Graph Problems

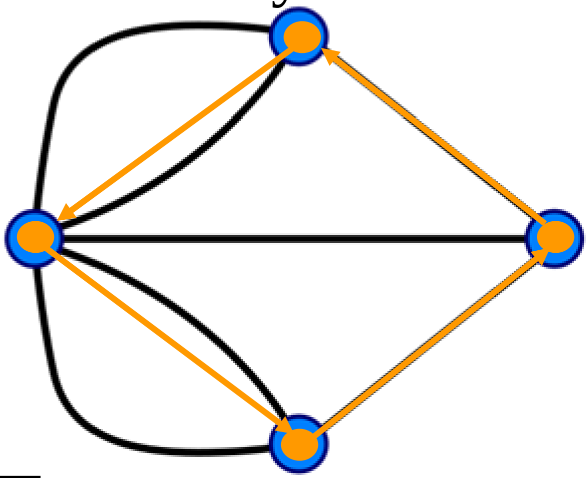
- Critical Path Analysis
- Maximum Flow Problem
- Other Difficult Problems

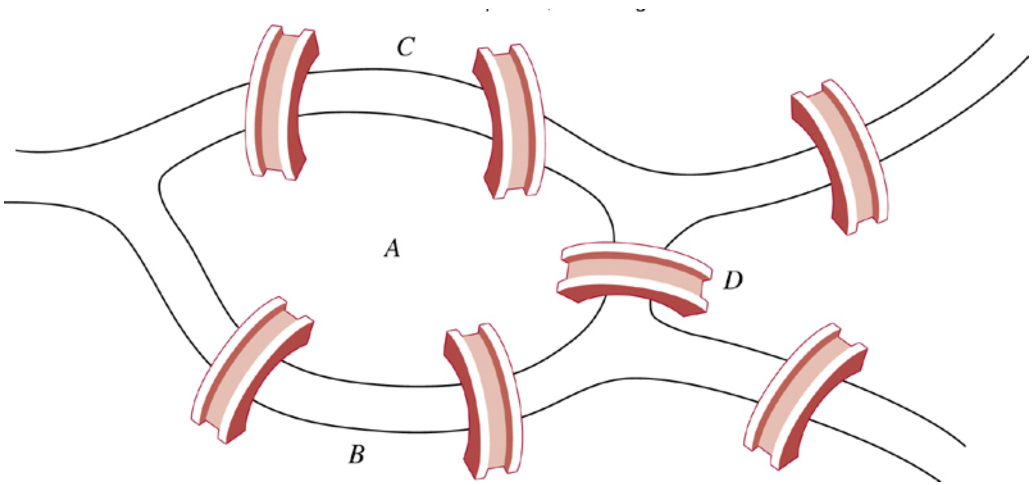
Carrano © Data Abstraction and Problem Solving with C++, 6th ed.

Other Difficult Problems

Eulerian circuit (Euler tour)

– Find a tour that would pass each <u>edge</u> exactly once and finally return to the starting vertex



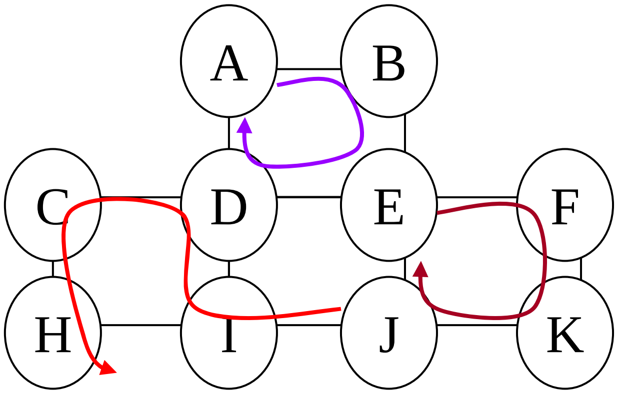


Hamilton circuit

– Find a tour that would visit each <u>vertex</u> exactly once and finally return to the starting vertex

Other Difficult Problems

- **Eulerian circuit (Euler tour)**
  - Find a tour that would pass each edge exactly once and finally return to the starting vertex
- **DFS-based Algorithm** **[Carl Hierholzer, 1873]**

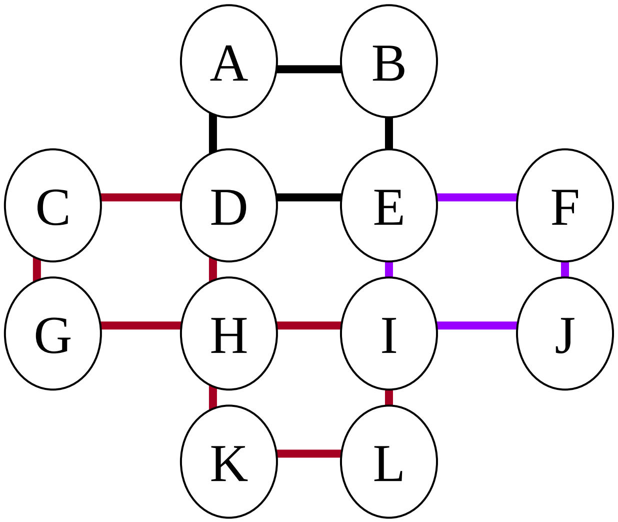


Not exist!

Path: A
B
E
F
K
J {I}
E
D {C, I}
A

Eulerian Circuit (Euler Tour)

DFS-based Algorithm



A B E D

A B E F J I E D

A B E F J I H D C G H K L I E D

Traveling Salesman Problem (TSP)

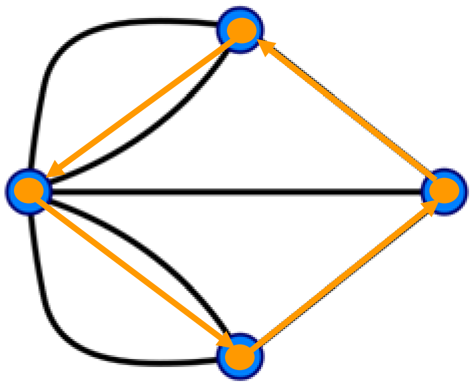
Data Structures

Hamilton circuit

- Find a tour that would visit each vertex exactly once and finally return to the starting vertex
- Decision problem (NP-complete)

Traveling Salesman Problem

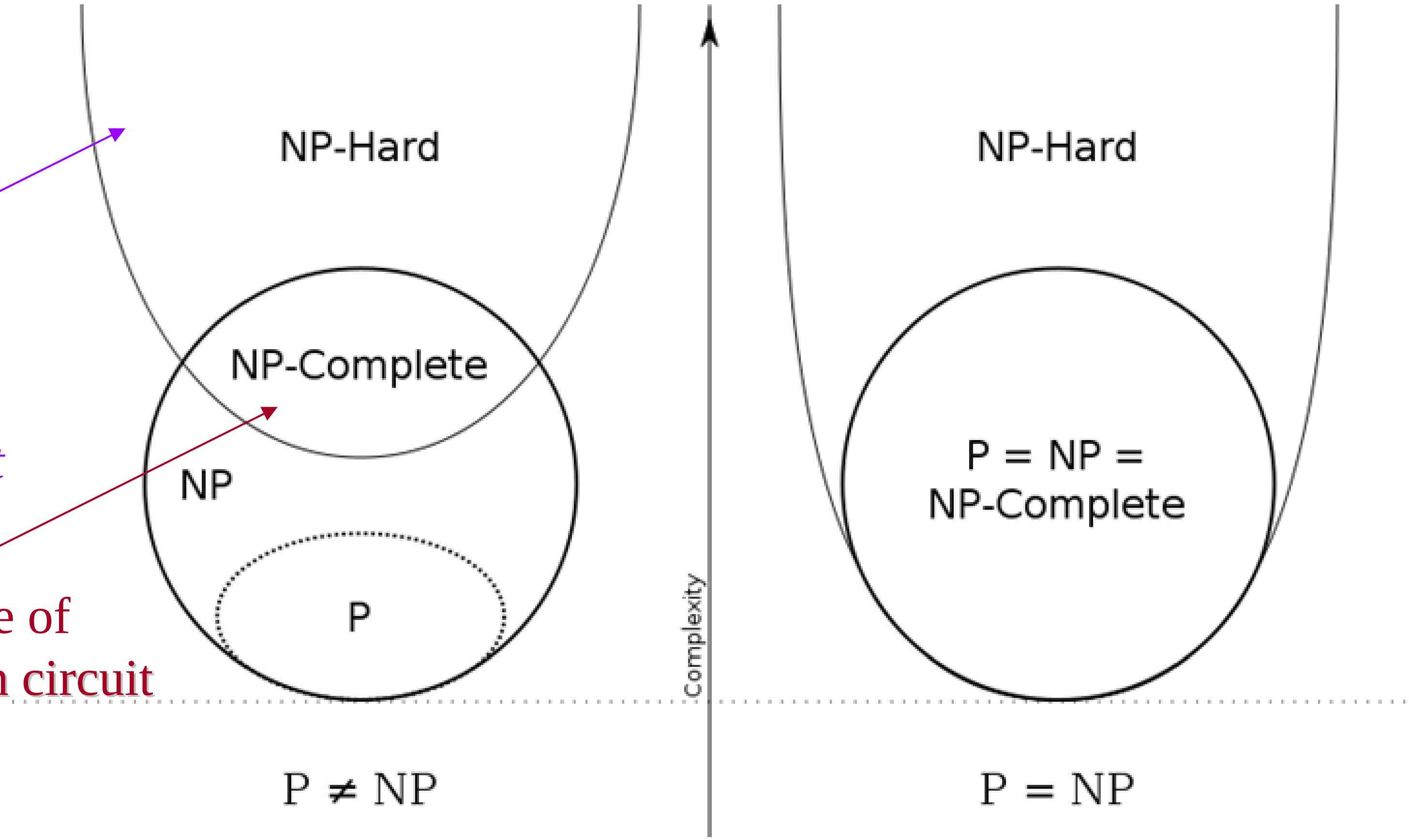
- Find the *shortest path* (Hamilton path) that would visit each vertex exactly once and finally return to the starting vertex
- Optimization problem (NP-hard)



Traveling Salesman Problem (TSP)

TSP: search of least expensive *Hamilton circuit*

Existence of Hamilton circuit



NP is the set of all *decision problems* for which the instances where the answer is "yes" have *efficiently verifiable proofs* of the fact that the answer is indeed "yes".

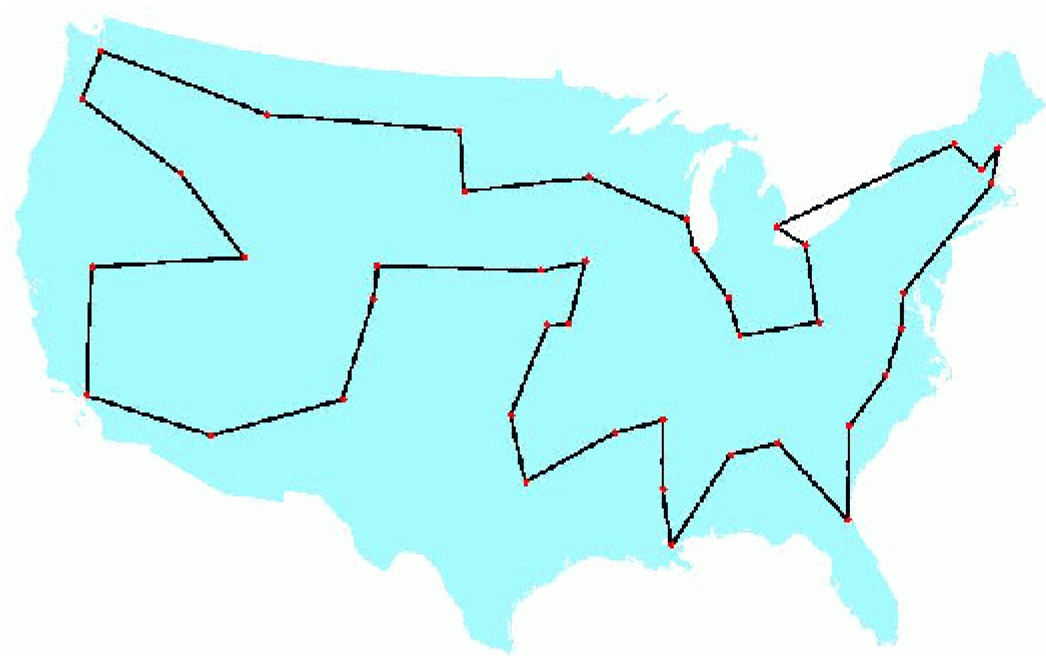
wikipedia ©

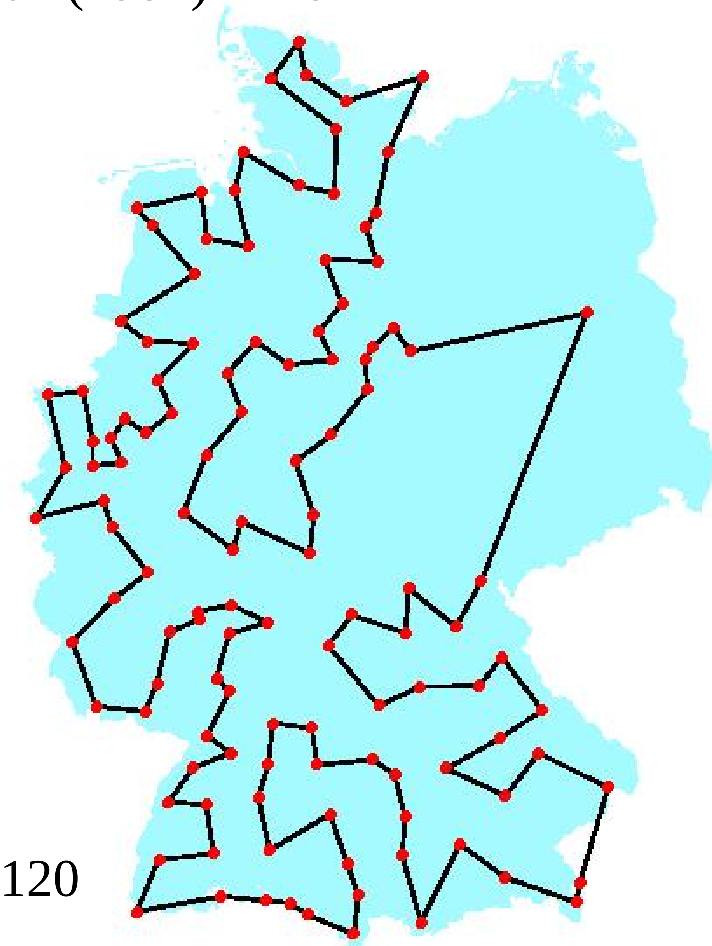
TSP on the Web

- **Georgia Tech.**
  - http://www.tsp.gatech.edu/index.html
- **Progress**
  - TSP is one of the most intensely studied problems in computational mathematics and yet no effective solution method is known for the *general* case.
  - A breakthrough came when George Dantzig, Ray Fulkerson, and Selmer Johnson (1954) published a description of a method for solving the TSP and illustrated the power of this method by solving an instance with 49 cities, an impressive size at that time.

TSP on the Web: *Milestones*

George Dantzig, Ray Fulkerson, and Selmer Johnson (1954) n=49

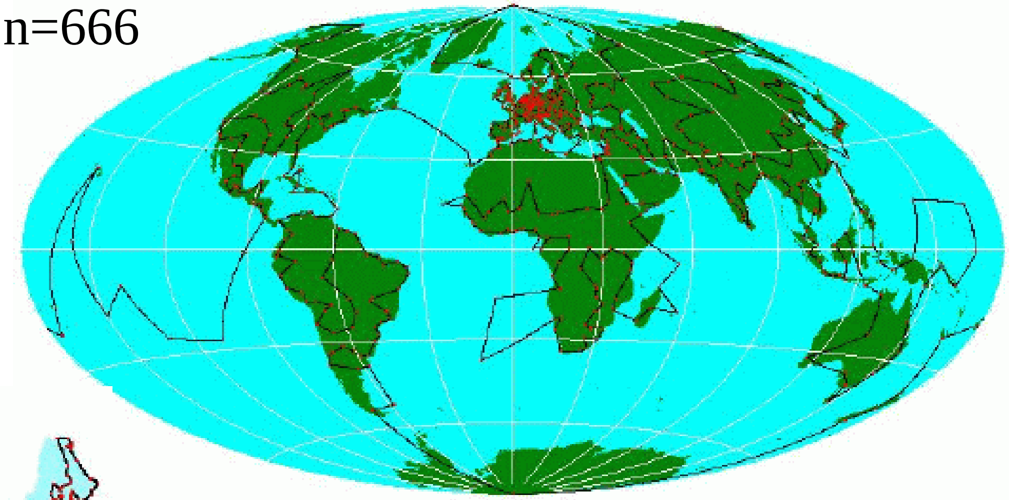


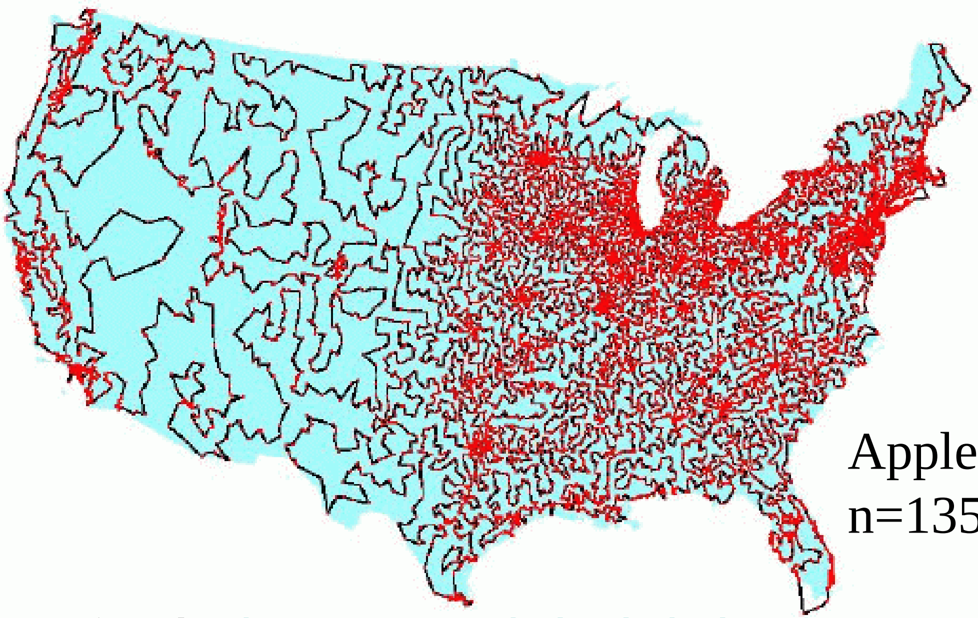


Groetschel (1977) n=120

TSP on the Web: *Milestones*

Groetschel and Holland (1987) n=666

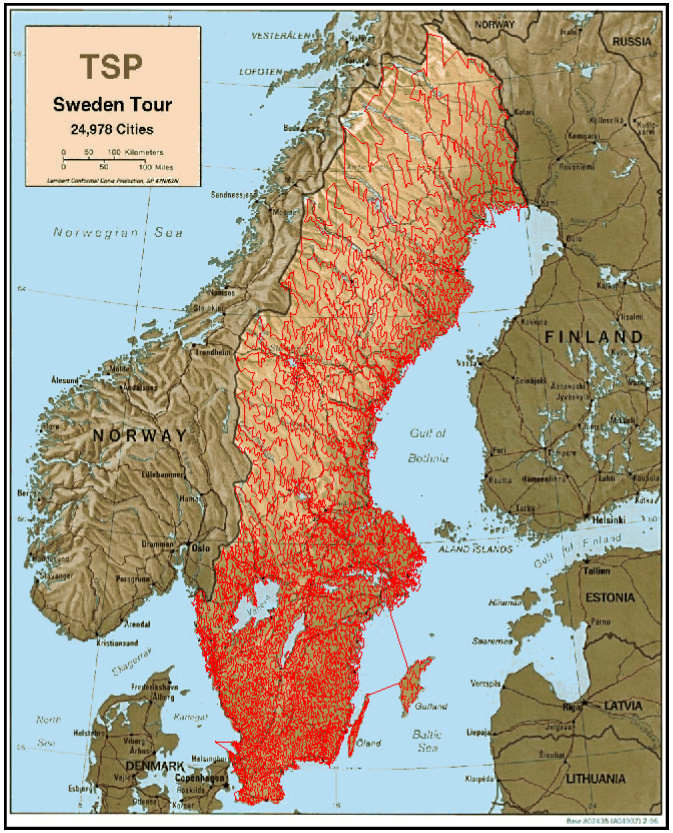


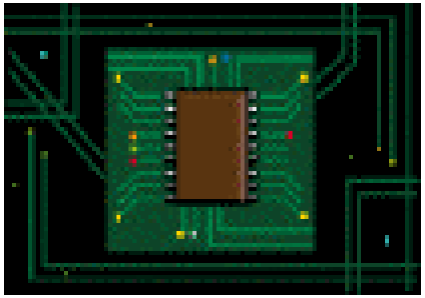


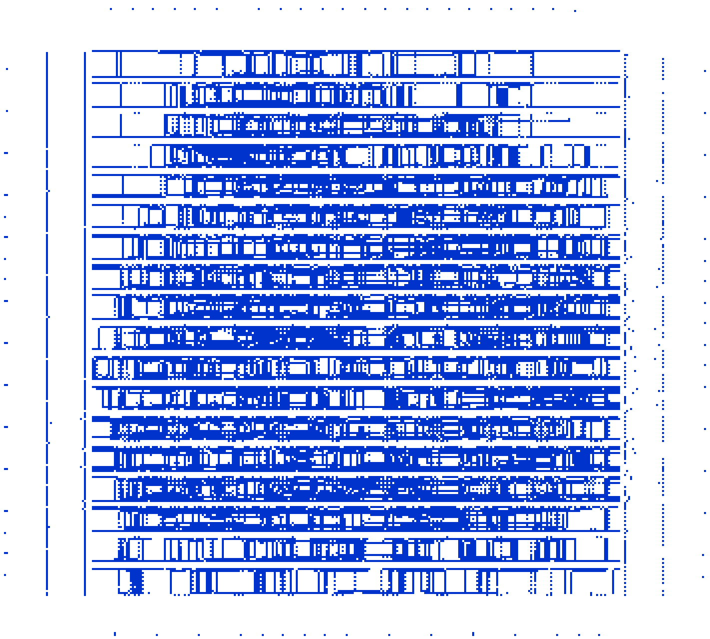
Applegate, Bixby, Chvátal, and Cook (1998) n=13509

TSP on the Web: *Milestones*

Applegate, Bixby, Chvátal, Cook, and Helsgaun
(2004) n=24978



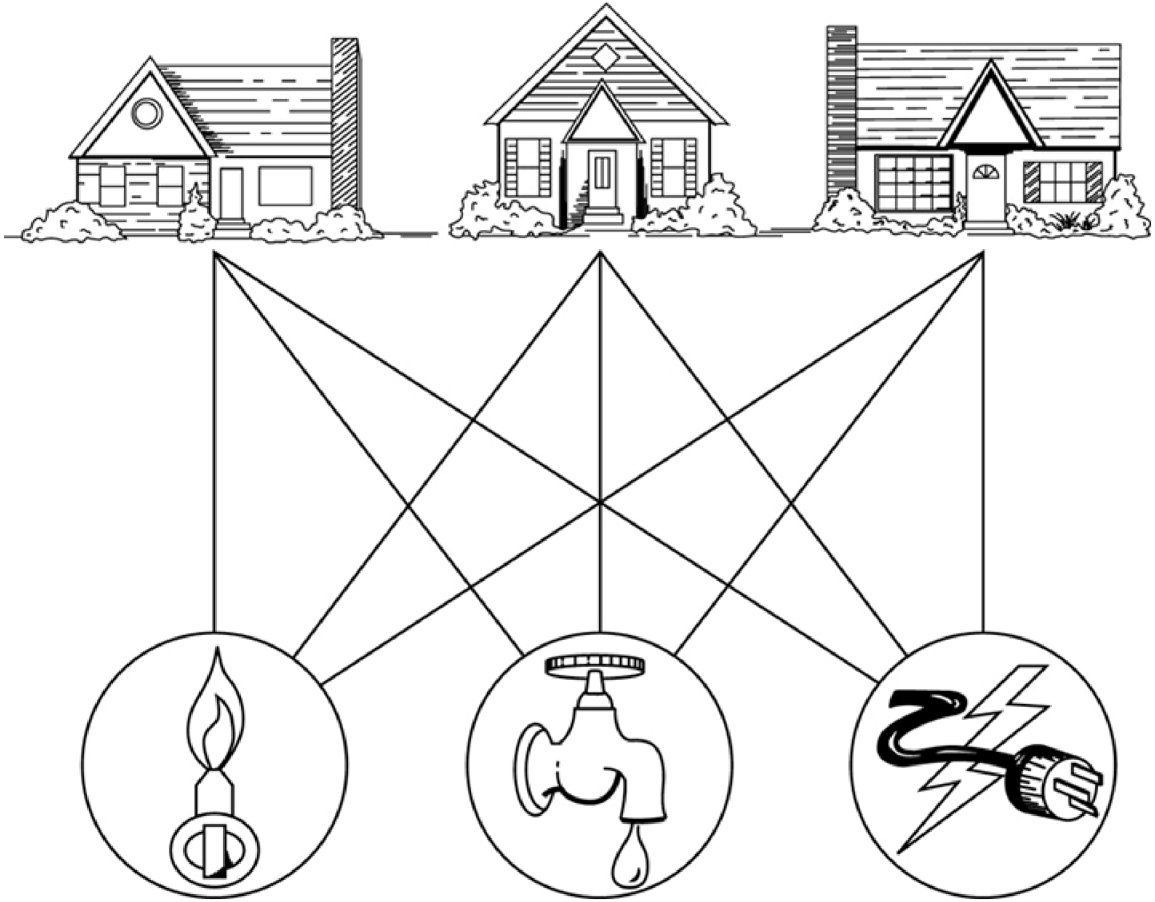




[The Traveling Salesman Problem: A Computational Study]. (2006) n=85900

3-utility problem: *Planar Graph*

© The McGraw-Hill Companies, Inc. all rights reserved.



Planar Graph: *Definitions*

Definition:

A graph that can be drawn in the plane without any of its edges intersecting is called a planar graph.

Definition:

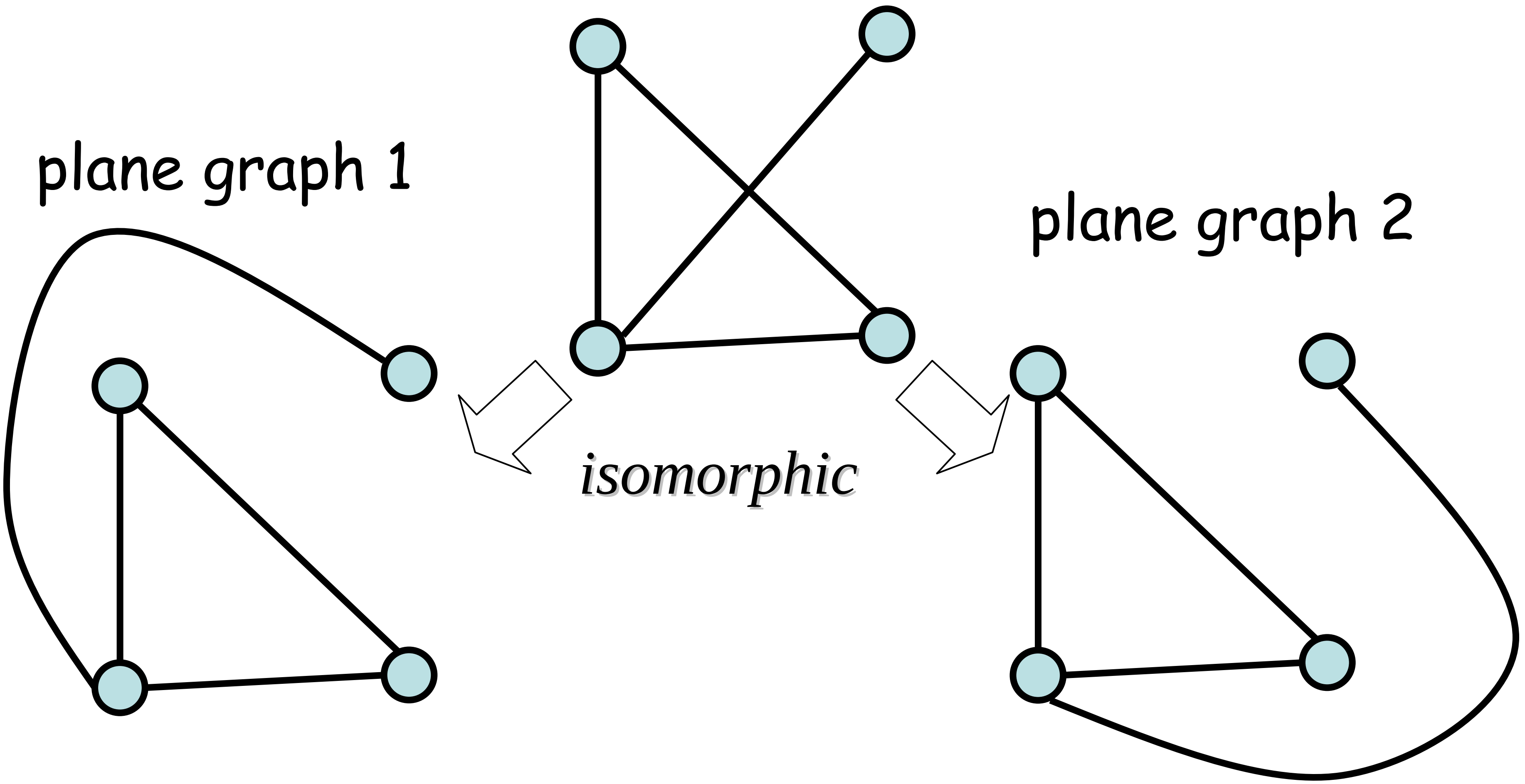
A planar graph G that is drawn in the plane so that no two edges intersect (that is, G is embedded in the plane) is called a plane graph.

(Euler's Formula)

If G is a connected plane graph with p vertices, q edges, and r regions, then

$$p - q + r = 2.$$

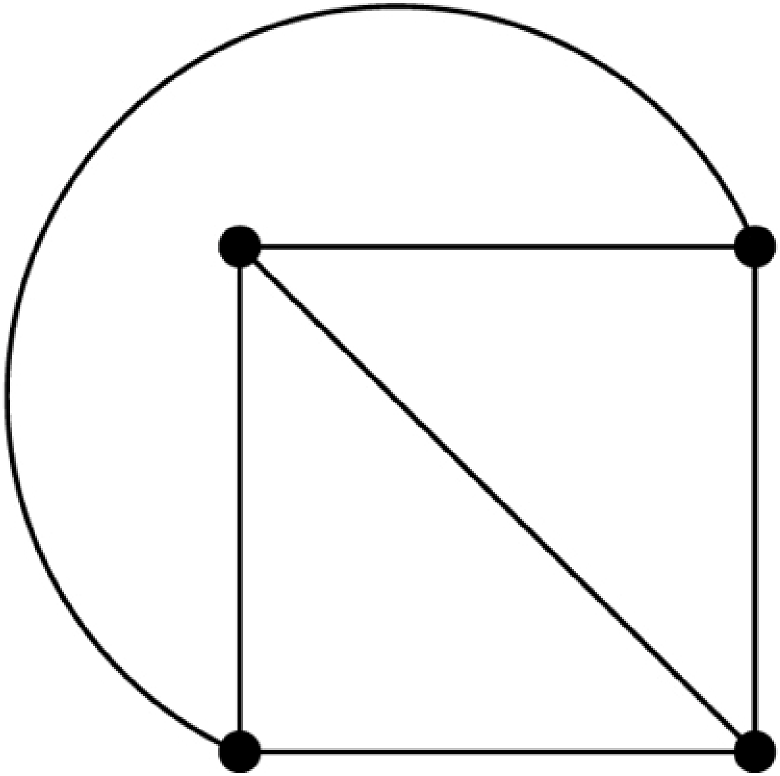
Plane Graph: *Examples*



$p-q+r=4-4+2=2$

Planar Graph: *Examples*

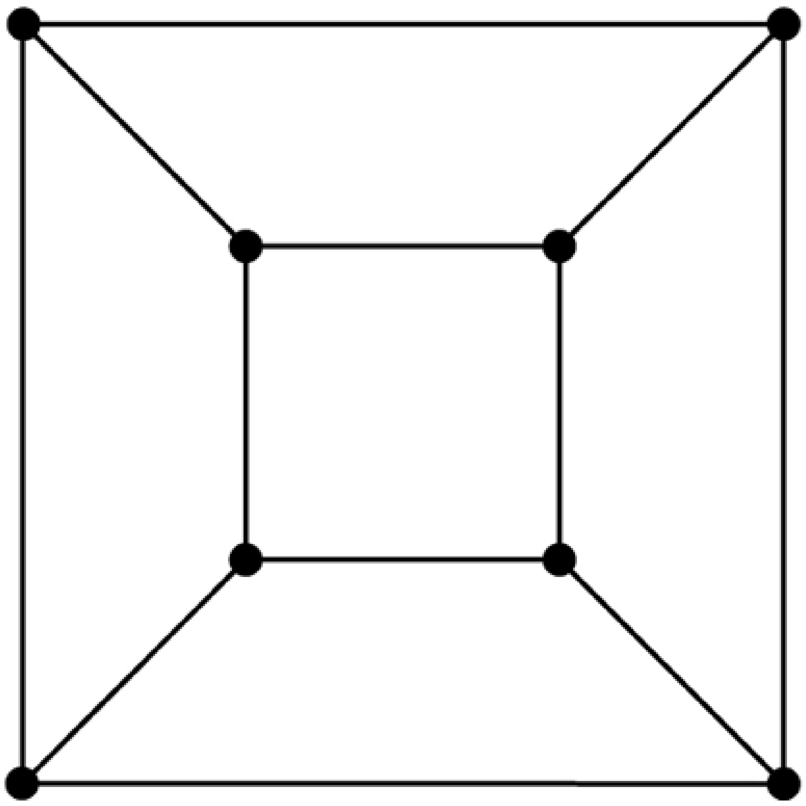
© The McGraw-Hill Companies, Inc. all rights reserved.



$p-q+r=4-6+4=2$

Planar Graph: *Examples*

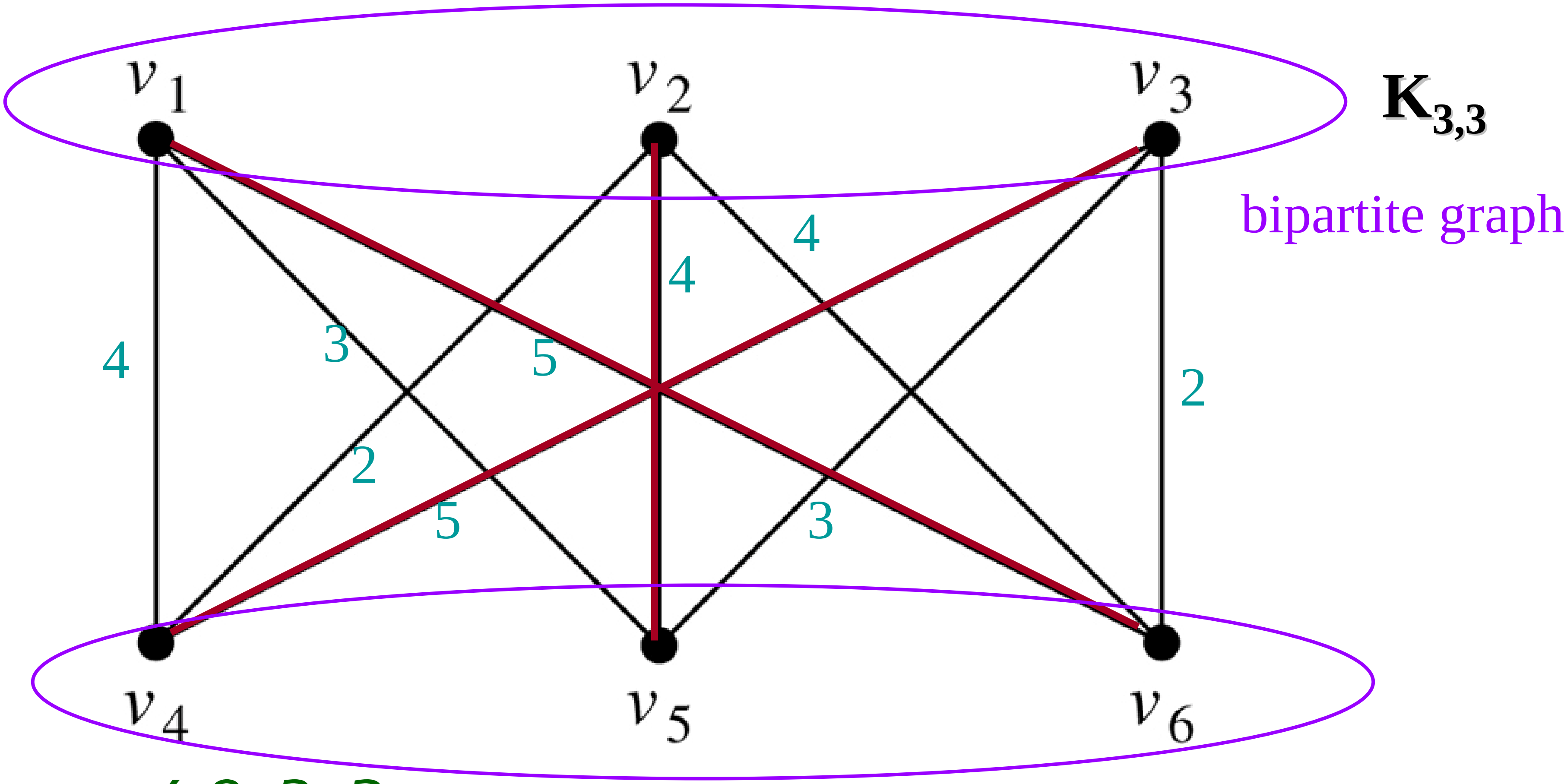
© The McGraw-Hill Companies, Inc. all rights reserved.



p-q+r=8-12+6=2

Planar Graph: *Examples*

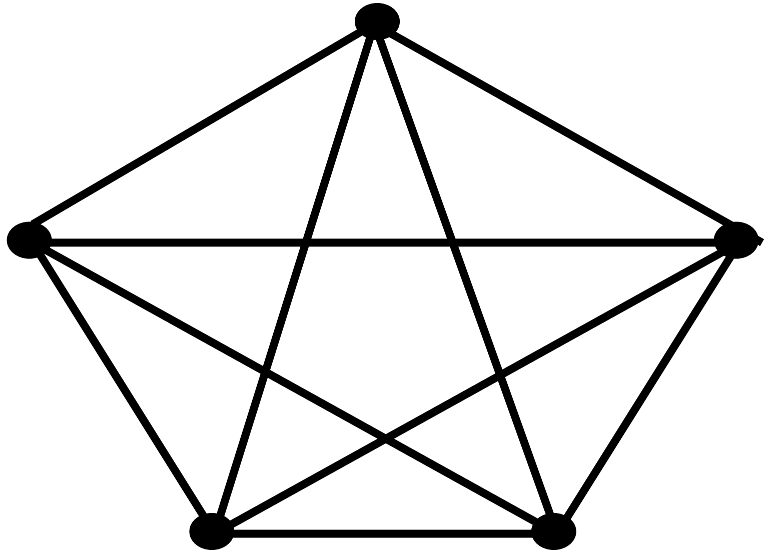
© The McGraw-Hill Companies, Inc. all rights reserved.



p-q+r=6-9+?=2

maximum match (14)

Planar Graph: *Examples*

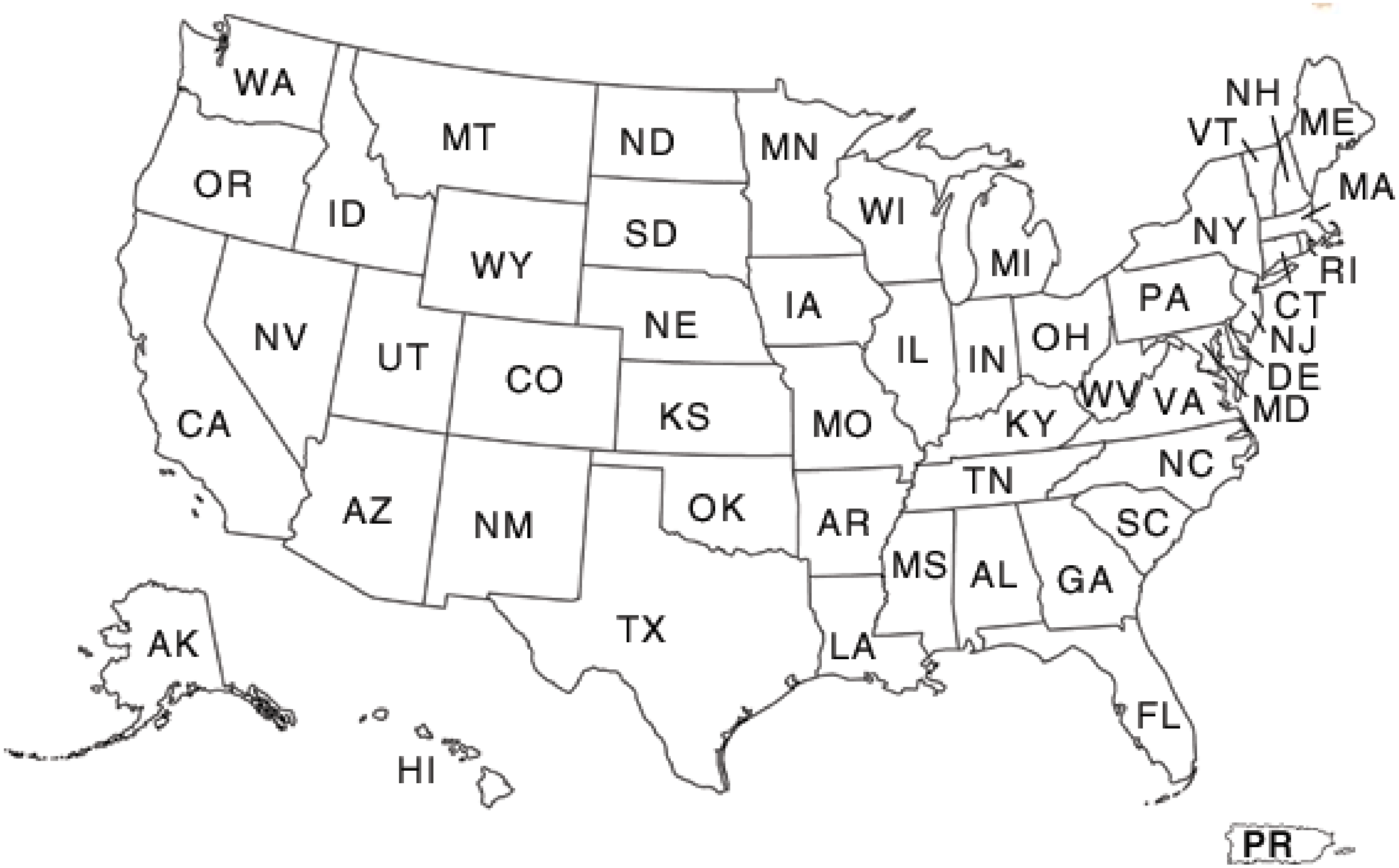


(Kuratowski's Theorem)
A graph is planar if and only if it contains no subgraph that is isomorphic to or is a subdivision of K_5 or $K_{3,3}$.

$3p-6=9<q=10$

4-color Problem: *Basics*

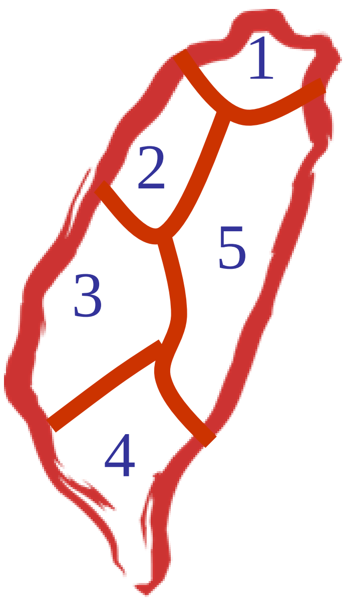
- **Story…**

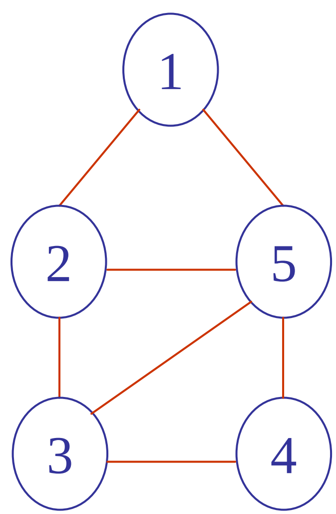


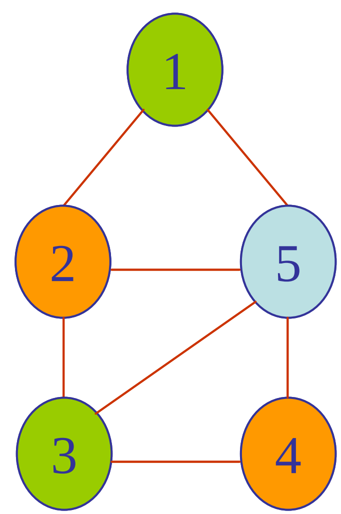
4-color Problem: *Basics*

- **At most *four* colors are required to color the vertices of a *planar graph* so that *no adjacent vertices have the same color***
  - Every planar graph is *four-colorable*

chromatic number = 3





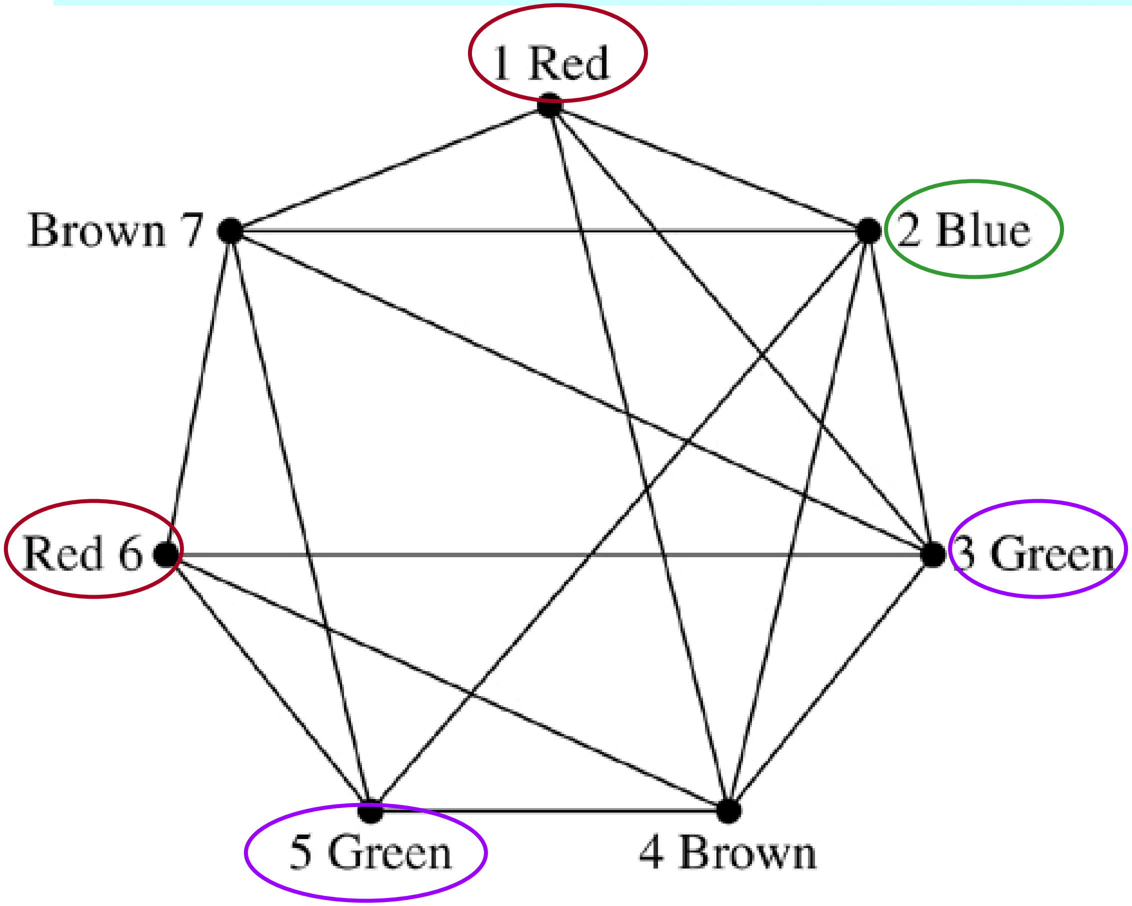


4-color Problem: *Basics*

- **After more than 120 years…**
  - Proven in **1976** by Kenneth Appel and Wolfgang Haken. It was the first major theorem to be proved using a computer.
    1. Showed by computer programs in UIUC that there is a particular set of 1,936 maps, each of which cannot be part of a smallest-sized *counterexample* to the four color theorem.
    2. Any map must have a portion that looks like one of these 1,936 maps.
    3. Concluded that no smallest *counterexamples* existed because any must contain, yet not contain, one of these 1,936 maps.

Graph Coloring Problem: *Application*

- **Scheduling of final exams for 7 courses**

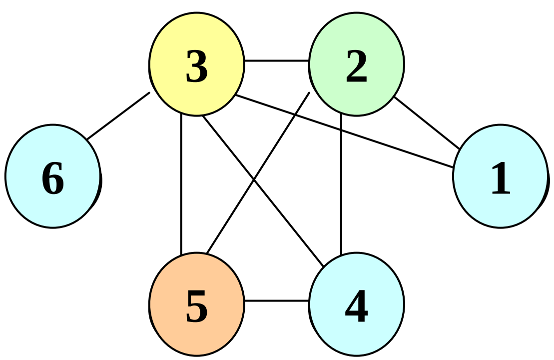


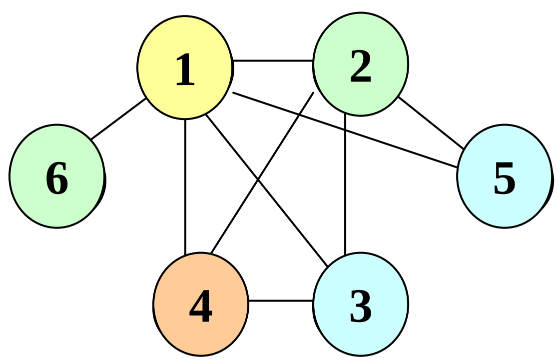
Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7

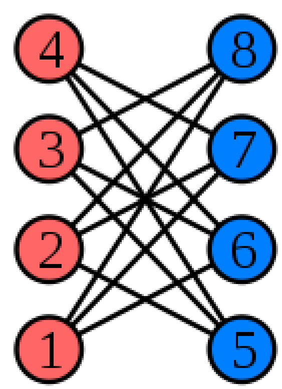
chromatic number = 4

Graph Coloring Problem: *Algorithm*

- **Vertex Coloring (Edge Coloring)**
- **Sequential ordering algorithms**
  - Heuristics for a specific ordering of vertices
    - **No guarantee on using the *least* number of colors**
  - *Welsh-Powell* algorithm (**greedy coloring**)
    - **max-degree first**

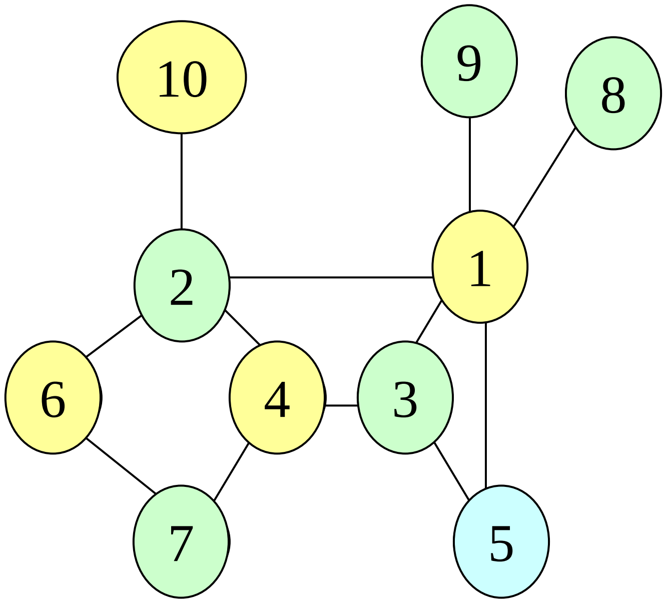






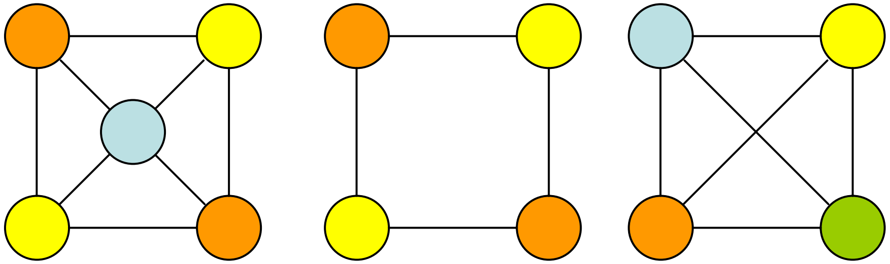
Greedy Coloring: *Example*

- **Color the graph by using as *less* colors as possible.**



chromatic number = 3

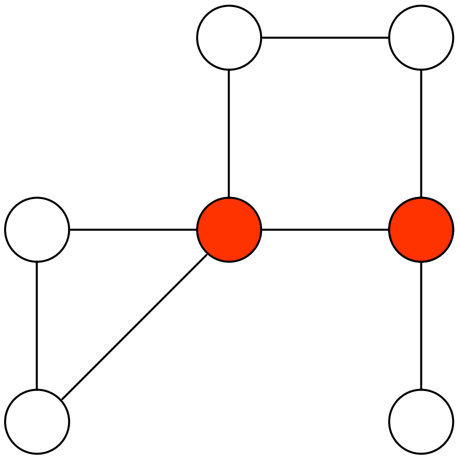
- **Wheel graph G_n = 3 or 4**



- **4-color game!**
  - http://www.gamedesign.jp/flash/fourcolor/fourcolor.html

Bi-connected Graph: *Algorithm*

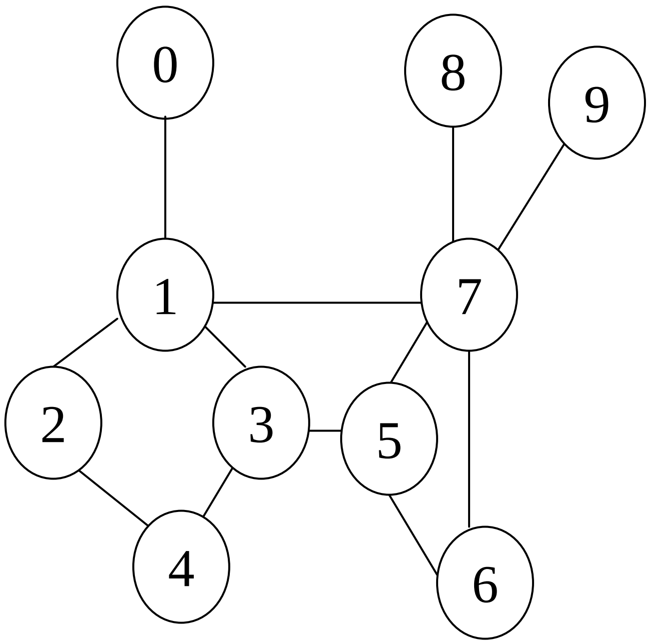
- **Articulation point**
  - A vertex *v* in G is an *articulation point* iff the deletion of *v*, together with the deletion of all edges incident to *v*, leaves behind a graph that has at least two connected components (*disconnected*)
- **Bi-connected graph**
  - A connected graph that has no *articulation point*

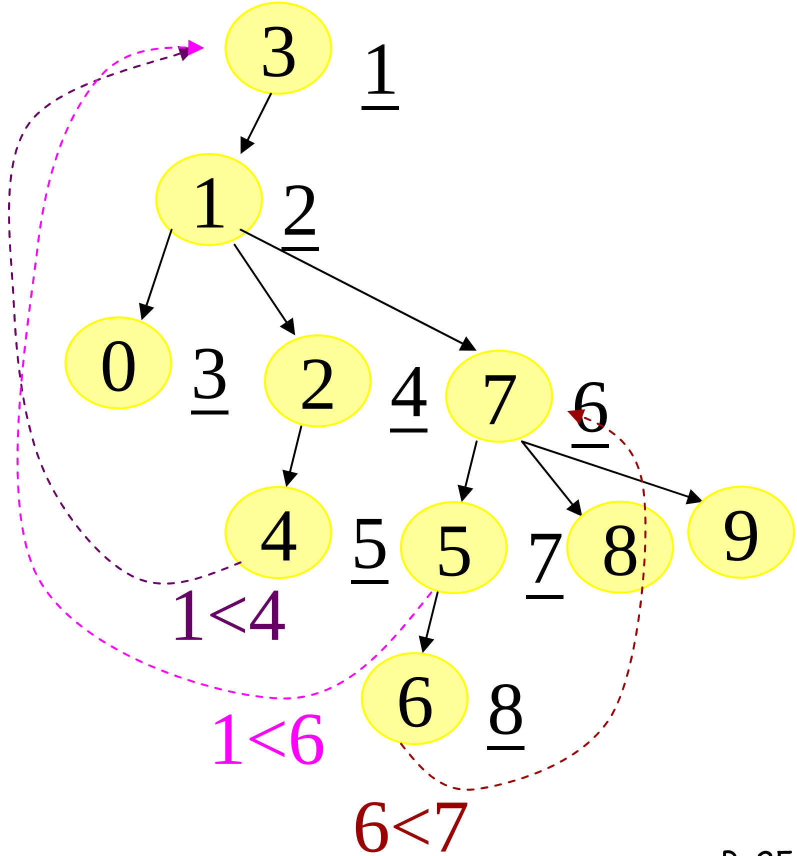


Bi-connected Graph: *Definitions*

Data Structures

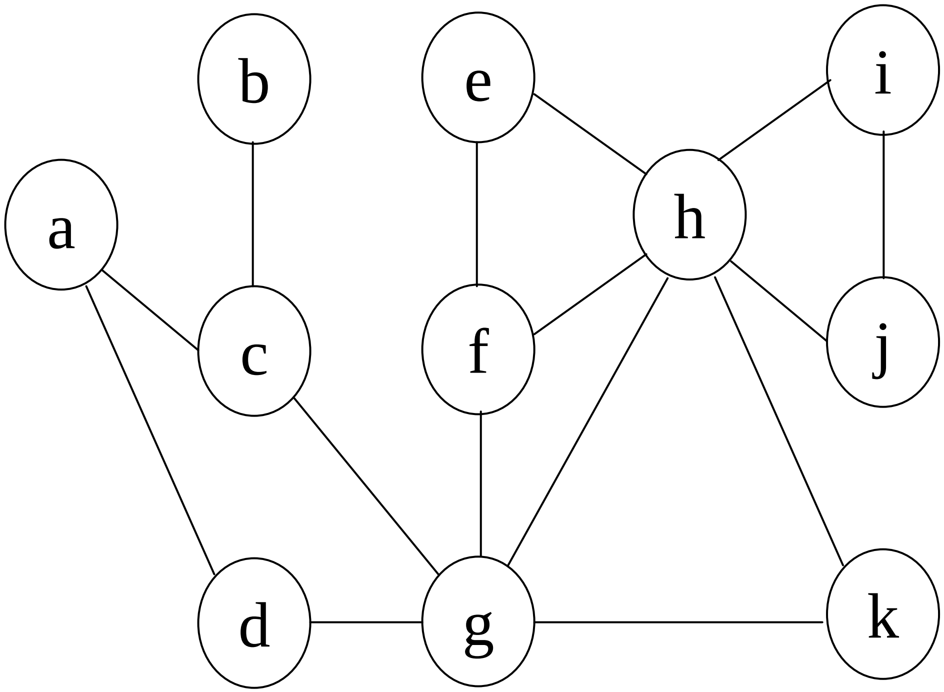
- **Finding the articulation points**
  - Graph traversal algorithm
  - DFS-tree based algorithm





Practice 13: *Coloring & Bi-connected*

1. **Color the graph by *max-degree first* algorithm**
2. **Find the *articulation points* by *DFS-tree based* algorithm**



Some *NP-Complete* Problems

- Boolean satisfiability problem
- Knapsack problem
- Hamiltonian path problem
- Travelling salesman problem
- Subgraph isomorphism problem
- Subset sum problem
- Clique problem
- Vertex cover problem
- Independent set problem
- Dominating set problem
- Graph coloring problem

wikipedia ©

Summary

Data Structures

- **An Euler circuit in an undirected graph is**
  - A cycle that begins at vertex *v*, passes through every edge in the graph exactly once, and terminates at *v*
- **A Hamilton circuit in an undirected graph is**
  - A cycle that begins at vertex *v*, passes through every vertex in the graph exactly once, and terminates at *v*