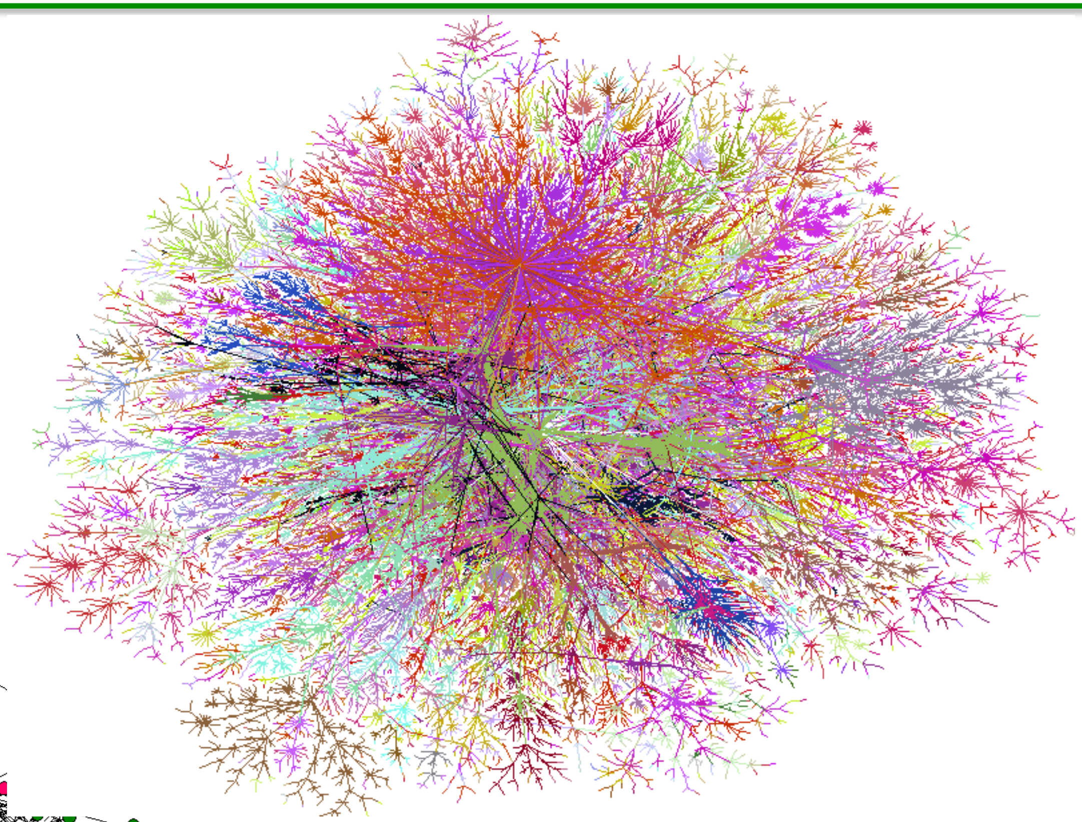
Graphs

CSCI-2270

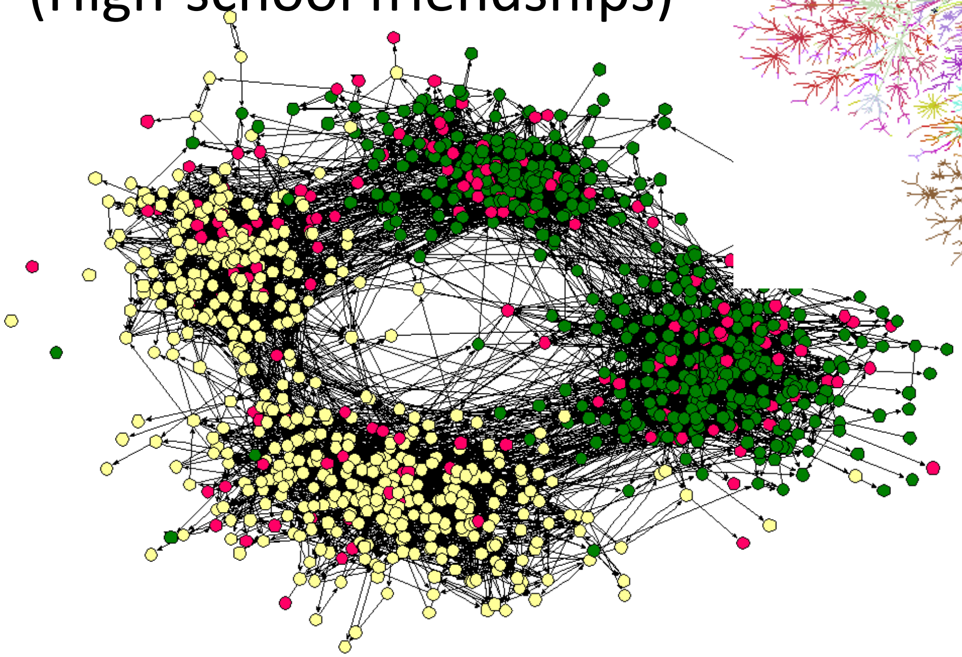
Elizabeth Boese

Dense Graphs

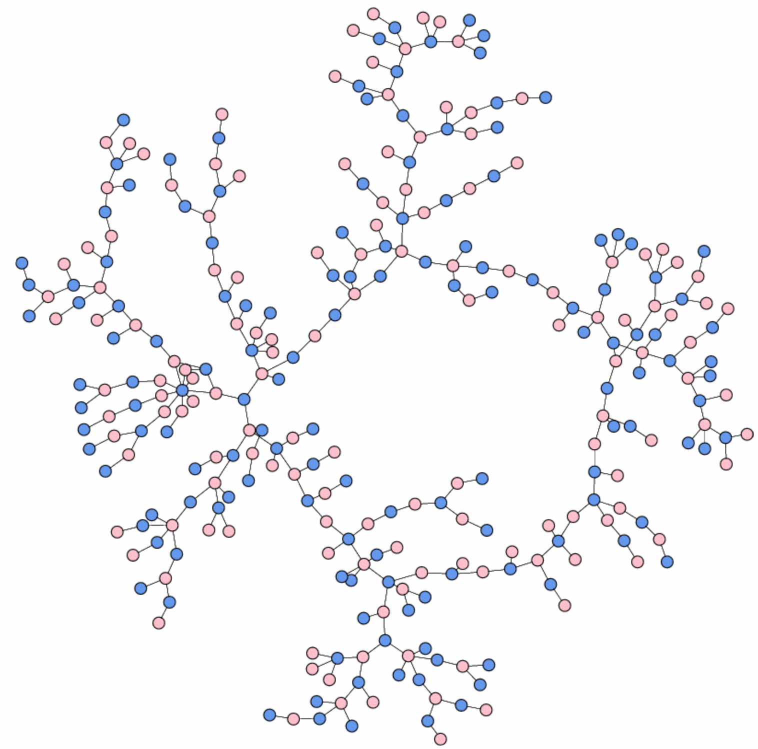


(The Internet)

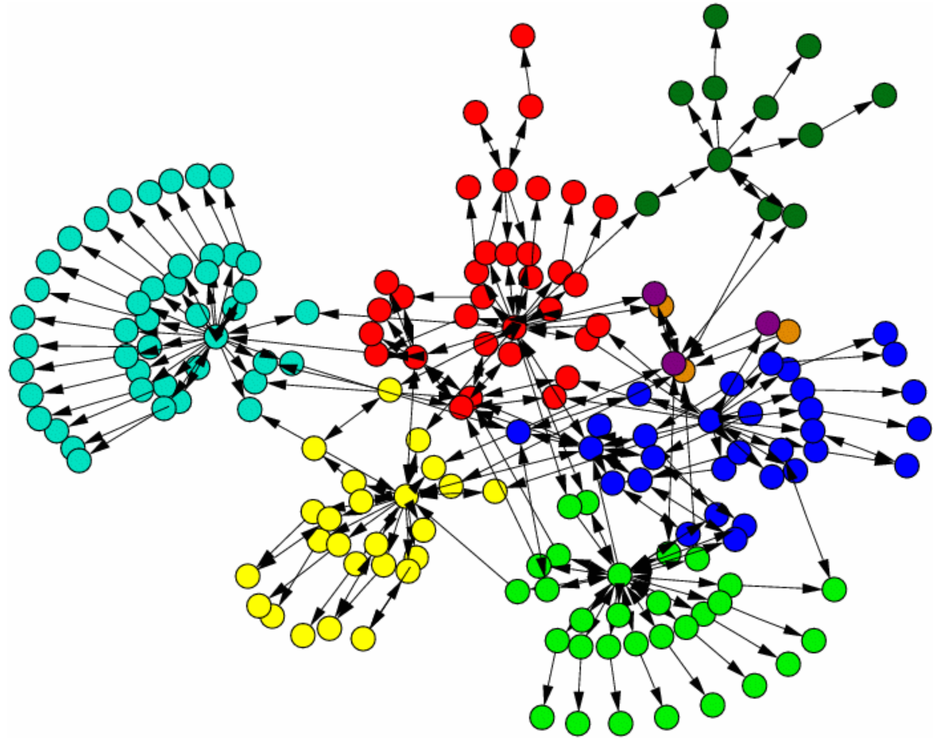
(High-school friendships)



Sparse Graphs

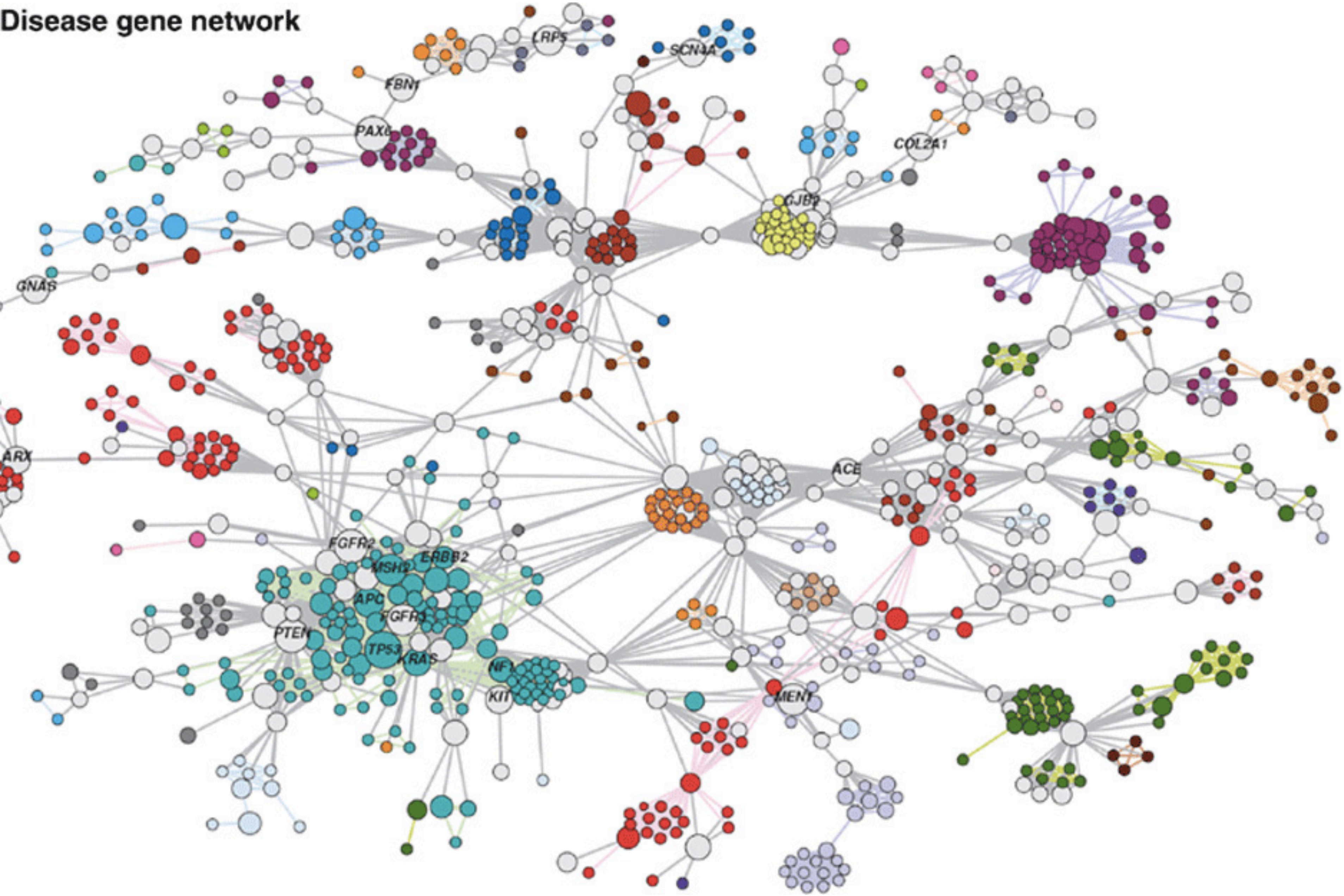


(High-school dating habits)

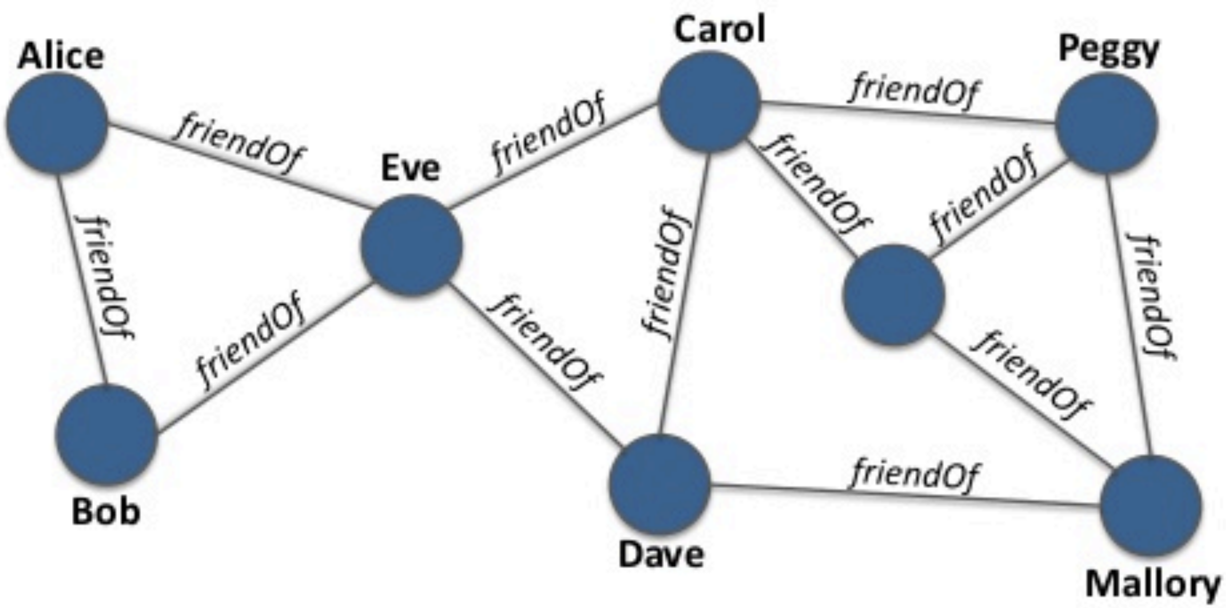


(Intra-website links)

Disease gene network



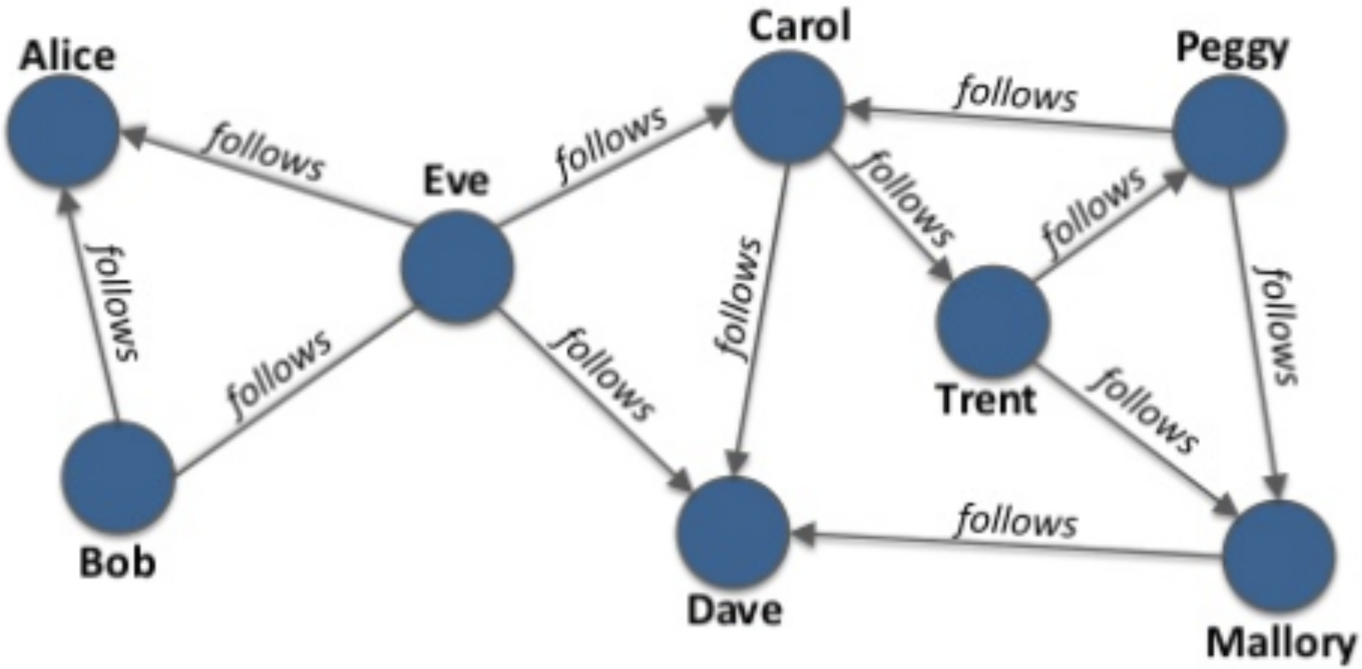
“Graphs are everywhere”



$$Graph = (\mathbf{Users}, Friendships)$$

http://www.slideshare.net/s1ck/meetup-big-data-user-group-dresden-gradoop-scalable-graph-analytics-with-apache-flink

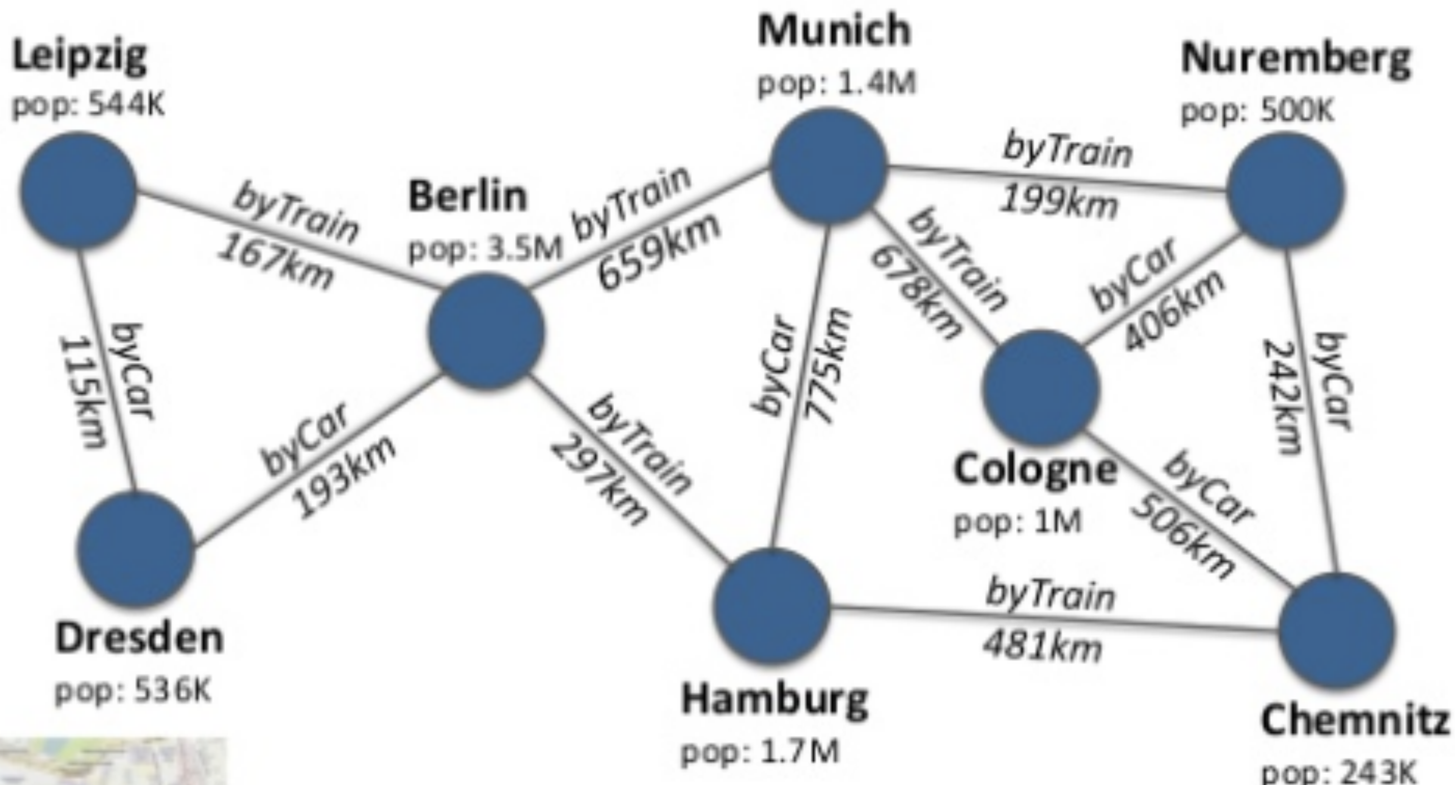
"Graphs are everywhere"



$$Graph = (\mathbf{Users}, Followers)$$

http://www.slideshare.net/s1ck/meetup-big-data-user-group-dresden-gradoop-scalable-graph-analytics-with-apache-flink

"Graphs are everywhere"



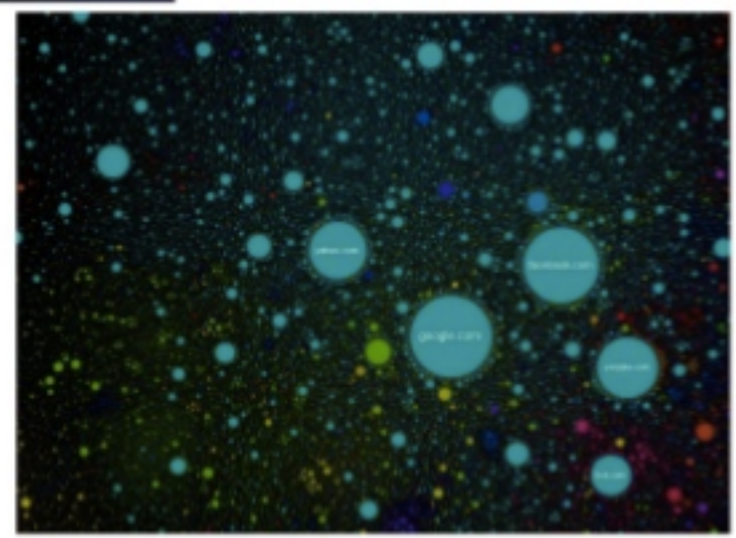
$$Graph = (\mathbf{Cities}, Connections)$$

http://www.slideshare.net/s1ck/meetup-big-data-user-group-dresden-gradoop-scalable-graph-analytics-with-apache-flink

“Graphs are large”

- Facebook
  - ca. 1.49 billion active users
  - ca. 340 friends per user

- World Wide Web
  - ca. 1 billion websites



http://www.slideshare.net/s1ck/meetup-big-data-user-group-dresden-gradoop-scalable-graph-analytics-with-apache-flink

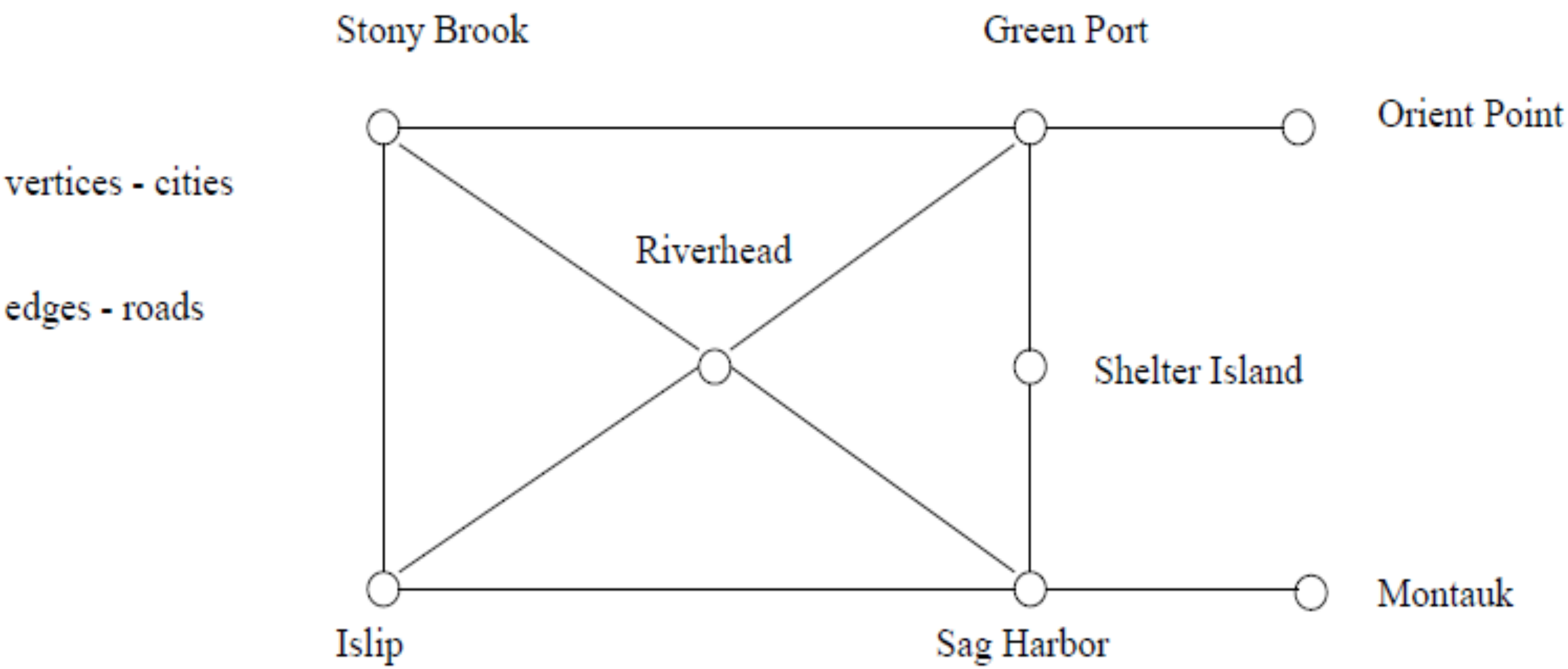
Graphs

- Graphs are one of the unifying themes of computer science.
- A graph **G = (V, E)** is defined by a
  - *set of vertices* ***V***
  - set of *edges* ***E*** *consisting of ordered or unordered pairs of* vertices from **V**

Initially prepared by Dr. Ilyas Cicekli; improved by various Bilkent CS202 instructors.

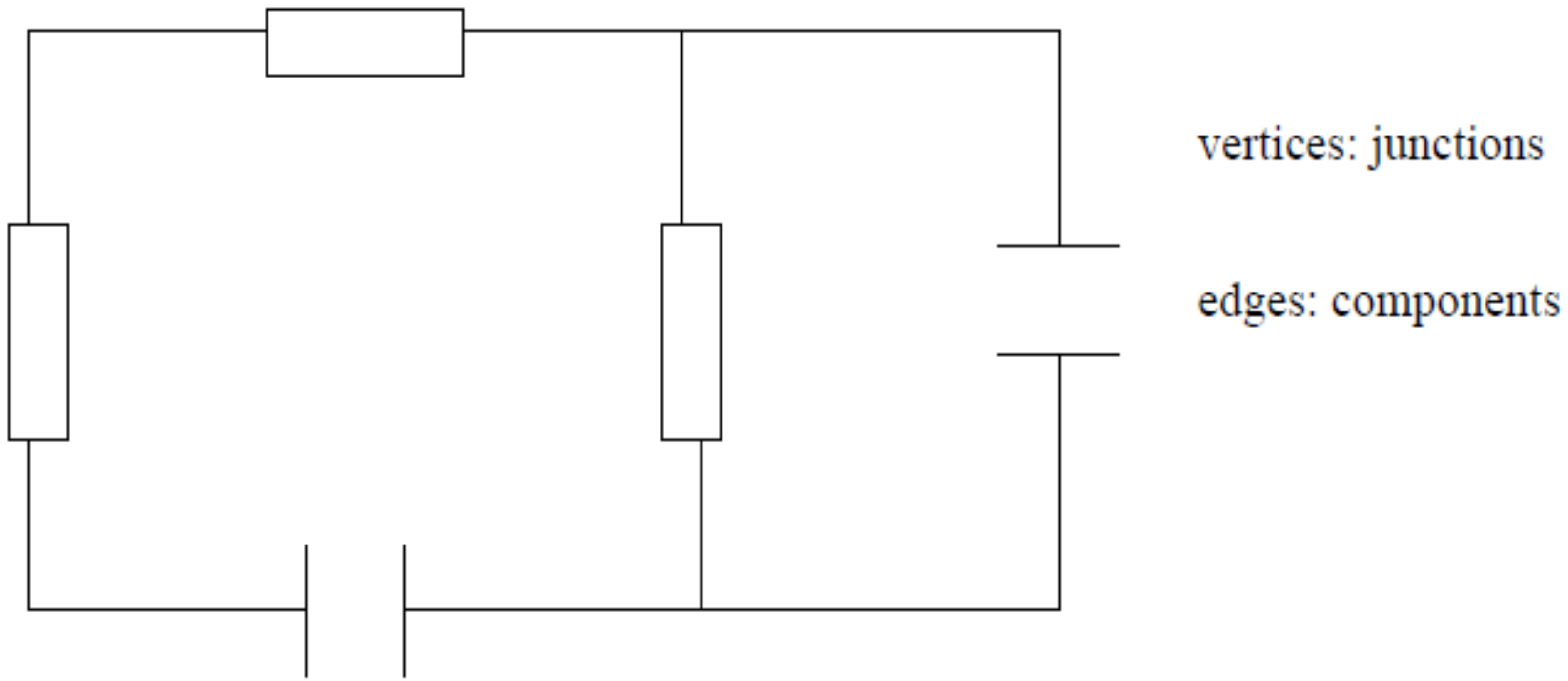
Road Networks

- In modeling a road network, the vertices may represent the cities or junctions, certain pairs of which are connected by roads/edges.



Electronic Circuits

- In an electronic circuit, with junctions as vertices and components as edges.



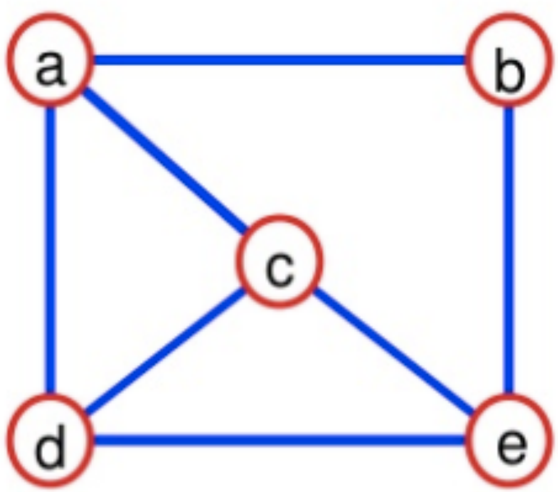
Applications

- Social networks (facebook ...)
- Courses with prerequisites
- Computer networks
- Google maps
- Airline flight schedules
- Computer games
- WWW documents
- ... (so many to list!)

Graphs – Definitions

- A **graph** G = (V, E) consists of
  - V: a set of ***vertices***
  - E: a set of ***edges***
    where each edge is a pair of vertices (v,w) s.t. $v,w \in V$

- Vertices are sometimes called **nodes**,
  edges are sometimes called **arcs**.

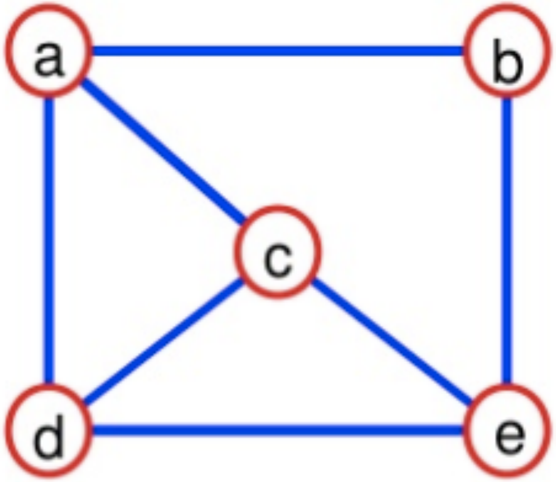


V= {a,b,c,d,e}

E= {(a,b),(a,c),(a,d),
(b,e),(c,d),(c,e), (d,e)}

Graphs

- **Size of a graph**
  - – # nodes
  - – Empty graph = size 0
  - – Example: size 5

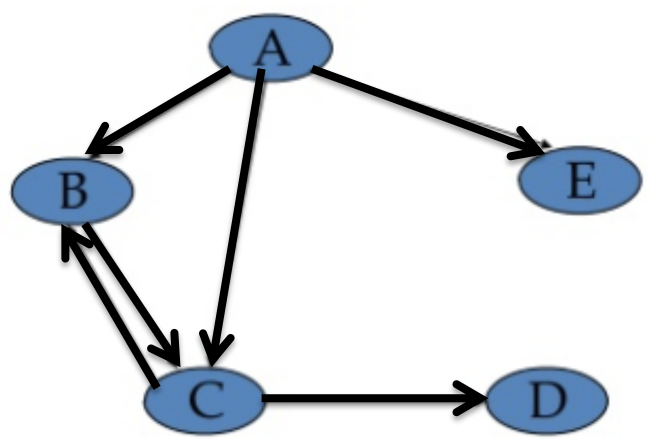


Graphs – Definitions

- **Directed graph (**also called ***digraphs)***
  - If the edge pair is **ordered**

- **Undirected graph**
  - normal graph (which is not a directed graph)
  - When we say **graph** we mean that it is an **undirected graph**.

undirected graph



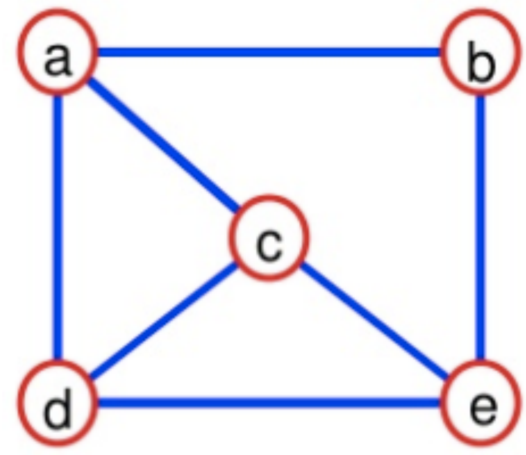
Directed Graph

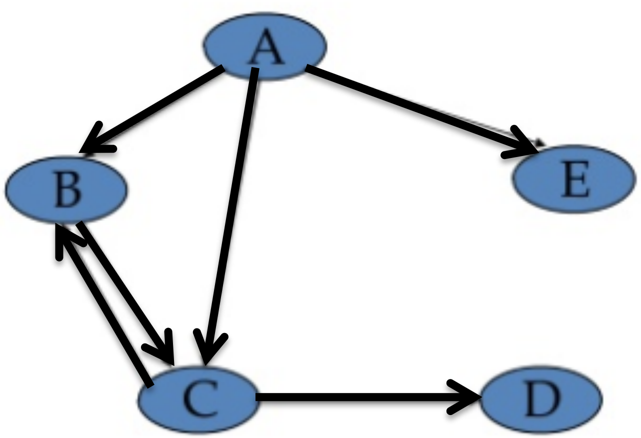
V = [A, B, C, D, E]
E = [<A,B>, <B,C>, <C,B>, <A,C>, <A,E>, <C,D>]

http://www.cs.cornell.edu/Courses/cs2112/2015fa/lectures/lecture.html?id=graphs

Graphs

- **Degree of a node**
  - # of edges
  - Example: degree of c = 3
- Degree of directed graphs
  - **in-degree of a node**:
    # of in-edges
    - E.g., B has 2 in-nodes
    - **Sources:** Vertices with an in-degree of zero
  - **out-degree of a node**:
    # of out-edges
    - E.g., D has 0 out-nodes
    - **Sinks:** Vertices with an out-degree of zero



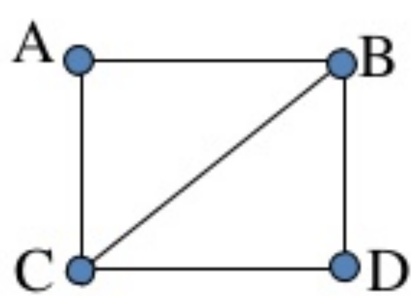


Graphs – Definitions

- Two vertices of a graph are **adjacent** if they are joined by an edge.
- Vertex w is **adjacent to** v iff $(v,w) \in E$.
  - In an undirected graph with edge (v, w) and hence (w,v) w is adjacent to v and v is adjacent to w.

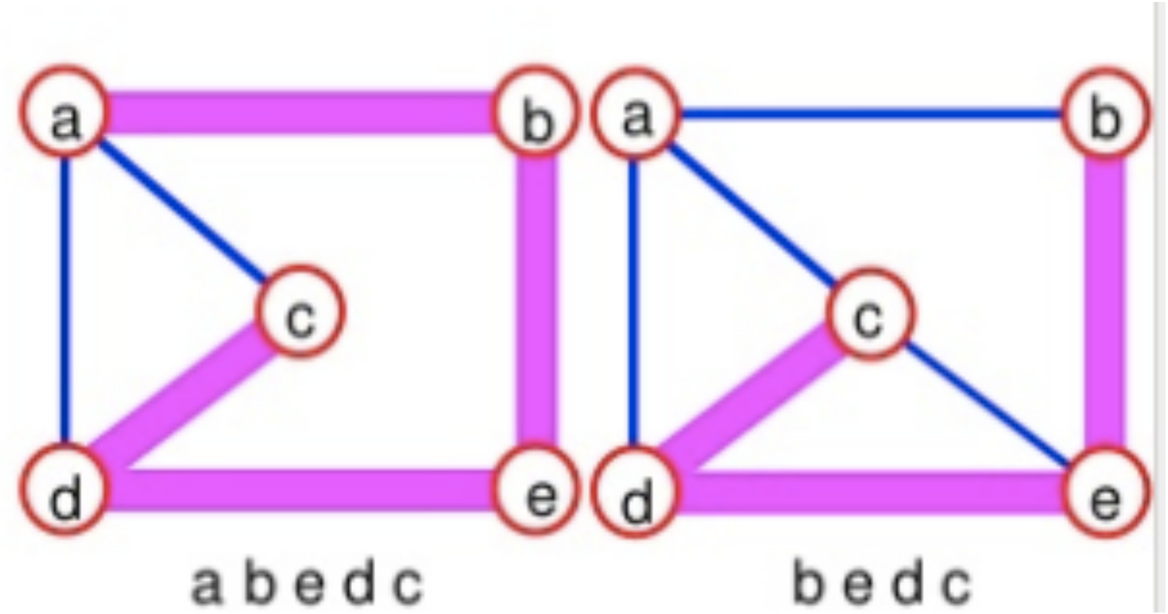
A and *B* are adjacent

A and *D* are not adjacent



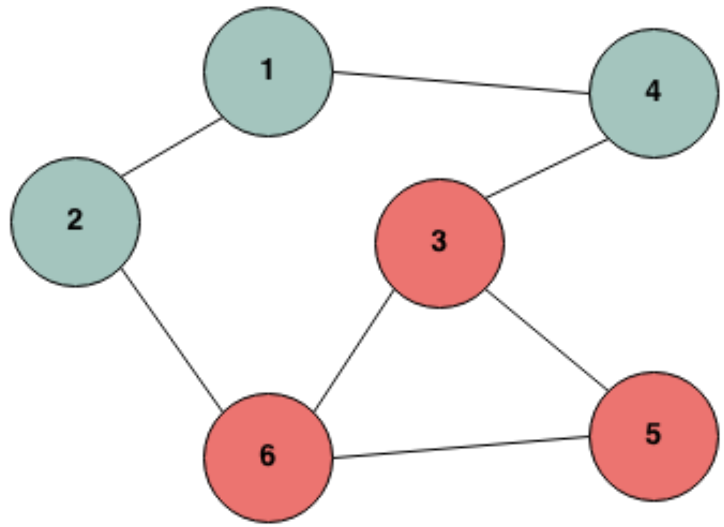
Graphs – Definitions

- A **path** between two vertices is a sequence of edges that begins at one vertex and ends at another vertex.
  - i.e. $w_1, w_2, \ldots, w_N$ is a path if $(w_i, w_{i+1}) \in E$ for $1 \leq i \leq. N-1$
  - A **simple path** passes through a vertex only once.

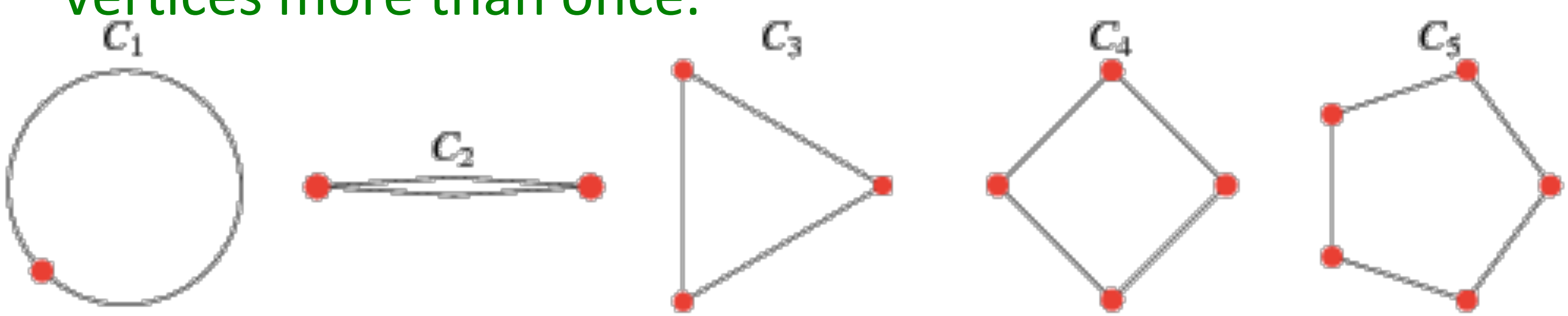


Graphs – Definitions

- A **cycle** is a path that begins and ends at the same vertex.

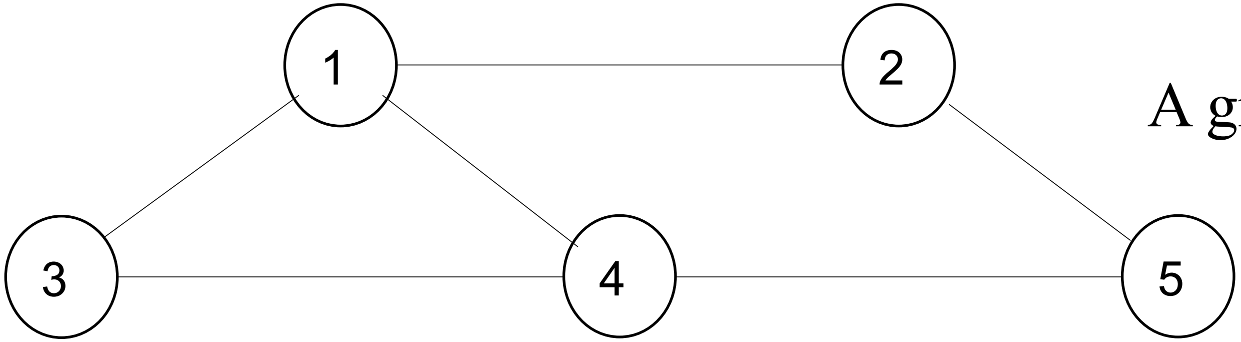


  - A **simple cycle** is a cycle that does not pass through other vertices more than once.



http://algorithmsandme.in/2014/06/graphs-detecting-cycle-in-undirected-graph/

Graphs – An Example



A graph G (undirected)

The graph G = (V, E) has 5 vertices and 6 edges:

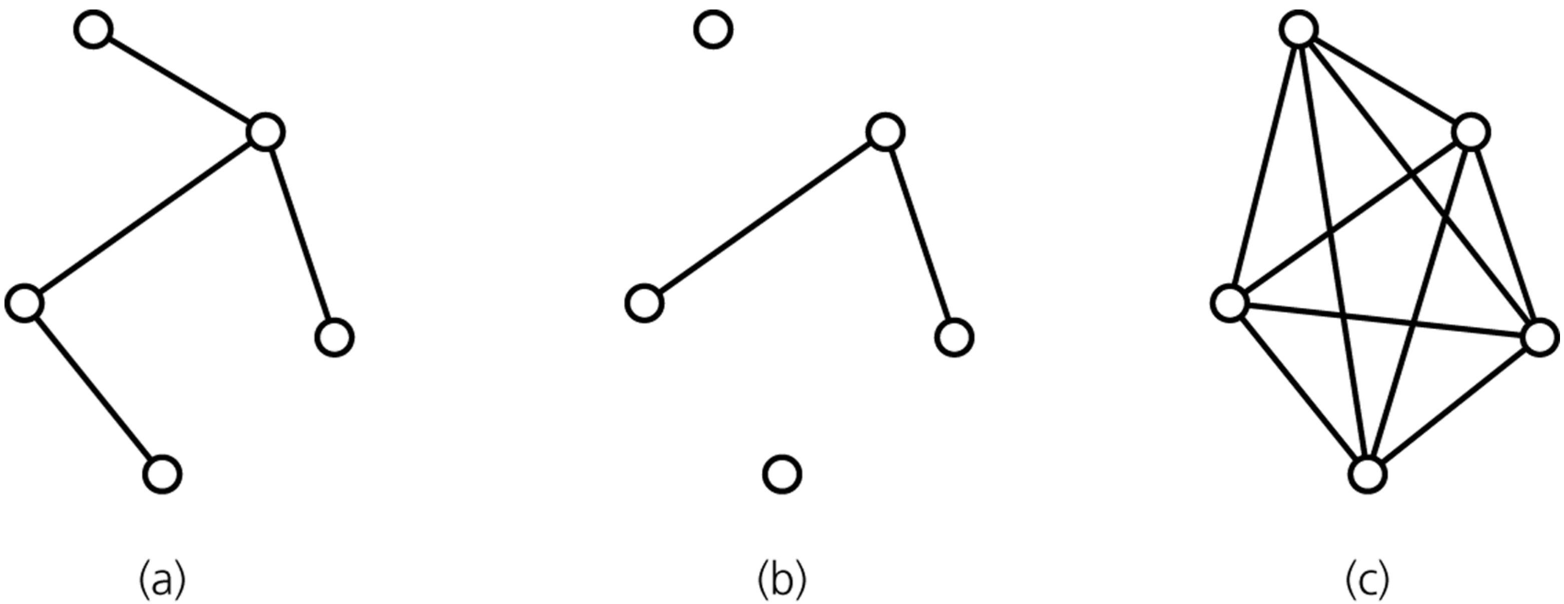
V = {1, 2, 3, 4, 5}

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

- ***Adjacent***:
  1 and 2 are adjacent -- 1 is adjacent to 2 and 2 is adjacent to 1
- ***Path***:
  1, 2, 5 (a simple path), 1, 3, 4, 1, 2, 5 (a path but not a simple path)
- ***Cycle***:
  1, 3, 4, 1 (a simple cycle), 1, 3, 4, 1, 4, 1 (cycle, but not simple cycle)

Graphs – Definitions

- A **connected graph** has a path between each pair of distinct vertices.
- A **complete graph** has an edge between each pair of distinct vertices.
  - A complete graph is also a connected graph.
    But a connected graph may not be a complete graph.

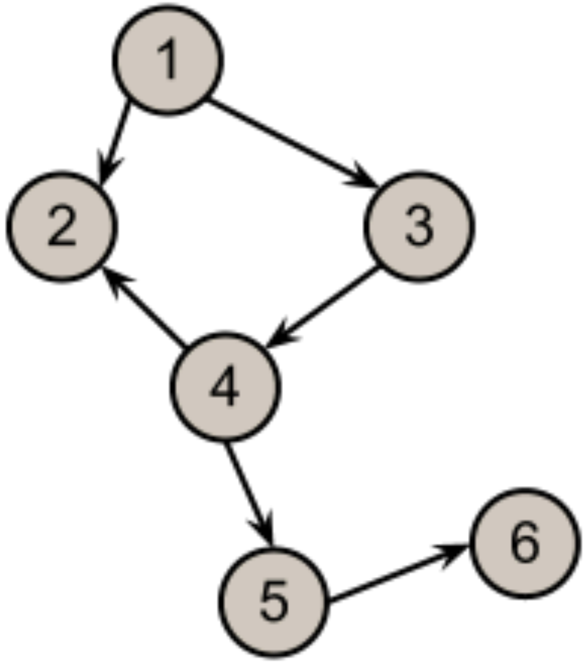


connected **disconnected** **complete**

DIRECTED GRAPHS

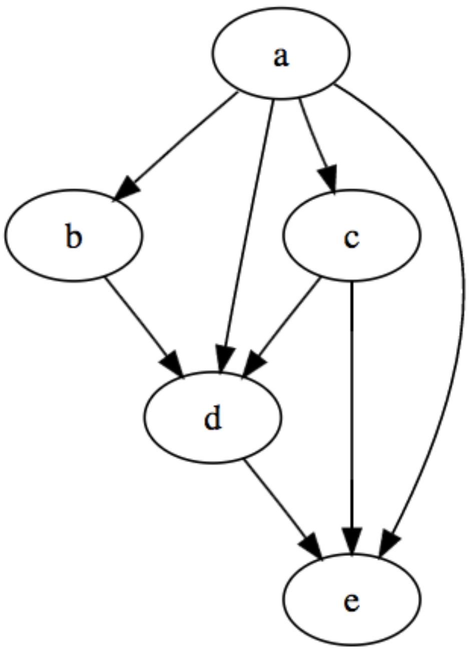
Directed Graphs

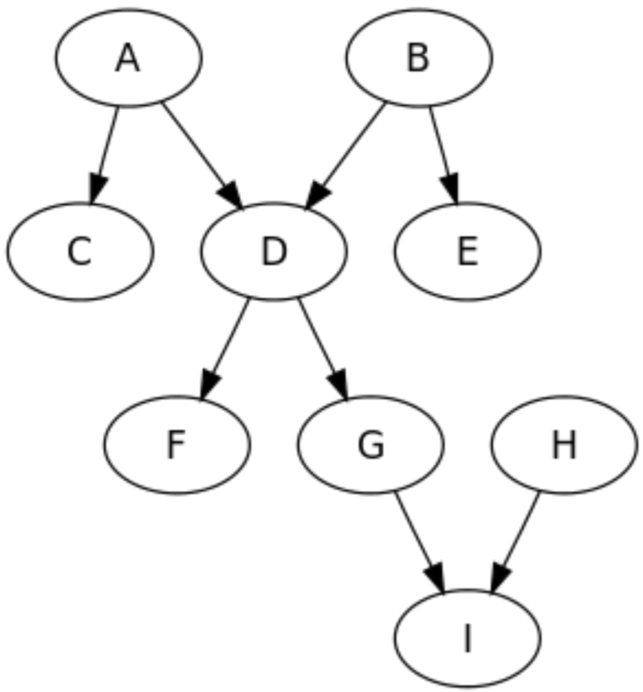
- A **directed path** between two vertices is a sequence of directed edges that begins at one vertex and ends at another vertex.
  - i.e. $w_1, w_2, \ldots, w_N$ is a path if $(w_i, w_{i+1}) \in E$ for $1 \leq i \leq N-1$
  - E.g.,
    3, 4, 5, 6

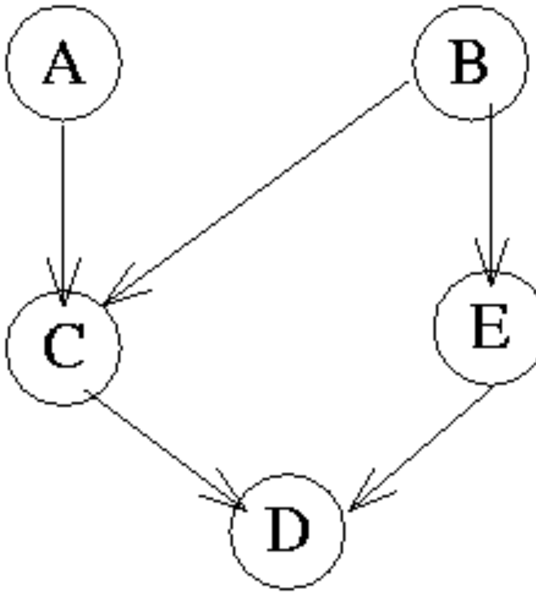


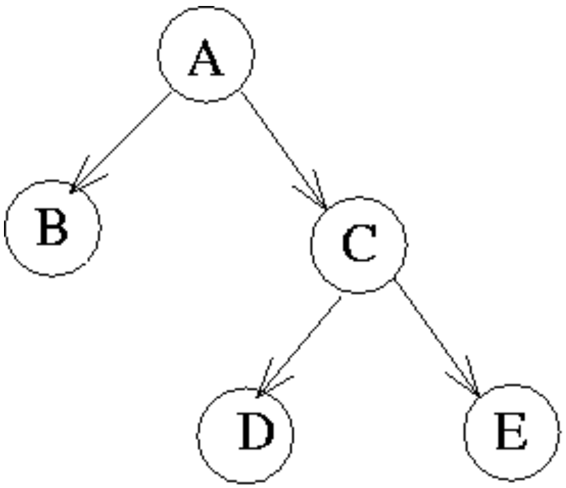
Directed Graphs

- A **directed acyclic graph** (***DAG***) is a directed graph with no cycles.



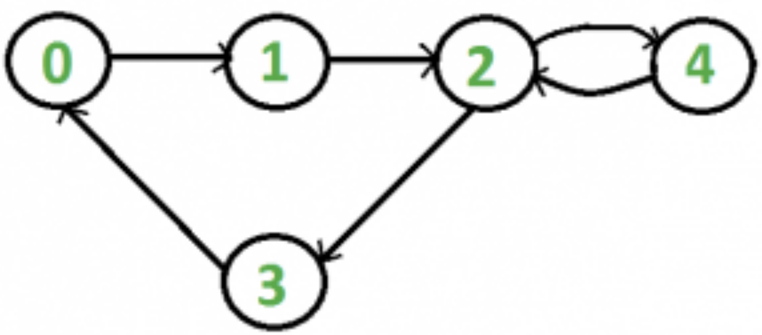




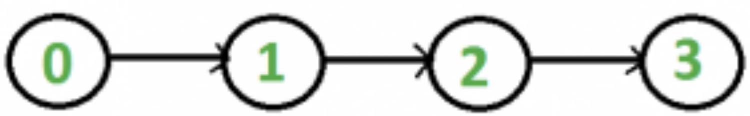


Directed Graphs

- An <u>undirected graph</u> is **connected** if there is a path from every vertex to every other vertex.
- A <u>directed graph</u> with this property is called **strongly connected**.
  - If a directed graph is not strongly connected, but the underlying graph (without direction to arcs) is connected then the graph is **weakly connected**.



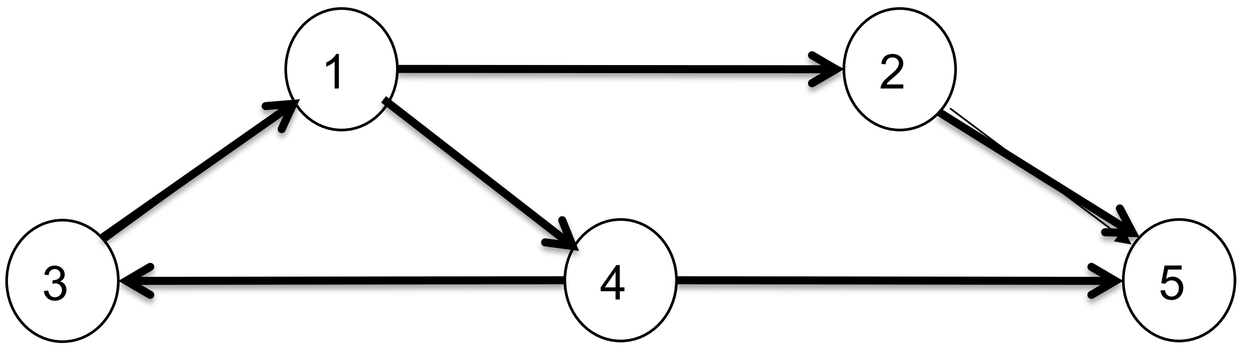
Strongly Connected



Not Strongly Connected

http://www.geeksforgeeks.org/connectivity-in-a-directed-graph/

Directed Graph – An Example



The graph G= (V, E) has 5 vertices and 6 edges:

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

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

- ***Adjacent***:

  2 is adjacent to 1, but 1 is NOT adjacent to 2
- ***Path***:

  1, 2, 5 (a directed path),
- ***Cycle***:

  1, 4, 3, 1 (a directed cycle),

Directed acyclic graphs

Applications of directed acyclic graphs include:

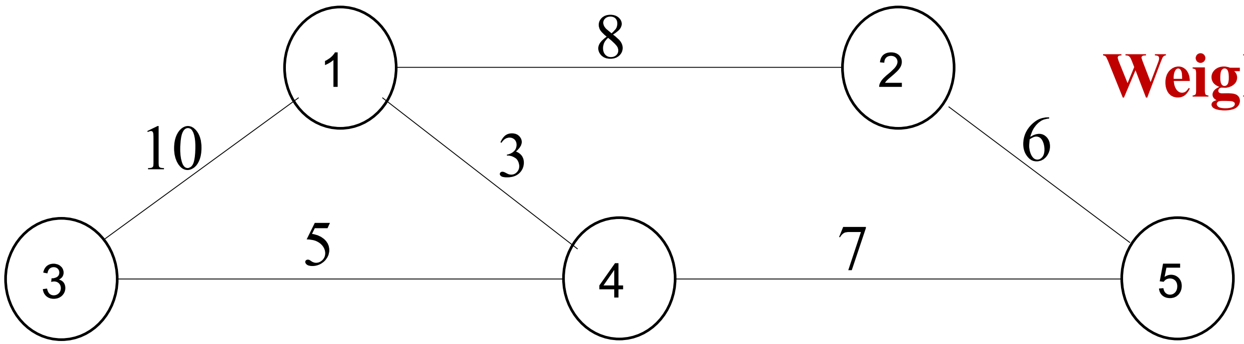
- The parse tree constructed by a compiler
- A reference graph that can be garbage collected using simple reference counting
- Dependency graphs such as those used in instruction scheduling and **`makefiles`**
- Dependency graphs between classes formed by inheritance relationships in object-oriented programming languages
- Information categorization systems, such as folders in a computer
- Directed acyclic word graph data structure to memory-efficiently store a set of strings (words)

Reference: http://en.wikipedia.org/wiki/Directed_acyclic_graph

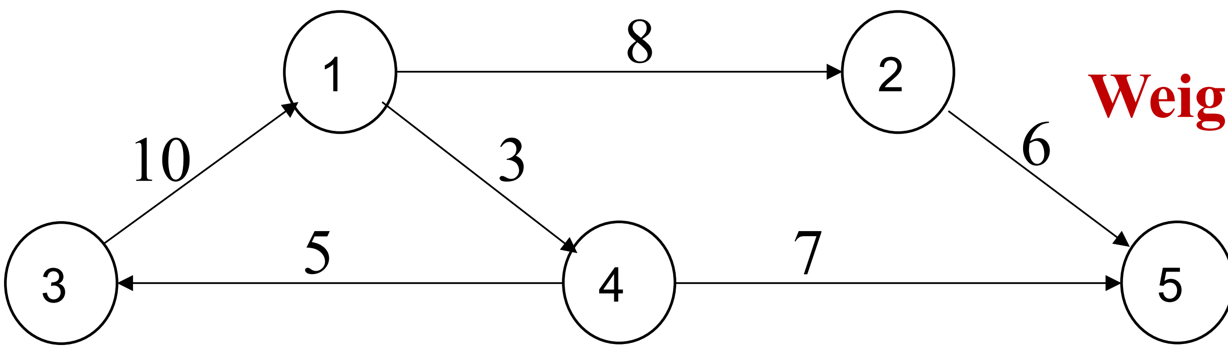
WEIGHTED GRAPHS

Weighted Graph

- We can label the edges of a graph with numeric values, the graph is called a **weighted graph**.



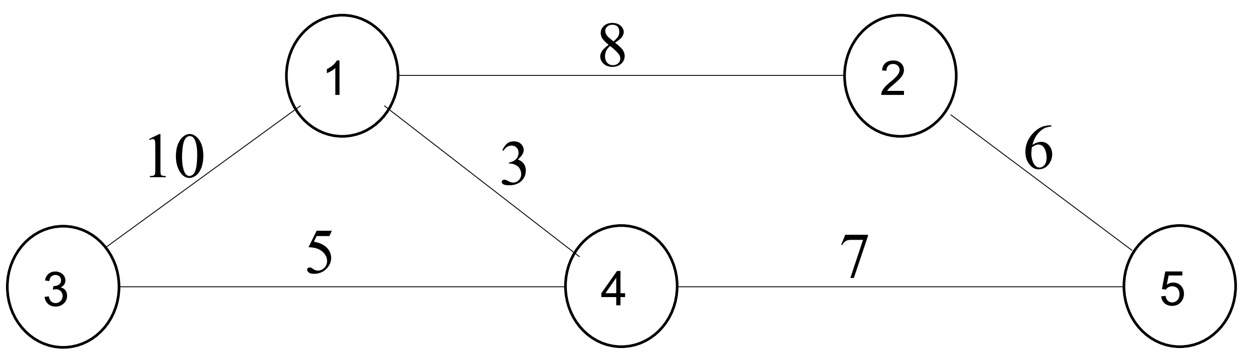
Weighted (Undirect) Graph



Weighted Directed Graph

Weighted graphs

- A weight may be associated with each edge in a graph
  - This could represent distance, energy consumption, cost, etc.
  - Such a graph is called a *weighted graph*
- The ***length*** of a path within a weighted graph is the sum of all of the edges which make up the path
  - The length of the path (1, 4, 5) in the following graph is 3+7= 10
  - The length of the path (1, 3, 4, 5) is 10 + 5 + 7 = 22

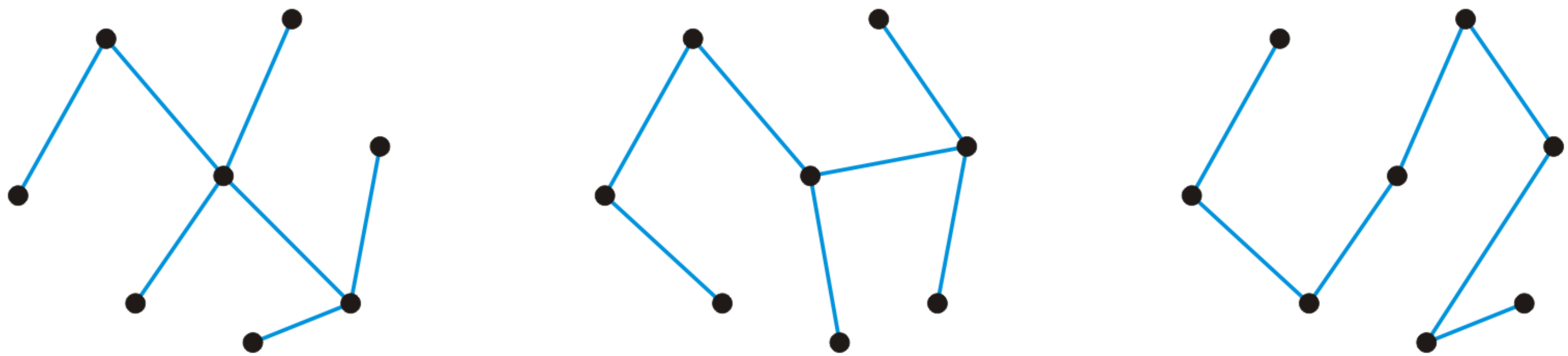


THE TREE IN THE FOREST

Trees

A graph is a tree if it is connected and there is a unique path between any two vertices

– Three trees on the same eight vertices



Consequences:

– The number of edges is $|E| = |V| - 1$
– The graph is *acyclic*, that is, it does not contain any cycles
– Adding one more edge must create a cycle
– Removing any one edge creates two disjoint non-empty sub-graphs

Trees

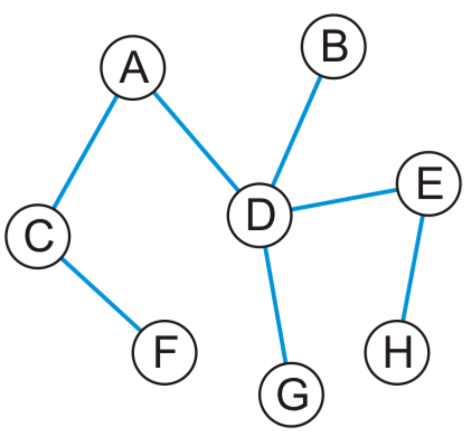
Any tree can be converted into a rooted tree by:

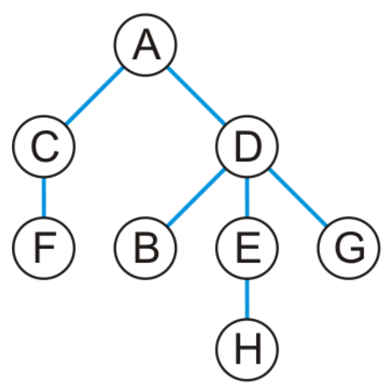
- Choosing any vertex to be the root
- Defining its neighboring vertices as its children

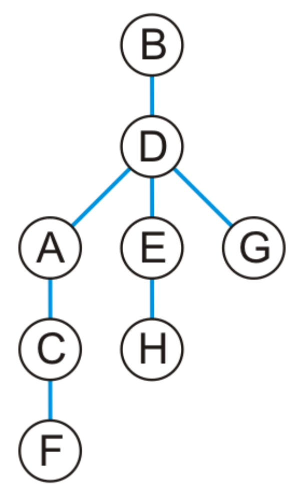
and then recursively defining:

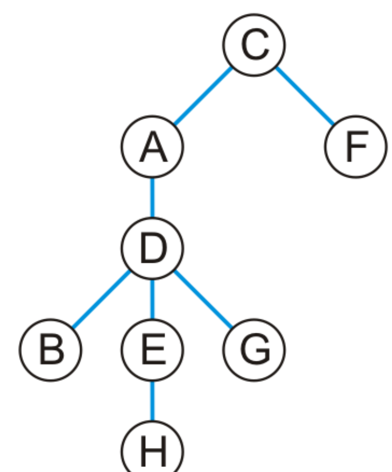
- All neighboring vertices other than that one designated its parent are now defined to be that vertices children

Given this tree, here are three rooted trees associated with it









Douglas W. Harder

Forests

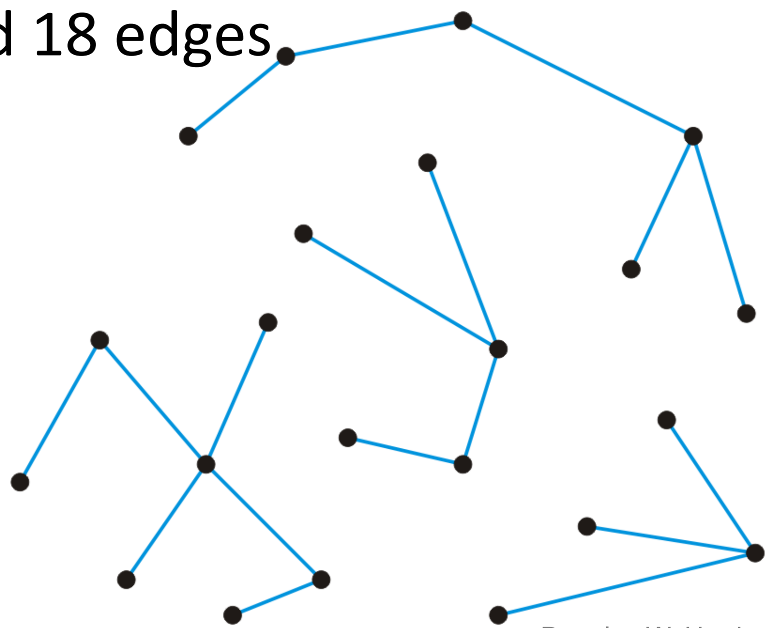
A forest is any graph that has no cycles

Consequences:

- The number of edges is $|E| < |V|$
- The number of trees is $|V| - |E|$
- Removing any one edge adds one more tree to the forest

Here is a forest with 22 vertices and 18 edges

- There are four trees



QUESTIONS TO PONDER

Questions

1. Which of the following are:
    a) Weakly connected
    b) Strongly connected
    c) Directed cycle