Graphs I: Notation and Basics

Reading

- Malik Magdon-Ismail. Discrete Mathematics and Computing.
  - Chapter 11

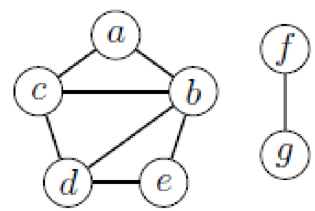
Overview

- Graph basics and notation
  - Equivalent graphs: isomorphism
- Degree sequence
  - Handshaking Theorem
- Trees
- Planar graphs
- Other types of graphs: multigraph, weighted, directed
- Problem solving with Graphs

Graph Basics and Notation

- Graphs model relationships:
  - friendships (e.g. social networks)
  - connectivity (e.g. cities linked by highways)
  - conflicts (e.g. radio-stations with listener overlap)

<u>Graph G</u>



Vertices (aka nodes): a b c d e f g $\qquad V = \{a, b, c, d, e, f, g\}$.

Edges: a–b, a–c, b–c, b–d, b–e, c–d, d–e, f–g

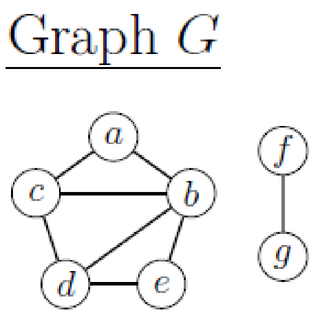
$$E = \begin{Bmatrix} (a,b), (a,c), (b,c), (b,d), \\ (b,e), (c,d), (d,e), (f,g) \end{Bmatrix}.$$

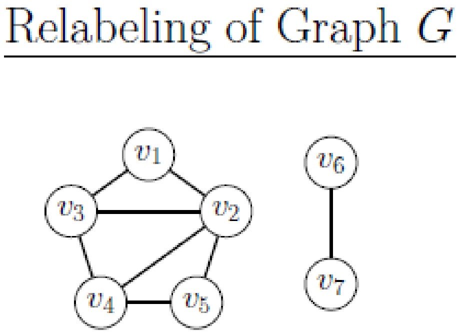
Degree: Number of relationships $\qquad e.g.$, degree$(b) = 4$.

Path: a—c—b—e—d—b $\qquad p = acbedb$.

Graph Isomorphism

- Suppose we relabel the nodes in G to $v_1, \ldots, v_7$





- Relabel nodes (i.e., give them new names) as follows:

$$a \to v_1, b \to v_2, c \to v_3, d \to v_4, e \to v_5, f \to v_6, g \to v_7$$

- What is the new set of vertices:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$

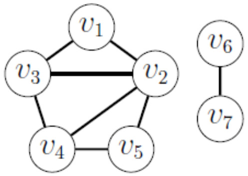
- How about edges?

$$E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_2, v_5), (v_3, v_4), (v_4, v_5), (v_6, v_7)\}$$

- If two graphs can be relabeled with $v_1, \ldots, v_n$, giving the *same* edge set, they are equivalent – *isomorphic*.
- **Practice**. Exercise 11.2.

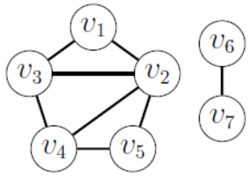
Paths and Connectivity

Graph, G



- A *path* from v_1 to v_2 is a sequence of vertices (start is v_1 and end is v_2):
  - e.g., $v_1 v_3 v_2 v_5 v_4 v_2$
- There is an edge in the graph between consecutive vertices in the path.
  - e.g., v_1 and v_2 are *connected*.
- The length of a path is the number of edges traversed (e.g., 5).
- *Cycle:* path that starts and ends at a vertex without repeating any edge:
  - e.g., $v_1 v_2 v_3 v_1$
- Vertices v_1 and v_6 are not connected by a path.
- The graph G is *not* connected (*every* pair of vertices must be connected by a path).
- How can we make *G* connected?
  - Add any edge from vertices $\{v_1, v_2, v_3, v_4, v_5\}$ to vertices $\{v_6, v_7\}$

Graph Representation



Graph, G

- Adjacency list

$$v_1: v_2, v_3$$
$$v_2: v_1, v_3, v_4, v_5$$
$$v_3: v_1, v_2, v_4$$
$$v_4: v_2, v_3, v_5$$
$$v_5: v_2, v_4$$
$$v_6: v_7$$
$$v_7: v_6$$

- Adjacency matrix

$$\begin{array}{c c} & \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

  – More wasted memory; faster algorithms.

  – Small redundancy: every edge is "represented" twice.

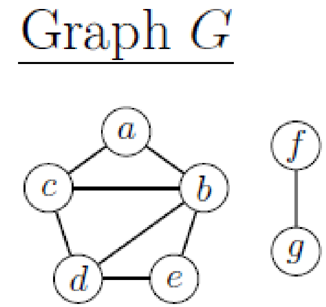
Degree Sequence

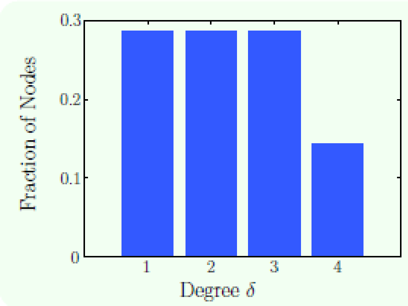
Rensselaer

- A node's degree is the number of its neighbors
  - degree δ_i = number of v_i's neighbors

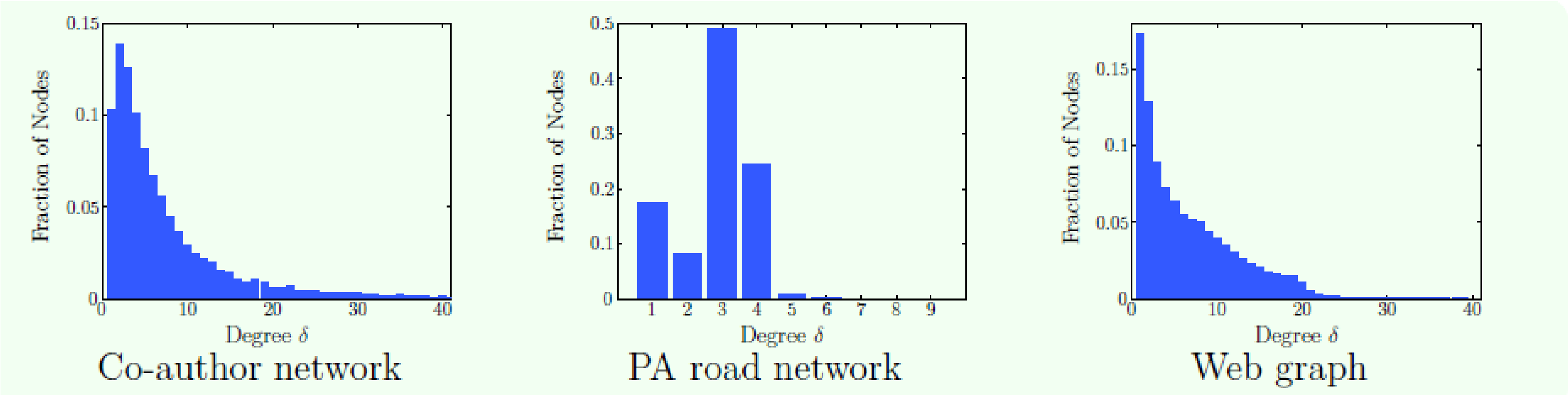
$$= \sum_{j}^{n} A_{ij}$$

$$\boldsymbol{\delta} = [4\ 3\ 3\ 2\ 2\ 1\ 1]$$





- Other examples



arxiv 1993-2003
(23K vertices, 98K edges)

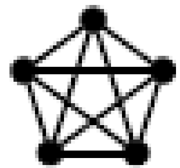
max degree 9!
(1M vertices, 1.5M edges)

Source: Google, 2002
(900K pages, 5M edges)

Graph Types

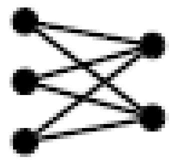
- Complete, K_5

$\boldsymbol{\delta} = [4,4,4,4,4]$



- Bipartite, $K_{3,2}$

$\boldsymbol{\delta} = [3,3,2,2,2]$

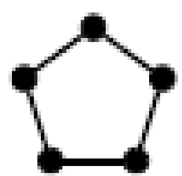


- Line, L_5

$\boldsymbol{\delta} = [2,2,2,1,1]$

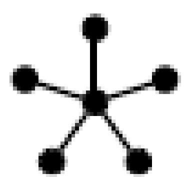
- Cycle, C_5

$\boldsymbol{\delta} = [2,2,2,2,2]$



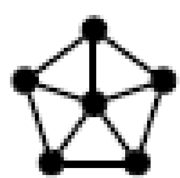
- Star, S_6

$\boldsymbol{\delta} = [5,1,1,1,1,1]$



- Wheel, W_6

$\boldsymbol{\delta} = [5,3,3,3,3,3]$

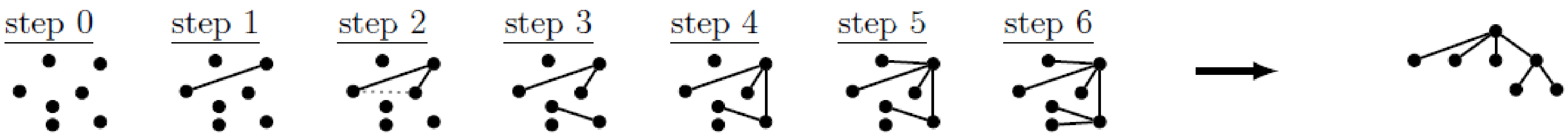


Handshaking Theorem

- **Exercise 1.** Construct a graph with degree sequence $\boldsymbol{\delta}$ = [3, 3, 3, 2, 1, 1].
- *Theorem [Handshaking Theorem]*. For any graph the sum of vertex-degrees equals twice the number of edges, $\sum_{i=1}^{n} \delta_i = 2|E|$.
- *Proof sketch.* Every edge contributes 2 to the sum of degrees. (Why?)
  - Every edge connects two vertices.
  - If there are $|E|$ edges, their contribution to the sum of degrees is $2|E|$.
- **Exercise.** Give a formal proof by induction on the number of edges in the graph.
- **Exercise 1 Answer.** Can't be done. Why?
  - sum of degrees is 3 + 3 + 3 + 2 + 1 + 1 = 13 (odd).
- **Exercise.** At a party a person is odd if they shake hands with an odd number of people.
  - Show that the number of odd people is even.

Trees (More General than RBTs)

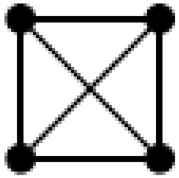
- *Definition [General Tree]*.
  - A tree is a *connected* graph with no cycles.
- Building a tree, one edge at a time.



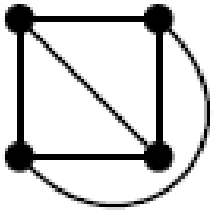
- Why is the dotted line in step 2 not allowed?
- Note that there is no designated root
  - If a root is desired, it is usually determined by the application
  - E.g., the starting state of your program
- **Exercise 11.6.** Every tree with n vertices has $n-1$ edges. (We proved this for RBTs.)

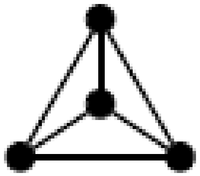
Planar Graphs

- A graph is planar if you can draw it without edge crossings.
- Consider a complete graph K_4



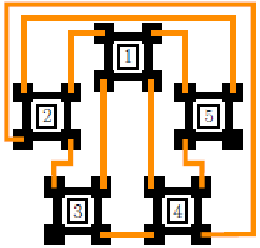
  – Can we draw it without crossings?



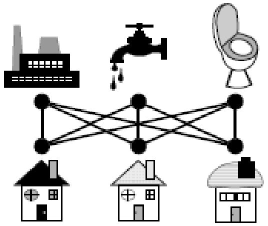


Why do we care about planar graphs?

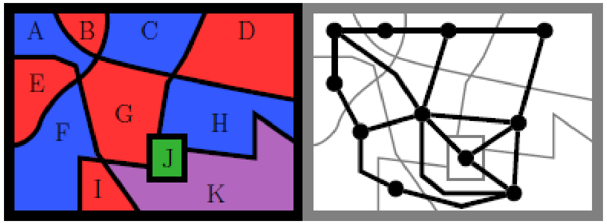
- **Chip design:** CPUs must be connected without wire-crossings (short circuit!)
  - Can you connect CPU 5 and 3?



- **Town planning:** connect utilities to homes without pipe-crossings (water and sewer)
  - Can it be done?



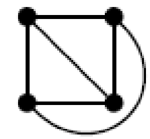
- **Map coloring:** adjacent countries sharing a border must have different colors. The map corresponds to a planar graph.
  - Four-color theorem says any map can be colored with 4 colors (excluding enclaves)

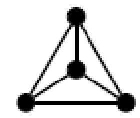


Planar Graphs Exercise

- **Exercise 11.7.** Euler's Invariant Characteristic: $F + V - E = 2$
  - Faces, F: outer region of 3D object
- Planar K_4

$$V = 4, E = 6, F = 4$$
$$4 + 4 - 6 = 2$$



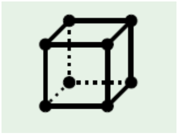


- Planar map

$$V = 11, E = 17, F = 8$$
$$8 + 11 - 17 = 2$$

- Pyramid
  - Same as planar K_4
- Cube (is it planar?)

$$V = 8, E = 12, F = 6$$
$$6 + 8 - 12 = 2$$

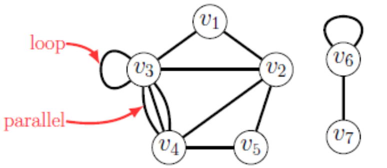


- Octohedron (is it planar?)

$$V = 6, E = 12, F = 8$$
$$8 + 6 - 12 = 2$$

Other Types of Graphs: Multigraph

Rensselaer

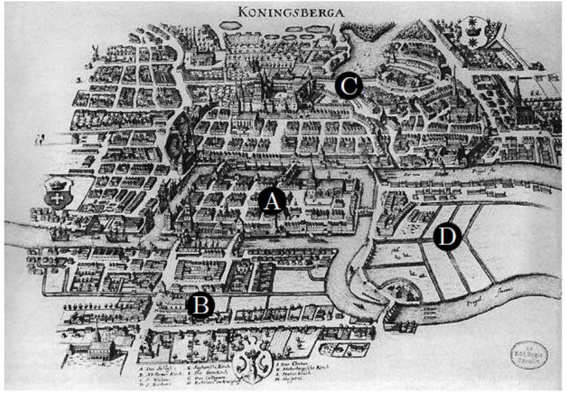


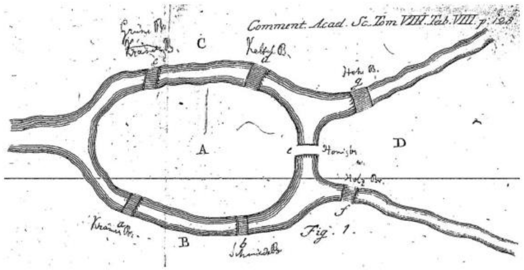
- Multigraphs
  - Allow for multiple edges between the same nodes
  - What are the vertices/edges, in the graph to the right?

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$

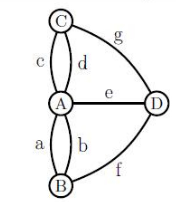
$$E = \{(v_1, v_2), (v_1, v_2), (v_2, v_3), (v_2, v_4), (v_2, v_5),$$
$$(v_3, v_3), (v_3, v_4), (v_3, v_4), (v_3, v_4), (v_4, v_5)\}$$

  - Multigraphs used to model complex systems where standard graph is redundant





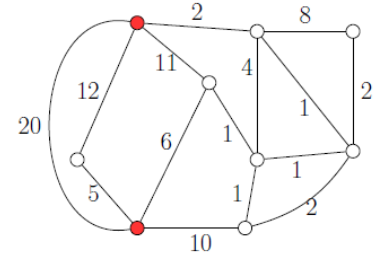
Euler's Multigraph



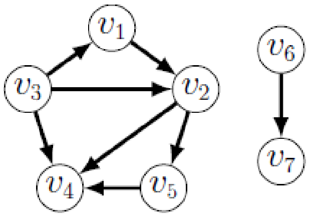
  - Königsberg was an old Prussian city (today's Kaliningrad)
    - Can you cross each bridge exactly once?
    - An Eulerian path is path through the graph that visits each edge only once
    - Can you find such a path in the graph to the right?

Other Types of Graphs: Weighted, Directed

Rensselaer



- Weighted graph
  - Each edge has a weight/cost
- Weighted graphs used to model a variety of scenarios
  - e.g., one internet service provider routing packages through another
    - weights correspond to the cost of sending the package
  - e.g., planning a route for your robot from point A to point B
    - weights model the physical length between nodes
- Directed graph
  - Each edge has a direction



$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$
$$E = \{(v_1, v_2), (v_2, v_4), (v_2, v_5), (v_3, v_1), (v_3, v_2), (v_3, v_4), (v_5, v_4), (v_6, v_7)\}$$

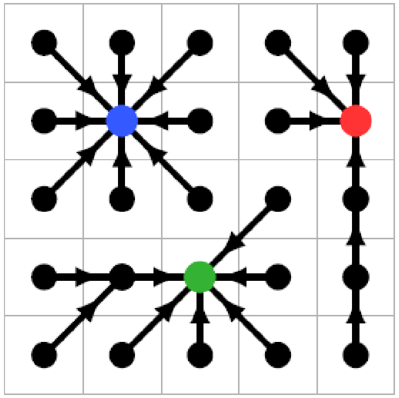
  - Used to model all sorts of asymmetric relationships
  - ancestry graphs, tournaments, one-way streets, partially ordered sets (Example 11.6)

Problem Solving with Graphs

Rensselaer

3	2	17	11	12
4	1	18	10	7
21	22	23	16	8
20	13	5	19	9
25	24	6	14	15

- Graphs are everywhere because relationships are everywhere
- On the right is elevation data in a park
  - One unit of rain falls on each grid-square
  - Water flows to a neighbor of lowest elevation (e.g., 17 → 1)
  - Where should we install drains and what should their capacities be?
- Model the problem as a directed graph.
  - Directed edges indicate how water flows: three disjoint trees.
  - Red, green and blue vertices are "sinks" (no out-going arrow)



- **Place drains at the sinks**
- **Drain capacities: blue=9 units, red=7 units and green=9 units**
- **The solution pops out once we formulate the problem as a graph**