Discrete Structures for Computer Science

PROF. NAVNEET GOYAL
COMPUTER SCIENCE DEPARTMENT
BITS-PILANI, PILANI CAMPUS

Broad Topics

- Set Theory & Relations
- Proving Techniques – Strong Mathematical Induction
- Recursion
- Data Structures – Graphs & Trees
- Algebraic Structures - Groups, Rings, Fields, & Vector Spaces

Why Study Discrete Structures?

Set Theory

- Complete Relational Database theory is built around Set Theory!
- Relational Database Theory uses the following characteristics of set theory:
  - Duplicate elements in a set are not useful
  - Ordering of elements is not important
  - Mathematical relations
  - Domains

Why Study Discrete Structures?

Recursion

- No. of valid expressions using 10 digits 0-9, and 4 arithmetic operators +, -, /, *.
- Syntax of programming language requires that the expression ends in a digit
- 2 valid expression can be combined using any of the 4 operators
- How many valid expression are there in this programming language?
- Pascal's Identity is another example of a recurrence relation
- Fibonacci Sequence
- Towers of Hanoi
- Sorting Algorithms
  - Merge-sort algorithm
  - Bubble-sort algorithm

Why Study Discrete Structures?

Relations & Digraphs

- Relations between input & output of a computer program
- Relations between data attributes in databases

Why Study Discrete Structures?

Graphs

G=(V,E), where E is a set of edges (ordered or unordered)

- Mainly used for modeling
  - Website
  - File system

Trees

- Simplified graphs
- Mainly used for modeling
  - Website
  - File system

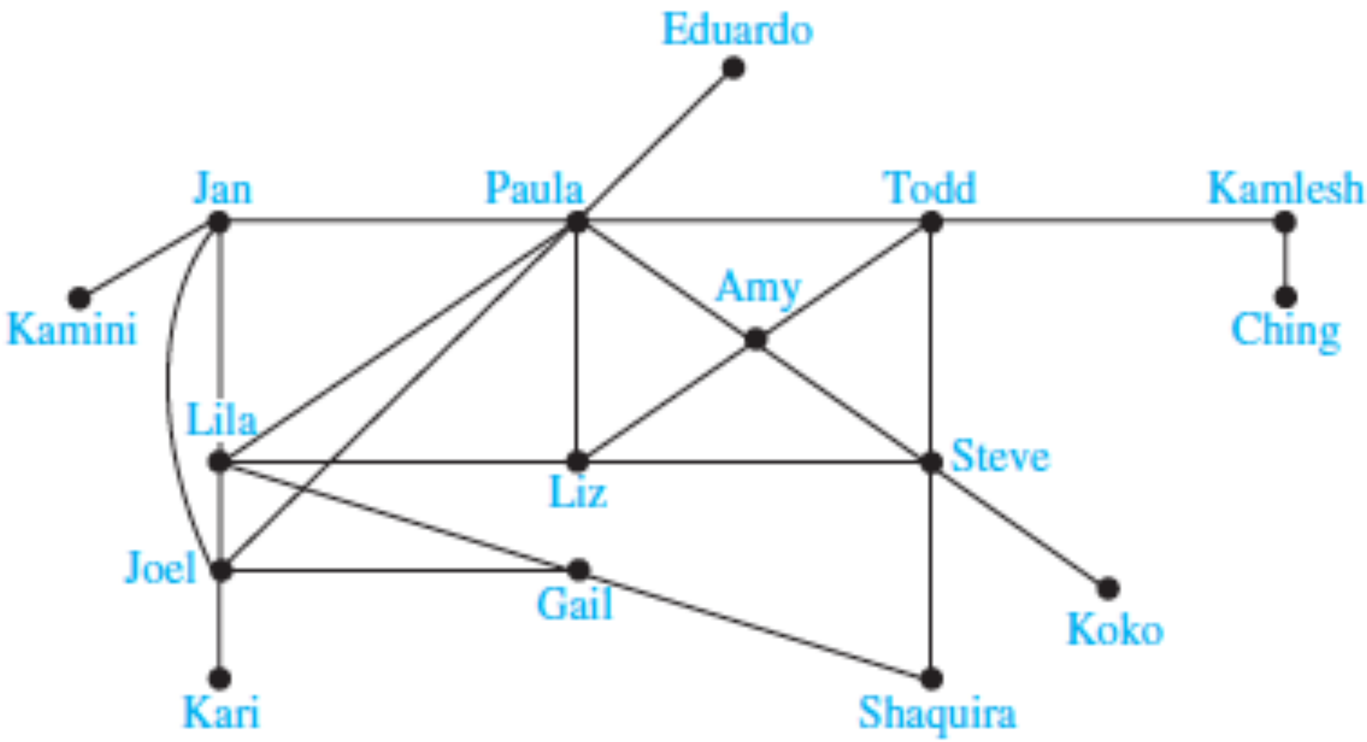
  Tree-based Indexing structures are very popular in RDBMSs

Types of Graphs

TABLE 1 Graph Terminology.

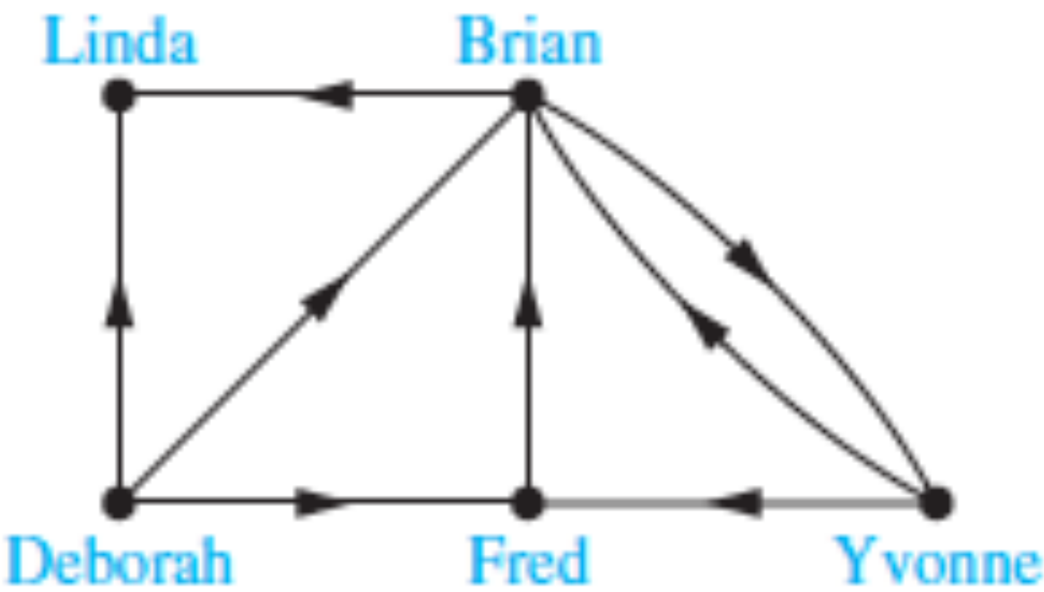
Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Graphs and Social Networks



Acquaintanceship Graph

Graphs and Social Networks



Influence Graph

Some Interesting Graphs

Collaboration Graphs

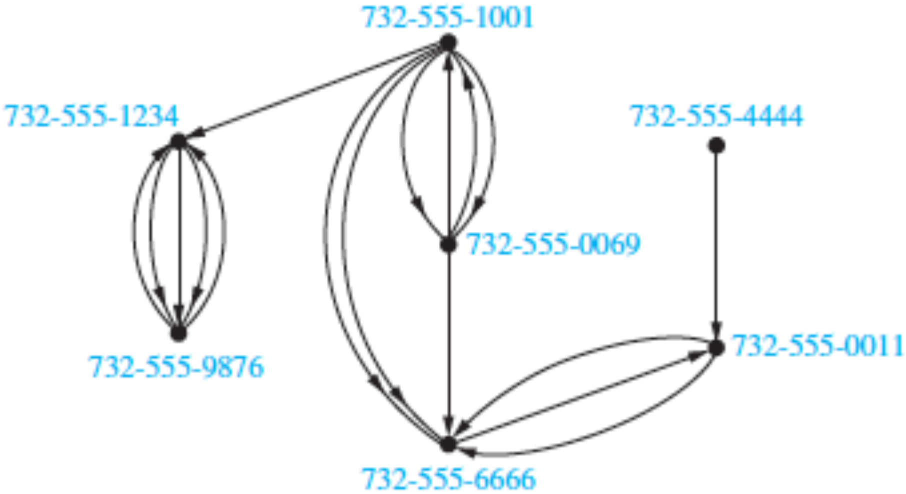
- Hollywood Collaboration Graph
- Academic collaboration graph

Call Graphs

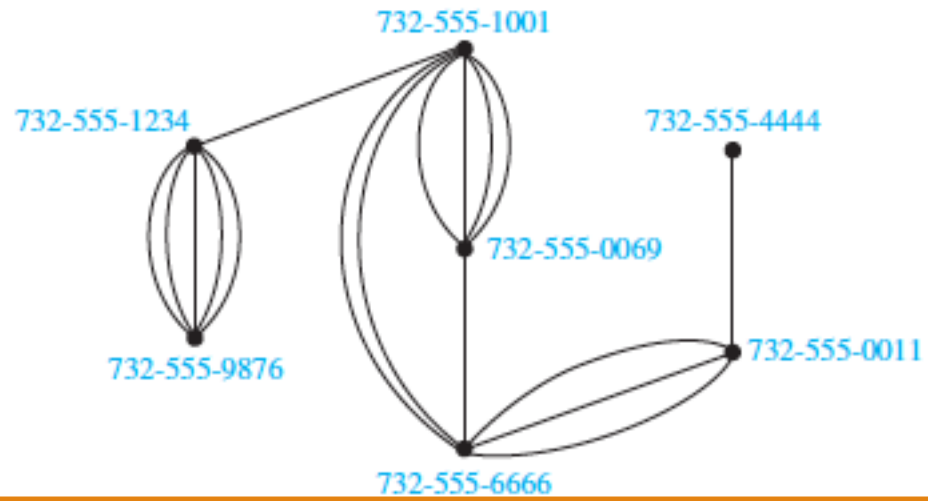
Citation Graphs

??

Call Graphs

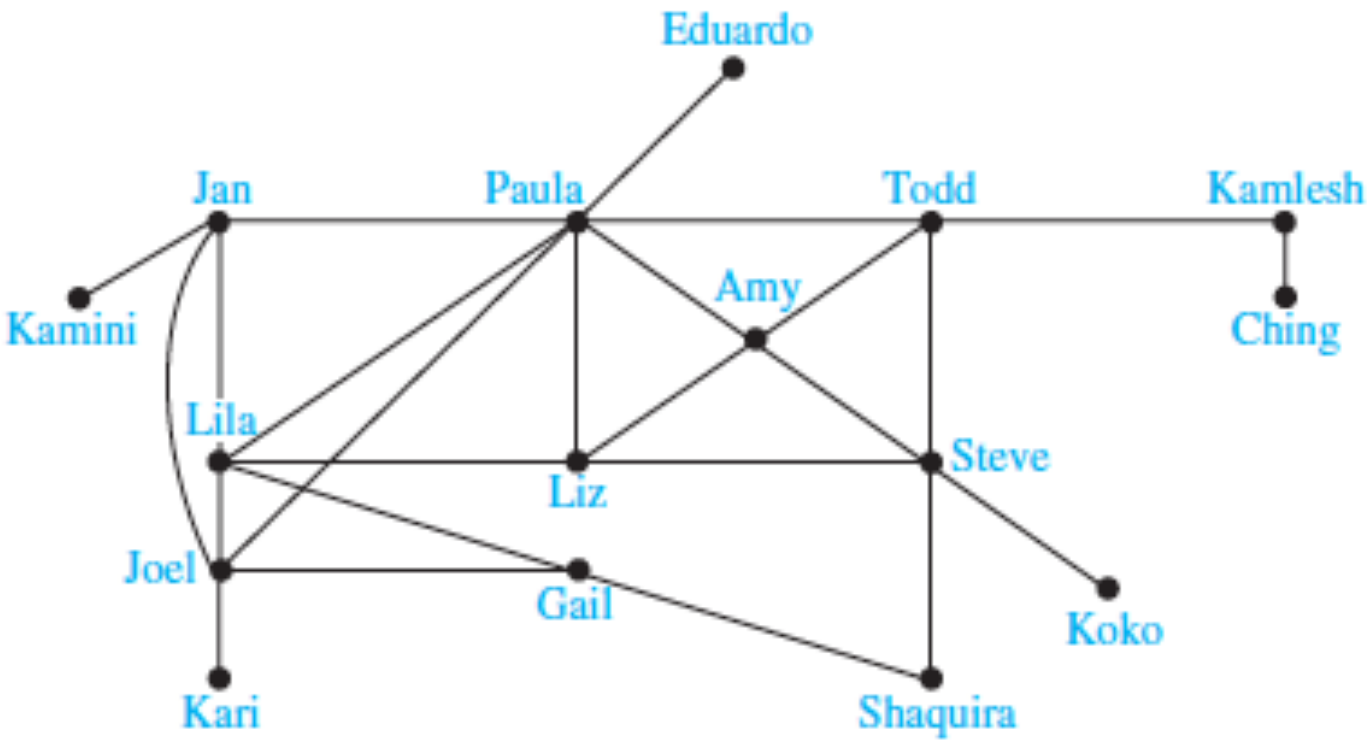


732-555-1001
732-555-1234
732-555-4444
732-555-0069
732-555-9876
732-555-0011
732-555-6666



732-555-1001
732-555-1234
732-555-4444
732-555-0069
732-555-9876
732-555-0011
732-555-6666

Paths in Social Networks



Acquaintanceship Graph

Paths in Collaboration Graphs

TABLE 1 The Number of Mathematicians with a Given Erdős Number (as of early 2006).

Erdős Number	*Number of People*
0	1
1	504
2	6,593
3	33,605
4	83,642
5	87,760
6	40,014
7	11,591
8	3,146
9	819
10	244
11	68
12	23
13	5

Paths in Hollywood Graph

TABLE 2 **The Number of Actors with a Given Bacon Number (as of early 2011).**

Bacon Number	*Number of People*
0	1
1	2,367
2	242,407
3	785,389
4	200,602
5	14,048
6	1,277
7	114
8	16

Why Study Discrete Structures?

Graph Isomorphism

f(a)=1

f(b)=6

f(*c*) = 8

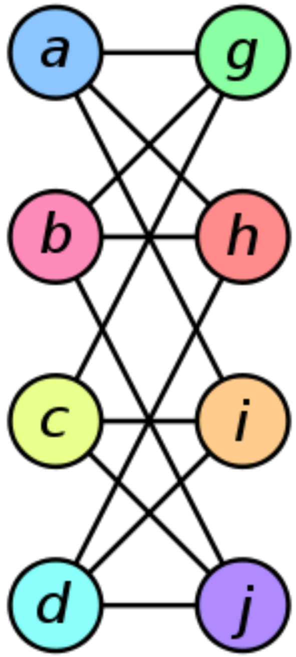
f(*d*) = 3

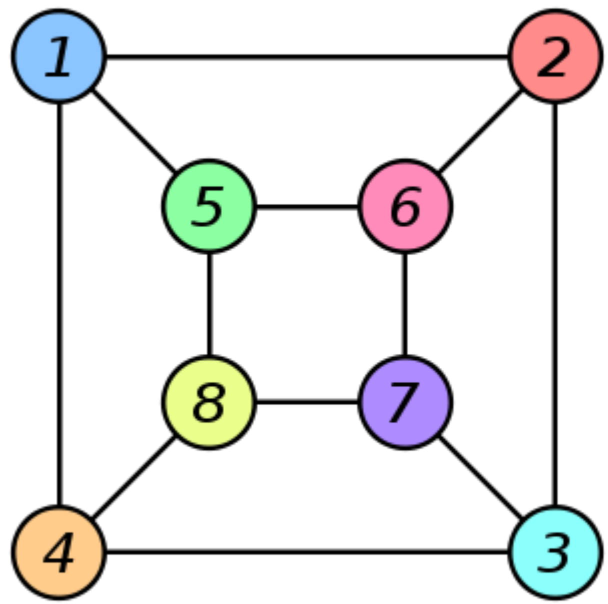
f(*g*) = 5

f(*h*) = 2

f(*i*) = 4

f(*j*) = 7





G & G' – isomorphic if there exists a fn f: V(G) ➔ V(G') if f is 1-1 onto and for each pair of vertices u & v of G belonging to E(G) iff f(u), f(v) belong to E(G')

Graph Coloring

Assignment of "*colors*" to certain objects in a graph subject to certain constraints

- Vertex coloring (the default)
- Edge coloring
- Face coloring (planar)

Vertex coloring

In its simplest form, it is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color

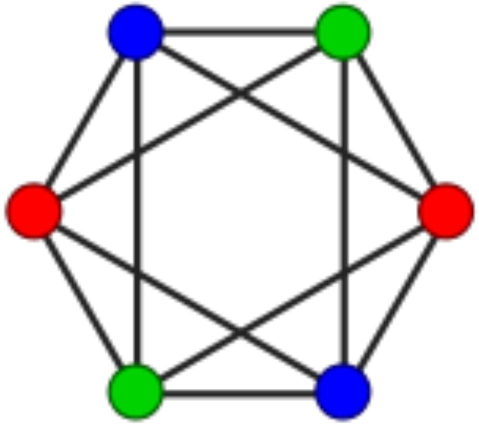
Edge and Face coloring can be transformed into Vertex version

Vertex Color example

Anything less results in adjacent vertices with the same color

- Known as “proper”

3-color example



Chromatic Number

χ - least number of colors needed to color a graph

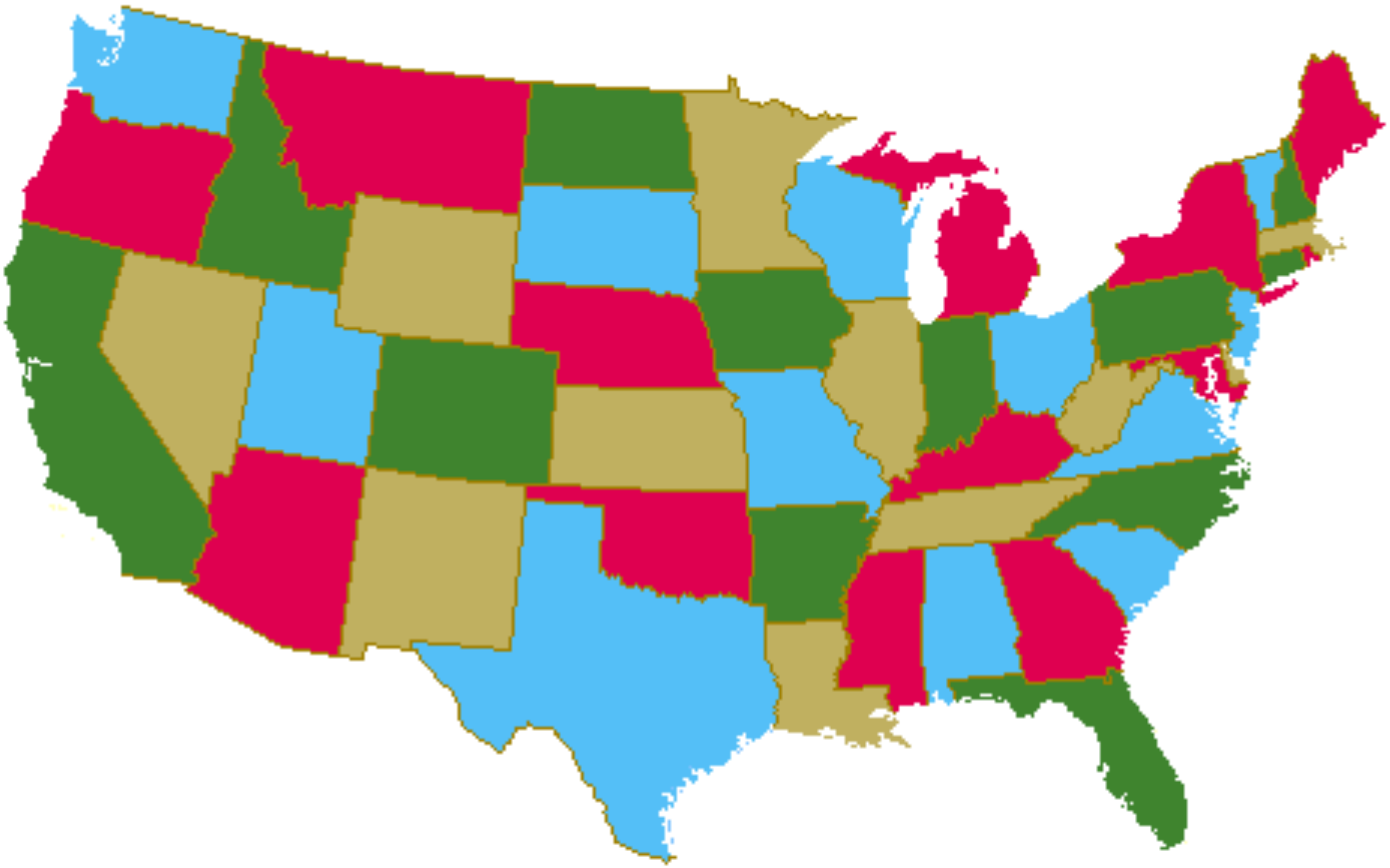
Four-color Theorem

Dates back to 1852 to Francis Guthrie

Any given plane separated into regions may be colored using no more than 4 colors

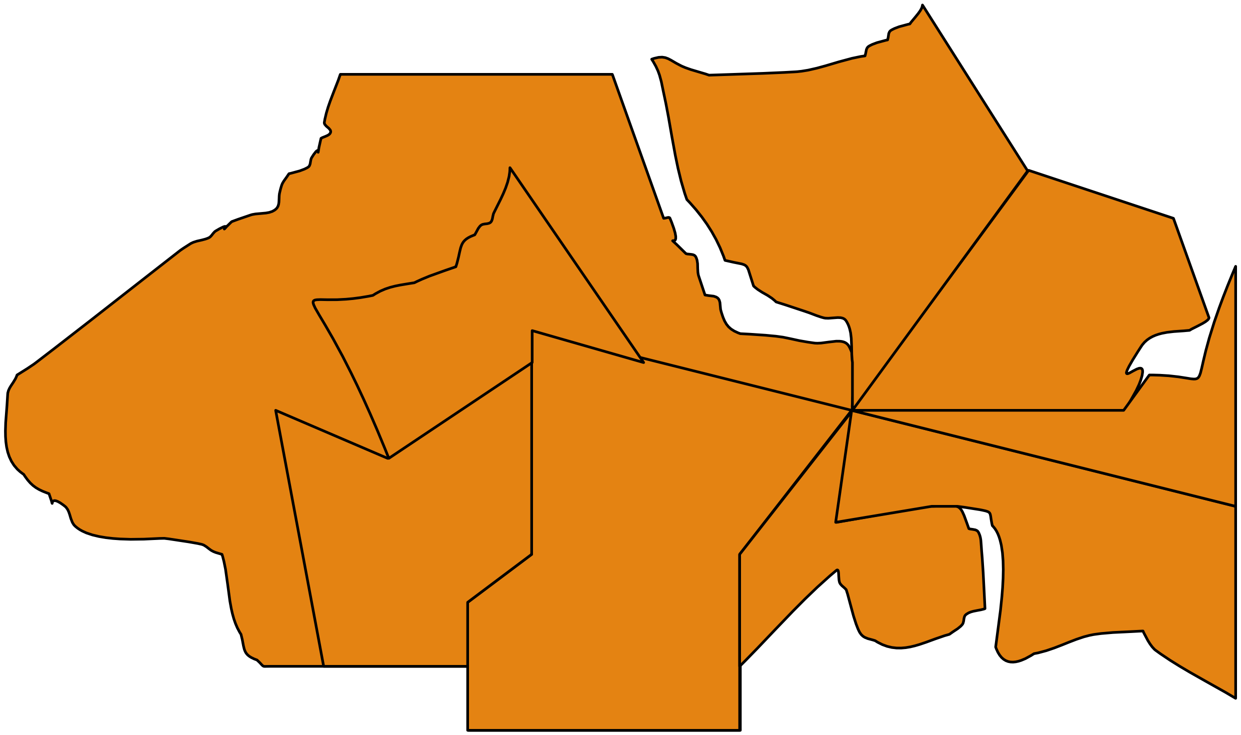
- Used for political boundaries, states, etc
- Shares common segment (not a point)

Four-color Theorem



Graph Coloring

Consider a fictional continent.



Map Coloring

Suppose removed all borders but still wanted to see all the countries.

1 color insufficient.

Map Coloring

So add another color. Try to fill in every country with one of the two colors.

Map Coloring

So add another color. Try to fill in every country with one of the two colors.

Map Coloring

So add another color. Try to fill in every country with one of the two colors.

Map Coloring

So add another color. Try to fill in every country with one of the two colors.

Map Coloring

PROBLEM: Two adjacent countries forced to have same color.

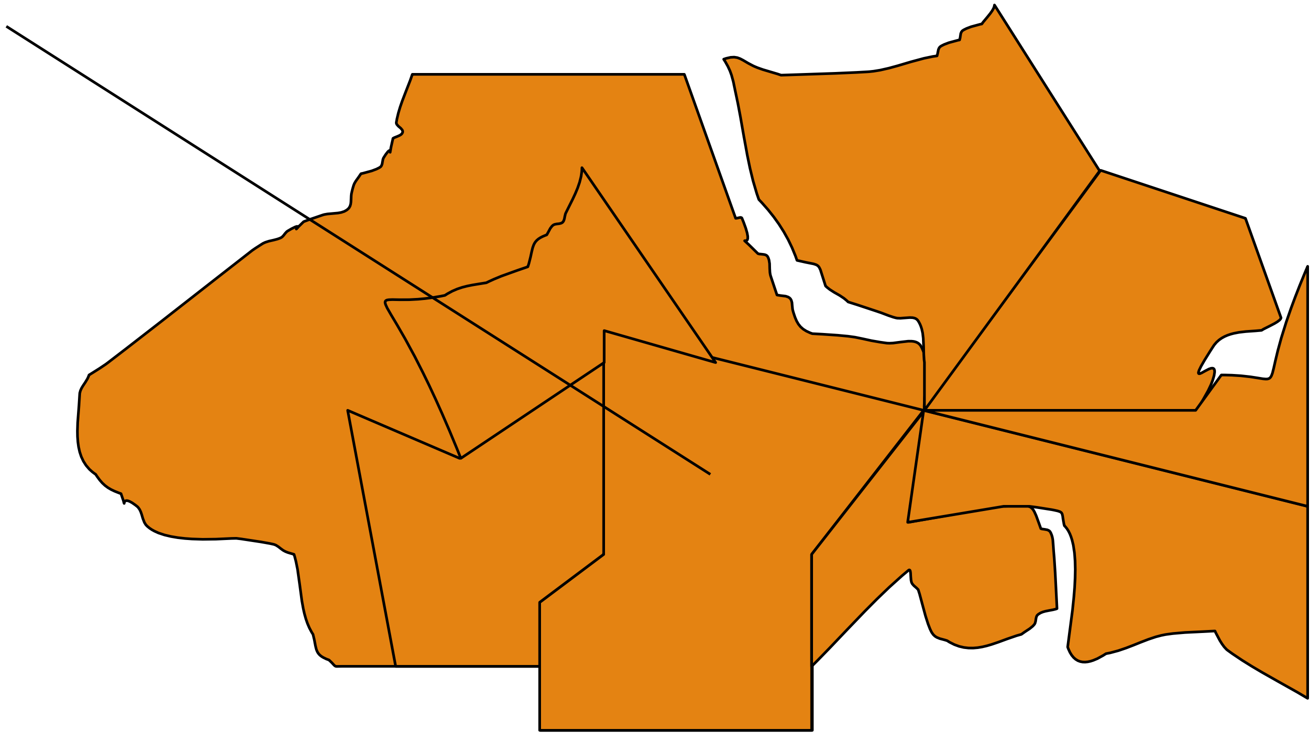
Border unseen.

Map Coloring

So add another color:

Map Coloring

Insufficient. Need 4 colors because of this country.



Map Coloring

With 4 colors, could do it.

4-Color Theorem

THM: Any planar map of regions can be depicted using 4 colors so that no two regions that share a positive-length border have the same color.

Proof by Haaken and Appel used exhaustive computer search.

Coloring a Graph - Applications

Sudoku

Scheduling

Mobile radio frequency assignment

Pattern matching

Planar Graphs

Planar graphs are graphs that can be drawn in the plane without edges having to cross.

Understanding planar graph is important:

Any graph representation of maps/ topographical information is planar.

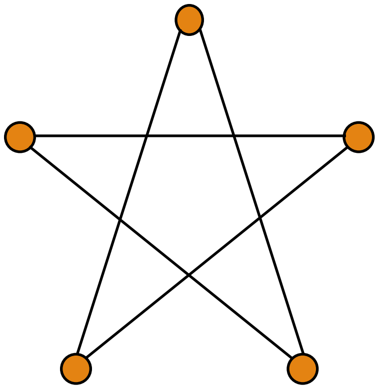
- graph algorithms often specialized to planar graphs (e.g. traveling salesperson)

Circuits usually represented by planar graphs

Planar Graphs
-Common Misunderstanding

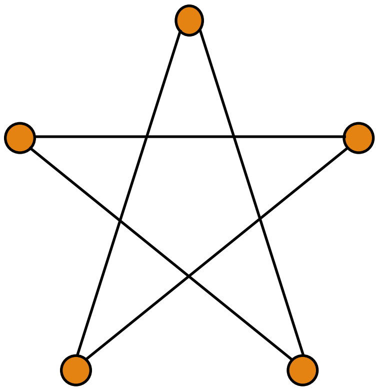
Just because a graph is drawn with edges crossing doesn't mean its not planar.

Q: Why can't we conclude that the following is non-planar?



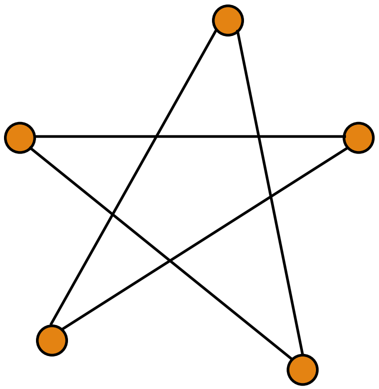
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



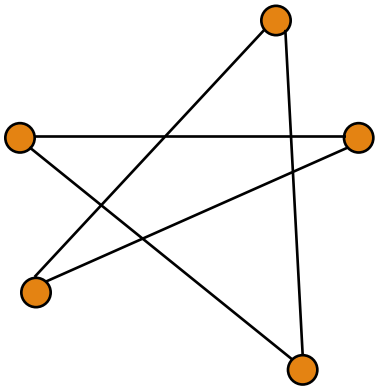
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



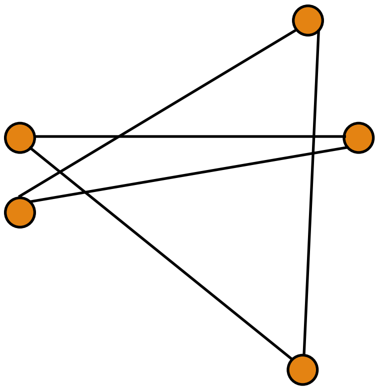
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



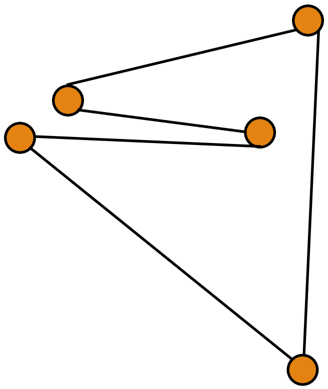
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



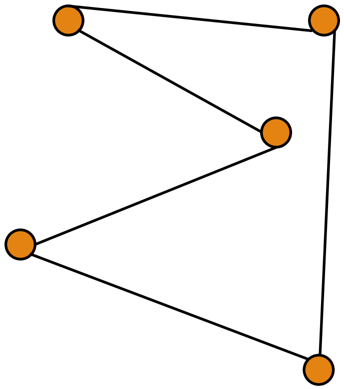
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



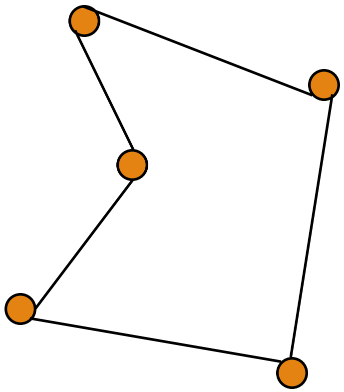
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



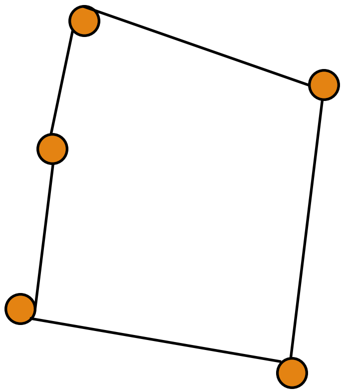
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



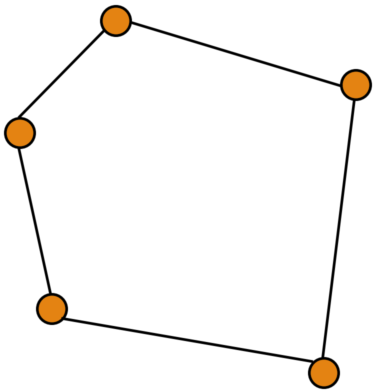
Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



Planar Graphs -Common Misunderstanding

A: Because it is isomorphic to a graph which *is* planar:



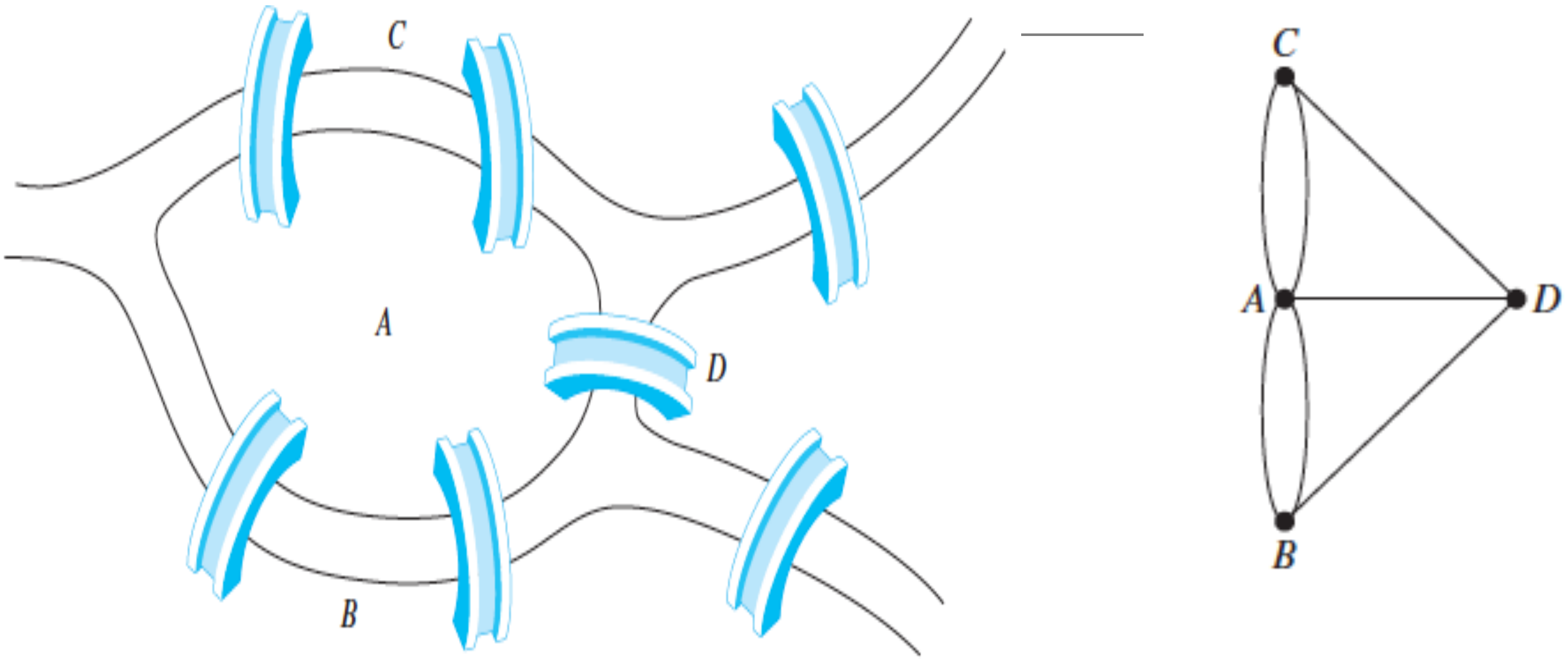
Proving Planarity

To prove that a graph is planar amounts to redrawing the edges in a way that no edges will cross. May need to move vertices around and the edges may have to be drawn in a very indirect fashion.

Some Interesting Problems

- The Konigsberg Bridges Problem
  - Multigraphs
  - Euler Walk
  - Traversability Problem
- The Travelling Salesman Problem (TSP)
  - Cheapest Hamiltonian Cycle
  - Cost associated with each edge

The Seven Bridges of Königsberg.

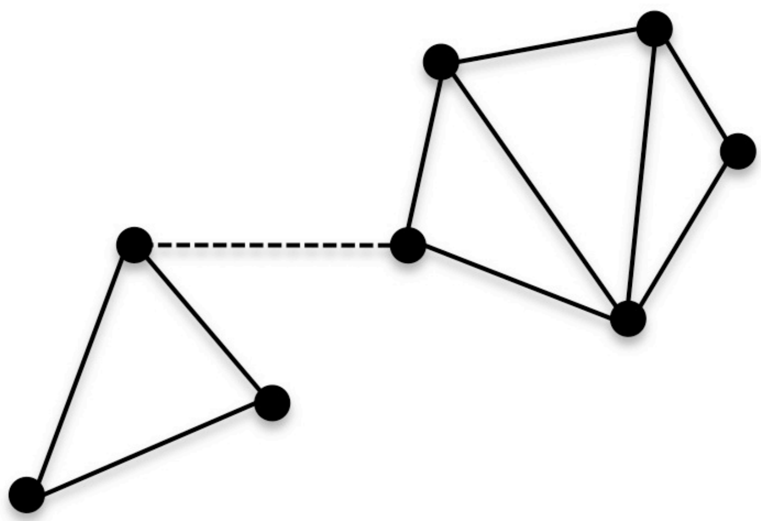


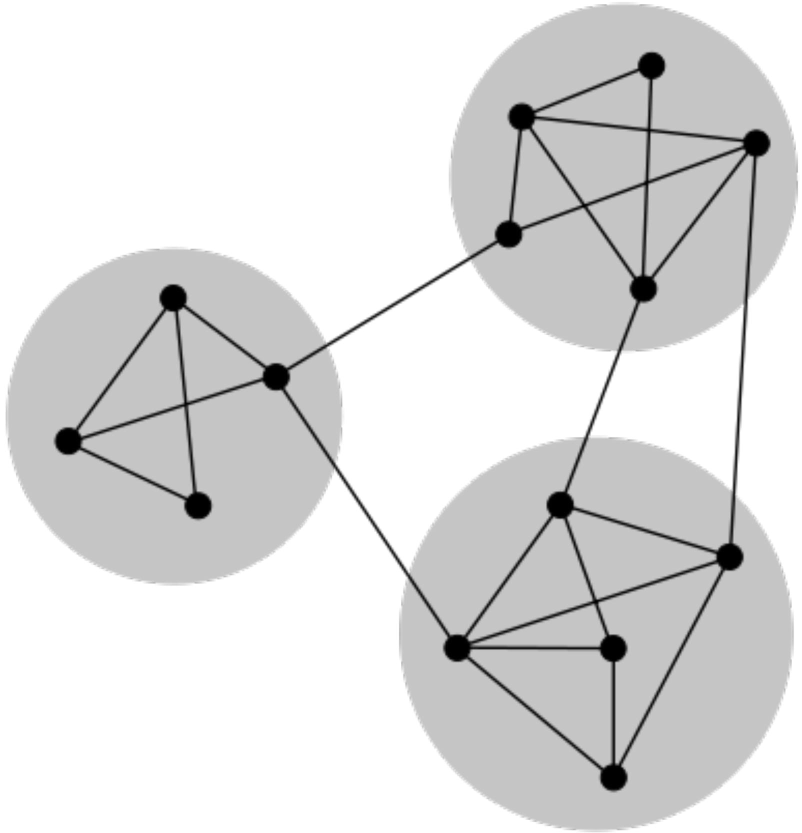
Can we start at any point in the town and return to the same location by crossing each bridge just once?

Euler & Hamiltonian Paths

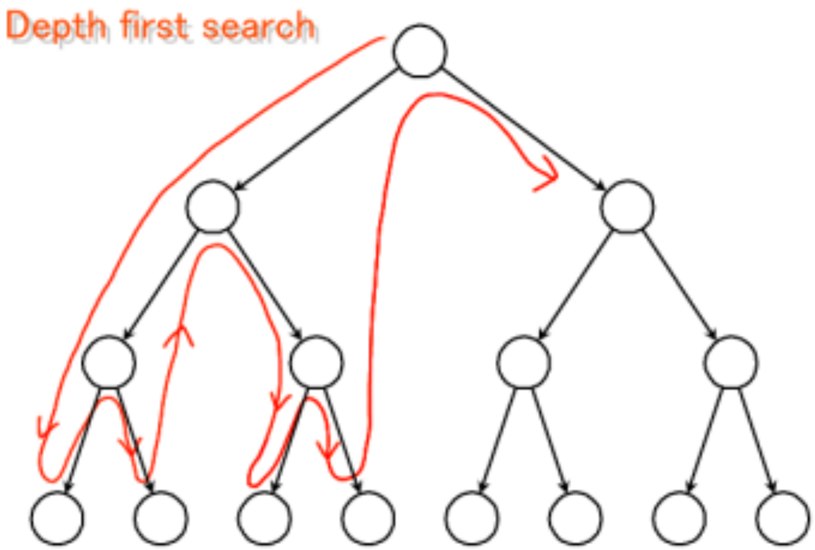
- Can we travel along the edges of a graph starting at a vertex and return to it after traversing each edge of the graph exactly once?
  - Euler circuit
- Similarly, can we travel along the edges of a graph starting at a vertex and return to it after traversing each vertex of the graph exactly once?
  - Hamiltonian circuit

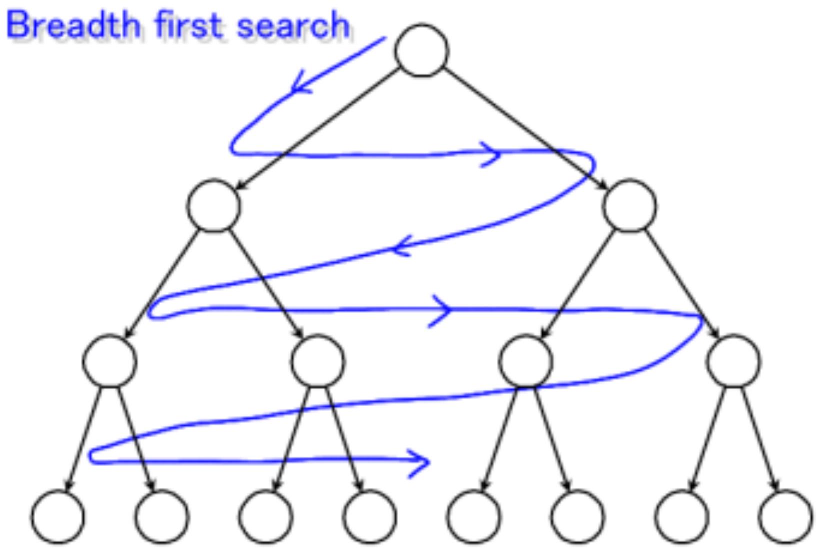
Connected Graphs





Graph Traversals





Trees

- A tree is a connected undirected graph with no cycles
- Represents parent-child relationships

Trees

- Oil Pipeline Problem
- 5 oil wells and a depot
- Required to build pipelines so that oil can be pumped from wells to the depot
- Oil can be pumped from one well to another easily (at very small cost)
- Major expense is building the pipelines
- Which pipelines are feasible to build?

Oil Pipeline Problem

- First impulse – connect depot to each well directly!!
- Cost: 26 lakhs!!
- Is this the cheapest solution?
- Feasibility graph to cheapest solution (tree)
- The conditions of the problem imply that the graph does not need to contain any cycle
- It must be connected though!

Oil Pipeline Problem

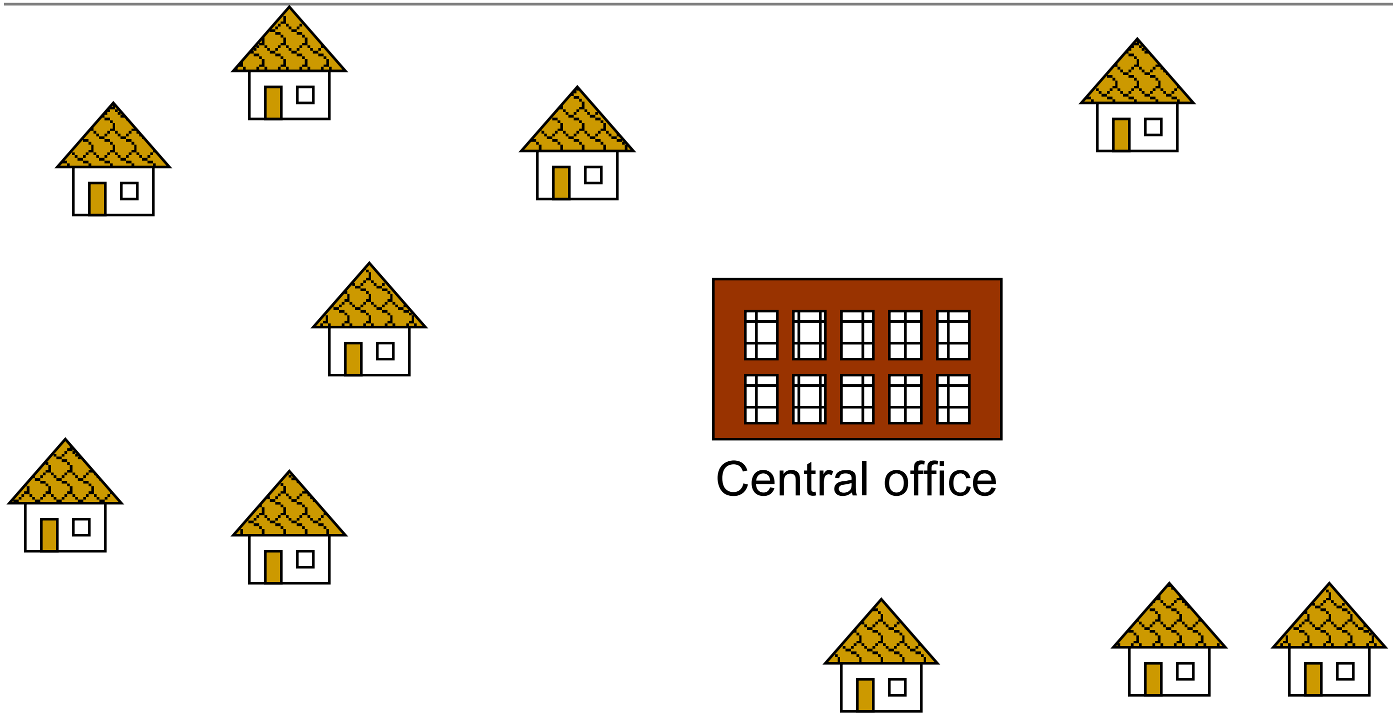
- A connected graph that contains no cycle is a TREE!!
- Solution of the problem is to find a subgraph of a given graph that is a TREE, and among those to find the one that is cheapest to build
- Spanning tree & Minimal spanning tree
- A spanning subgraph of a graph G is one that contains every vertex of G
- A spanning tree in a graph is a spanning subgraph that is a tree

Oil Pipeline Problem

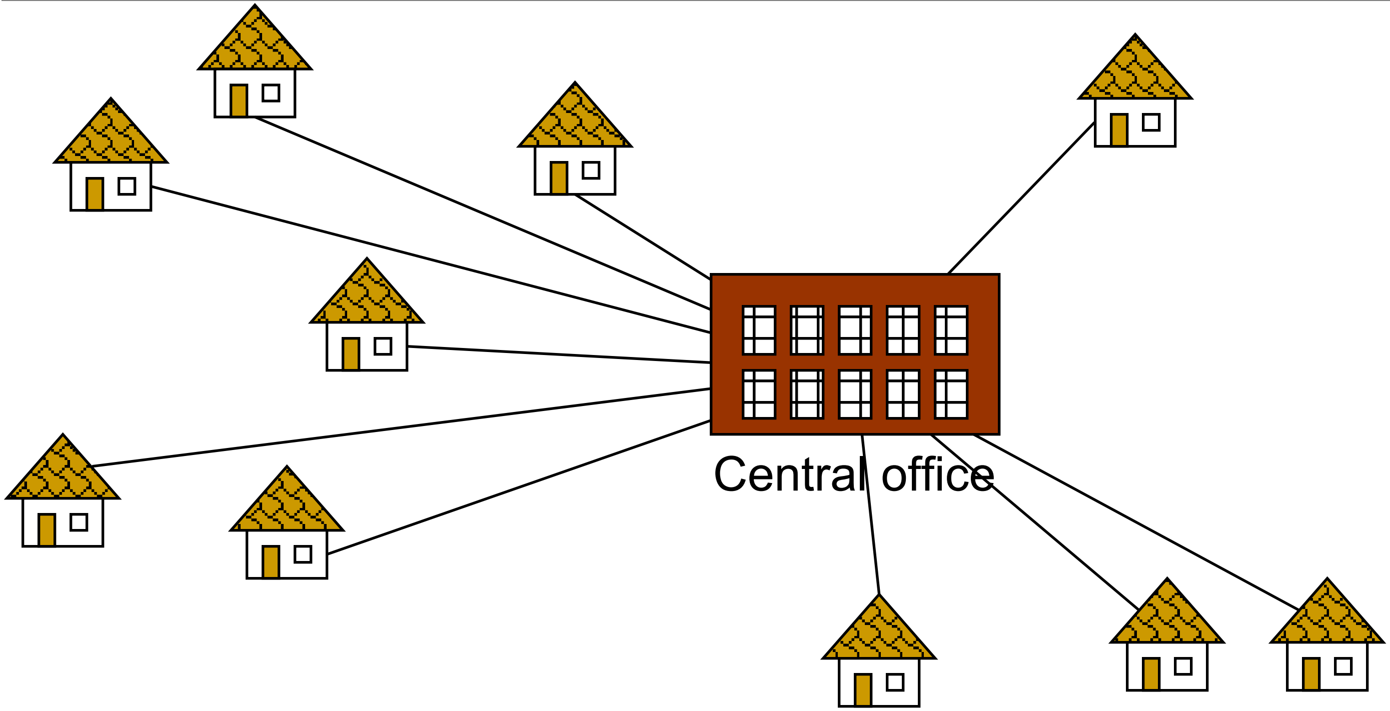
- In many applications, each edge of a graph has a weight/cost associated with it
- It is sometimes desirable to find a spanning tree such that the weight of the tree is minimum
- Minimum Spanning Tree (MST) !!

Problem taken from "A Beginner's Guide to Discrete Mathematics" by W D Wallis

Problem: Laying Telephone Wire

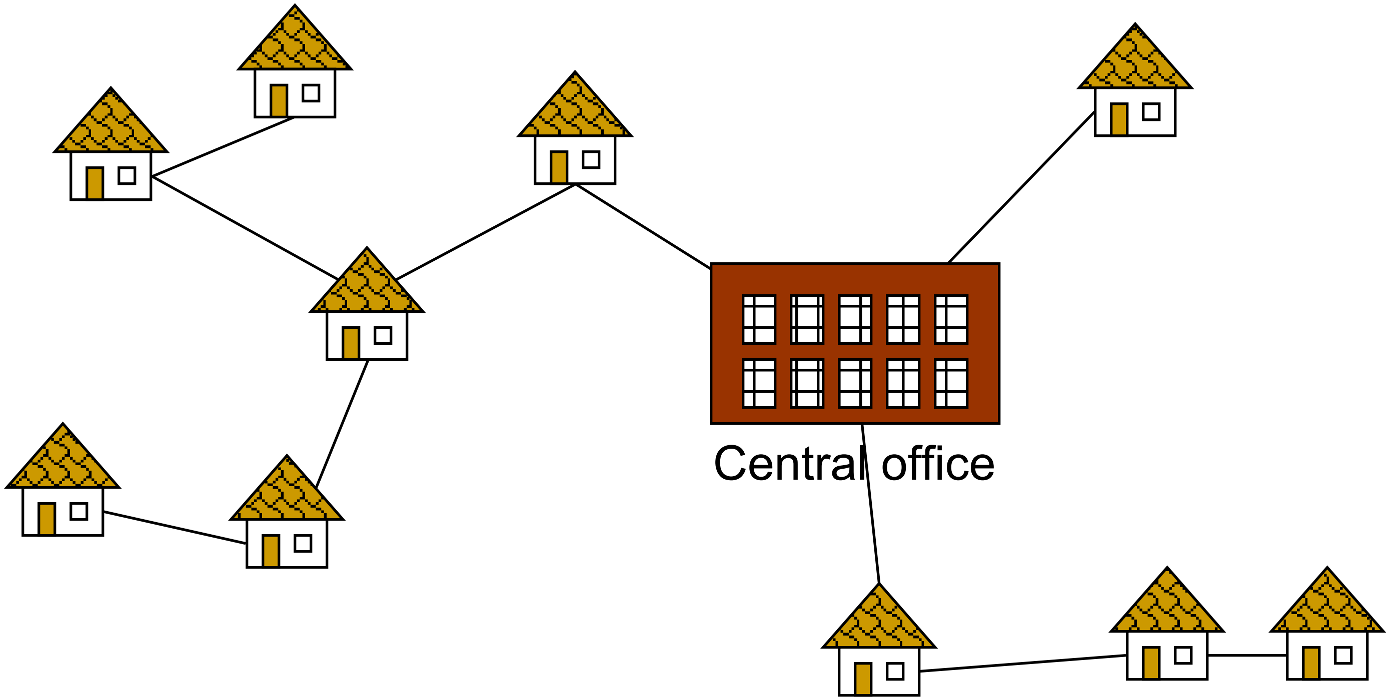


Wiring: Naïve Approach



Expensive!

Wiring: Better Approach



Minimize the total length of wire connecting the customers

Minimum Spanning Tree (MST)

A **minimum spanning tree** is a subgraph of an undirected weighted graph ***G***, such that

- it is a tree (i.e., it is acyclic)
- it covers all the vertices ***V***
  - contains ***|V| - 1*** edges
- the total cost associated with tree edges is the minimum among all possible spanning trees
- not necessarily unique

Applications of MST

Any time you want to visit all vertices in a graph at minimum cost (e.g., wire routing on printed circuit boards, sewer pipe layout, road planning…)

How to find MST?

- Kruskal's Algorithm (A greedy algorithm!!)
- Prim's Algorithm (A refinement of Kruskal's algorithm)

Groups, Rings, Fields, & Vector Scpaces

- Important Algebraic structures