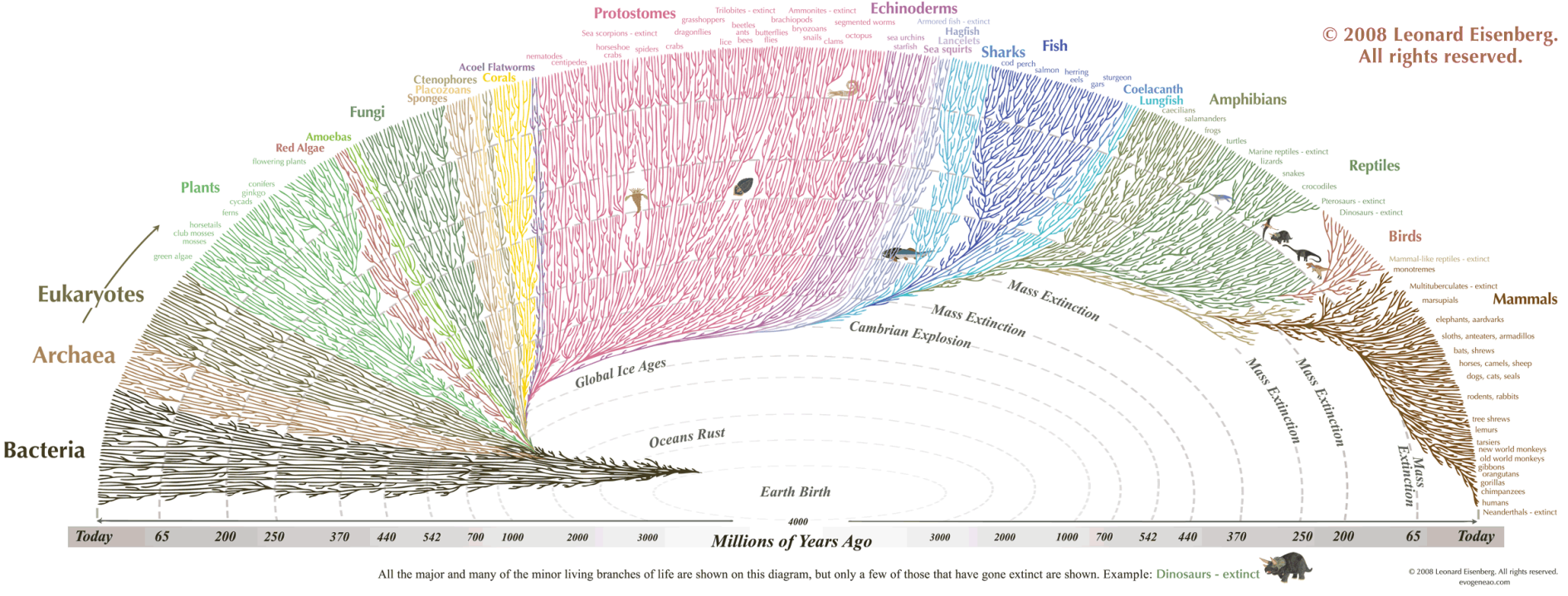


http://www.cs.ubc.ca/~tmm/papers/tj/#video

2D, RADIAL, EXPLICIT EDGES



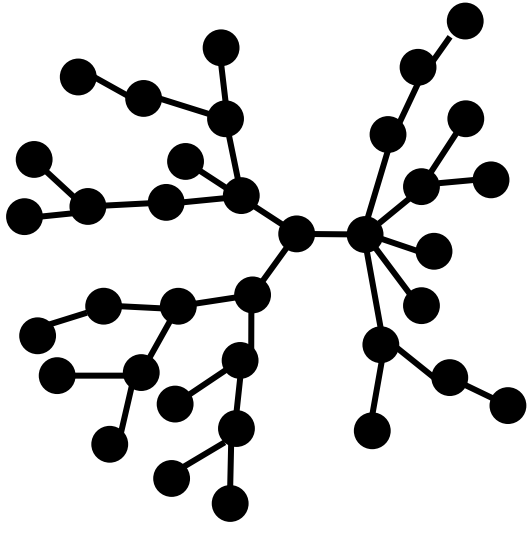
“RADIAL” NODE-LINK



All the major and many of the minor living branches of life are shown on this diagram, but only a few of those that have gone extinct are shown. Example: Dinosaurs - extinct

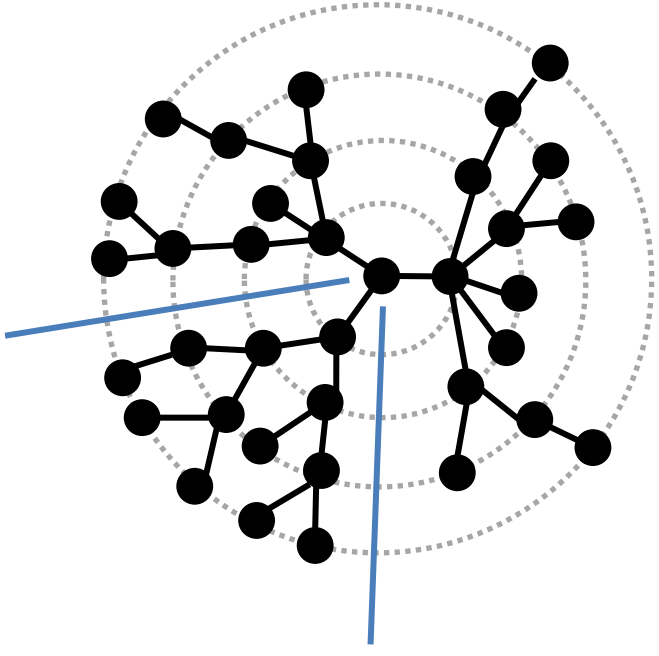
RADIAL NODE-LINK DRAWING

variation of layered drawing from what we saw before



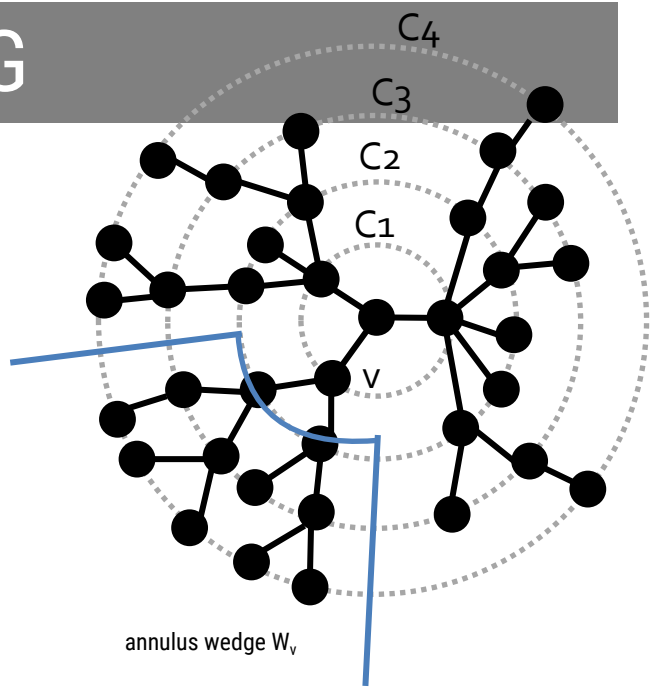
RADIAL NODE-LINK DRAWING

- nodes drawn on concentric circles
- nodes drawn within wedges of the circular layout



RADIAL NODE-LINK DRAWING

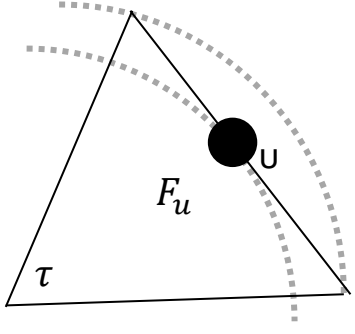
- radius of C_i given by function p(i)
- subtree of v drawn within W_v
- to guarantee planarity (no edge crossings), wedge has to be convex
- several algorithms exist for figuring out the correct angles, e.g.



annulus wedge W_v

$$\beta_u = \min\left(\frac{\ell(u)\beta_v}{\ell(v)}, \tau\right)$$

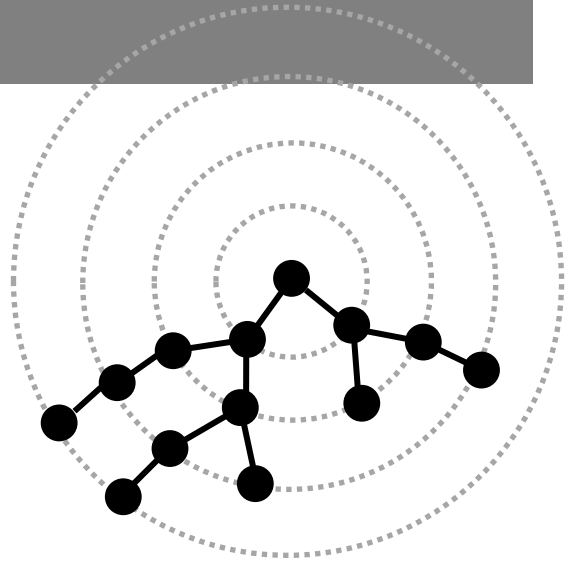
for each child **u** of **v**:
- $\boldsymbol{\beta_u}$ is the angle of $\boldsymbol{W_u}$
- $\boldsymbol{\tau}$ is the angle formed by region $\boldsymbol{F_u}$
- **l(v)** is number of leaves in subtree rooted at **v**
- place **u** at center of $\boldsymbol{W_u}$

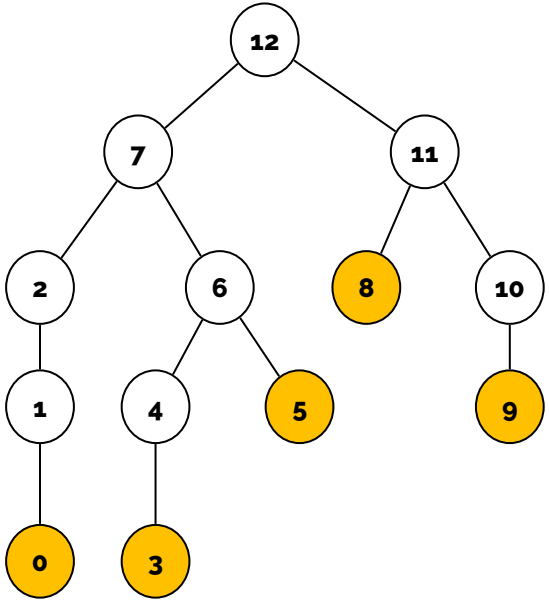


resource: Battista et al. Graph Drawing

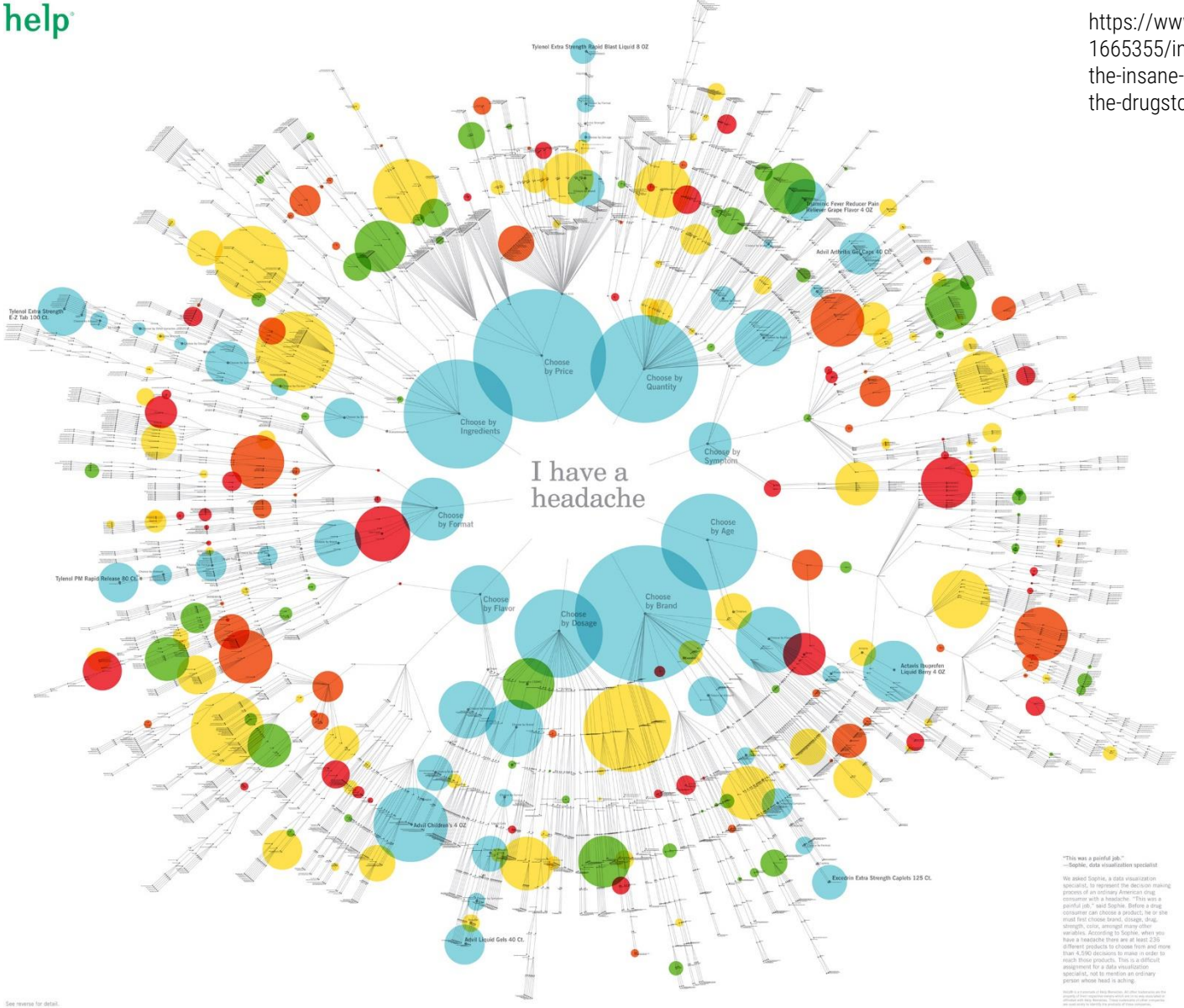
RADIAL NODE-LINK DRAWING

- alternatively, you can adjust the simple node-link diagram approach for angles



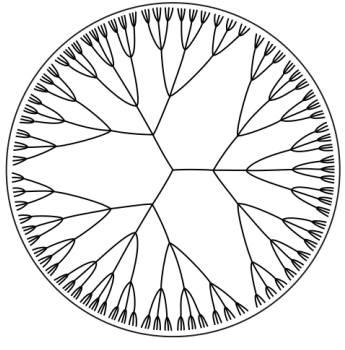


https://www.fastcompany.com/1665355/infographic-of-the-day-the-insane-choices-you-face-at-the-drugstore

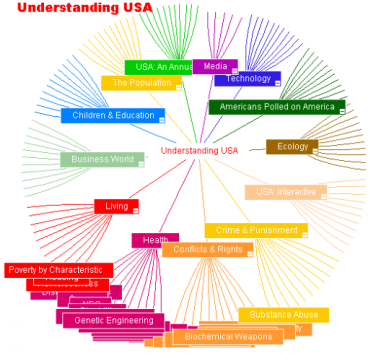


HYPERBOLIC BROWSER

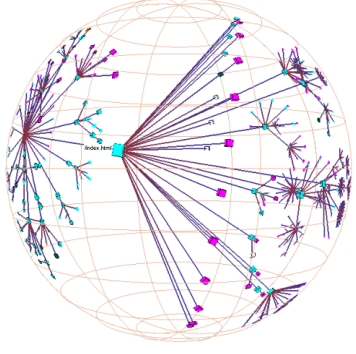
- uses hyperbolic geometry (a non-Euclidean geometry)
- a hyperbolic plane can be displayed using the Poincaré disk model
  - a tree structure of any size fits within a finite area (circle)
  - node is displayed in center
  - all other nodes move away from center and become exponentially smaller



(a) Uniform hyperbolic tree.

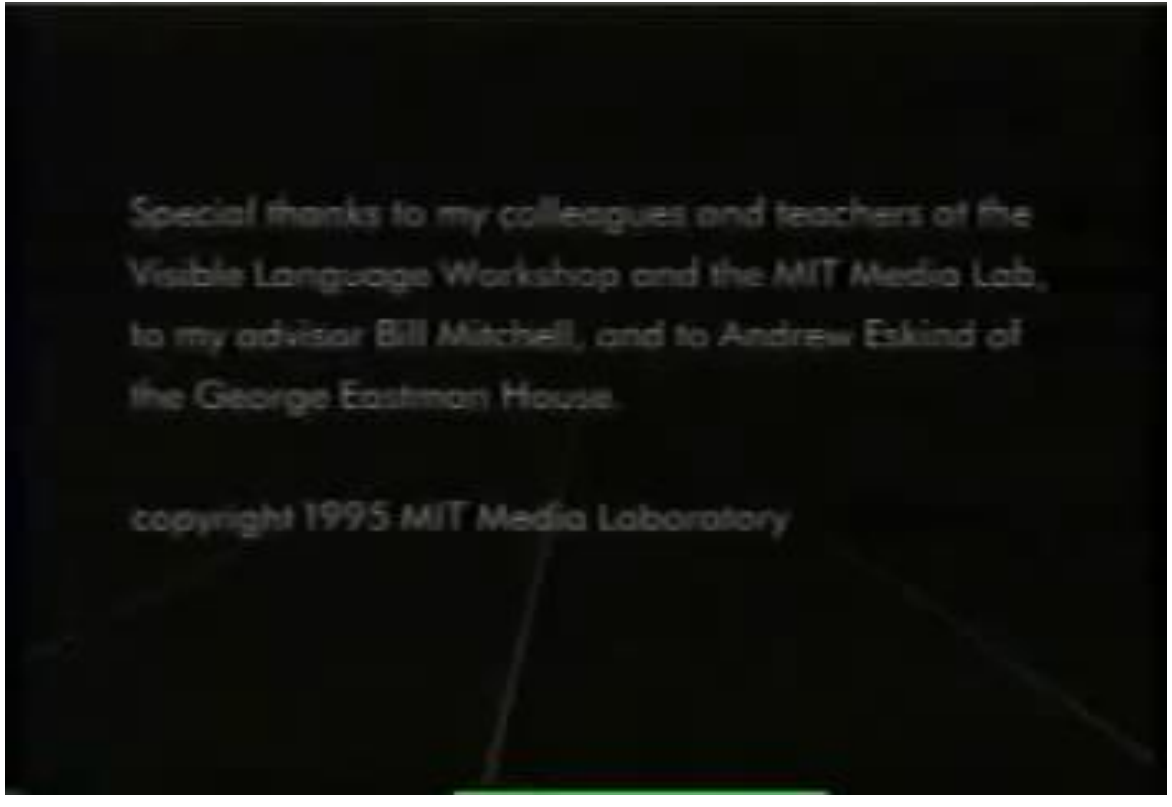


(b) StarTree by Inxight Software.



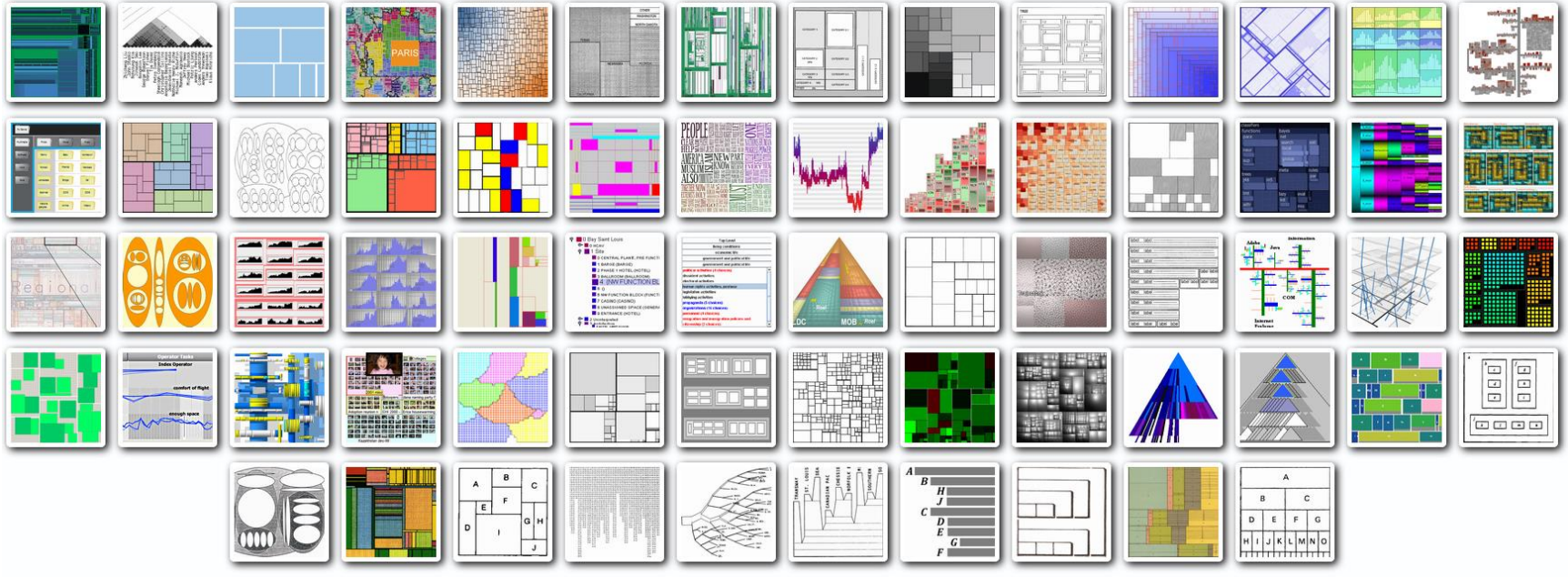
(c) H3 Browser.

CHI 1995 VIDEO OF HYPERBOLIC BROWSER



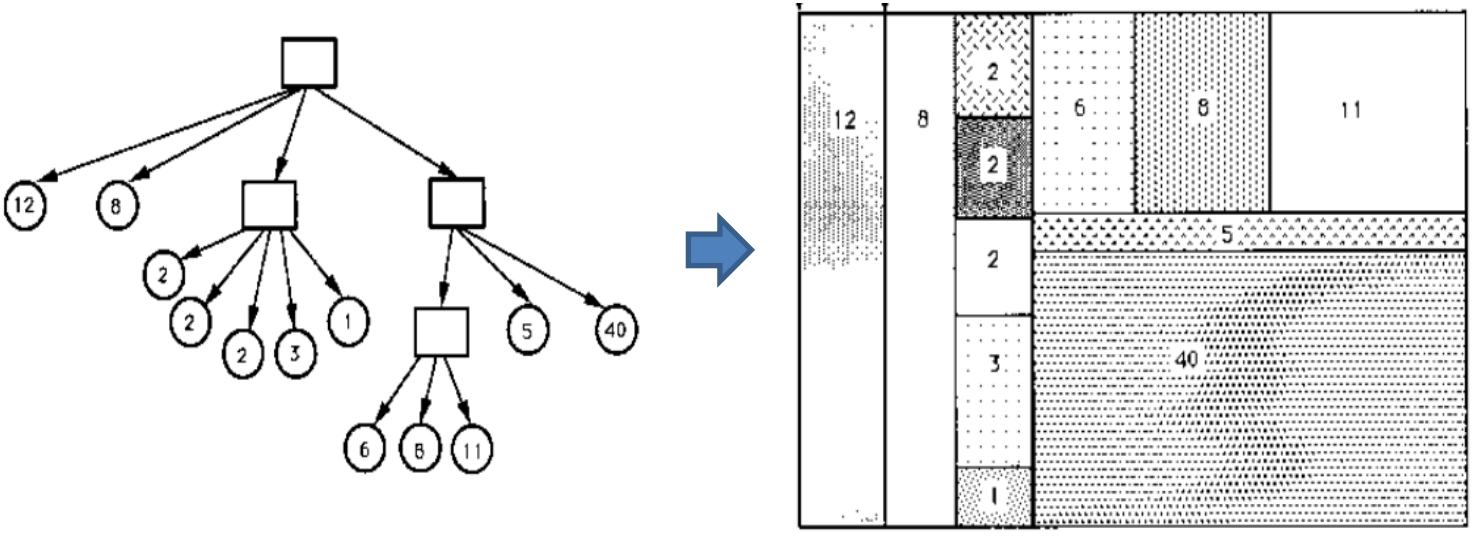
https://www.youtube.com/watch?v=8bhq08BQLDs

2D, AXIS-PARALLEL, IMPLICIT EDGES



A CLASSIC CONTAINMENT LAYOUT

- example tree to rebuild with **treemap** algorithm
- size of each node as numbers in leaves



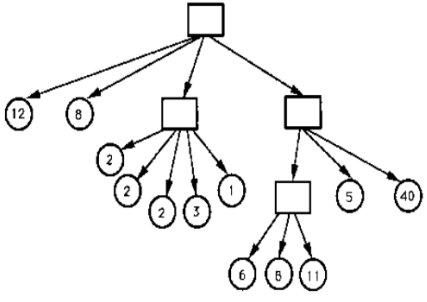
TREEMAP ALGORITHM

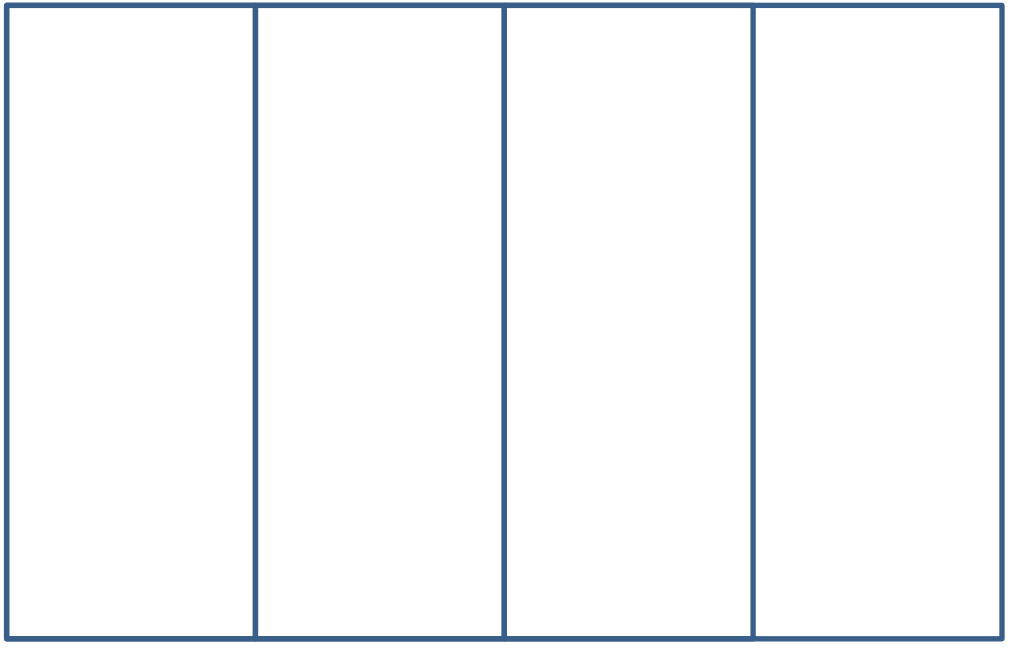
- take a rectangular display area $P_1(x_1,y_1)$, $Q_1(x_2,y_2)$
- this area represents the root of the tree

TREEMAP ALGORITHM

recursive algorithm

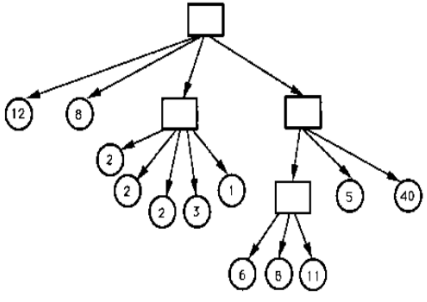
- the number of children of the current node define the number of partitions of the current node

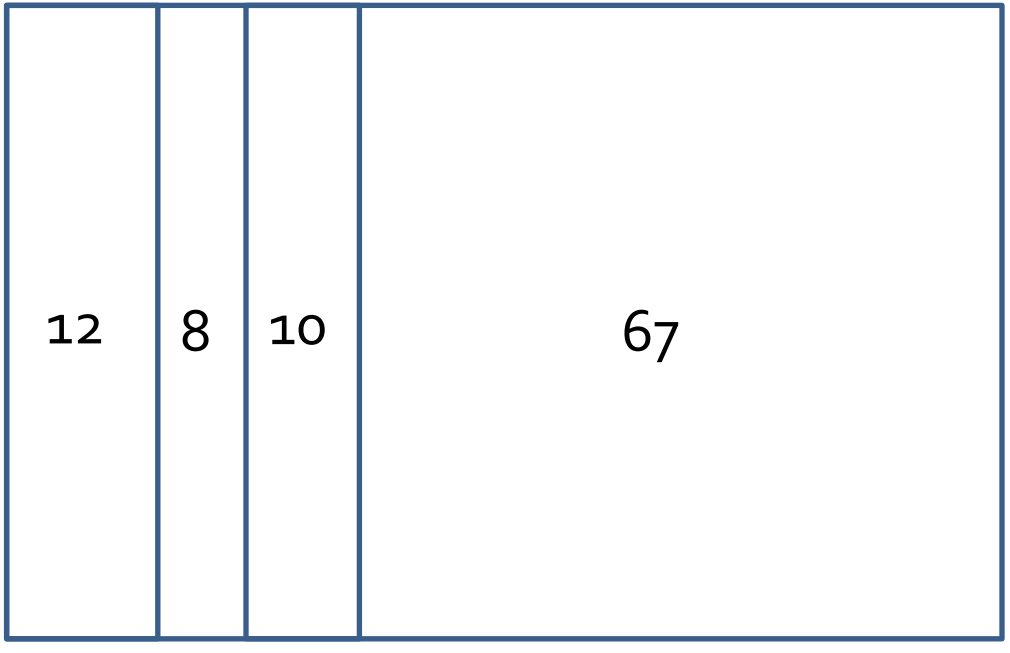




TREEMAP ALGORITHM

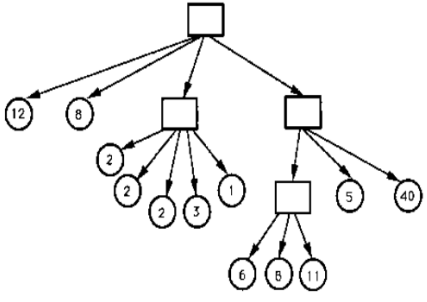
the **weight** of each node determines the size of each partition

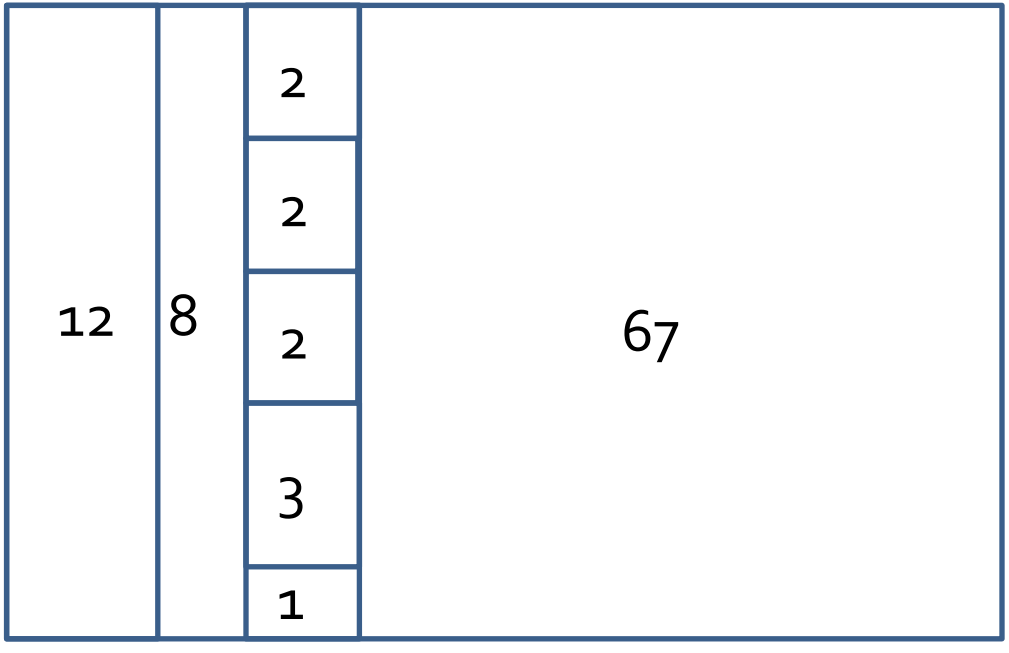




TREEMAP ALGORITHM

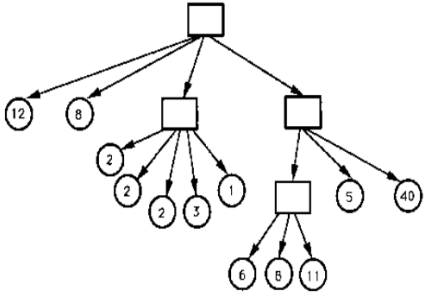
at each change of level, **rotate orientation** of split by 90 degrees

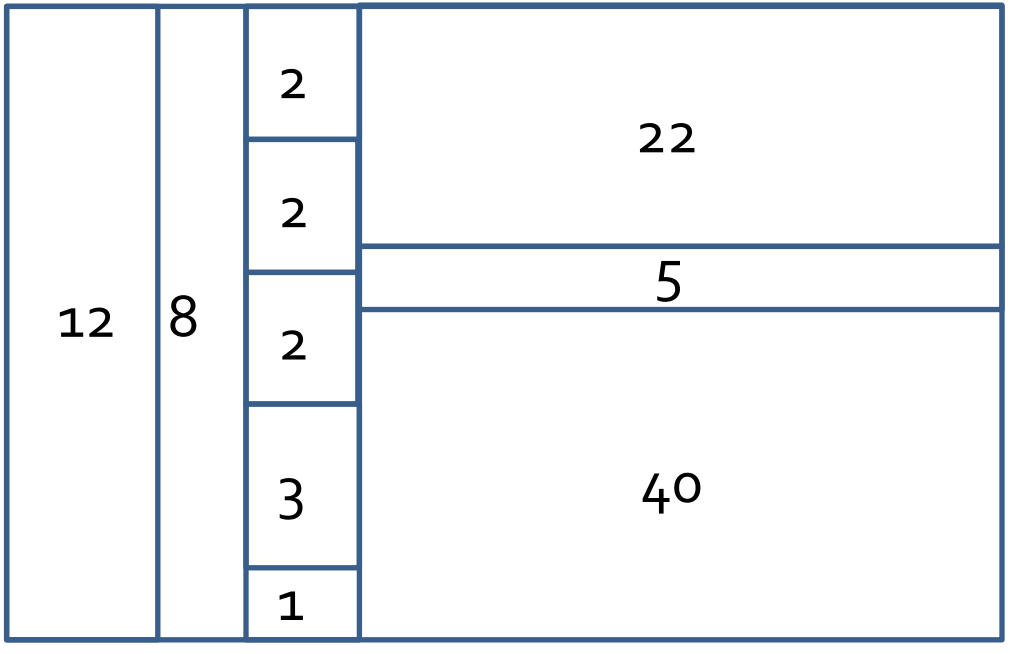




TREEMAP ALGORITHM

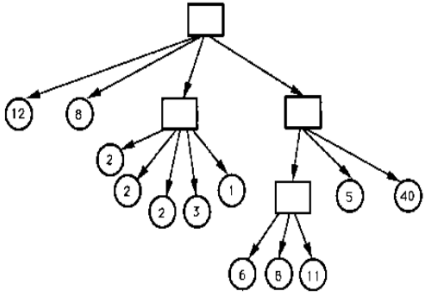
at each change of level, **rotate orientation** of split by 90 degrees

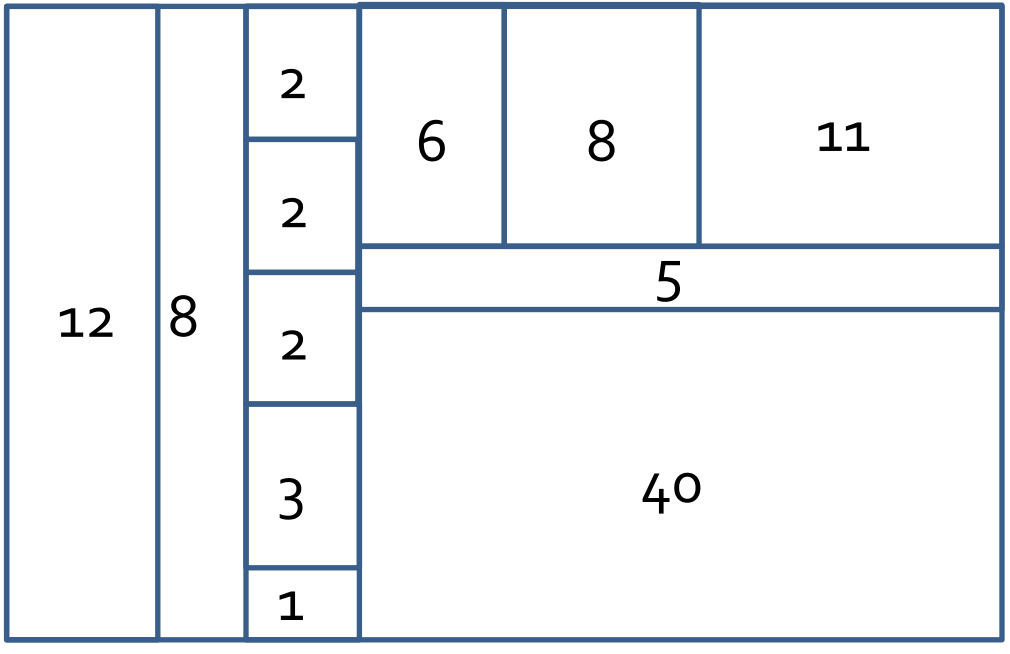




TREEMAP ALGORITHM

at each change of level, **rotate orientation** of split by 90 degrees



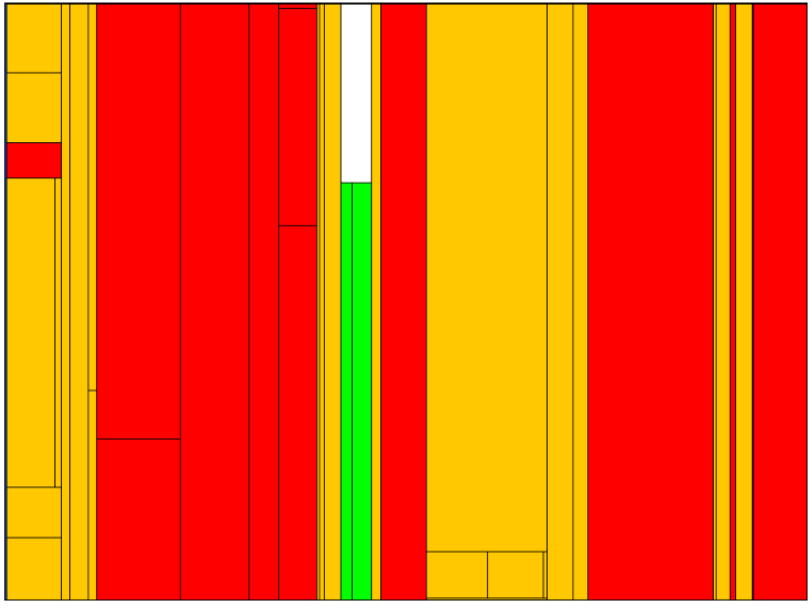


TREEMAP

- a 2-D space-filling layout
- for further references and to try out a treemap in various applications: http://www.cs.umd.edu/hcil/treemap-history/

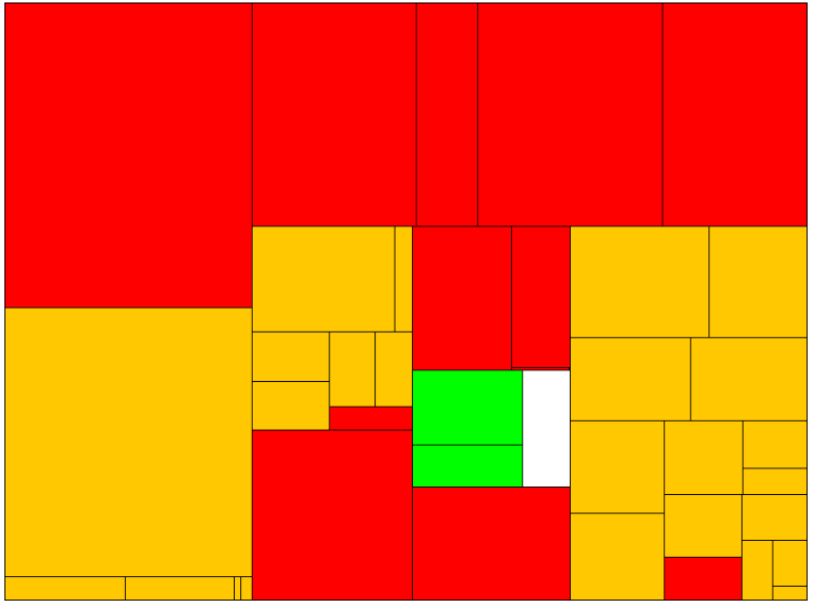
TREEMAP VARIATIONS

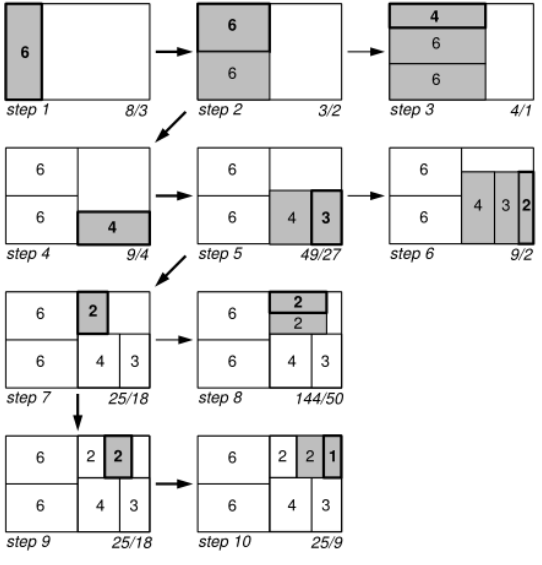
- problem with original treemap: lots of long stripes
- for long stripes the areas are difficult to compare



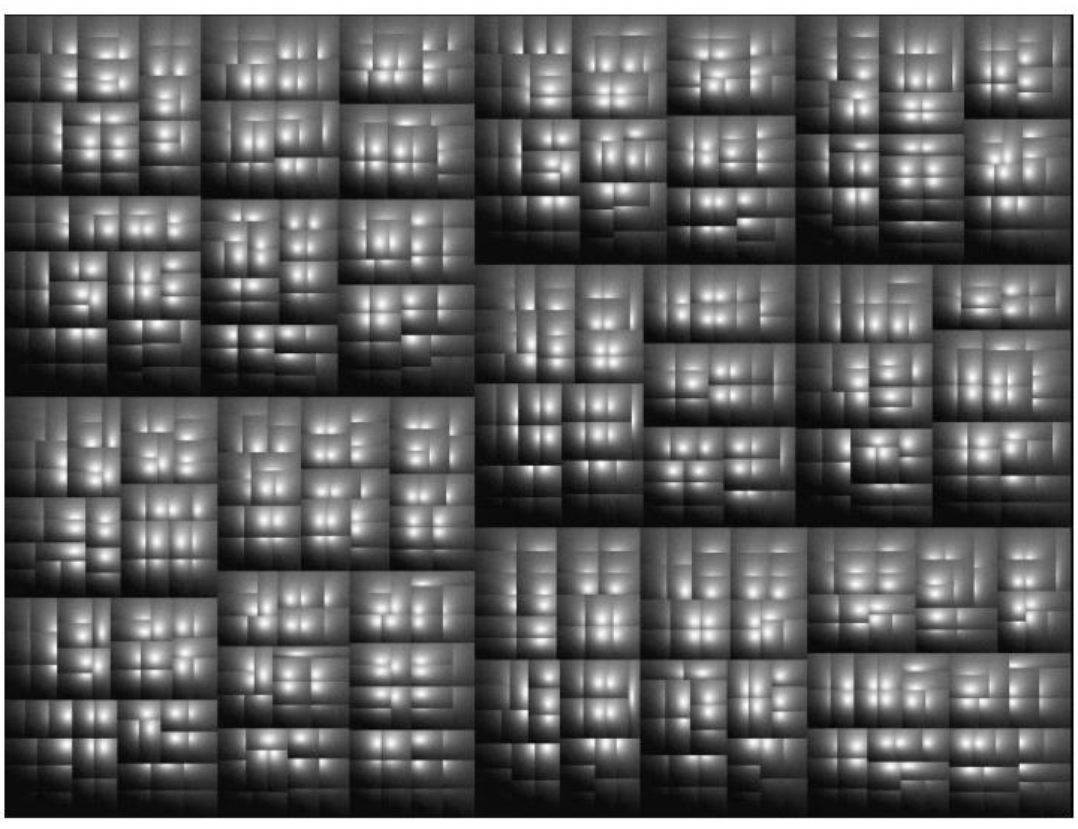
SQUARIFIED TREEMAP

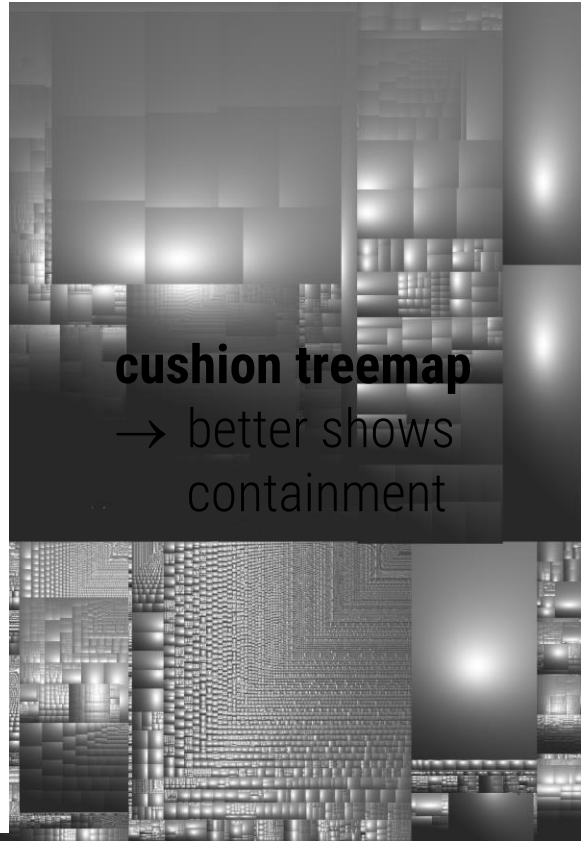
- calculates more squared regions
- problem: order not as easily read,
  not very stable with dynamically changing data





SQUARIFIED TREEMAP

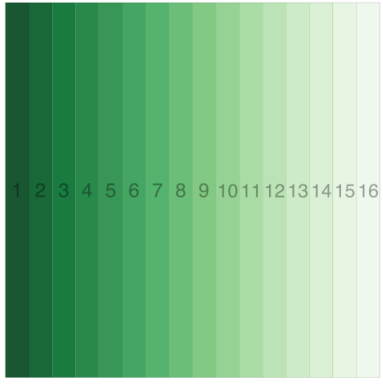




https://www.win.tue.nl/sequoiaview/

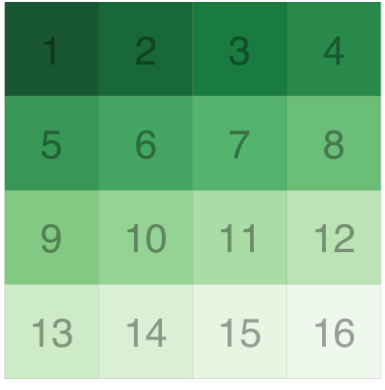
ORDERED TREEMAP

several algorithms in comparison



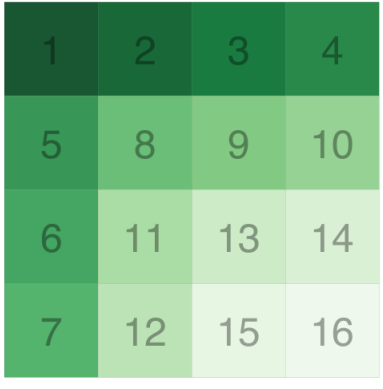
slice and dice

B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. ACM Transactions on Graphics, 11:92–99, 1992.



strip

B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. ACM Transactions on Graphics, 21:833–854, 2002.



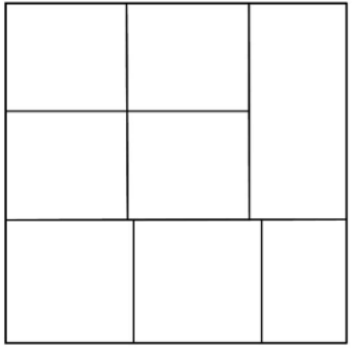
squarified

M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. EuroVis, pages 33–42, 2000.

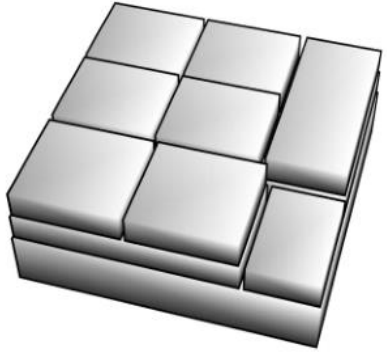
ordered squarified

B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In Infovis01, pages 73–78, 2001.

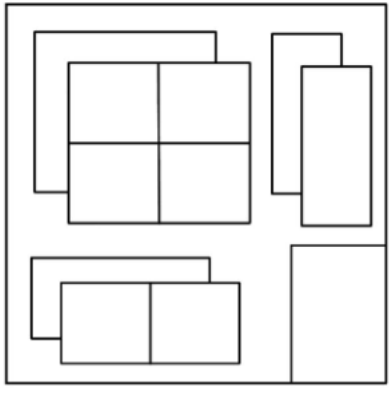
OTHER VARIATIONS OF TREEMAPS



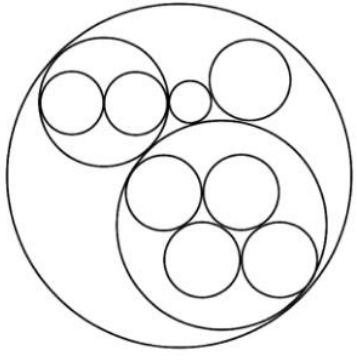
original squarified:
emphasizes leafs and their attributes



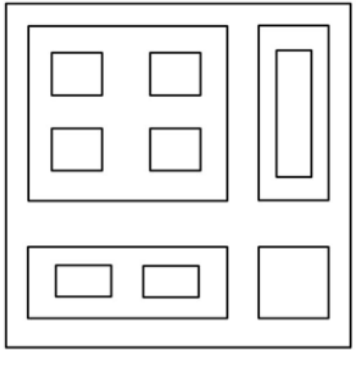
steptree:
emphasizes structure with extrusion



cascaded layout:
emphasizes structure with overlap



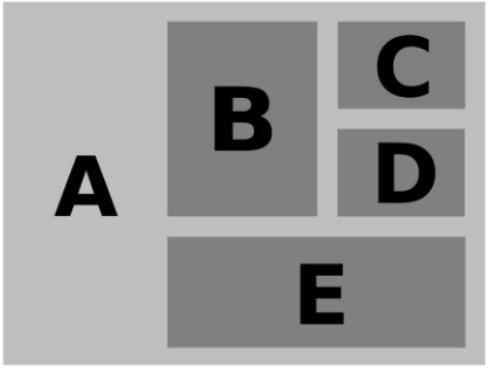
circular treemap:
emphasizes structure with
non-space-filling primitive



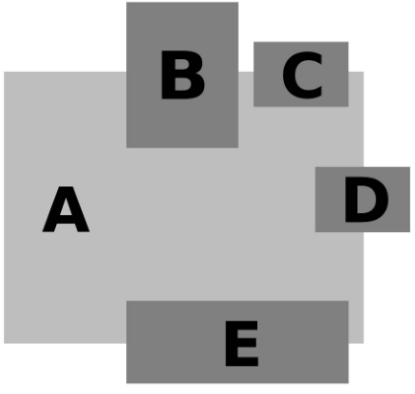
nested layout:
emphasizes structure with whitespace

TVCG, 17(4), The Design Space of Implicit Hierarchy Visualization: A Survey
https://doi.org/10.1109/TVCG.2010.79

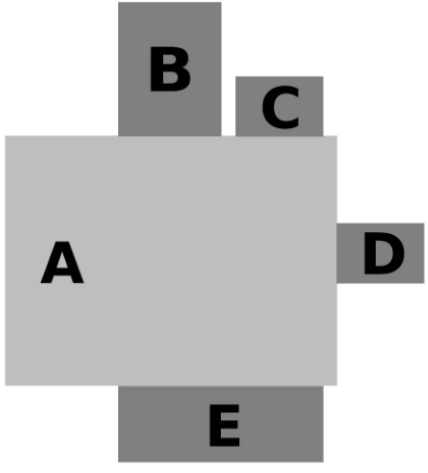
OTHER IMPLICIT TECHNIQUES FOR SHOWING HIERARCHICAL RELATIONSHIPS



nesting



overlap

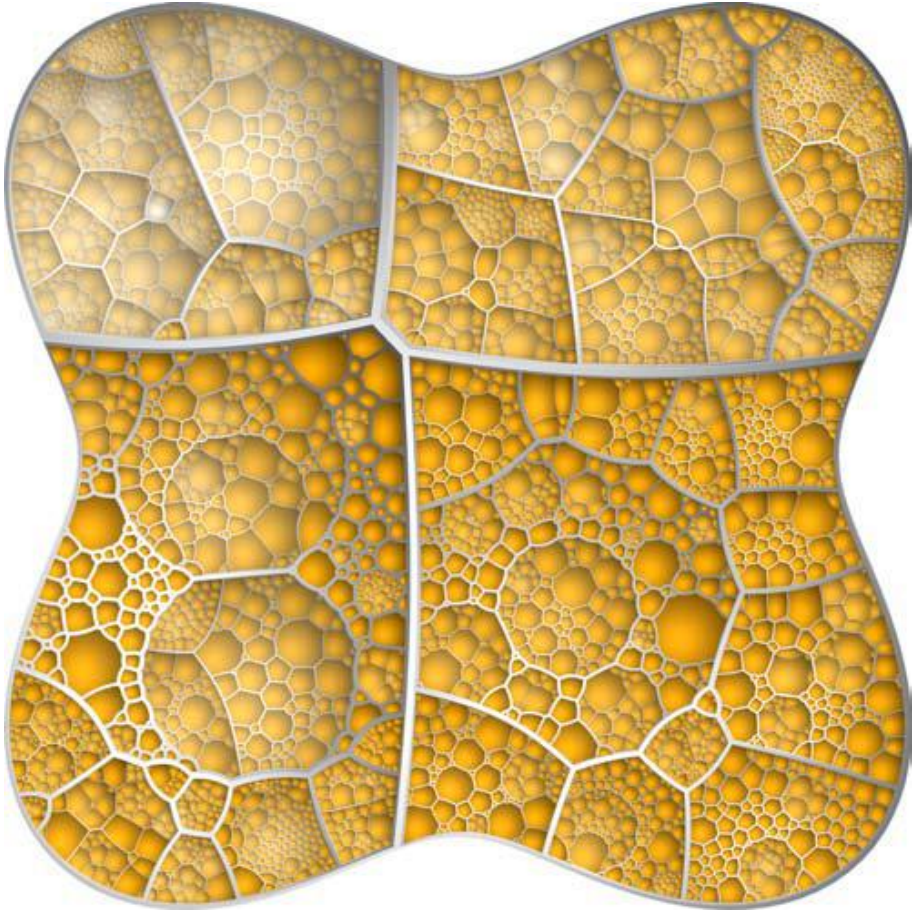


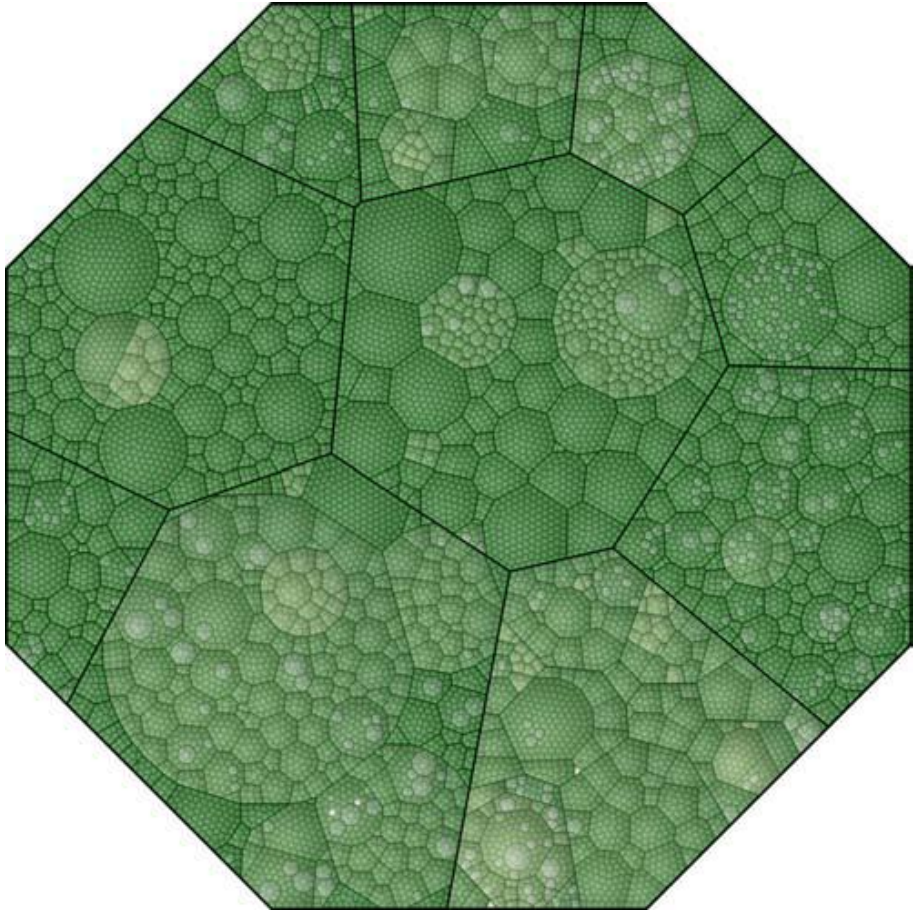
adjacency

HTTP://NEWSMAP.JP/

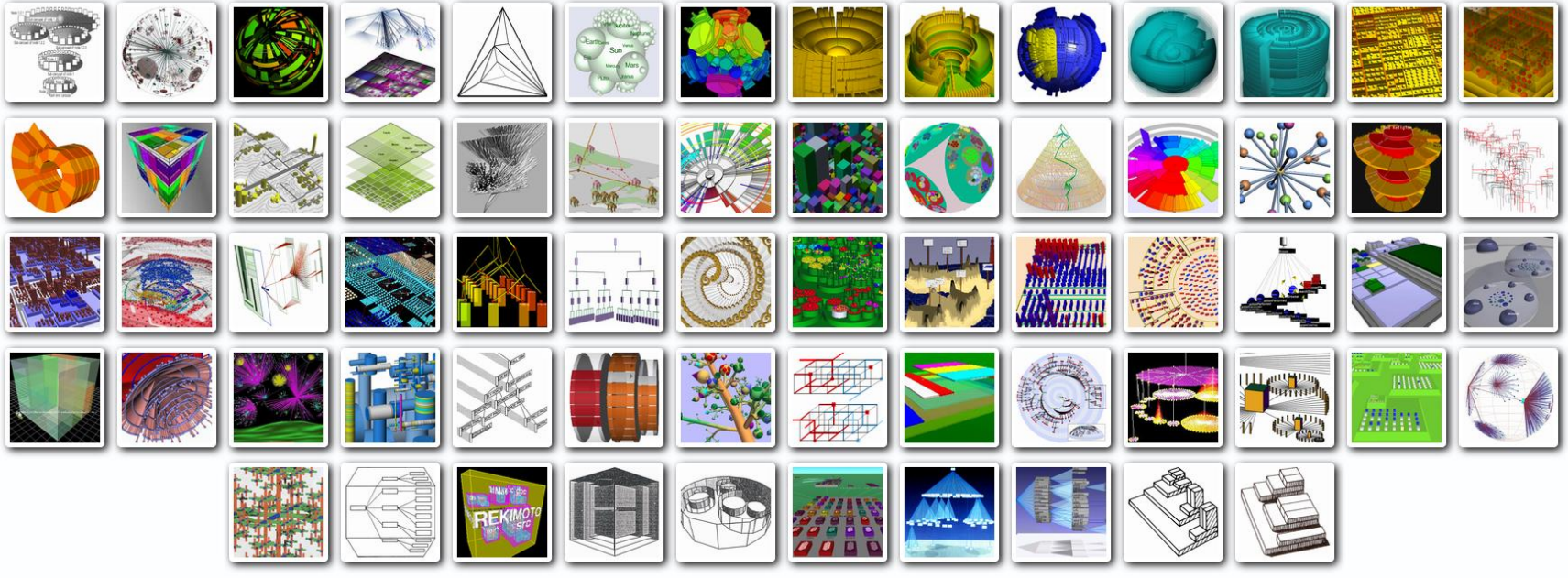
(unfortunately requires Flash)

OTHER TREEMAP VARIATIONS

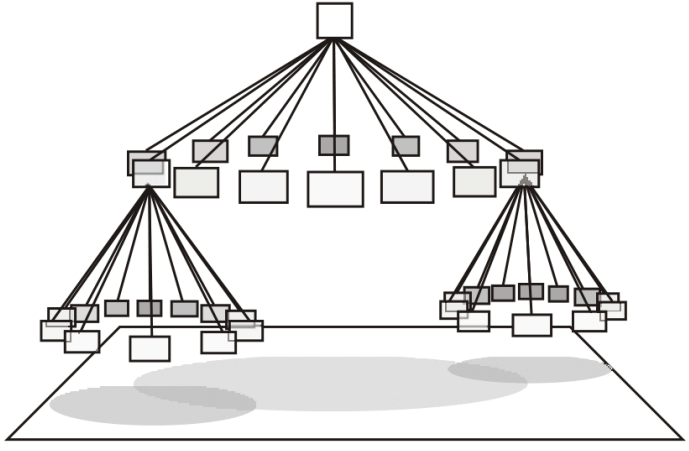




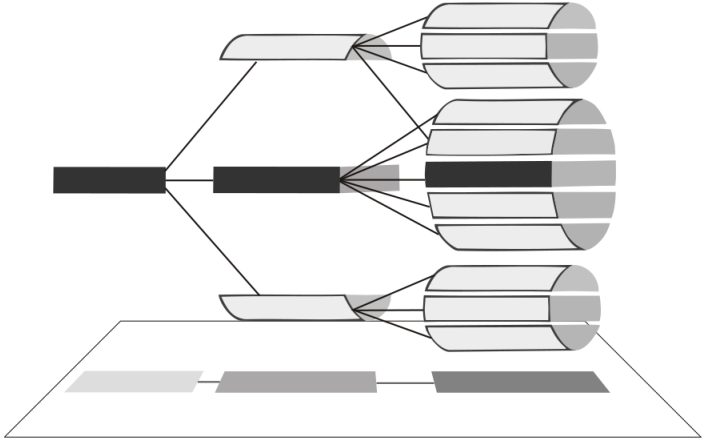
3D LAYOUTS



HISTORIC EXAMPLE: CONETREE / CAMTREE



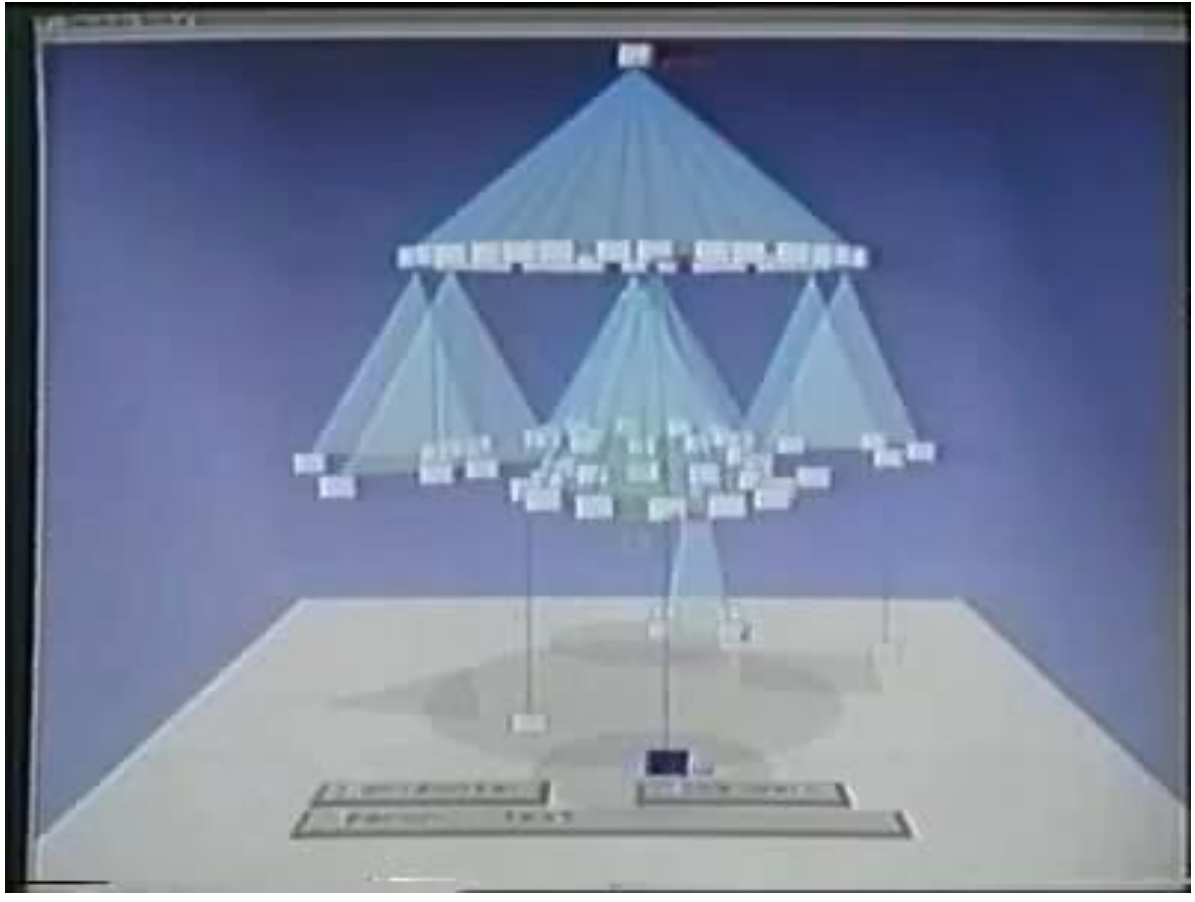
ConeTree



CamTree

[Robertson et al., 1991] George Robertson, Jock D. Mackinlay, Stuart Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of the ACM CHI 91* https://doi.org/10.1145/108844.108883

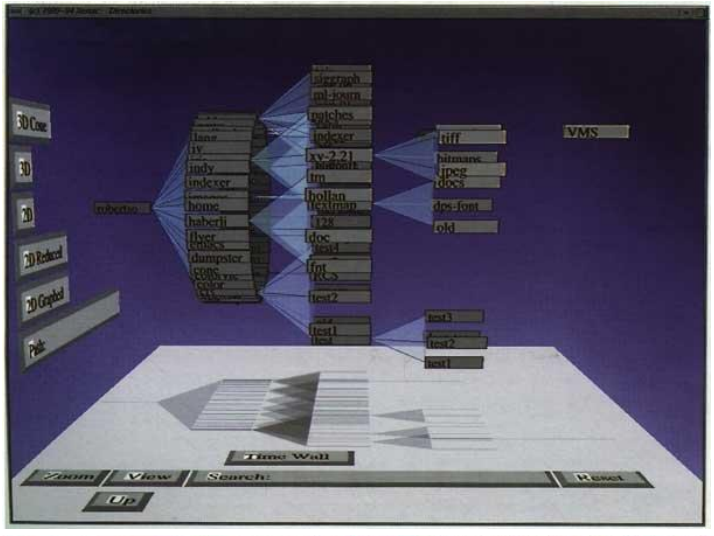
CONETREE

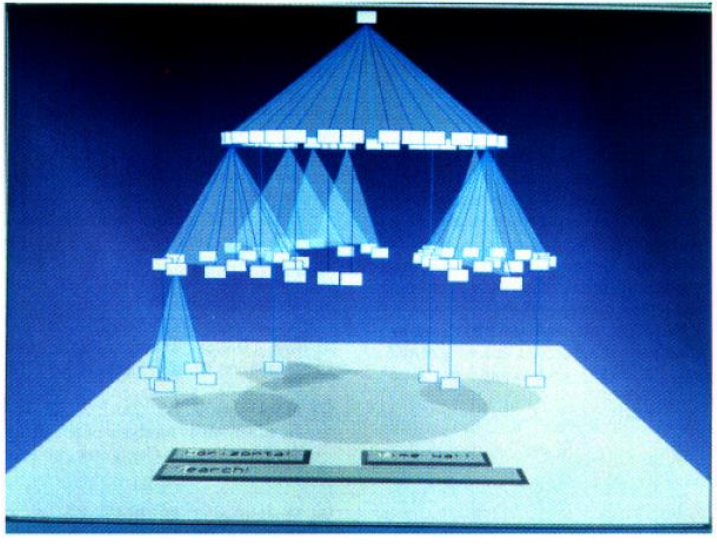


[Robertson et al., 1991] George Robertson, Jock D. Mackinlay, Stuart Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of the ACM CHI 91* https://doi.org/10.1145/108844.108883

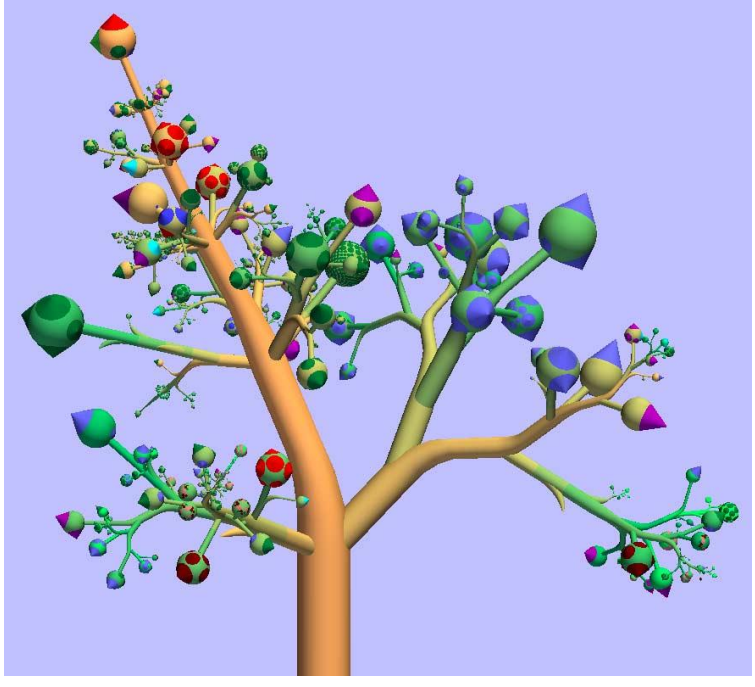
CONE/CAMTREE

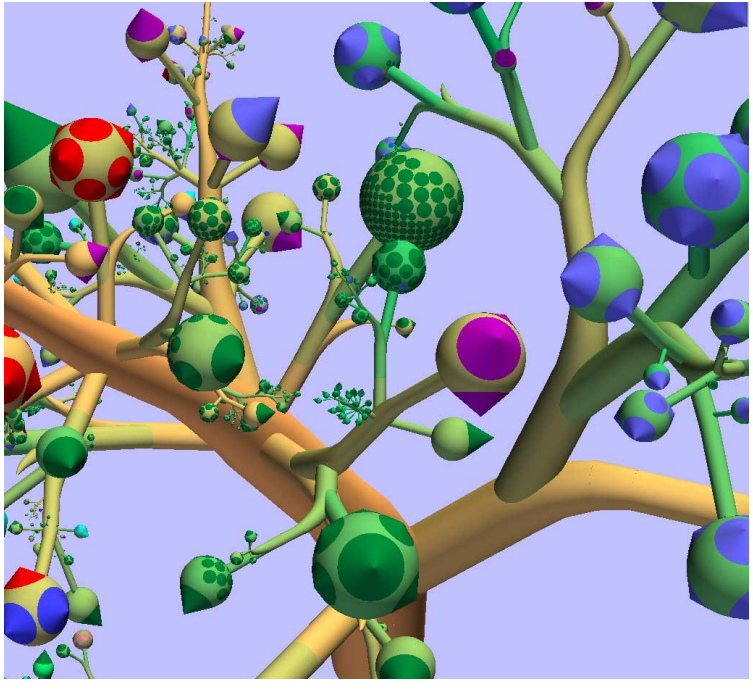
- children of a node are laid out in a cylinder “below” the parent
- siblings located on the same 2D circle
- use of animation
- shadows to enhance structure





BOTANICAL VISUALIZATION OF HUGE HIERARCHIES





Visualization of Unix home directory – Kleiberg et al., Proc. InfoVis 2001; https://doi.org/10.1109/INFVIS.2001.963285

3D LAYOUTS

- advantages
  - fit more data into same aspect ratio
  - can be aesthetically pleasing
  - can be meaningful for specific domains
- disadvantages
  - occlusion
  - requires interaction or animation
  - no overviews

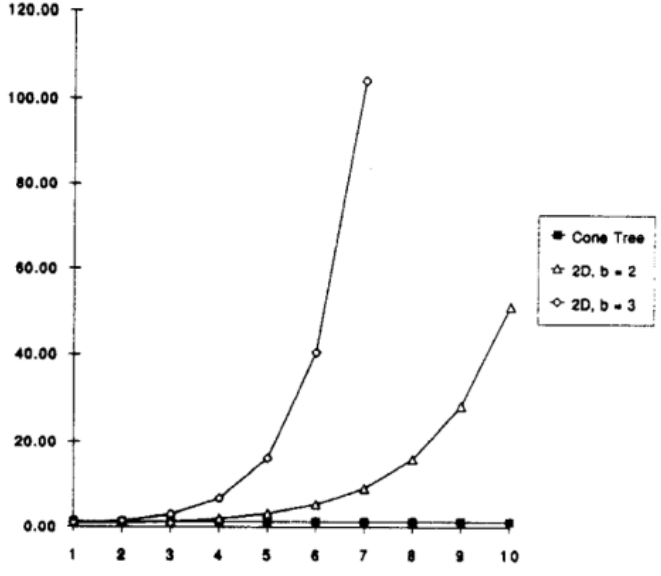


Figure 1: Aspect Ratio of 2D and 3D Trees.

TREE VISUALIZATION SUMMARY

- there are lots of tree visualizations
  - there is also lots of free software, try it out (see links earlier in the lecture)
  - there are a few overview articles, e.g.:
    - A Generative Layout Approach for Rooted Tree Drawings by Hans-Jörg Schulz, Zabed Akbar, and Frank Maurer; IEEE PacificVis 2013 https://doi.org/10.1109/PacificVis.2013.6596149
    - The Design Space of Implicit Hierarchy Visualization: A Survey by Hans-Jörg Schulz, Steffen Hadlak, and Heidrun Schumann; IEEE TVCG 17(4) https://doi.org/10.1109/TVCG.2010.79
  - also see the interactive overview at https://treevis.net/

TREE VISUALIZATION SUMMARY

- can be categorized by
  - edge representations (implicit, explicit)
  - dimensionality of layout
  - radial vs. axis-parallel

- can be modified by
  - layout parameters
  - which marks are used
  - visual variables on marks (which meta-data is represented?)

GRAPHS / NETWORKS

DEFINITION GRAPH

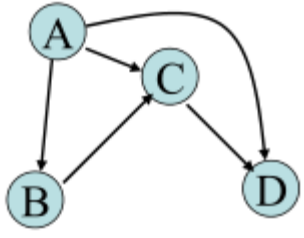
- a set of vertices $V = \{v_i\}$
- a set of edges $E = \{e_{ij}\}$ with $e_{ij} = \{v_i, v_j\}$
- when the order of vertices of an edge is meaningful, the graph is directed

GRAPH MEASURES

- SIZE = # of nodes
- DENSITY = edges/vertices (roughly)
- PATH = sequence of edges
  connecting (different) vertices
- VERTEX DEGREE = # of edge connections
- DISTANCE = # of hops between vertices

TWO CLASSICAL VISUAL REPRESENTATIONS

node-link diagram

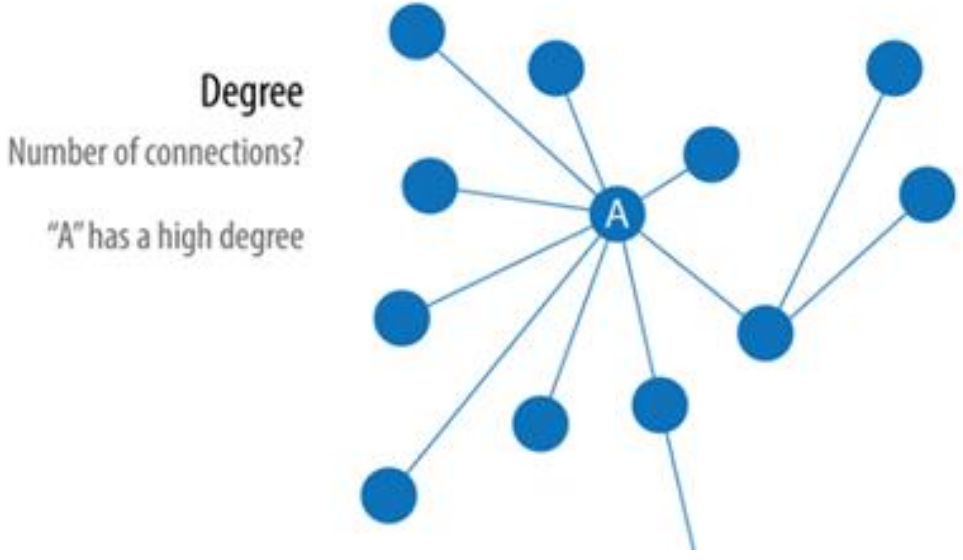


adjacency matrix

↗	A	B	C	D
A		X	X	X
B			X	
C				X
D				

TASKS

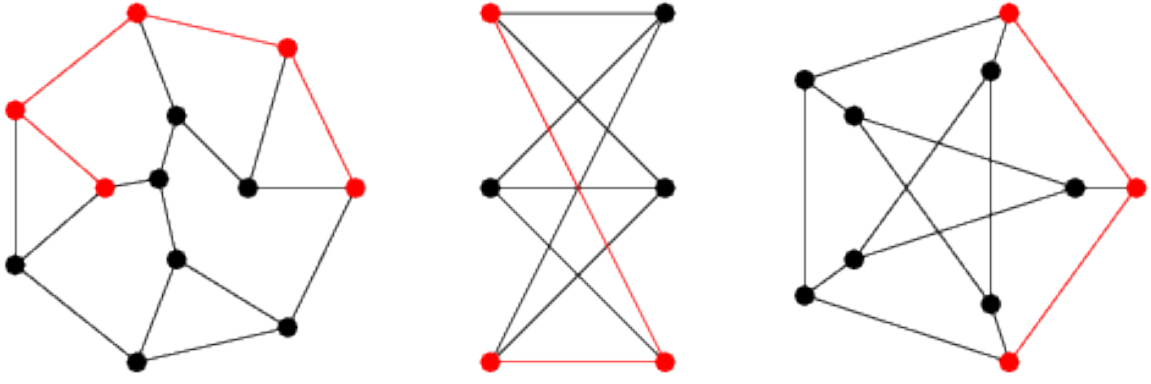
find # of neighbors of a vertex



https://combattb.org/combat-tb-neodb/images/centalityalgos.png

TASKS

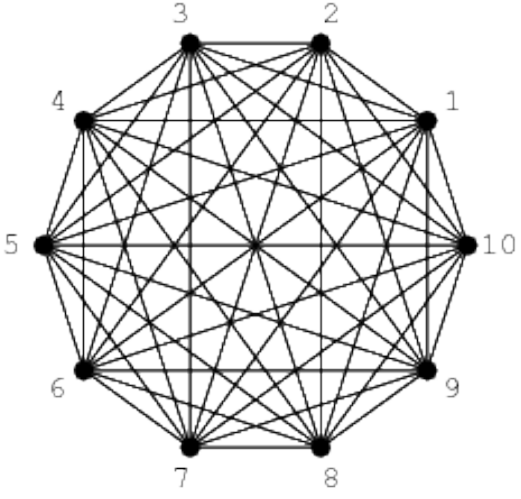
see paths (overviews, shortest, cycles)

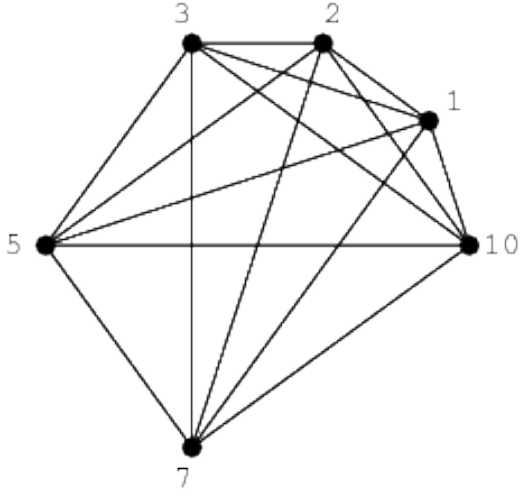


http://mathworld.wolfram.com/GraphGeodesic.html

TASKS

identify sub-graphs





http://mathworld.wolfram.com/Vertex-InducedSubgraph.html

TASKS

HIGHER-LEVEL

involves many elements
involves more human judgment

- Which nodes are important?
- Where are clusters?
- What are attribute and connection correlations?
- How does the network change over time?

TASKS

- many, many more specific tasks
- each application domain will add more

Task Taxonomy for Graph Visualization

Bongshin Lee, Catherine Plaisant,
Cynthia Sims Parr
Human-Computer Interaction Lab
University of Maryland,
College Park, MD 20742, USA
+1-301-405-7445
{bongshin, plaisant, csparr}@cs.umd.edu

Jean-Daniel Fekete,
Nathalie Henry
INRIA Futurs/LRI Bat. 490
Université Paris-Sud,
91405 ORSAY, France
+33-1-69153460
Jean-Daniel.Fekete@inria.fr, nhenry@lri.fr

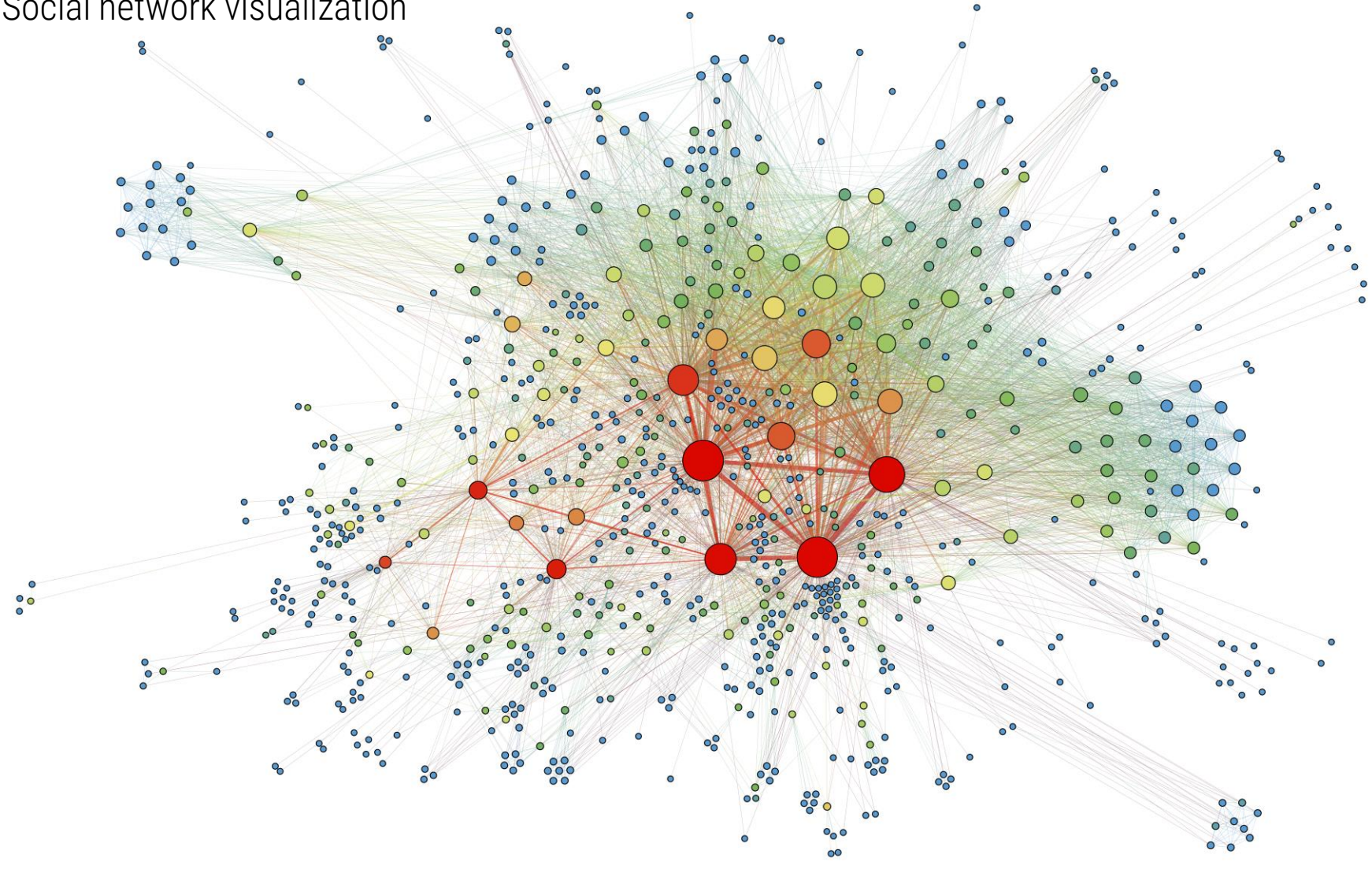
ABSTRACT

Our goal is to define a list of tasks for graph visualization that has enough detail and specificity to be useful to: 1) designers who want to improve their system and 2) to evaluators who want to compare graph visualization systems. In this paper, we suggest a list of tasks we believe are commonly encountered while analyzing graph data. We define graph specific objects and demonstrate how all complex tasks could be seen as a series of low-level tasks performed on those objects. We believe that our user studies of graph visualization techniques and extracted the tasks used in those studies.

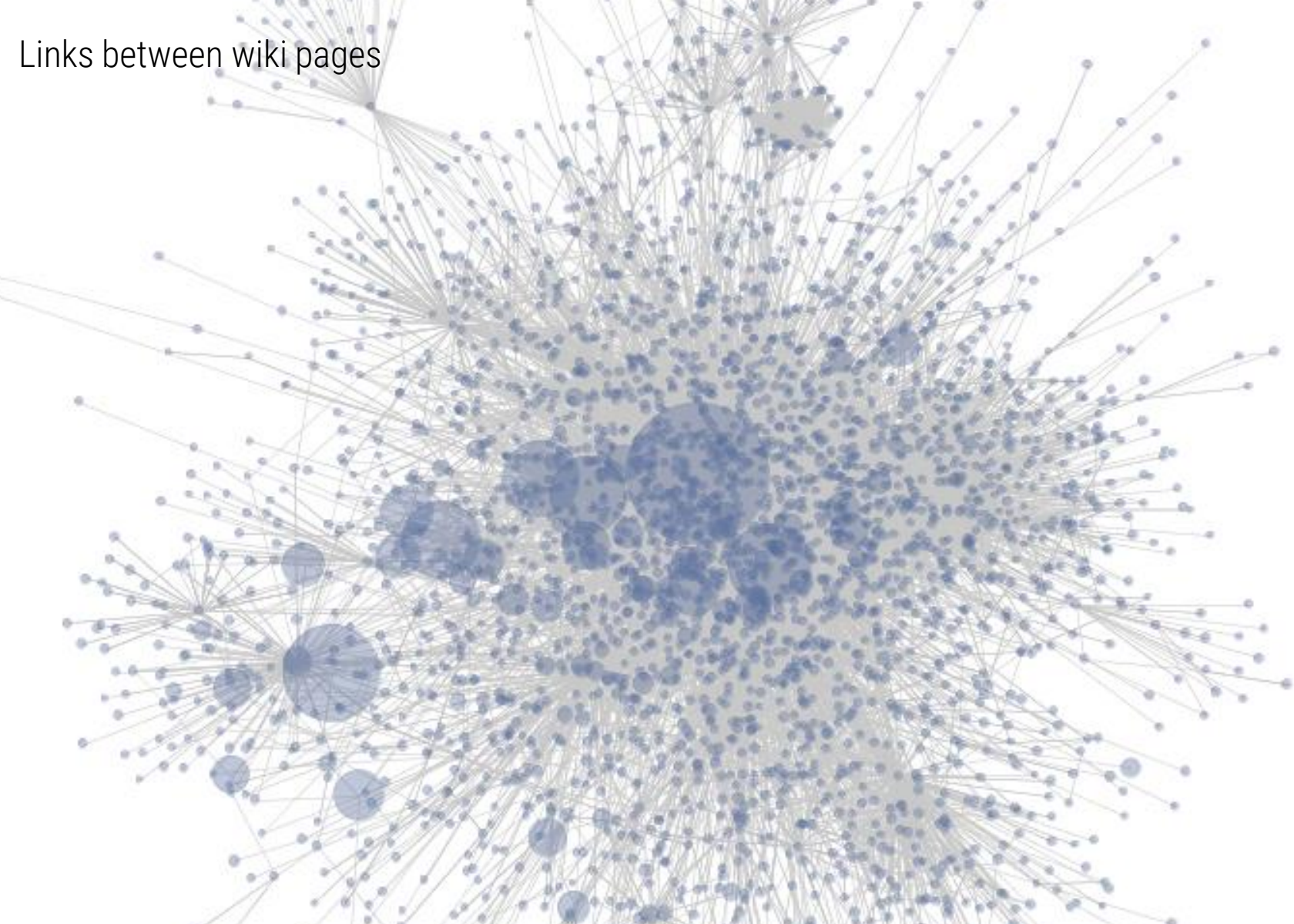
After making those two lists, we considered the set of low-level Visual Analytics tasks proposed by Amar *et al.* [2]. These tasks were extracted from a corpus of questions about tabular data. We realized that our tasks all seem to be compound tasks made up of Amar *et al*'s primitive tasks applied to the graph objects. When some tasks could not be represented with those tasks and objects, we added either an object or a low-level task. In this paper, we

https://doi.org/10.1145/1168149.1168168

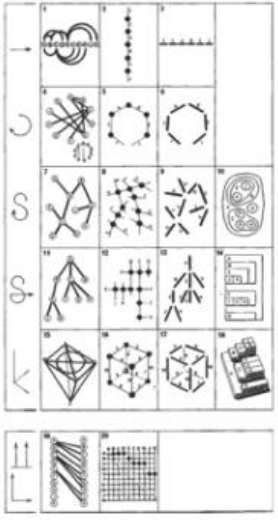
Social network visualization



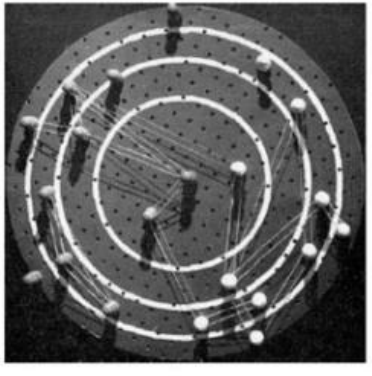
Links between wiki pages



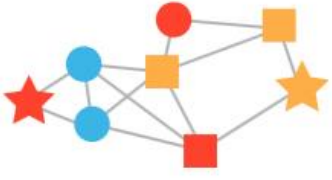
GRAPH VISUALIZATION CHALLENGES



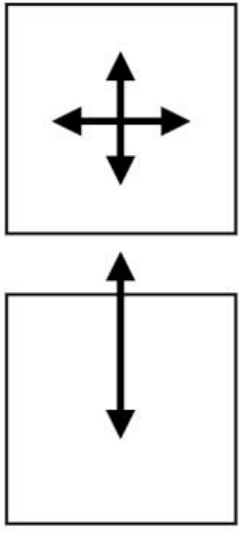
Representation



Layout



Types + Attributes



Navigation

SOCIAL NETWORK ANALYSIS

real example:
https://datascientistinsights.com/2014/02/18/art-of-resistance-the-social-network-anatomy-of-a-kinetic-activist-group/

- determine if Greenpeace was or could become a significant disruptive geopolitical force

- first: identify who/what to concentrate resources on, i.e., find data

SOCIAL NETWORK ANALYSIS

real example:
https://datascientistinsights.com/2014/02/18/art-of-resistance-the-social-network-anatomy-of-a-kinetic-activist-group/

Data Scientist Insights

EXPLORING THE DARKEST PLACES ON EARTH

DATA SCIENCE | FIELD NOTE | TOOLS | VISUALIZATION | BIG DATA | R | CASE STUDY

HOME > DATA SCIENCE > ART OF RESISTANCE – THE SOCIAL NETWORK ANATOMY OF A KINETIC ACTIVIST GROUP

Art of Resistance – The Social Network Anatomy of a Kinetic Activist Group

BY DR. J *on* FEBRUARY 18, 2014 · (0)

As a data scientist that works in the intelligence community, we are often asked to help identify where intelligence gathering and analysis resources should be allocated. Governmental and non-governmental

SOCIAL NETWORK ANALYSIS

1) get Facebook data using Netvizz

Studying Facebook via Data Extraction: The Netvizz Application

Bernhard Rieder
University of Amsterdam
Turfdraagsterpad 9
1012TX Amsterdam
rieder@uva.nl

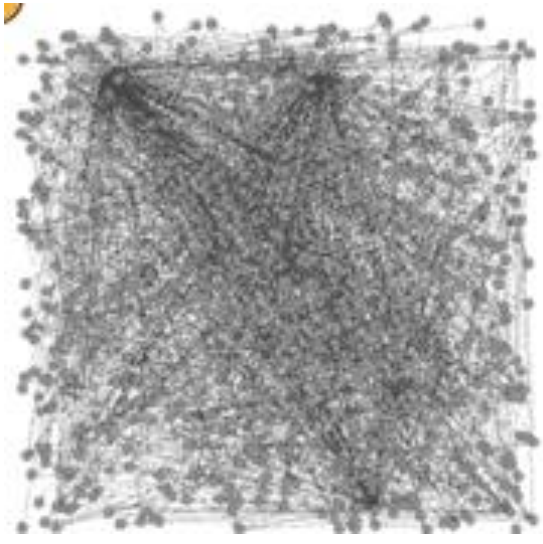
ABSTRACT
This paper describes Netvizz, a data collection and extraction application that allows researchers to export data in standard file formats from different sections of the Facebook social networking service. Friendship networks, groups, and pages can thus be analyzed quantitatively and qualitatively with regards to demographical, post-

numerous publications employing conceptual and/or critical approaches. While traditional empirical methods such as interviews, experiments, and observations are widely used, a growing number of studies rely on what the authors call "data crawling", i.e. "gleaning information about users from their profiles without their active participation" [19]. This paper presents a software tool, Netvizz, designed to

SOCIAL NETWORK ANALYSIS

2) load the data into Gephi

https://gephi.org/



585 nodes, interconnected by 1788 edges.
“Somewhere in that spaghetti is a potential bad guy, but where?”

SOCIAL NETWORK ANALYSIS

3) choose a layout algorithm that makes sense for social networks

Force Atlas 2

provides some transparency into the network but still lacks any real clarity around behavioral importance

SOCIAL NETWORK ANALYSIS

4) map an attribute to size of the nodes

betweenness centrality
(number of shortest paths from all vertices to all others that pass through that node)

bigger nodes are more central to behavioral dynamics

several nodes become central figures in the overall network

SOCIAL NETWORK ANALYSIS

5) highlight communities

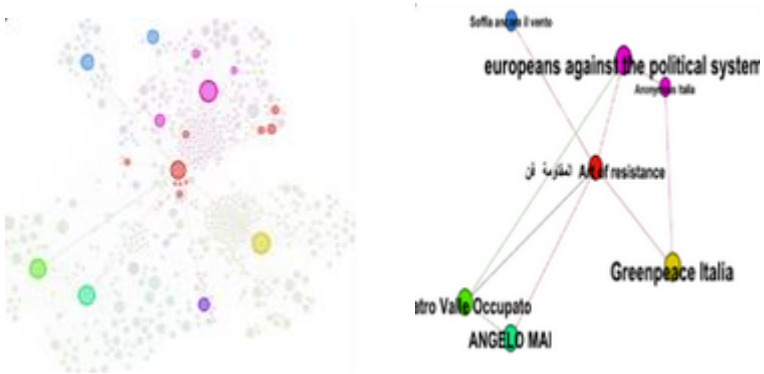
color nodes by modularity / clusters

We now begin to see a clearer picture of who is doing what with whom.

What becomes really interesting at this stage is understanding some of the more nuanced relationships.

SOCIAL NETWORK ANALYSIS

6) filter, explore, label



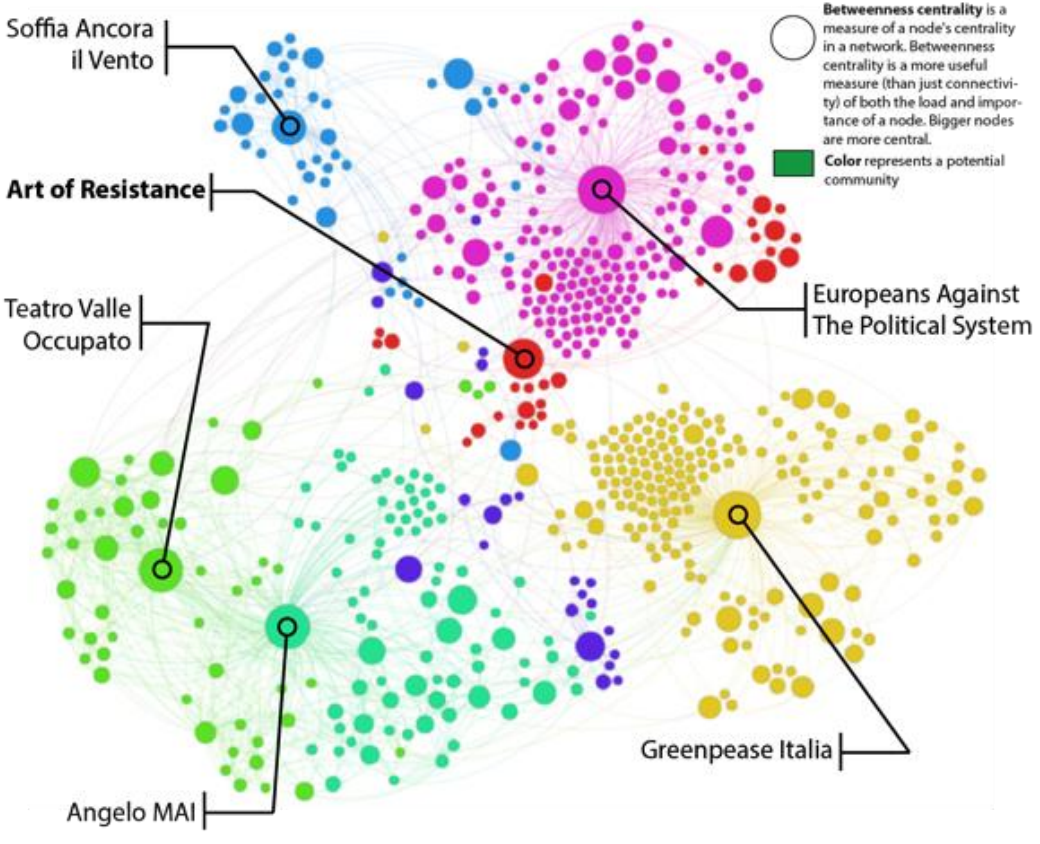
Five outlying nodes in the network (blue, maroon, yellow, dark green, and light green).

Center: an equally important red node

Emergence of a previously un-recognized activism player: [Art of Resistance](#).

SOCIAL NETWORK ANALYSIS

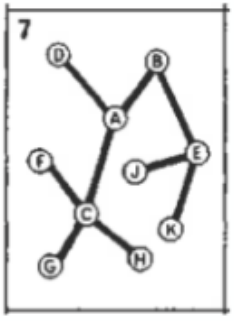
7) communicate & explain



LAYOUTS

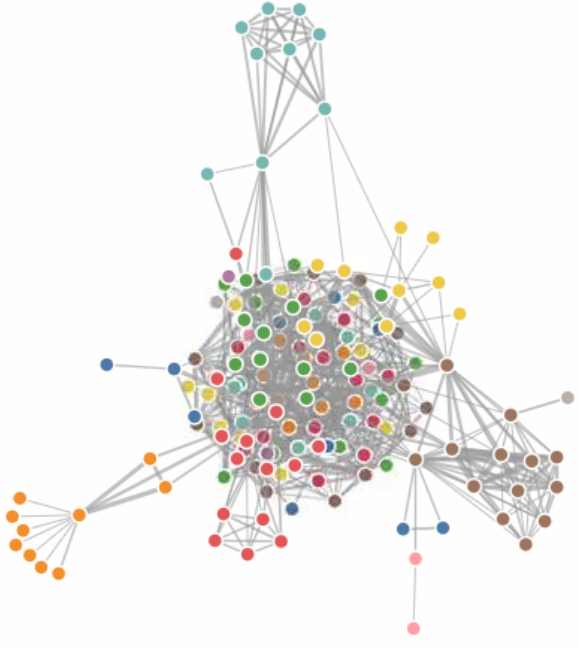
Important to the success of your analysis

FORCE-DIRECTED LAYOUT



- physical forces
- proximity based

- spring model
- Kamanda & Kawai
- Frucherman & Reingold
- Davidson & Harel
- LinLog



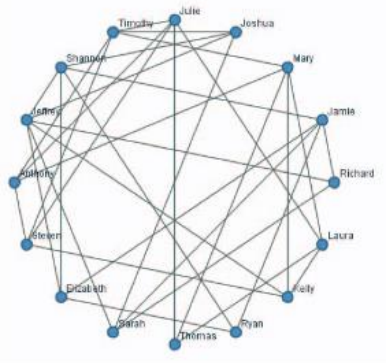
MANUAL LAYOUT

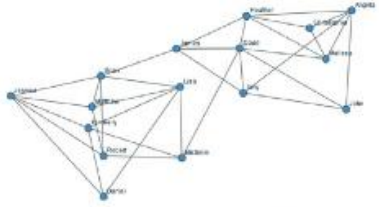
aesthetic criteria

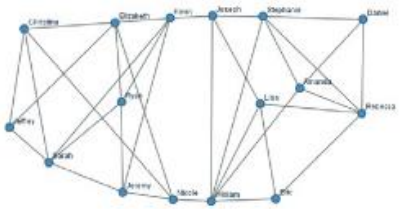
- reduce number of edge crossing
- foster symmetry
- uniform edge length
- aspect ratio
- equal angles
- ...

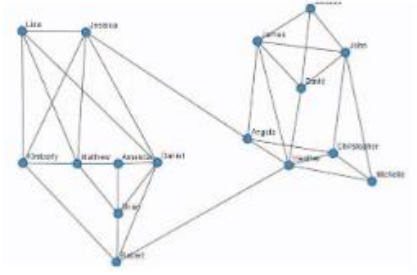
} GRAPH DRAWING

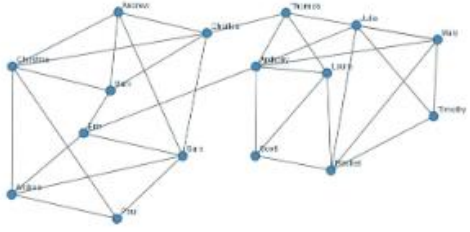
MANUAL LAYOUT

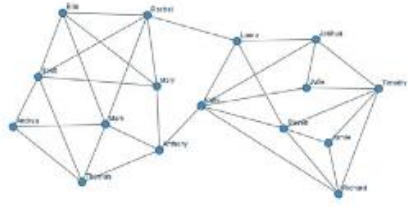


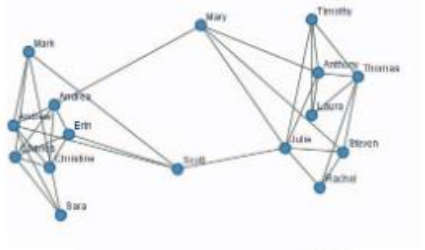


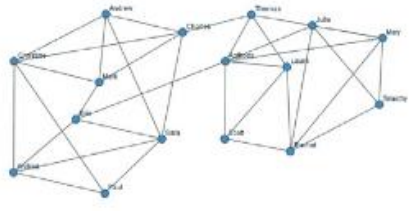






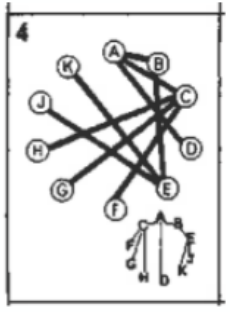


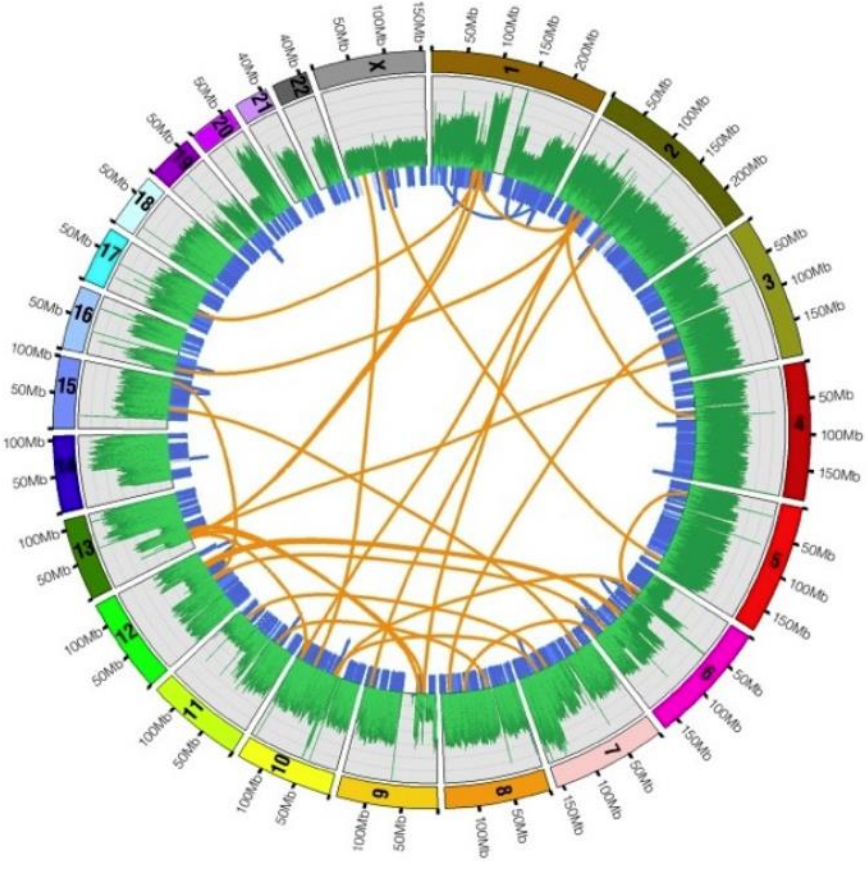




Perceptual organization in user-generated graph layouts
van Ham, F.J.J.; Rogowitz, B.

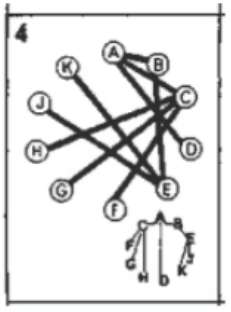
LAYOUT CIRCULAR

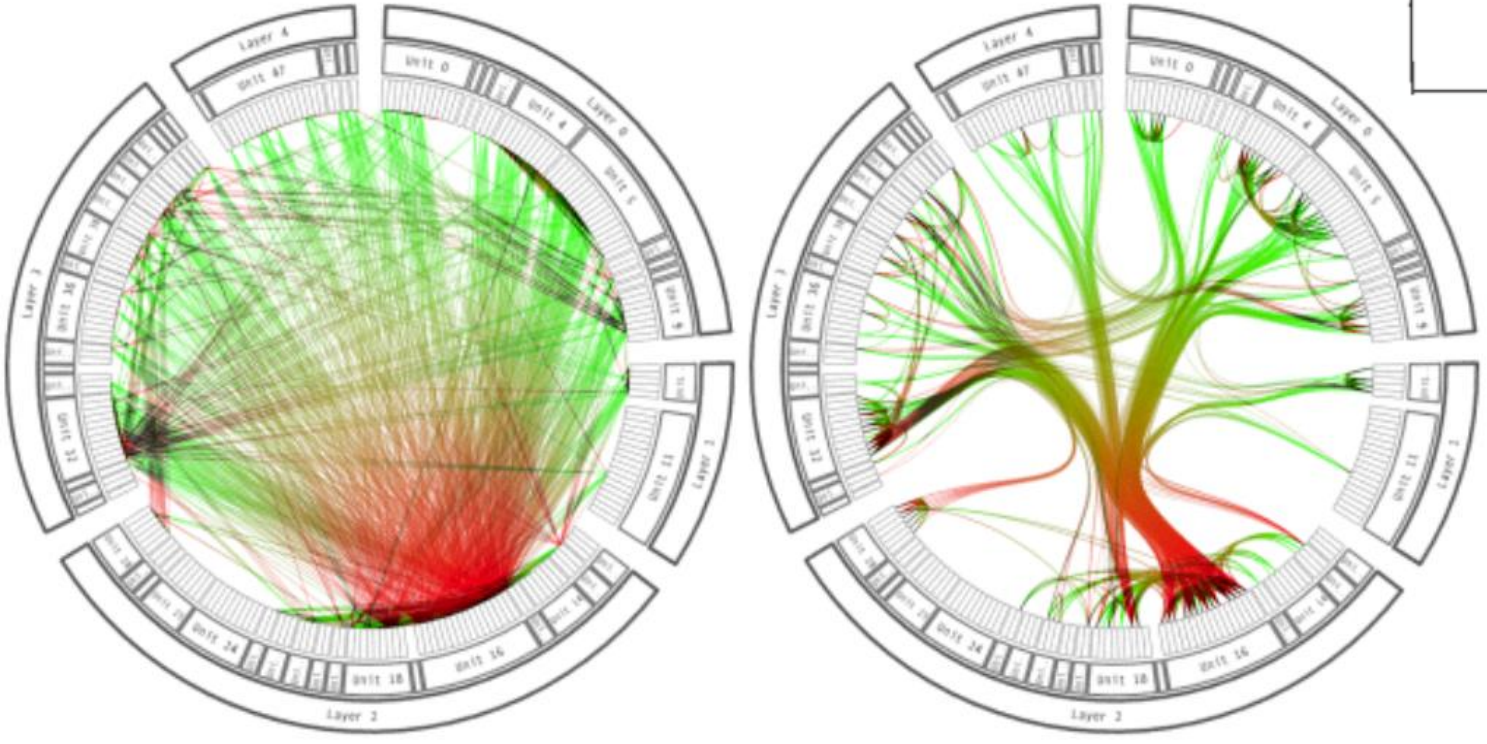




- edges on the inside
- vertices & attributes on the outside
- ordering possible

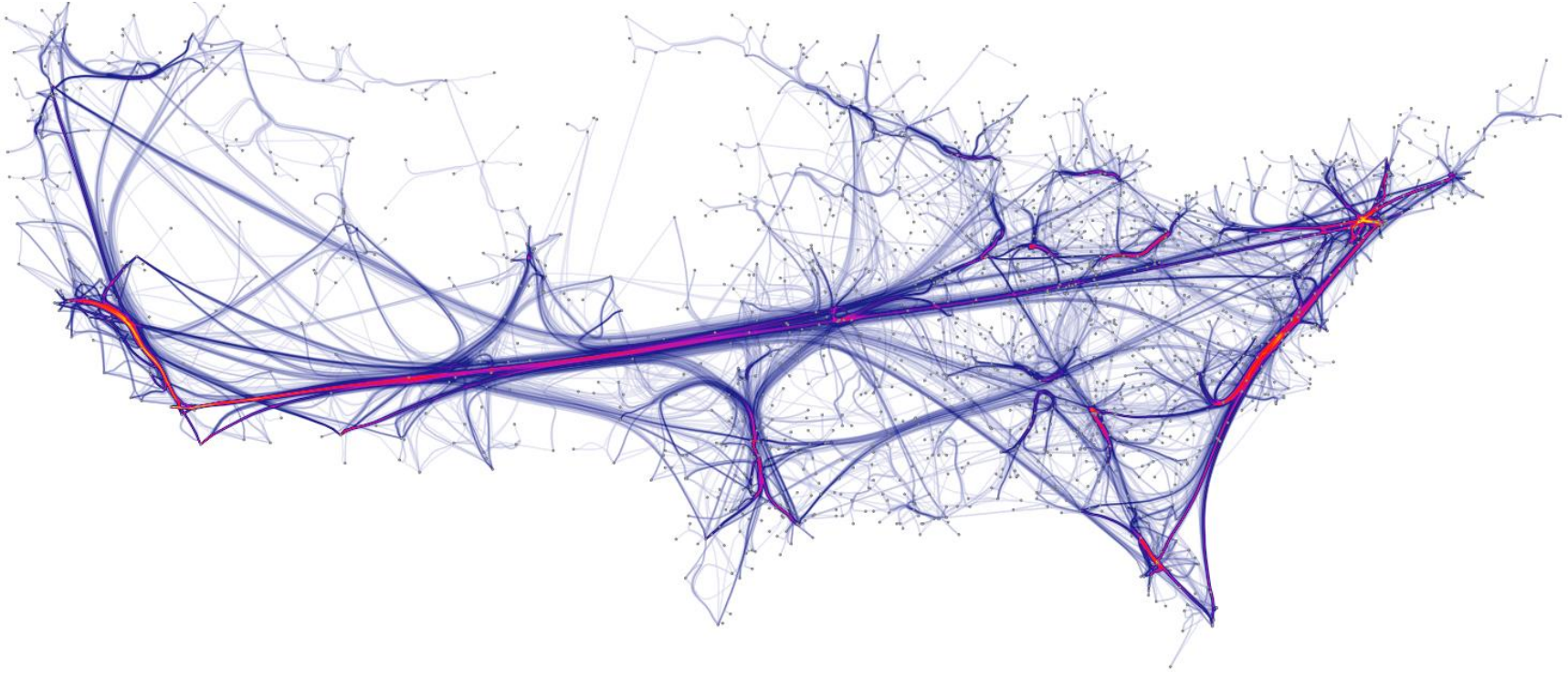
EDGE BUNDLING





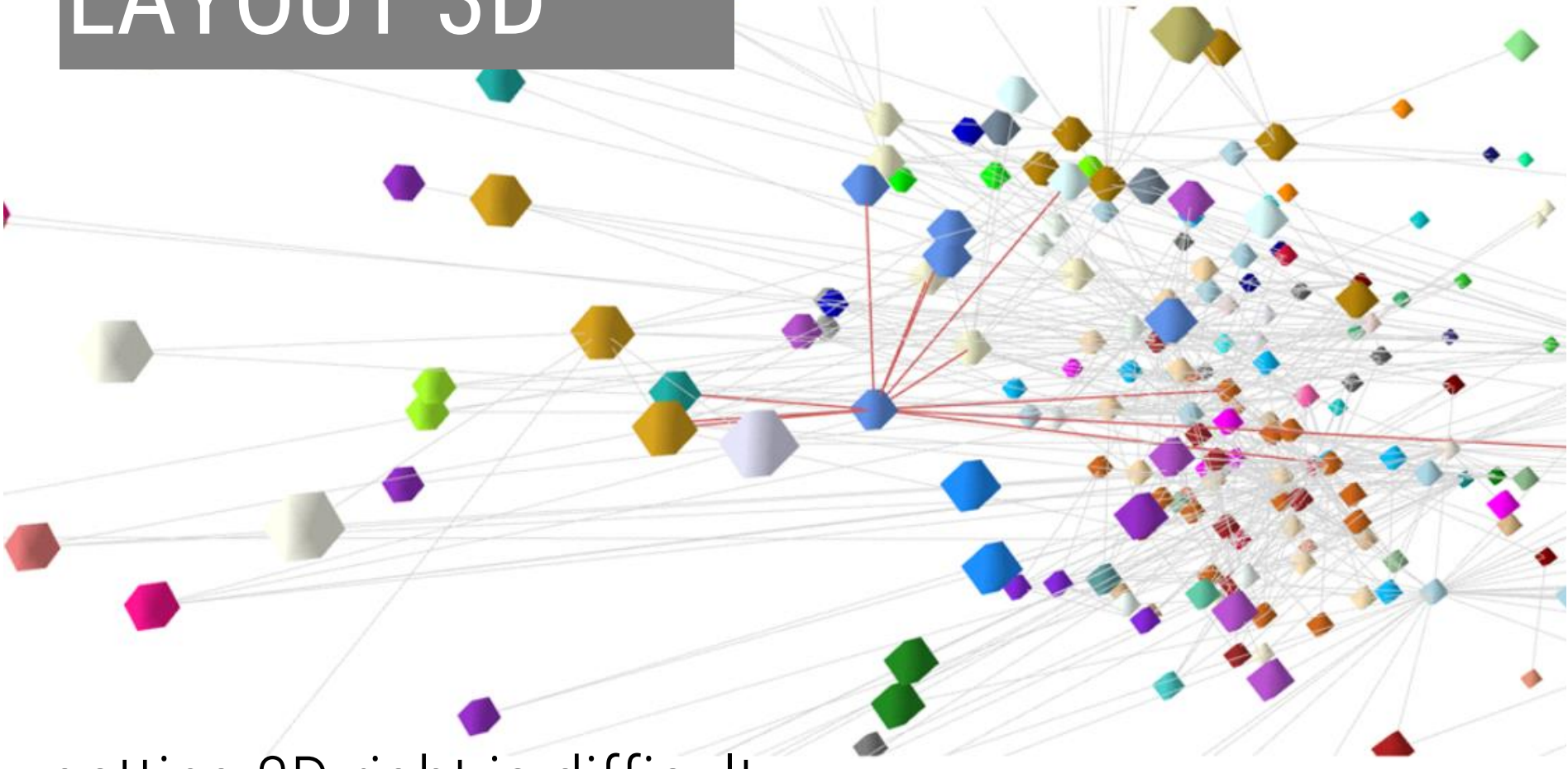
Edge Bundling
Holten 2006

EDGE BUNDLING



Holten and van Wijk, 2009

LAYOUT 3D



getting 3D right is difficult

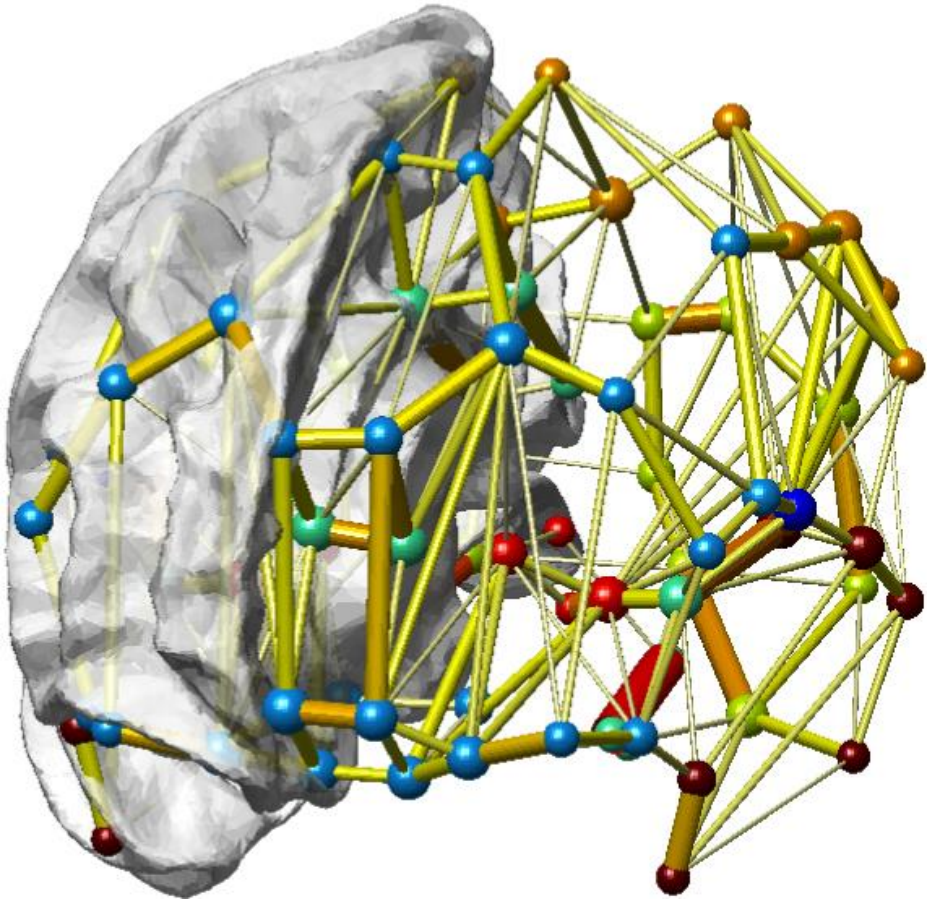
https://fwaris.wordpress.com/2012/07/08/a-simple-technique-for-creating-3d-graphs-from-2d-ones/

LAYOUT 3D

sometimes beneficial

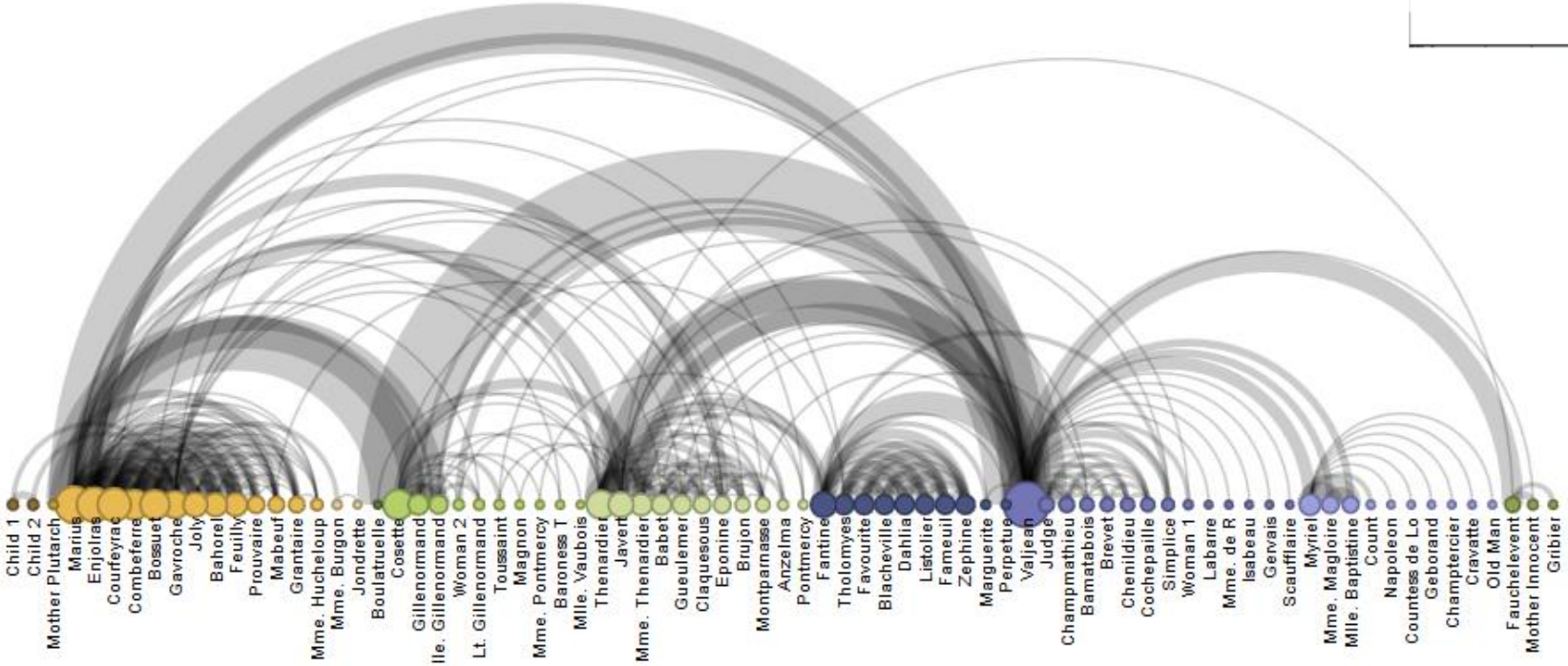
but: common issues

- occlusion
- perspective distortion
- different views of same setup perceived differently



http://gephi.forumatic.com/viewtopic.php?f=26&t=2616

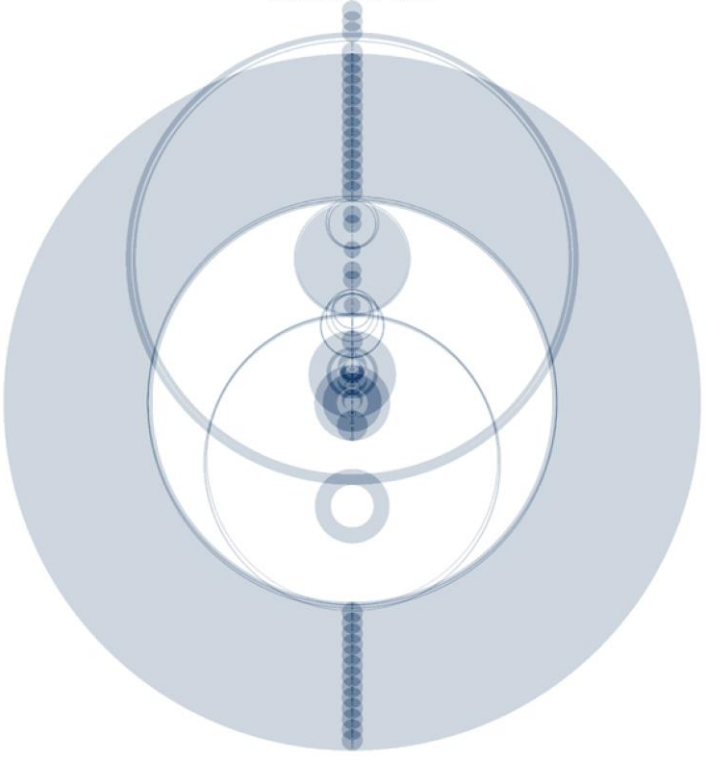
LAYOUT LINEAR



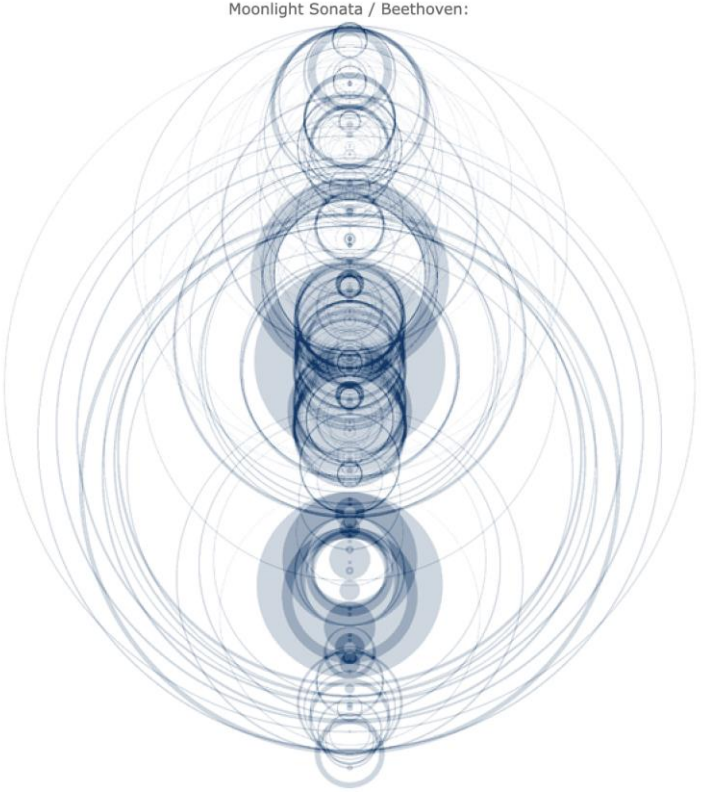
http://mbostock.github.io/protovis/ex/arc-full.html

LAYOUT LINEAR

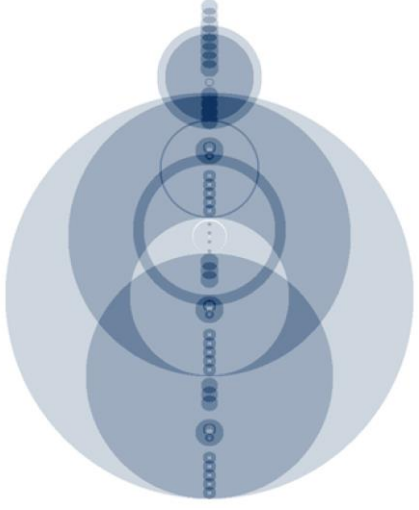
Koyaanisqatsi / Glass:



Moonlight Sonata / Beethoven:

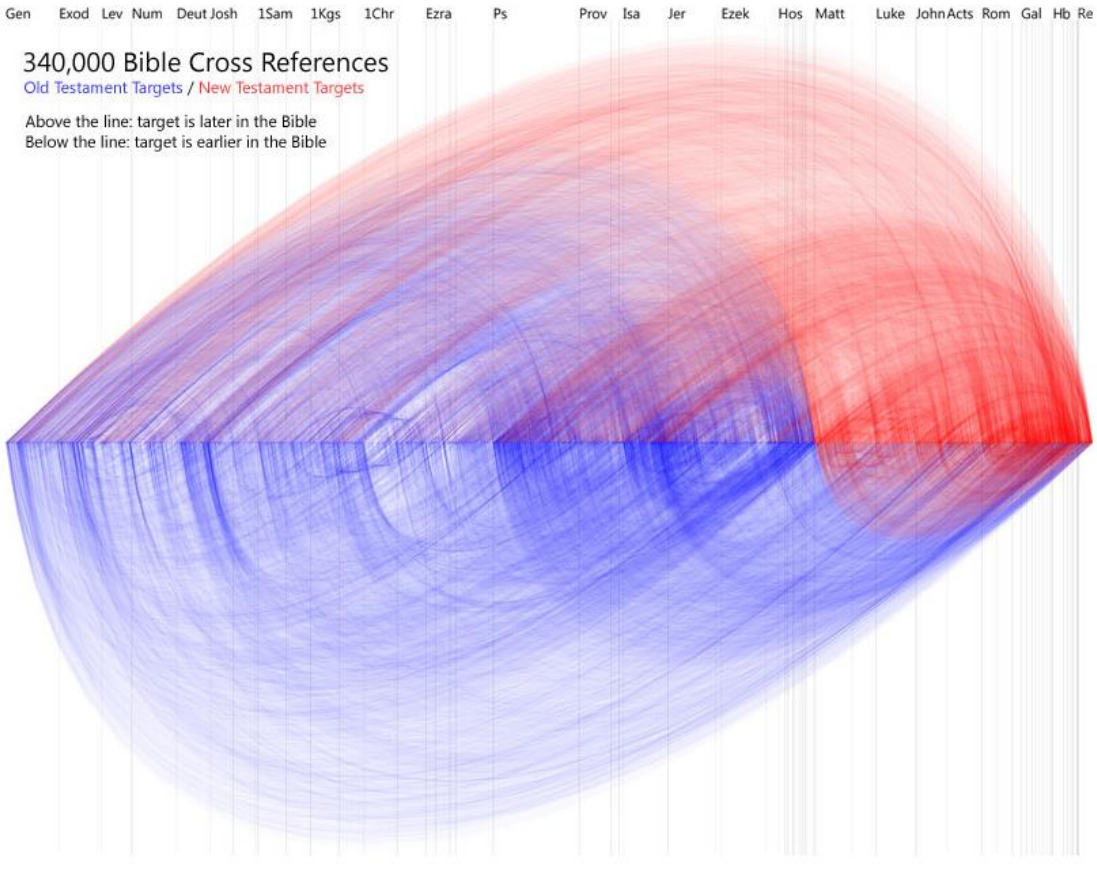


As She Was / Talking Heads:



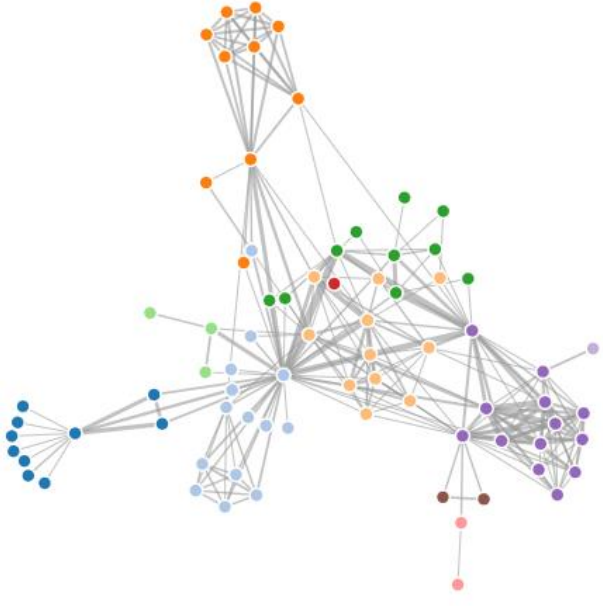
http://www.bewitched.com/song.html

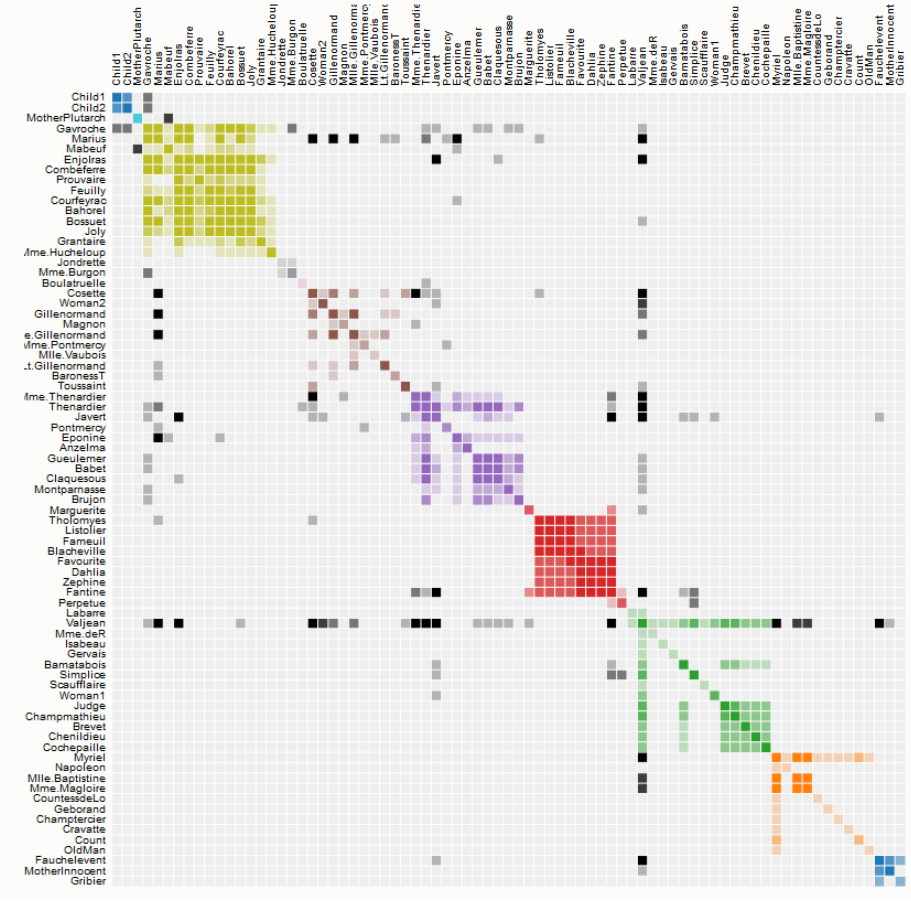
LAYOUT LINEAR



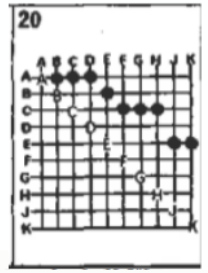
https://www.openbible.info/labs/cross-references/

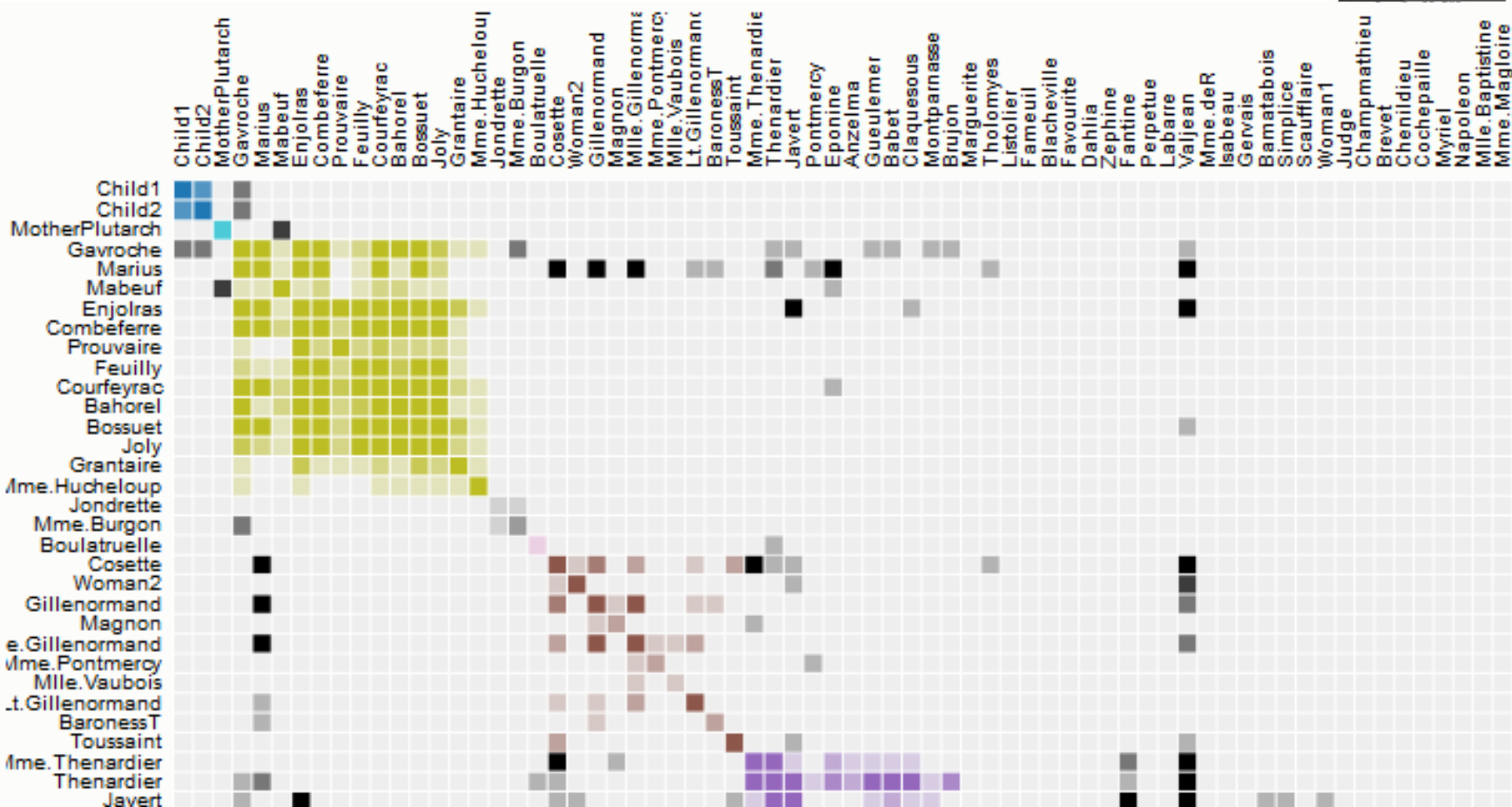
LAYOUT ADJACENCY MATRIX





ADJACENCY MATRIX





PROS/CONS

matrix

- no vertex/edge overlap or crossings
- readable for dense graph
- fast navigation

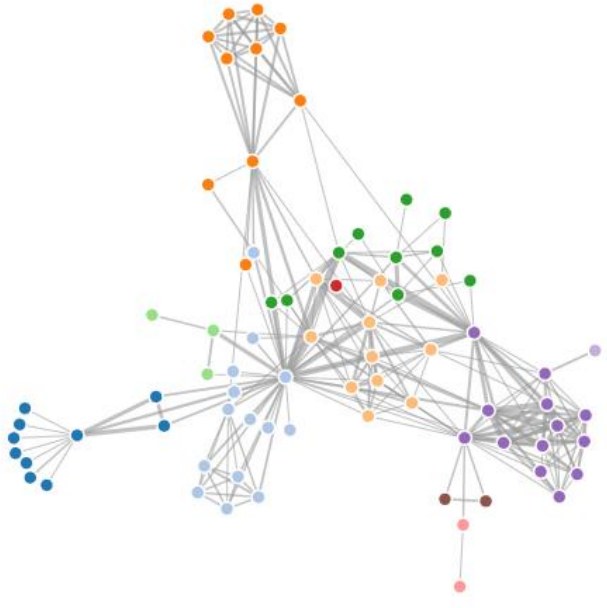
- less familiar
- space intensive
- weak for path following tasks

node-link

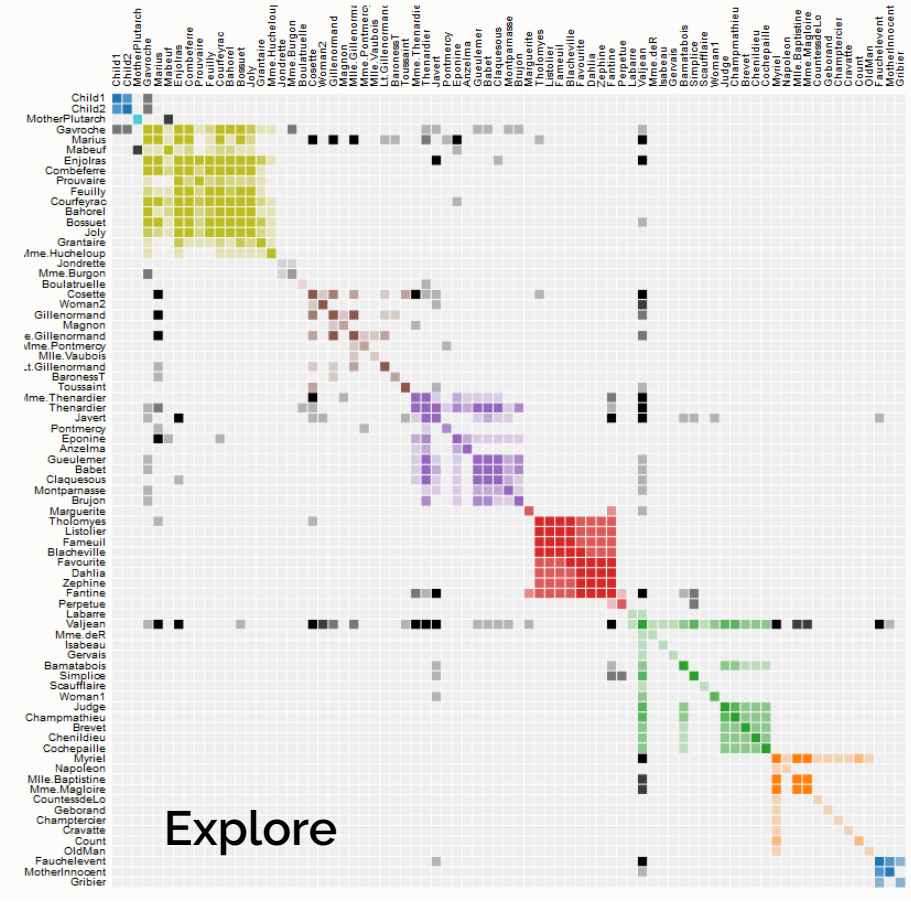
- familiar
- compact
- path following easier
- effective for small and sparse graphs

- useless without layout
- not readable for dense graphs
- manipulation requires layout computation

LAYOUT ADJACENCY MATRIX



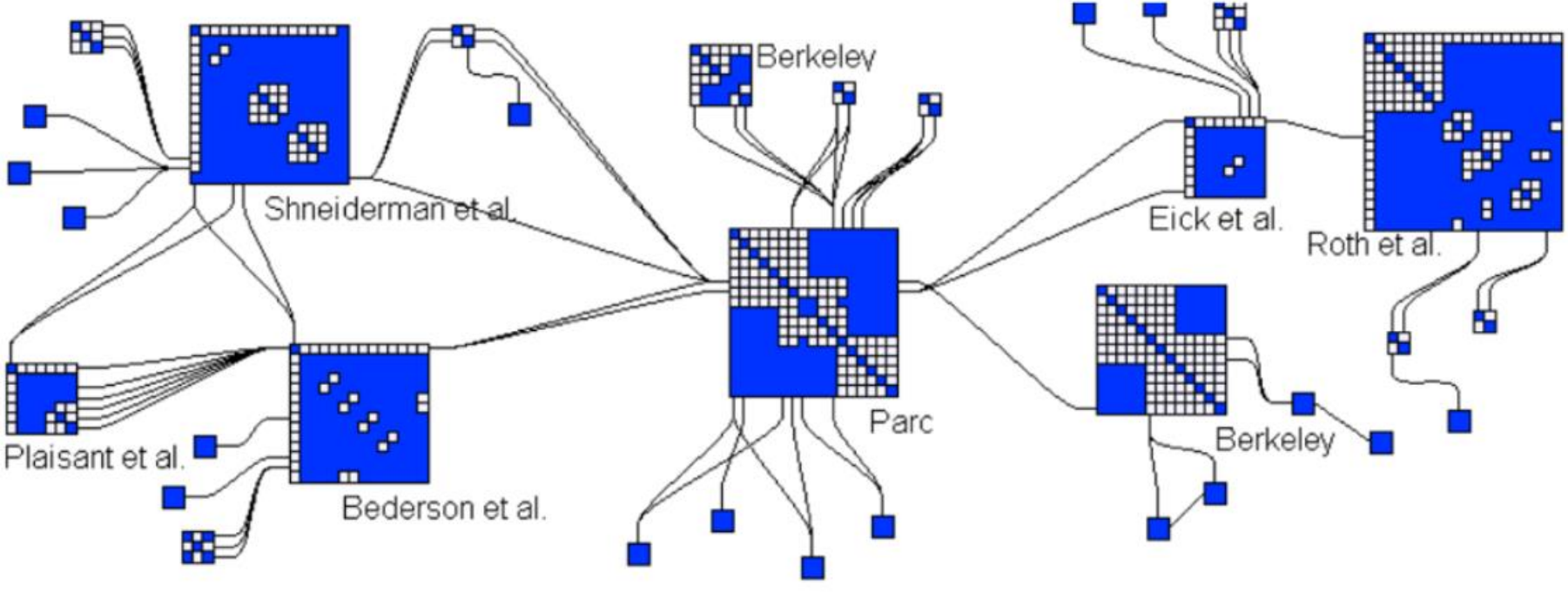
Communicate



Explore

HYBRID

Henry et al., NodeTrix



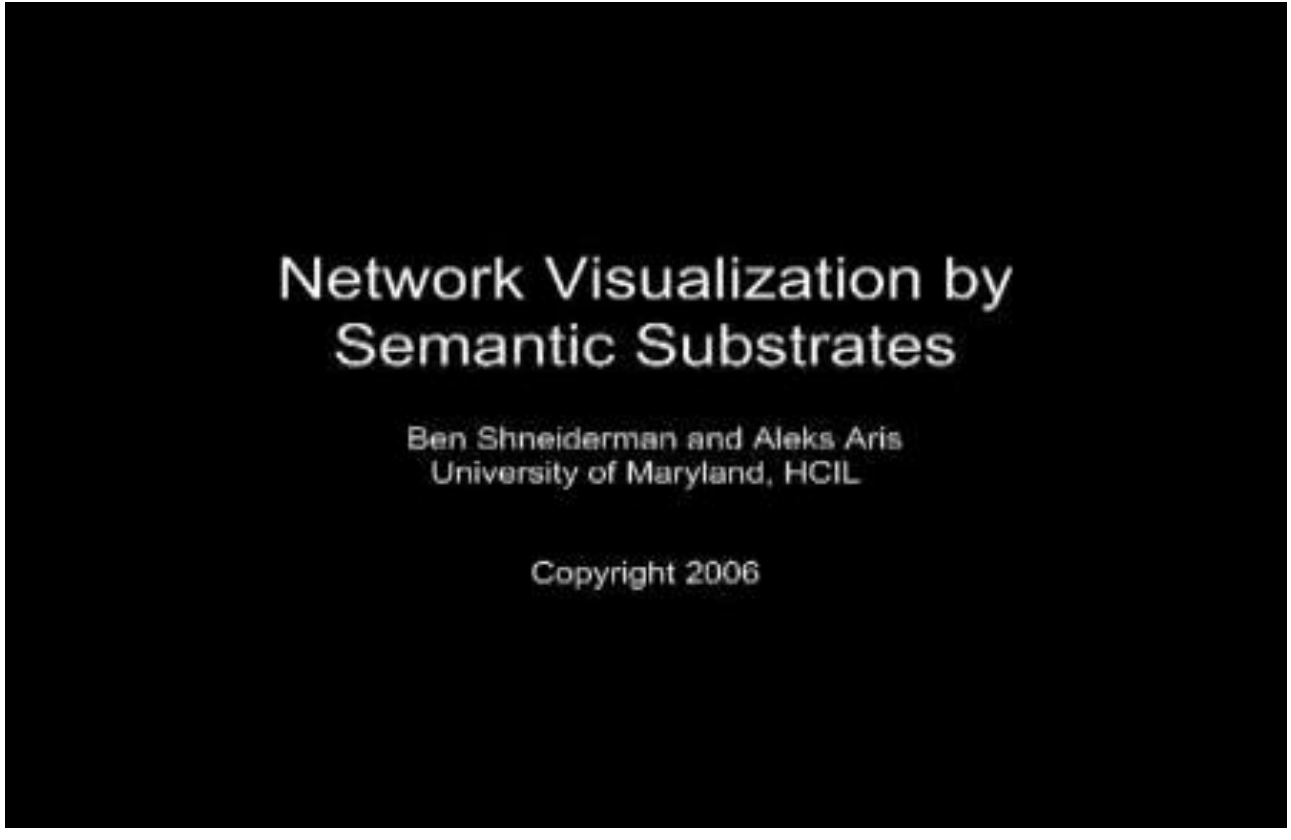
Infovis Coauthorship (133 actors)

dense = matrices, sparse = node-link

MULTIVARIATE NETWORKS

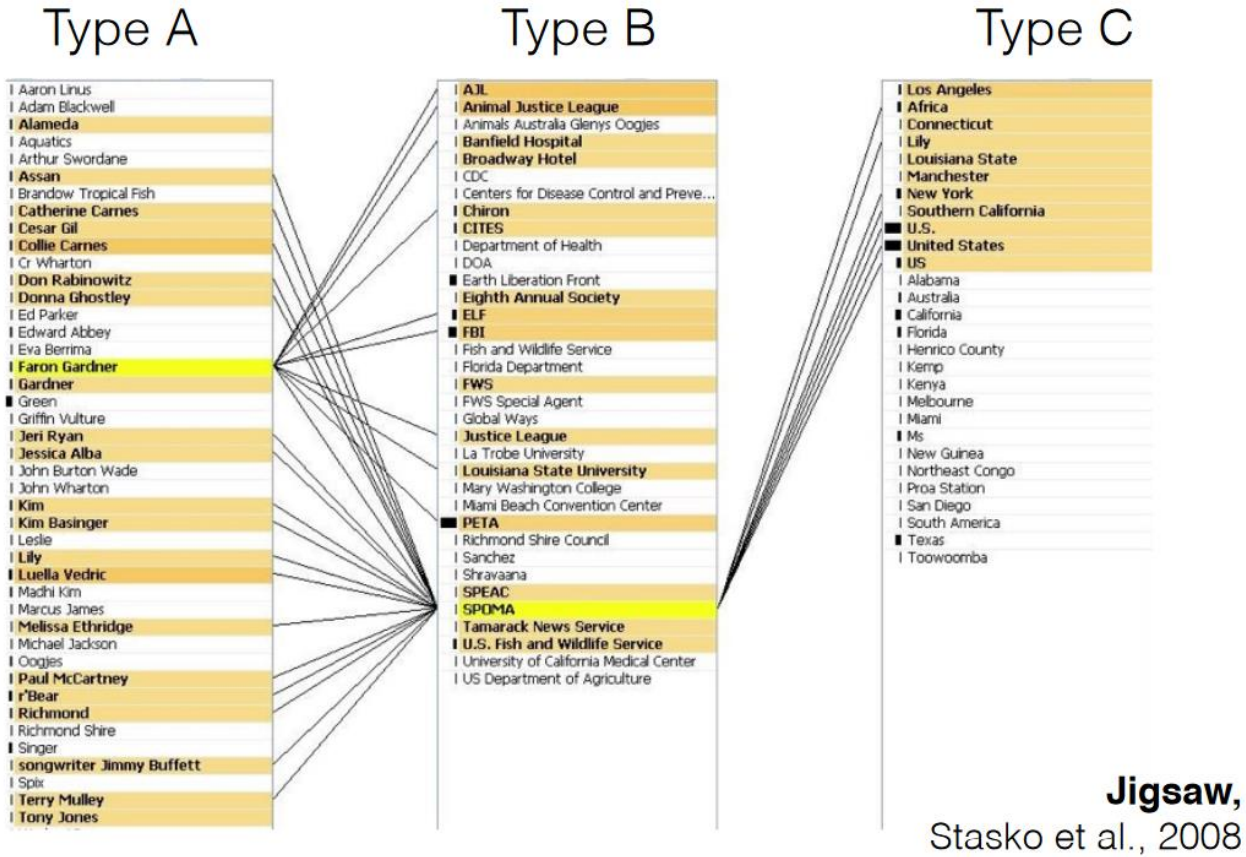
- if the network data has additional properties; e.g.,
  - nodes or edges that are categorized
  - nodes or edges that have several additional properties
- many possibilities

MULTIVARIATE NETWORKS



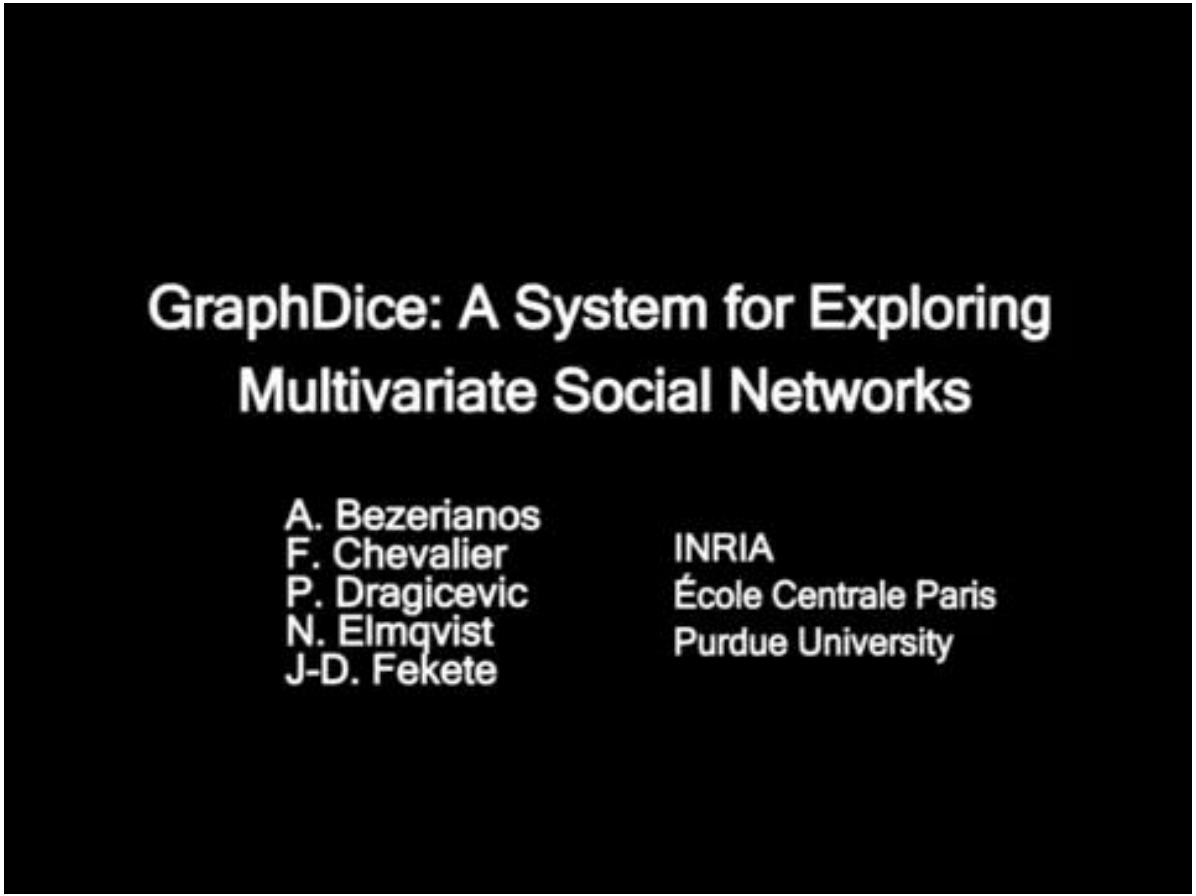
https://www.youtube.com/watch?v=f3hmn7gvocQ

MULTIVARIATE NETWORKS



Jigsaw,
Stasko et al., 2008

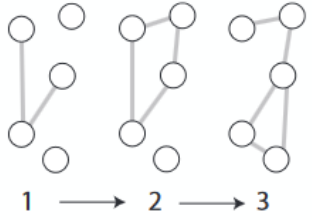
MULTIVARIATE NETWORKS



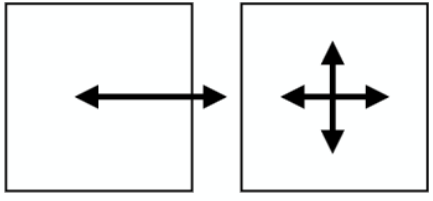
https://youtu.be/NLiHw5B0Mco

ADDITIONAL CHALLENGES

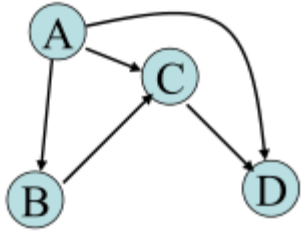
- TIME



- INTERACTION



- EDGE DIRECTION



SUMMARY

- trees and graphs: nodes and connections
- layout: many options with pros & cons
  - saw some simple approaches (many more exist)
  - often aesthetic decisions
  - often contradicting goals
  - often application-, task- & data-dependent
- overview; many more resources to go further

ADDITIONAL RESOURCES