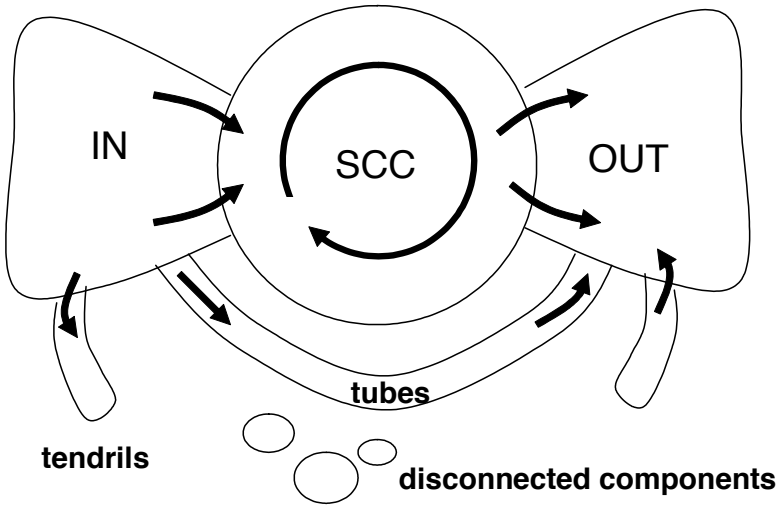
Network Metrics, Planar Graphs, and Software Tools

Based on materials by Lala Adamic, UMichigan

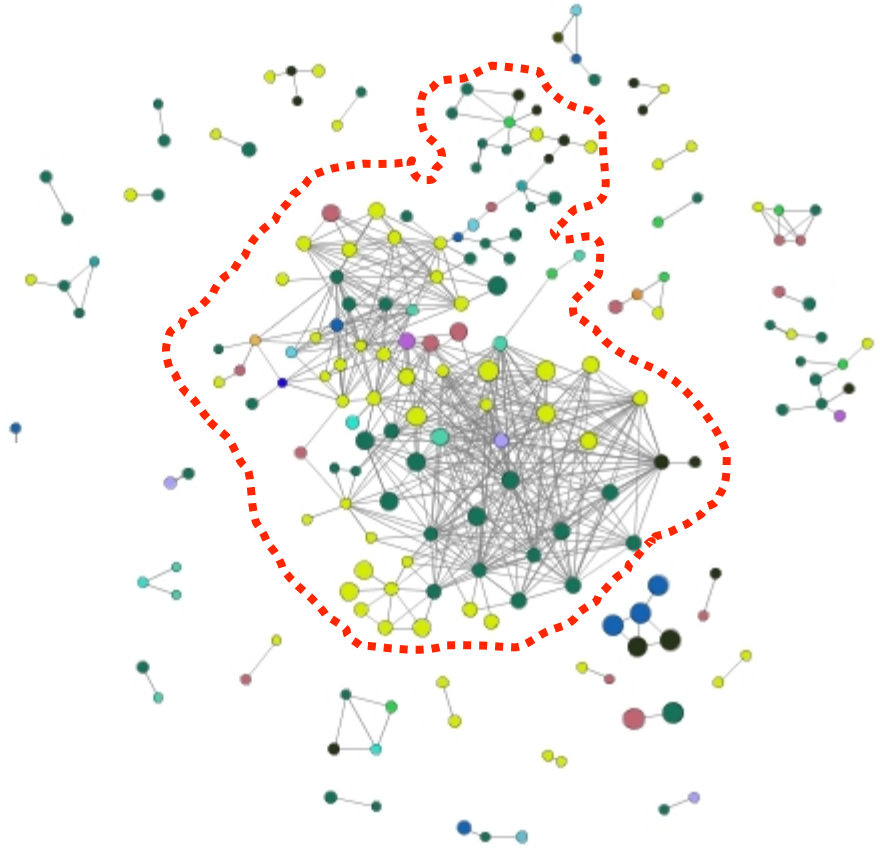
Network Metrics: Bowtie Model of the Web

- The Web is a directed graph:
  - webpages link to other webpages
- The connected components tell us what set of pages can be reached from any other just by surfing (no 'jumping' around by typing in a URL or using a search engine)
- Broder et al. 1999 – crawl of over 200 million pages and 1.5 billion links.
- SCC – 27.5%
- IN and OUT – 21.5%
- Tendrils and tubes – 21.5%
- Disconnected – 8%

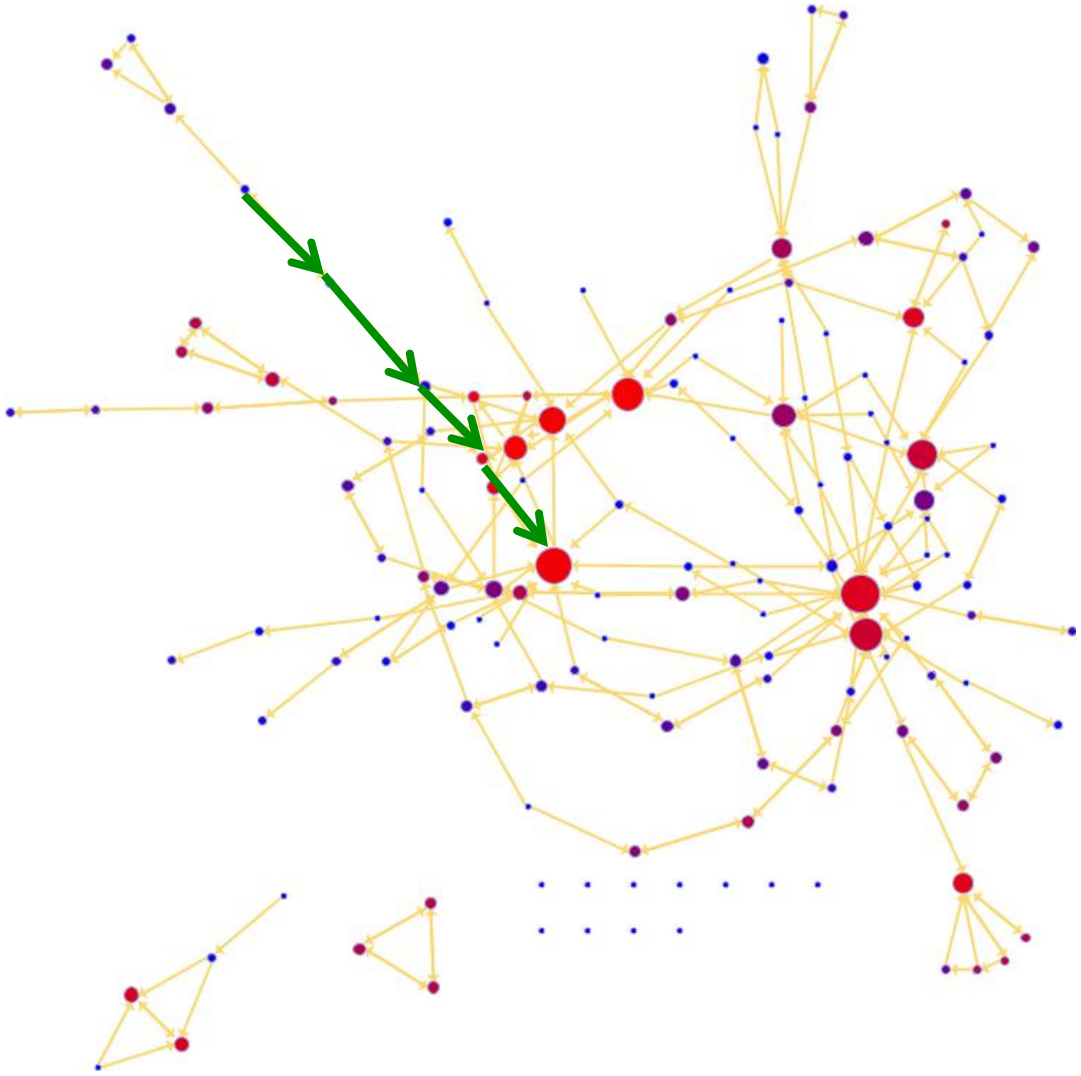


Network Metrics: Size of Giant Component

- if the largest component encompasses a significant fraction of the graph, it is called the **giant component**

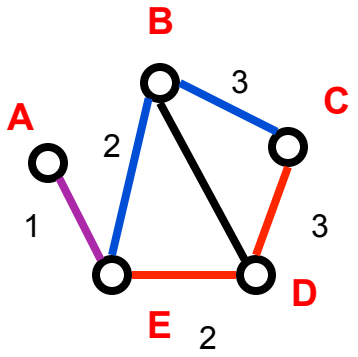


Characterizing Networks: How far apart are things?



Network Metrics: Shortest Paths

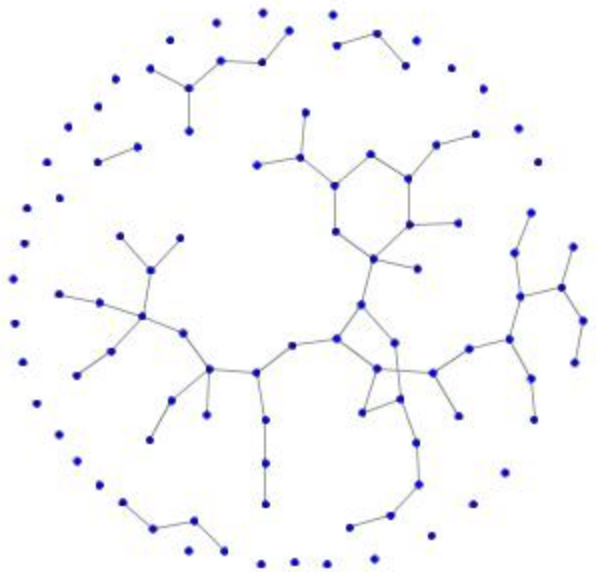
- Shortest path (also called a geodesic path)
  - The shortest sequence of links connecting two nodes
  - Not always unique
    - A and C are connected by 2 shortest paths
      - A – E – B - C
      - A – E – D - C

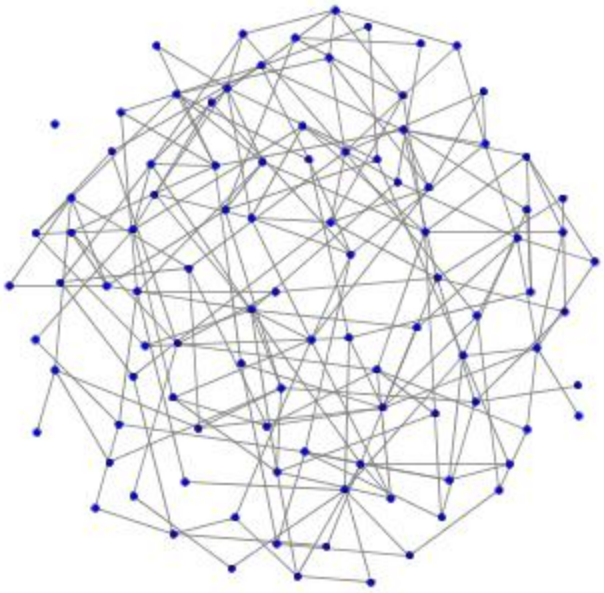


- Diameter: the largest geodesic distance in the graph
  - The distance between A and C is the maximum for the graph: 3

- Caution: some people use the term ‘diameter’ to be the average shortest path distance, in this class we will use it only to refer to the maximal distance

Characterizing Networks: How Dense Are They?

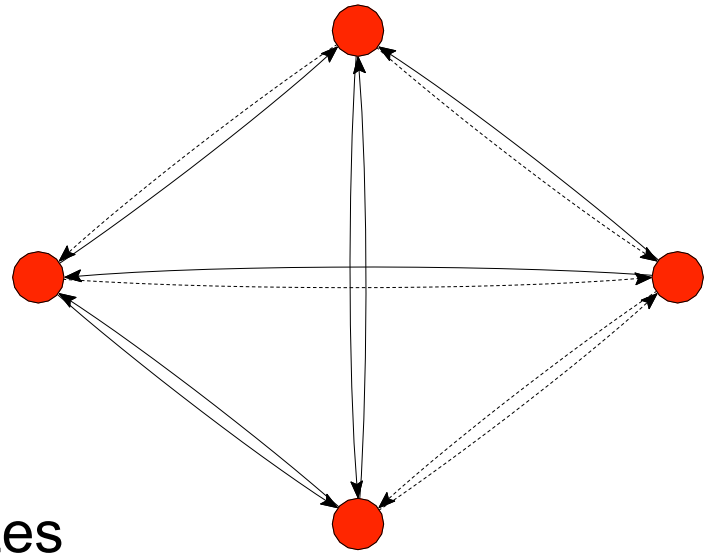




Network Metrics: Graph Density

- Of the connections that may exist between n nodes
  - directed graph
    $e_{max} = n*(n-1)$
    each of the n nodes can connect to (n-1) other nodes
  - undirected graph
    $e_{max} = n*(n-1)/2$
    since edges are undirected, count each one only once

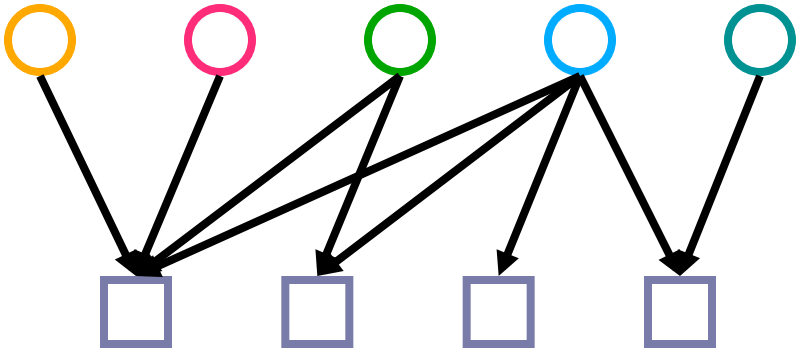
- What fraction are present?
  - density = e/ e_{max}
  - For example, out of 12 possible connections, this graph has 7, giving it a density of 7/12 = 0.583



- Would this measure be useful for comparing networks of different sizes (different numbers of nodes)?

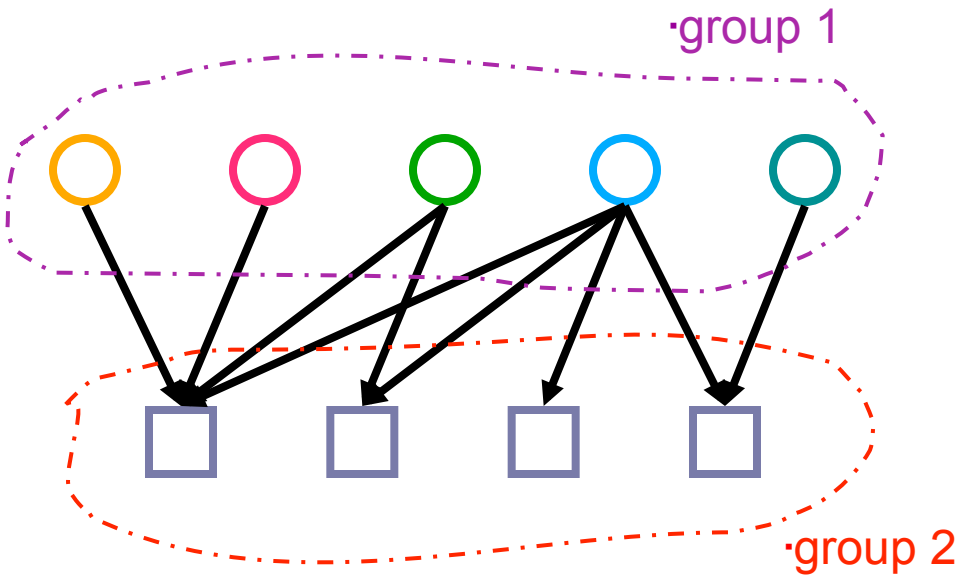
Bipartite (Two-mode) Networks

- edges occur only between two groups of nodes, not within those groups
- for example, we may have individuals and *events*
  - directors and boards of directors
  - customers and the items they purchase
  - metabolites and the reactions they participate in

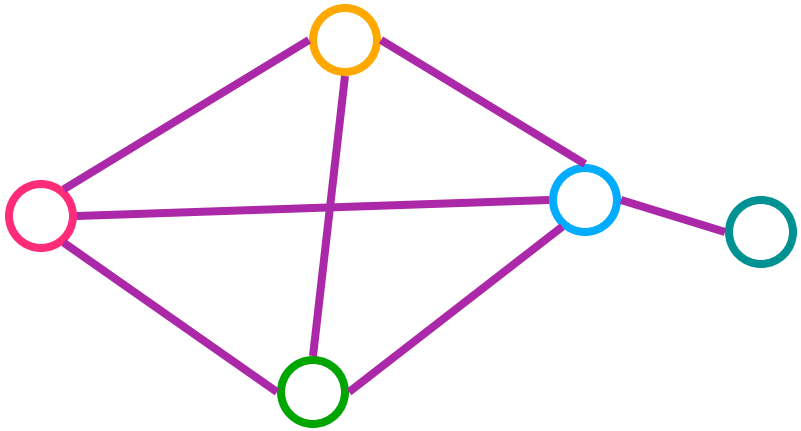


Going From A Bipartite To A One-mode Graph

- Two-mode network

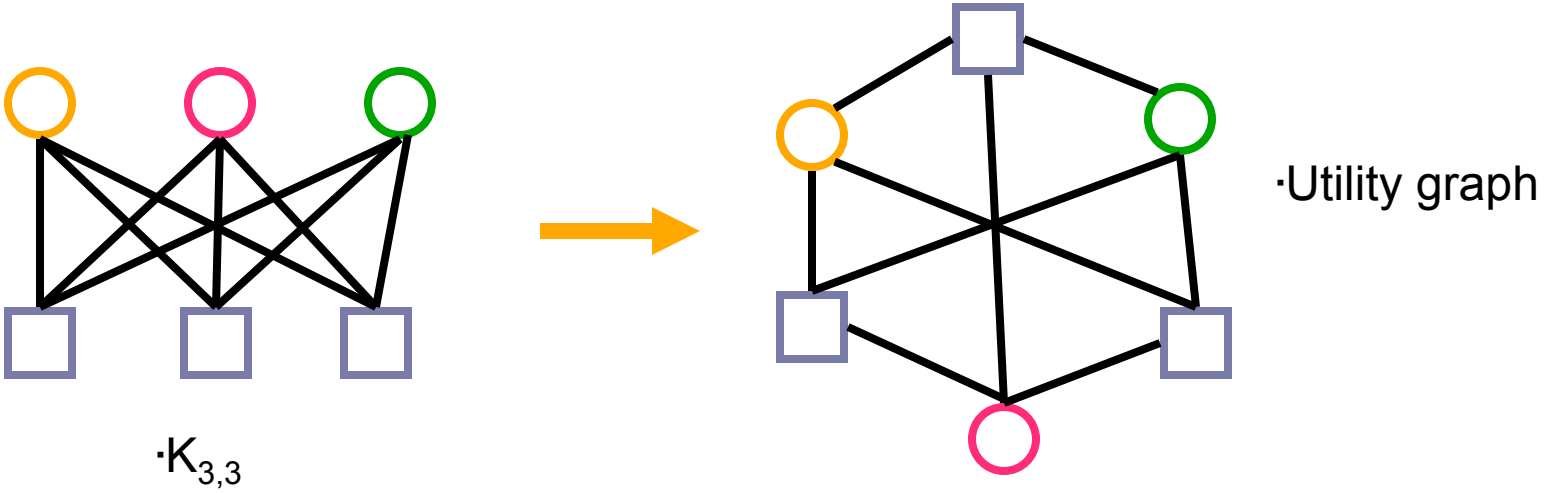


- One mode projection
  - two nodes from the first group are connected if they link to the same node in the second group
  - some loss of information
  - naturally high occurrence of cliques



Bi-cliques (Cliques In Bipartite Graphs)

- $K_{m,n}$ is the complete bipartite graph with **m** and **n** vertices of the two different types
- $K_{3,3}$ maps to the utility graph
  - Is there a way to connect three utilities, e.g. gas, water, electricity to three houses without having any of the pipes cross?

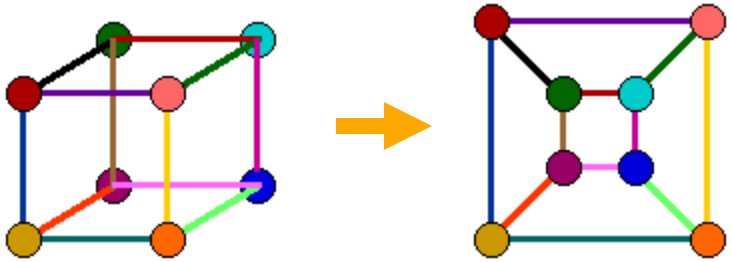


·$K_{3,3}$

·Utility graph

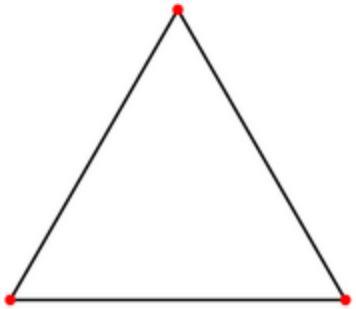
Planar graphs

- A graph is planar if it can be drawn on a plane without any edges crossing

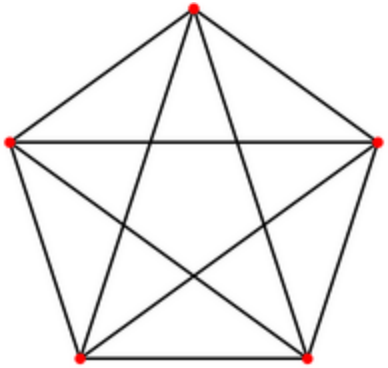


Cliques and complete graphs

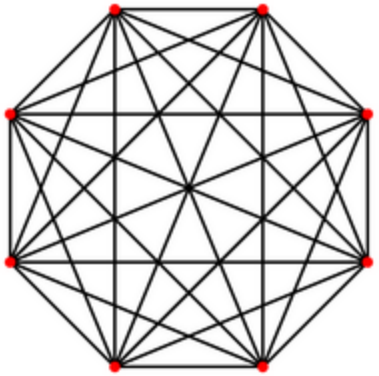
- K_n is the complete graph (clique) with K vertices
  - each vertex is connected to every other vertex
  - there are n*(n-1)/2 undirected edges



·K_3

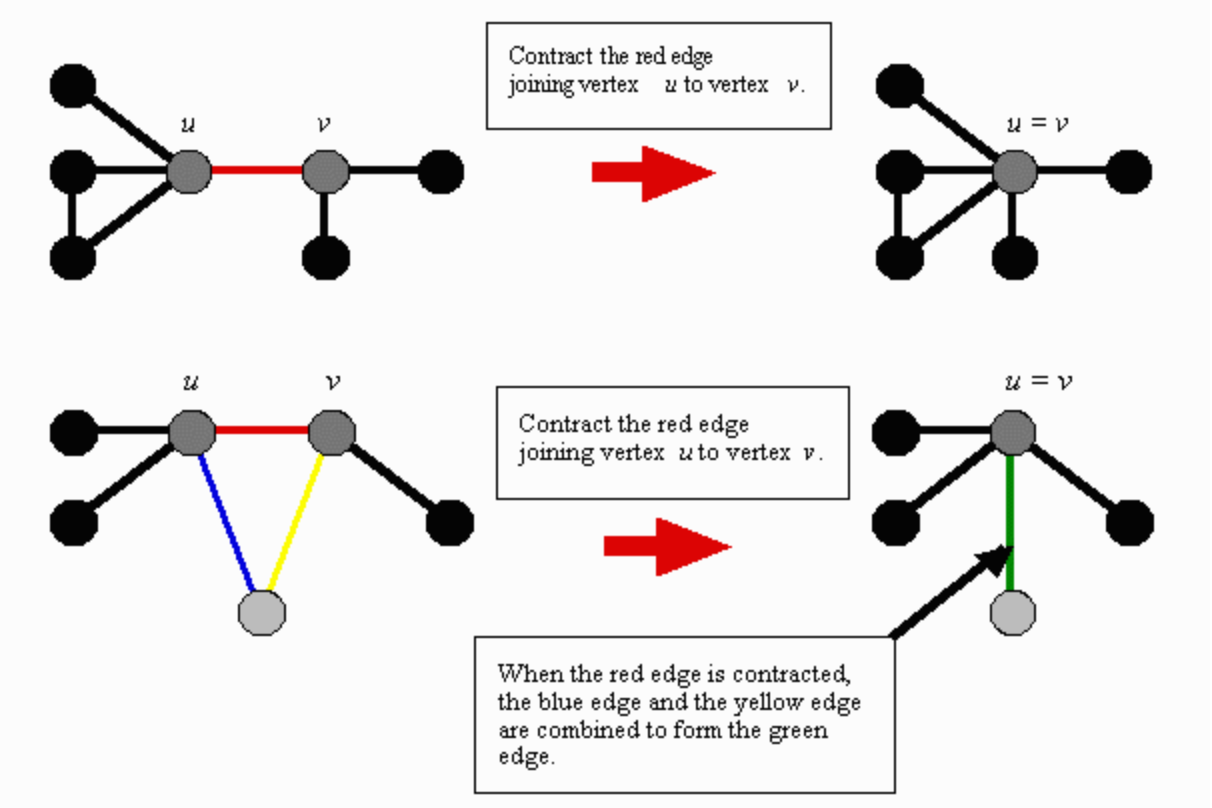


·K_5



·K_8

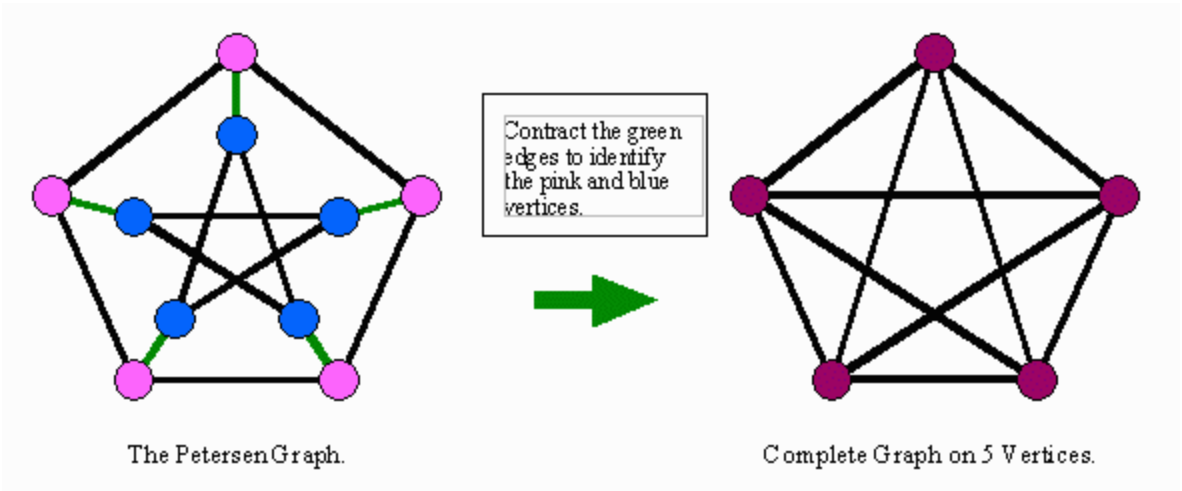
Edge contractions defined



- A finite graph *G* is planar if and only if it has no subgraph that is homeomorphic or edge-contractible to the complete graph in five vertices (K_5) or the complete bipartite graph $K_{3,3}$. **(Kuratowski's Theorem)**

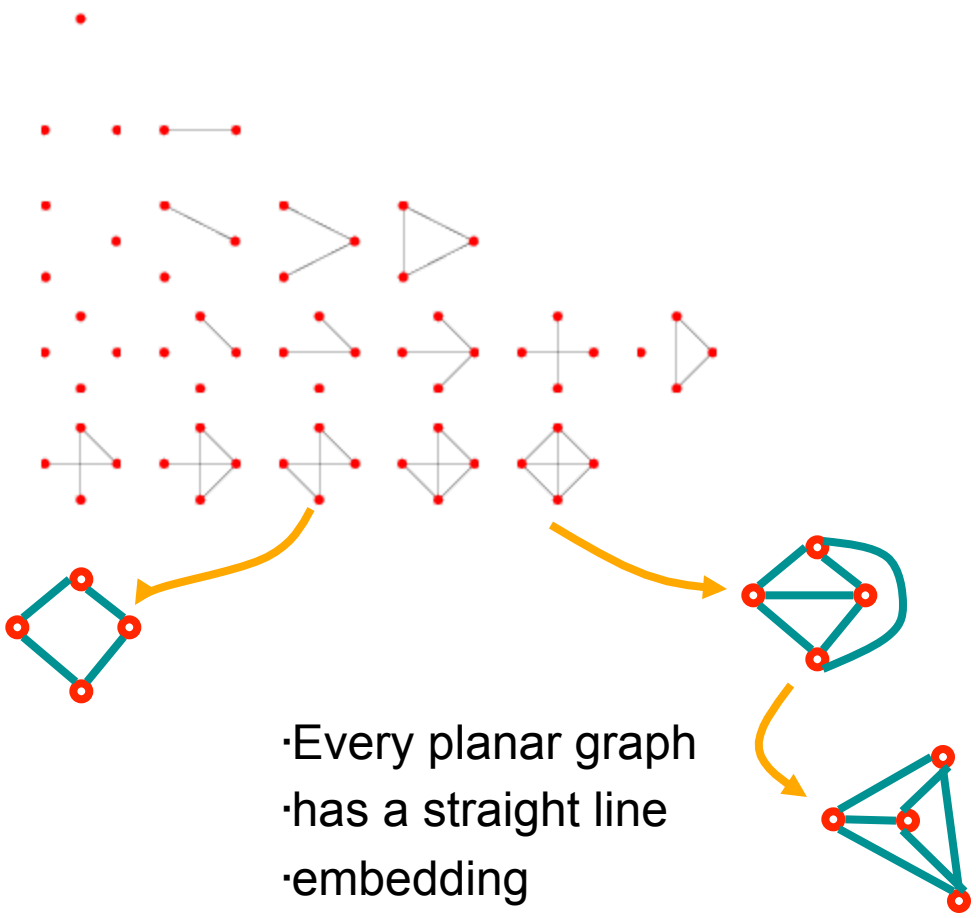
Peterson graph

- Example of using edge contractions to show a graph is not planar



#s of Planar Graphs of Different Sizes

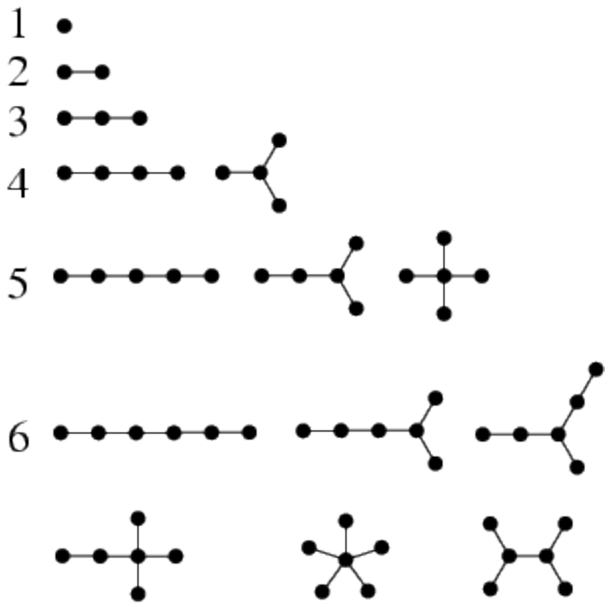
- 1:1
- 2:2
- 3:4
- 4:11



- Every planar graph
- has a straight line
- embedding

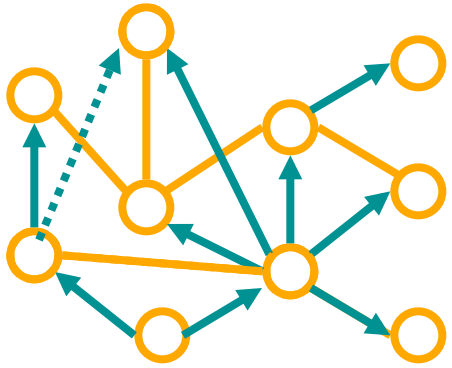
Trees

- Trees are undirected graphs that contain no cycles



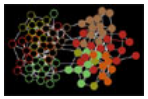
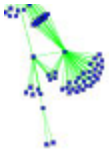
Examples of Trees

- In nature
- Man made
- Computer science
- Network analysis



NETWORK VISUALIZATION AND ANALYSIS SOFTWARE

Overview of Network Analysis Tools

	**Pajek**	network analysis and visualization, menu driven, suitable for large networks	platforms: Windows (on linux via Wine) download
	**Netlogo**	agent based modeling recently added network modeling capabilities	platforms: any (Java) download
	**GUESS**	network analysis and visualization, extensible, script-driven (jython)	platforms: any (Java) download

Other software tools that we will not be using but that you may find useful:

visualization and analysis:

UCInet - user friendly social network visualization and analysis software (suitable smaller networks)

iGraph - if you are familiar with R, you can use iGraph as a module to analyze or create large networks, or you can directly use the C functions

Jung - comprehensive Java library of network analysis, creation and visualization routines

Graph package for Matlab (untested?) - if Matlab is the environment you are most comfortable in, here are some basic routines

SIENA - for p* models and longitudinal analysis

SNA package for R - all sorts of analysis + heavy duty stats to boot

NetworkX - python based free package for analysis of large graphs

InfoVis Cyberinfrastructure - large agglomeration of network analysis tools/routines, partly menu driven

visualization only:

GraphViz - open source network visualization software (can handle large/specialized networks)

TouchGraph - need to quickly create an interactive visualization for the web?

yEd - free, graph visualization and *editing* software

specialized:

fast community finding algorithm

motif profiles

CLAIR library - NLP and IR library (Perl Based) includes network analysis routines

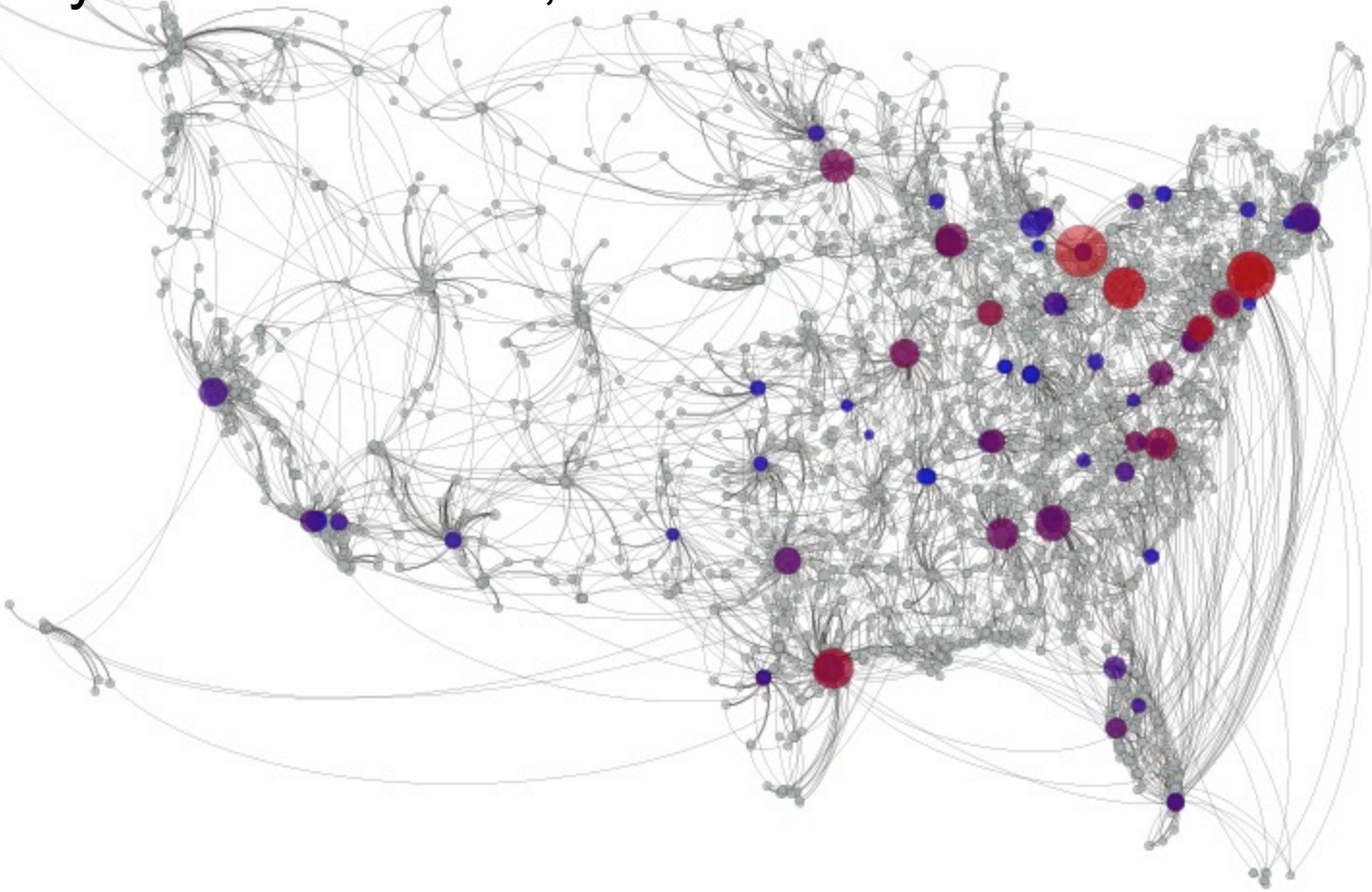
finally: INSNA long list of SNA packages

Common Tools

- **Pajek:** extensive menu-driven functionality, including many, many network metrics and manipulations
  - but… not extensible
- **Guess:** extensible, scriptable tool of exploratory data analysis, but more limited selection of built-in methods compared to Pajek
- **NetLogo:** general agent based simulation platform with excellent network modeling support
- **iGraph:** libraries can be accessed through R or python. Routines scale to millions of nodes.

Other Tools: Visualization Tool: gephi

- http://gephi.org
- primarily for visualization, has some nice touches



Visualization Tool: GraphViz

- Takes descriptions of graphs in simple text languages
- Outputs images in useful formats
- Options for shapes and colors
- Standalone or use as a library

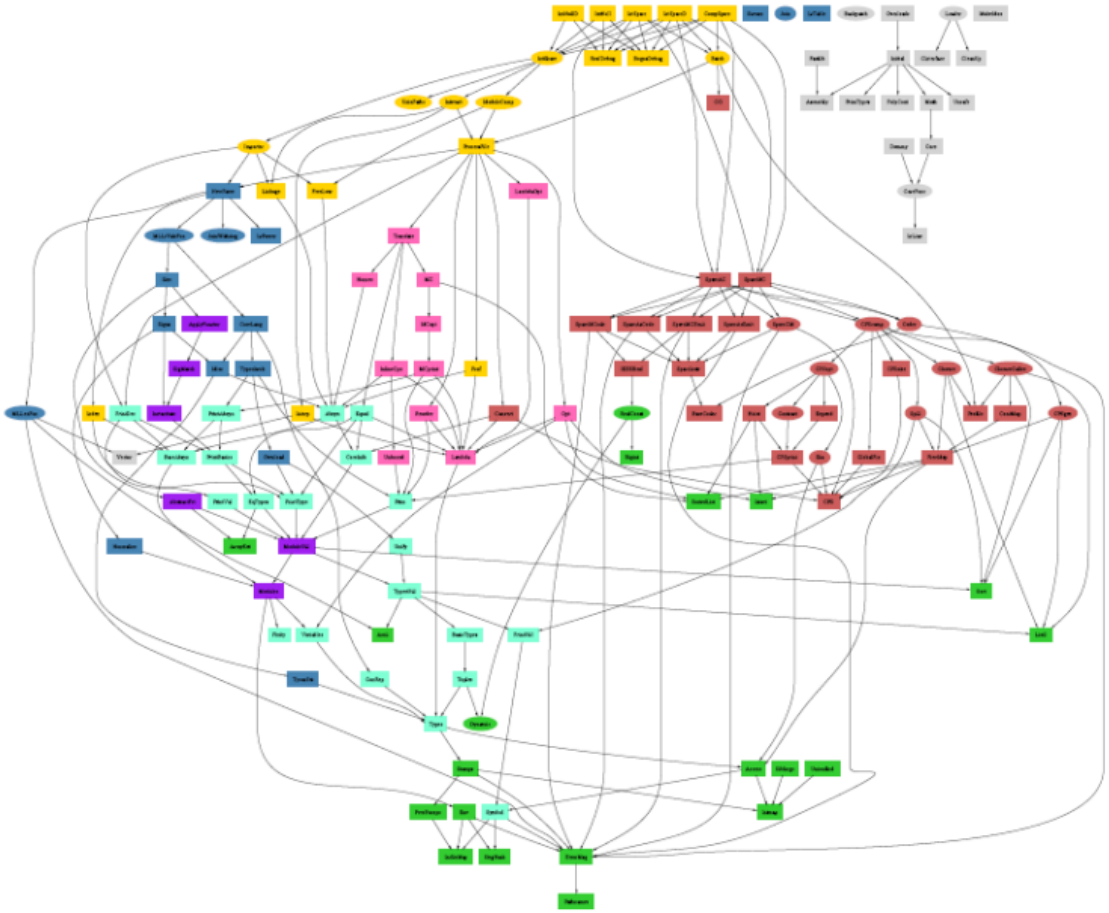
- dot: hierarchical or layered drawings of directed graphs, by avoiding edge crossings and reducing edge length
- neato (Kamada-Kawai) and fdp (Fruchterman-Reinhold with heuristics to handle larger graphs)
- twopi – radial layout
- circo – circular layout

http://www.graphviz.org/

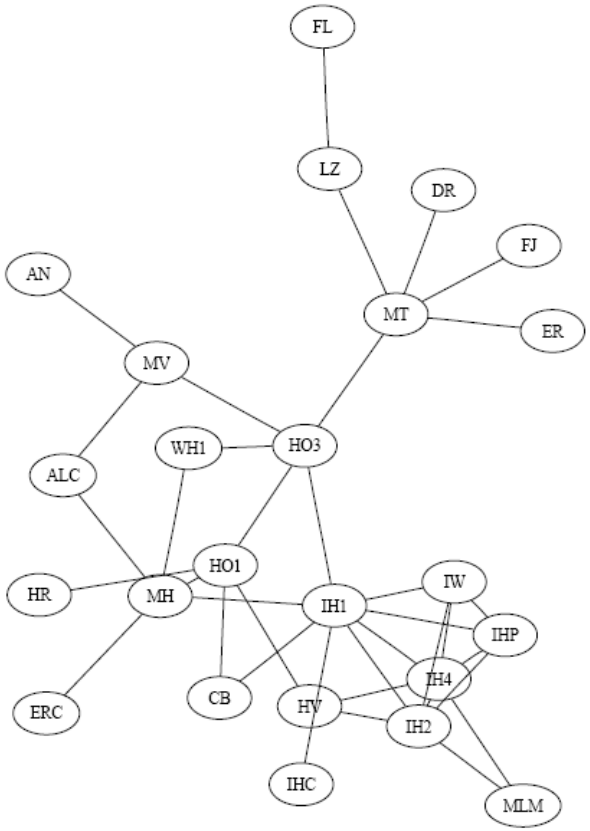
GraphViz: dot language

```
digraph G {
  ranksep=4
  nodesep=0.1
  size="8,11"
ARCH531_20061 [label="ARCH531",style=bold,color=yellow,style=filled]
ARCH531_20071 [label="ARCH531",gstyle=bold,color=yellow,style=filled]
BIT512_20071 [label="BIT512",gstyle=bold,color=yellow,style=filled]
BIT513_20071 [label="BIT513",gstyle=bold,color=yellow,style=filled]
BIT646_20064 [label="BIT646",gstyle=bold,color=yellow,style=filled]
BIT648_20064 [label="BIT648",gstyle=bold,color=yellow,style=filled]
DESCI502_20071 [label="DESCI502",gstyle=bold,color=yellow,style=filled]
ECON500_20064 [label="ECON500",gstyle=bold,color=yellow,style=filled]
…
…
SI791_20064->SI549_20064[weight=2,color=slategray,style="setlinewidth(4)"]SI791_20064-
      >SI596_20071[weight=5,color=slategray,style=bold,style="setlinewidth(10)"]SI791_20064-
      >SI616_20071[weight=2,color=slategray,style=bold,style="setlinewidth(4)"]SI791_20064-
      >SI702_20071[weight=2,color=slategray,style=bold,style="setlinewidth(4)"]SI791_20064-
      >SI719_20071[weight=2,color=slategray,style=bold,style="setlinewidth(4)"]
```

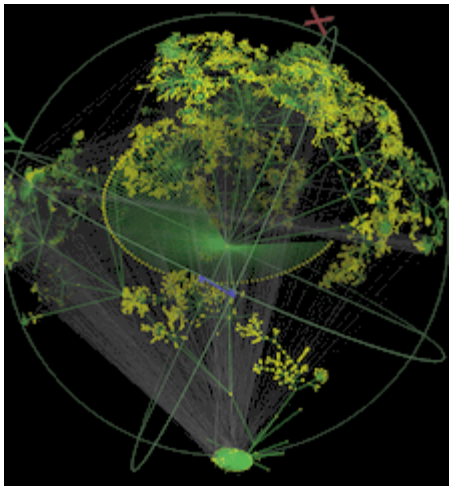
Dot (GraphViz)



Neato (Graphviz)



Other visualization tools: Walrus



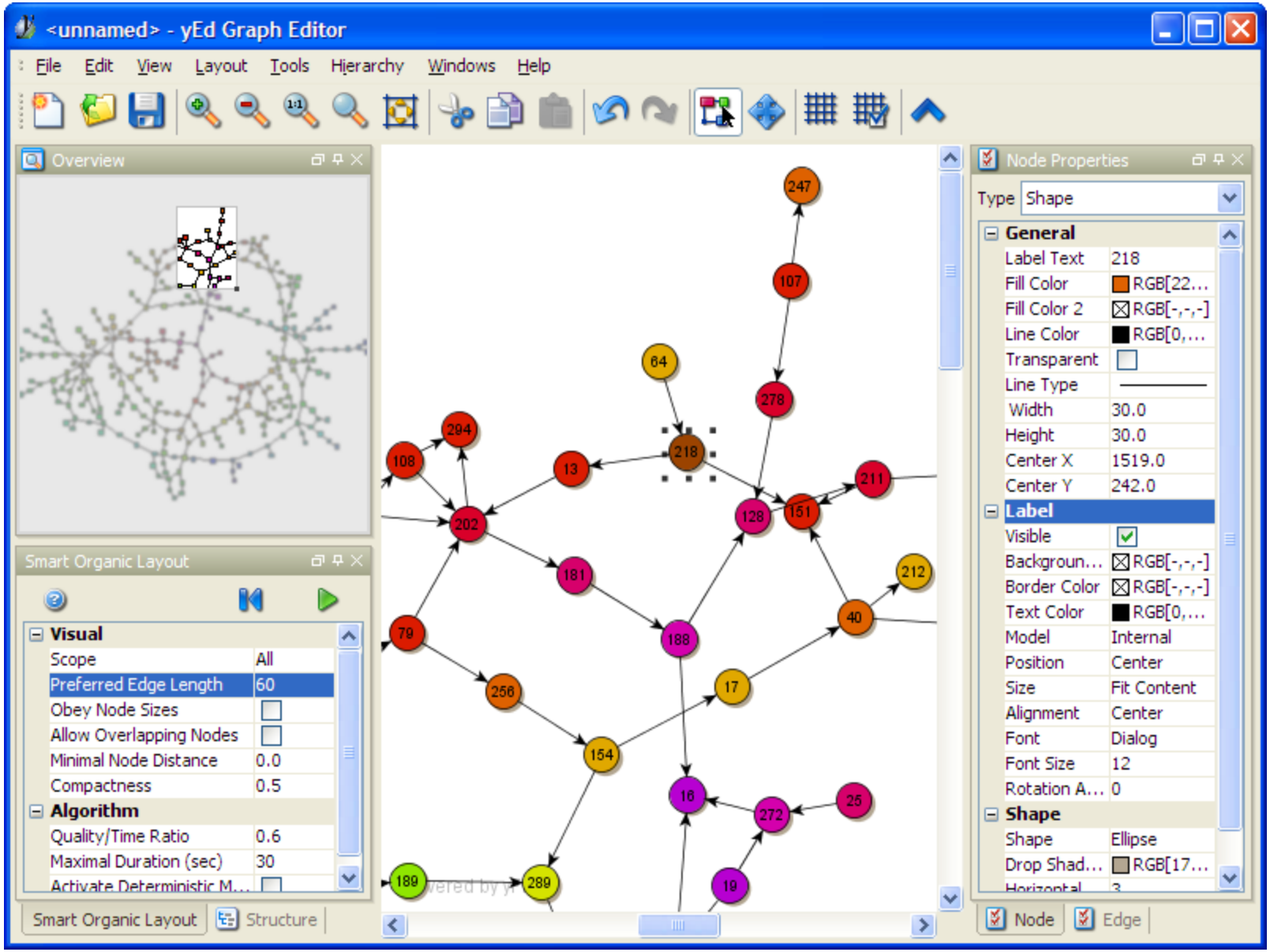
- developed at CAIDA available under the GNU GPL.
- "…best suited to visualizing moderately sized graphs that are nearly trees. A graph with a few hundred thousand nodes and only a slightly greater number of links is likely to be comfortable to work with."
- Java-based
- Implemented Features
  - rendering at a guaranteed frame rate regardless of graph size
  - coloring nodes and links with a fixed color, or by RGB values stored in attributes
  - labeling nodes
  - picking nodes to examine attribute values
  - displaying a subset of nodes or links based on a user-supplied boolean attribute
  - interactive pruning of the graph to temporarily reduce clutter and occlusion
  - zooming in and out

Source: CAIDA, http://www.caida.org/tools/visualization/walrus/

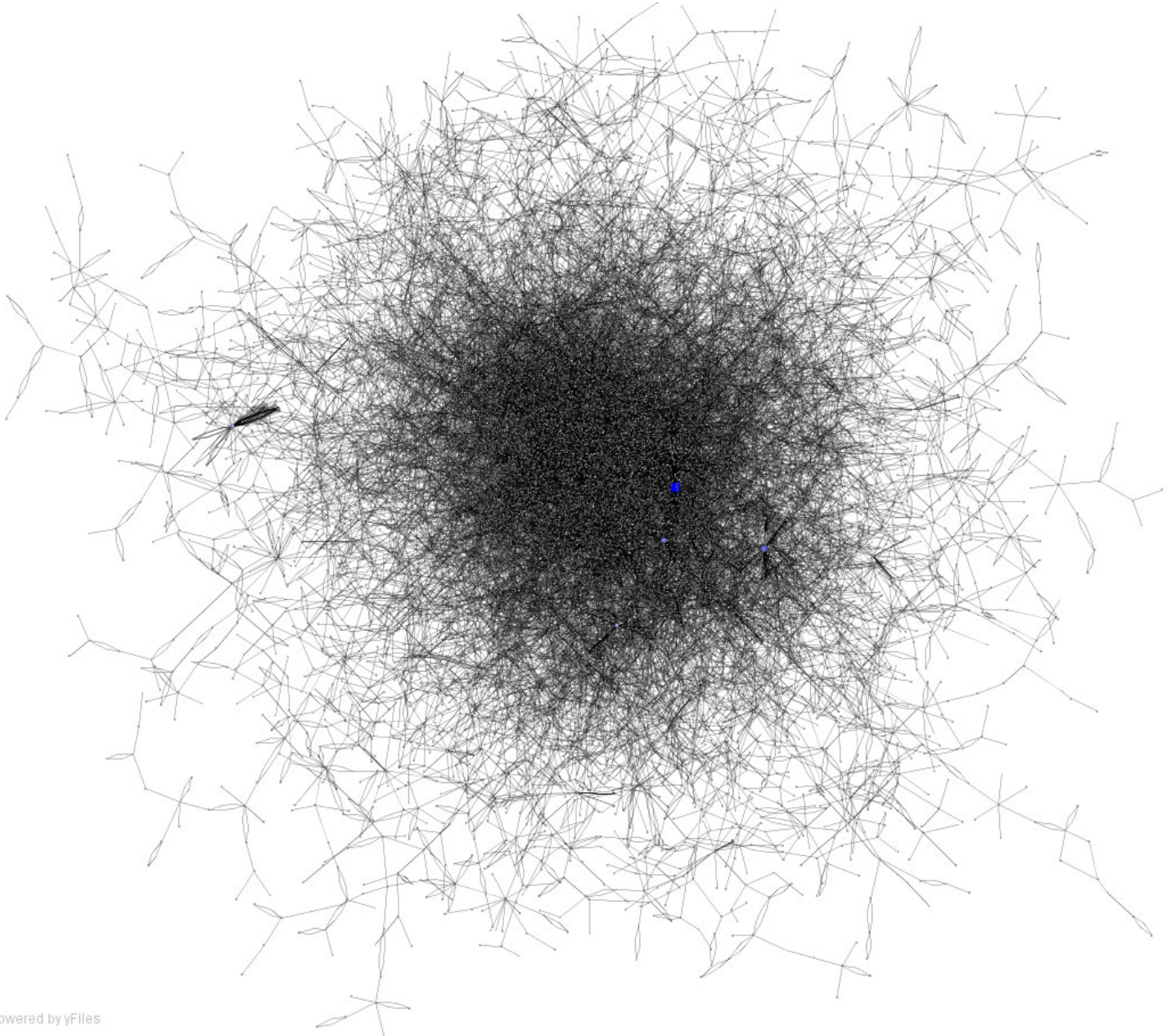
Visualization Tools: yEd - JavaTM Graph Editor

http://www.yworks.com/en/products_yed_about.htm

(good primarily for layouts, maybe free)



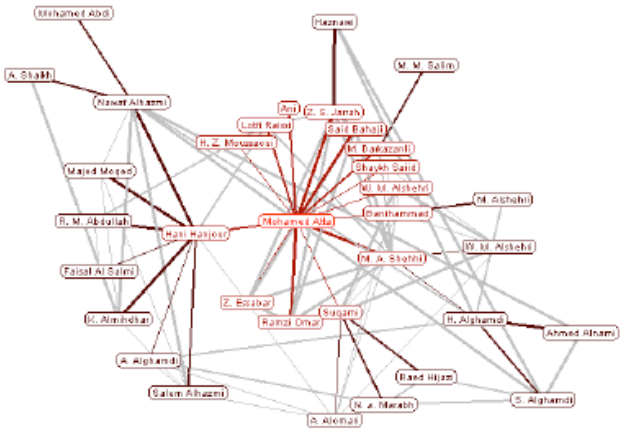
yEd and 26,000 nodes (takes a few seconds)



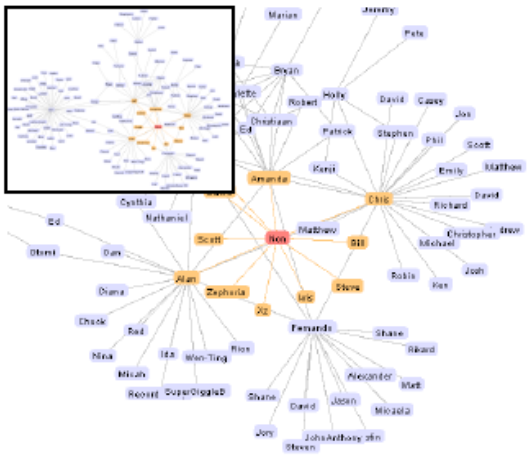
Visualization Tools: Prefuse

- (free) user interface toolkit for interactive information visualization
  - built in Java using Java2D graphics library
  - data structures and algorithms
  - pipeline architecture featuring reusable, composable modules
  - animation and rendering support
  - architectural techniques for scalability
- requires knowledge of Java programming
- website: http://prefuse.sourceforge.net/
  - CHI paper http://guir.berkeley.edu/pubs/chi2005/**prefuse**.pdf

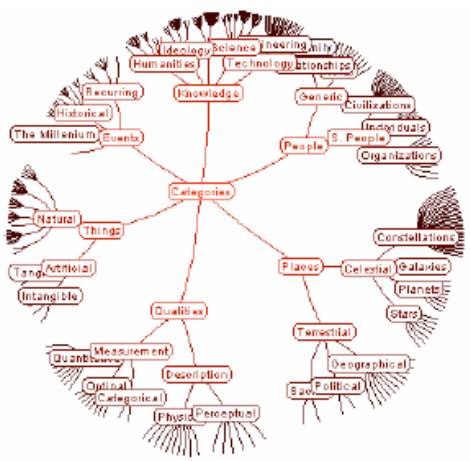
Simple prefuse visualizations



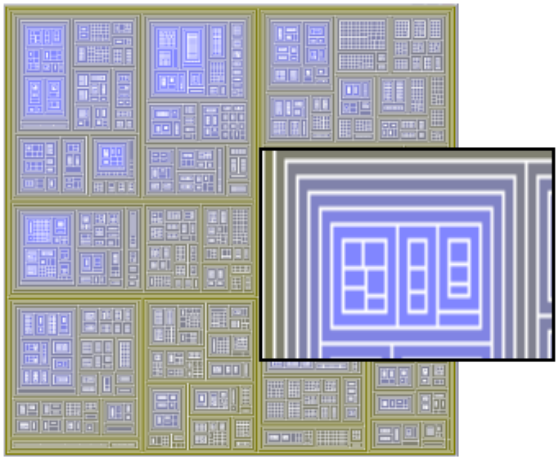
(a) Animated radial layout.



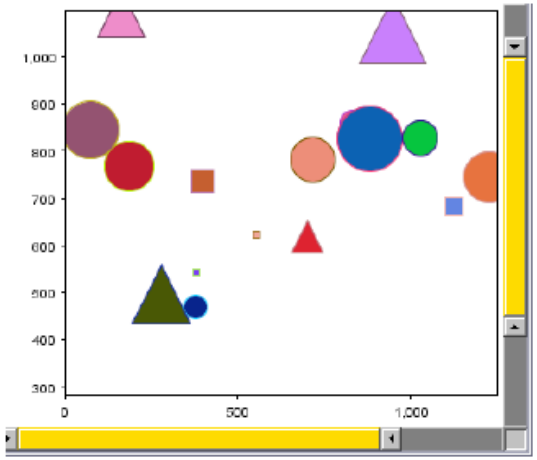
(b) Force-directed layout with overview.



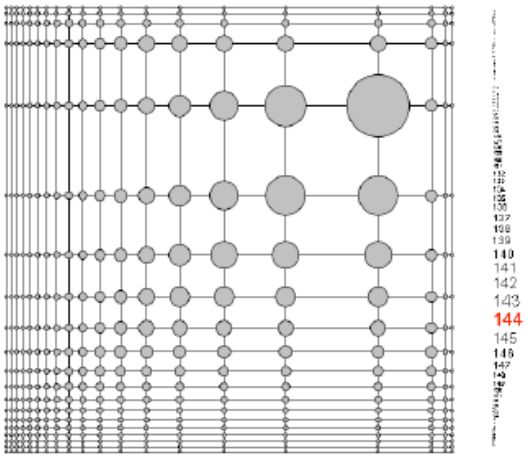
(c) Hyperbolic tree.



(d) TreeMap.



(e) SpotPlot scatterplot.

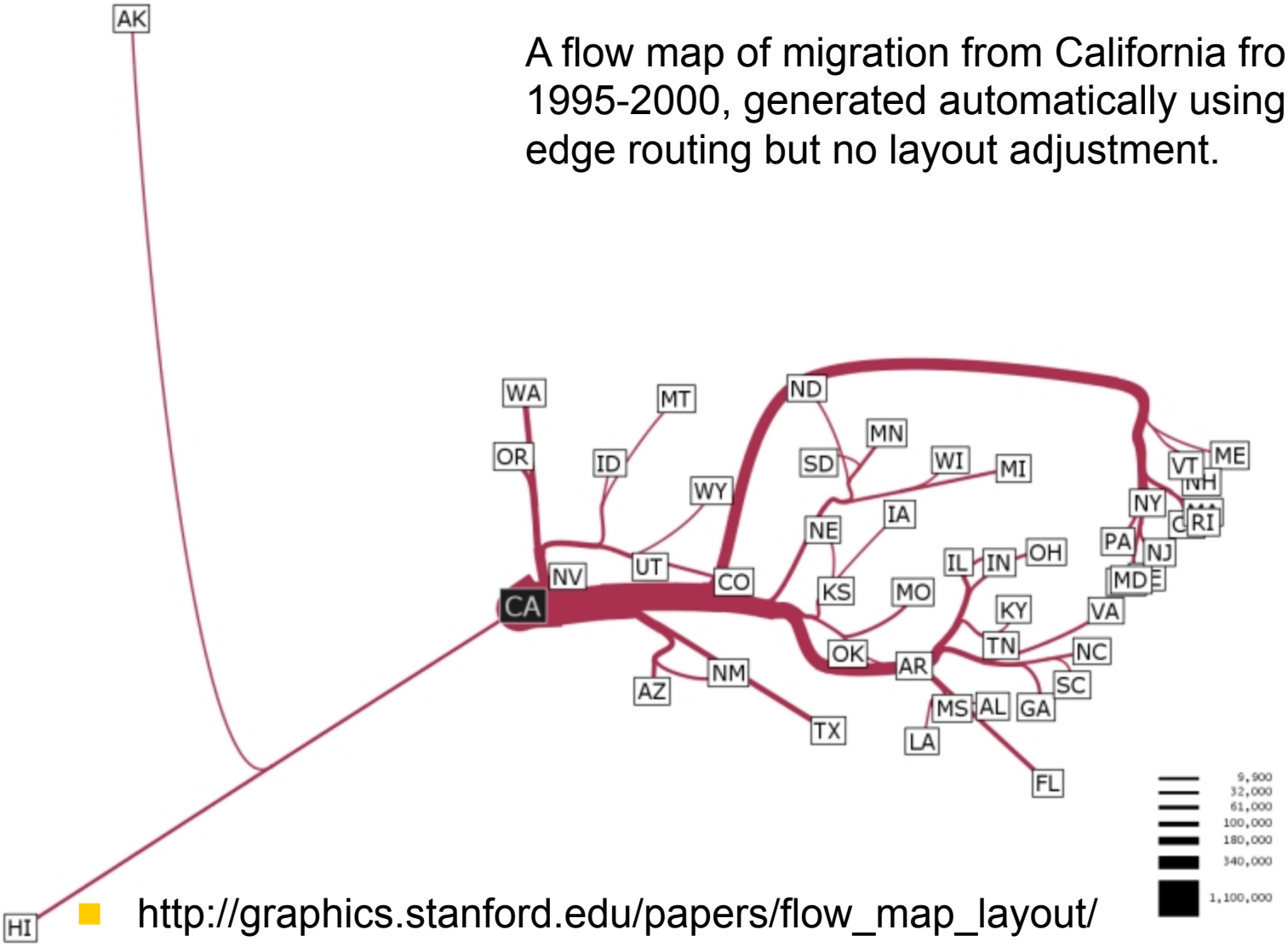


(f) Fisheye graph. (g) Fisheye menu.

Source: Prefuse, http://prefuse.sourceforge.net/

Examples of prefuse applications: flow maps

A flow map of migration from California from 1995-2000, generated automatically using edge routing but no layout adjustment.



- http://graphics.stanford.edu/papers/flow_map_layout/

Examples of prefuse applications: vizster

- http://jheer.org/vizster/