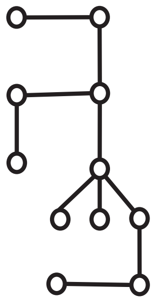
CompSci 102
Discrete Math for Computer Science



April 19, 2012

Prof. Rodger

Lecture adapted from Bruce Maggs/Lecture developed at Carnegie Mellon, primarily by Prof. Steven Rudich.

Announcements

- Read for next time Chap 10.3-4
- Last homework out, due Tuesday
- No Recitation on FRIDAY this week after all
- Graphs

Approximating Graphs and Solving Systems of Linear Equations

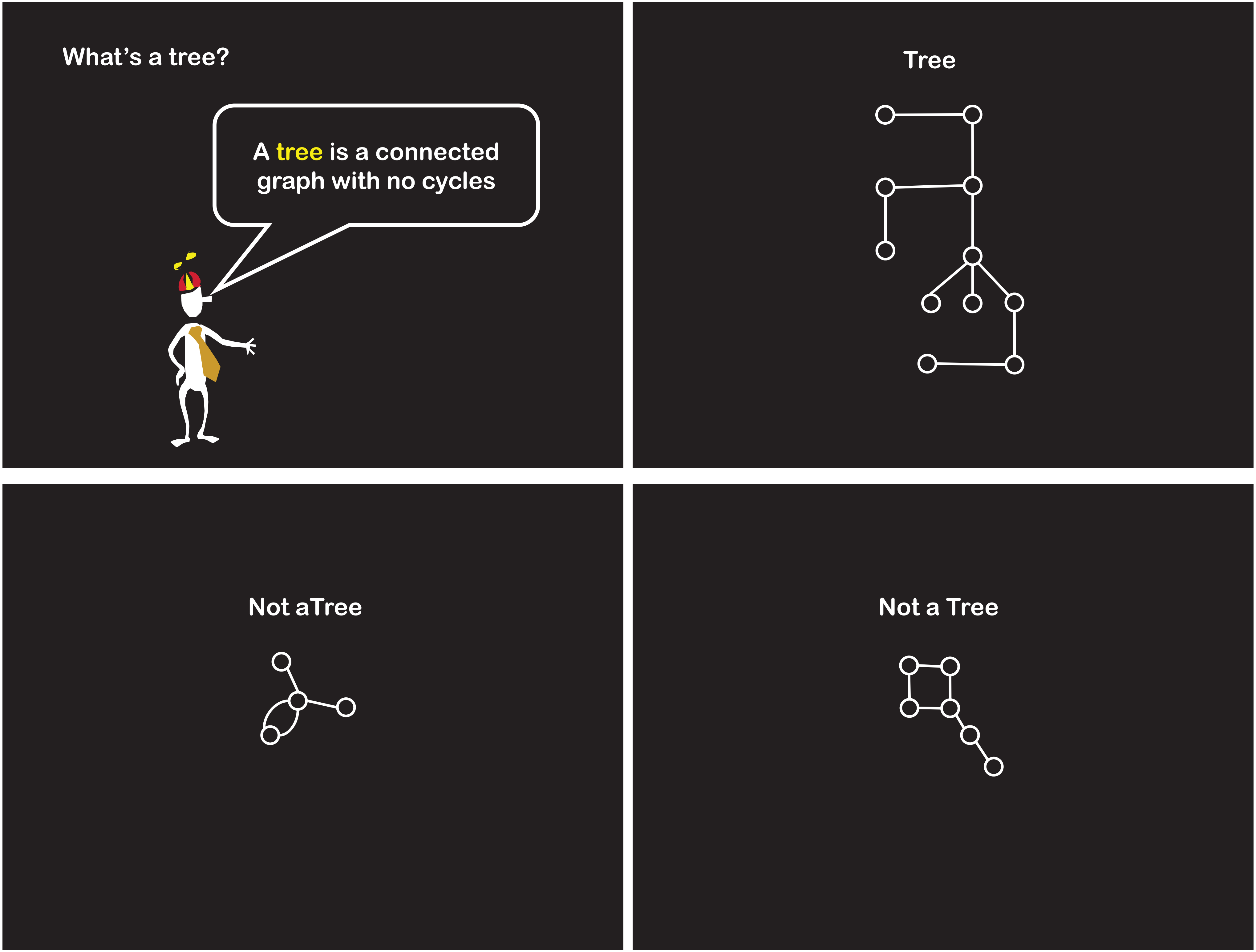
Speaker:	Daniel Spielman Yale University
Date:	Monday, April 23, 2012
Time:	4:00pm - 5:00pm
Location:	130A North Building, Duke

Abstract

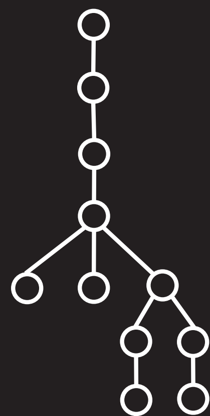
We survey recent advances in the design of algorithms for solving linear equations in the Laplacian matrices of graphs. These algorithms motivate and rely upon fascinating concepts in graph theory, the most important of which is a definition of what it means for one graph to approximate another. This definition leads to the problem of sparsification: the approximation of one graph by a sparser graph.

The sparsest possible approximation of a graph is a spanning tree. The average stretch of a spanning tree will be revealed to be a good measure of its approximation quality. We explain how every graph on n vertices can be well-approximated by a graph with O (n) edges. To solve linear equations, we will require something in between: approximations by graphs with n + O (n / log n) vertices. To build these sparse approximations, we employ low-stretch spanning trees, random matrix theory, spectral graph theory, and graph partitioning algorithms.

Graphs


What’s a tree?
A tree is a connected graph with no cycles
Tree
Not aTree
Not a Tree

Tree



How Many n-Node Trees?

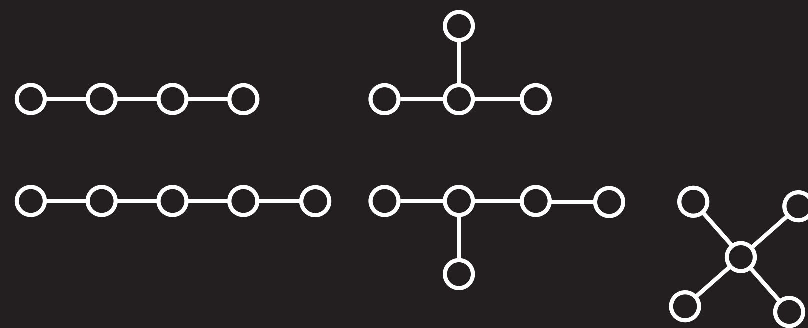
1:

2:

3:

4:

5:



Notation

In this lecture:

n will denote the number of nodes in a graph

e will denote the number of edges in a graph

Theorem: Let G be a graph with n nodes and e edges

The following are equivalent:

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path
3. G is connected and $n = e + 1$
4. G is acyclic and $n = e + 1$
5. G is acyclic and if any two non-adjacent points are joined by adding a new edge, the resulting graph has exactly one cycle

To prove this, it suffices to show

$$1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1$$

$1 \Rightarrow 2$

1. G is a tree (connected, acyclic)
2. Every two nodes of G are joined by a unique path

Proof: (by contradiction)

Assume G is a tree that has two nodes connected by two different paths:

Then there exists a cycle!

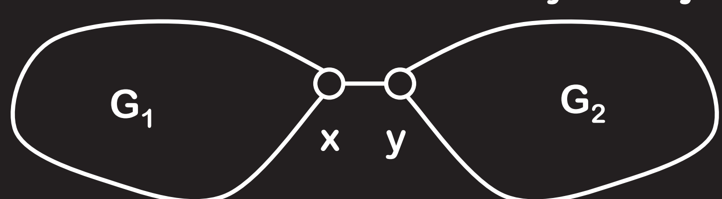
$2 \Rightarrow 3$

2. Every two nodes of G are joined by a unique path
3. G is connected and $n = e + 1$

Proof: (by induction)

Assume true for every graph with < n nodes

Let G have n nodes and let x and y be adjacent



Let n_1, e_1 be number of nodes and edges in G_1

Then $n = n_1 + n_2 = (e_1 + 1) + (e_2 + 1) = e + 1$

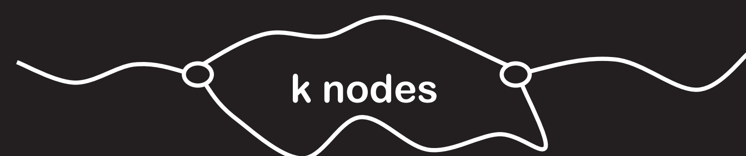
$= (e_1 + 1 + e_2) + 1 = e + 1$

$3 \Rightarrow 4$

3. G is connected and $n = e + 1$
4. G is acyclic and $n = e + 1$

Proof: (by contradiction)

Assume G is connected with $n = e + 1$, and G has a cycle containing k nodes



Note that the cycle has k nodes and k edges

Starting from cycle, add other nodes and edges until you cover the whole graph

Number of edges in the graph will be at least n

Corollary: Every nontrivial tree has at least two endpoints (points of degree 1)

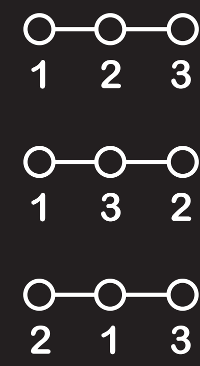
Proof:

Assume all but one of the points in the tree have degree at least 2

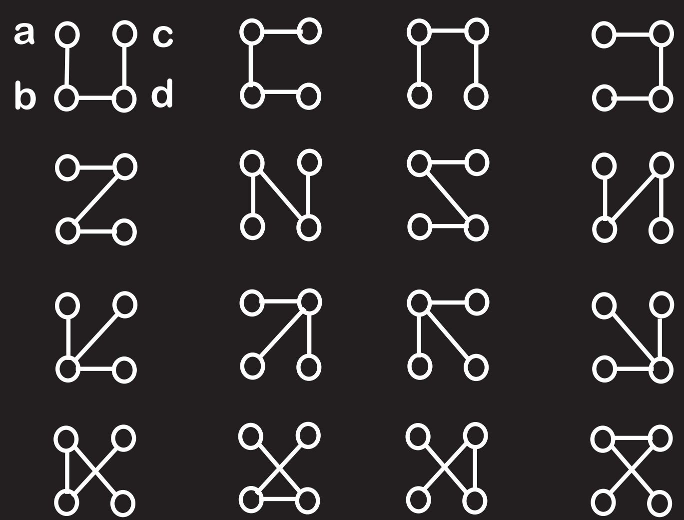
In any graph, sum of the degrees = 2e

Then the total number of edges in the tree is at least (2n-1)/2 = n - 1/2 > n - 1

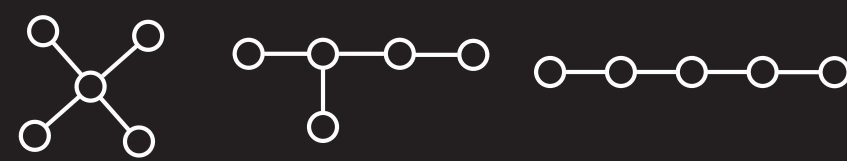
How many **labeled** trees are there with three nodes?



How many **labeled** trees are there with four nodes?



How many **labeled** trees are there with five nodes?



5 labelings

$5\binom{4}{3}3$ labelings

$5\binom{4}{2}2$ labelings

125 labeled trees

How many labeled trees are there with n nodes?

3 labeled trees with 3 nodes

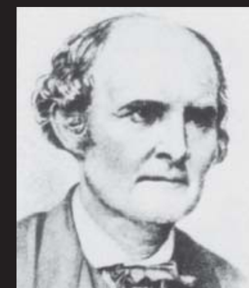
16 labeled trees with 4 nodes

125 labeled trees with 5 nodes

n^{n-2} labeled trees with n nodes

Cayley's Formula

The number of labeled trees on n nodes is n^{n-2}



The proof will use the correspondence principle

Each labeled tree on n nodes

corresponds to

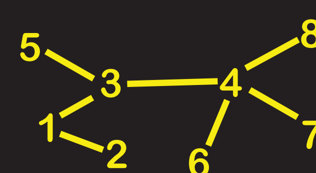
A sequence in $\{1,2,\ldots,n\}^{n-2}$ (that is, n-2 numbers, each in the range [1..n])

How to make a sequence from a tree?

Loop through i from 1 to n-2

- Let L be the degree-1 node with the lowest label
- Define the i^{th} element of the sequence as the label of the node adjacent to L
- Delete the node L from the tree

Example:



1 3 3 4 4 4

How to reconstruct the unique tree from a sequence S:

Let I = {1, 2, 3, …, n}

Loop until S is empty

Let i = smallest # in I but not in S

Let s = first label in sequence S

Add edge {i, s} to the tree

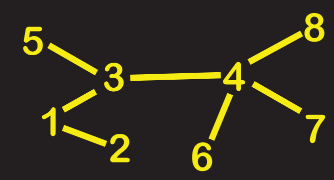
Delete i from I

Delete s from S

Add edge {a,b}, where I = {a,b}

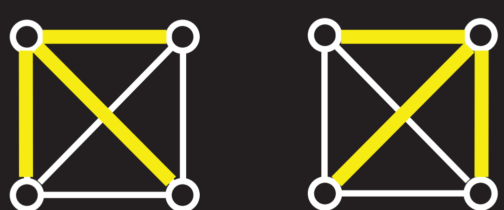
S: 1 3 3 4 4 4

I: 1 2 3 4 5 6 7 8

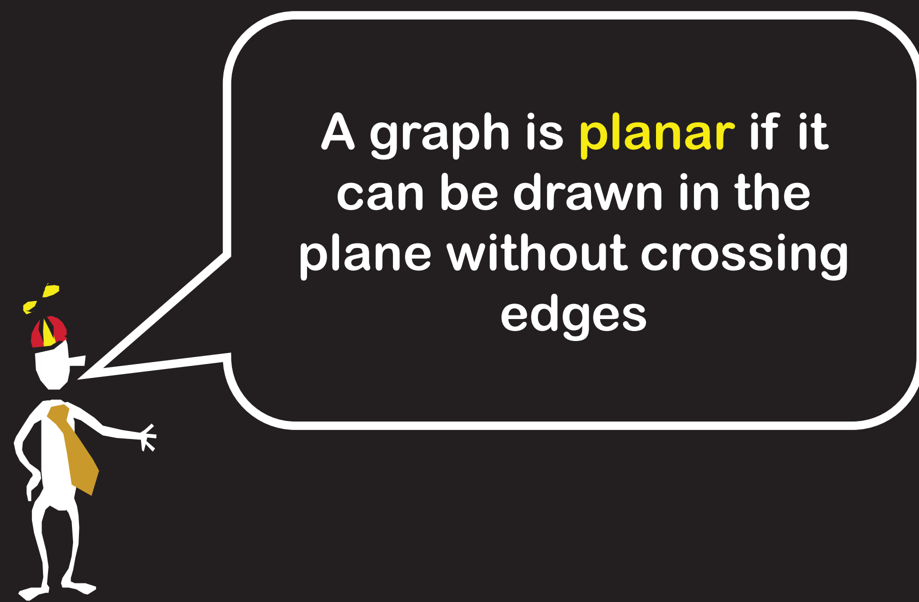


Spanning Trees

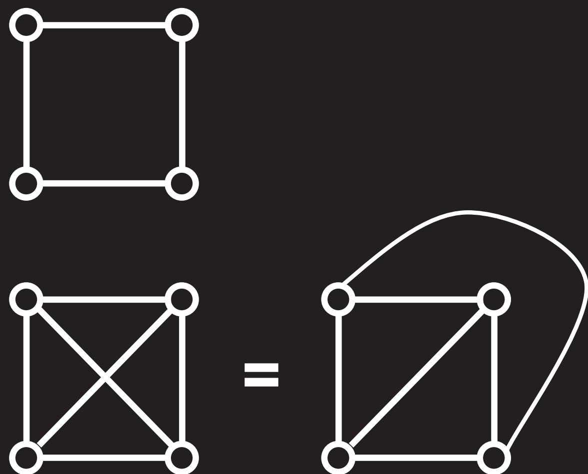
A spanning tree of a graph G is a tree that touches every node of G and uses only edges from G



Every connected graph has a spanning tree



Examples of Planar Graphs



Faces

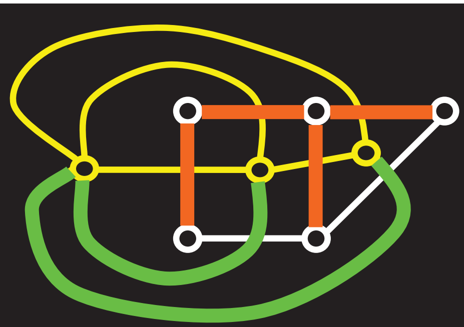
A planar graph splits the plane into disjoint faces

Euler's Formula

If G is a connected planar graph with n vertices, e edges and f faces, then $n - e + f = 2$

Rather than using induction, we'll use the important notion of the **dual graph**

Dual = put a node in every face, and an edge between every adjacent face



Let G^* be the dual graph of G

Let T be a spanning tree of G

Let T^* be the graph where there is an edge in dual graph for each edge in G – T

Then T^* is a spanning tree for G^*

$$n = e_T + 1 \qquad n + f = e_T + e_{T^*} + 2$$

$$f = e_{T^*} + 1 \qquad = e + 2$$

Corollary: Let G be a simple planar graph with n > 2 vertices. Then:

1. G has a vertex of degree at most 5
2. G has at most 3n – 6 edges

Proof of 1 (by contradiction):

In any graph, (sum of degrees) = 2e

Assume all vertices have degree ≥ 6

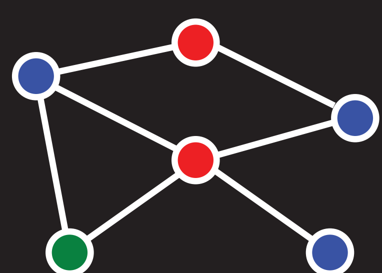
Then $e \geq 3n$

Furthermore, since G is simple, $3f \leq 2e$

So $3n + 3f \leq 3e$, $3n + 3f = 3e + 6$, and $3e + 6 \leq 3e$

Graph Coloring

A coloring of a graph is an assignment of a color to each vertex such that no neighboring vertices have the same color



Graph Coloring

Arises surprisingly often in CS

Register allocation: assign temporary variables to registers for scheduling instructions. Variables that interfere, or are simultaneously active, cannot be assigned to the same register

Theorem: Every planar graph can be 6-colored

Proof Sketch (by induction):

Assume every planar graph with less than n vertices can be 6-colored

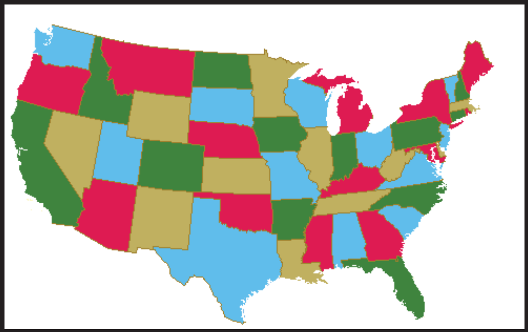
Assume G has n vertices

Since G is planar, it has some node v with degree at most 5

Remove v and color by Induction Hypothesis

Not too difficult to give an inductive proof of 5-colorability, using same fact that some vertex has degree ≤ 5

4-color theorem remains challenging!



Implementing Graphs

Adjacency Matrix

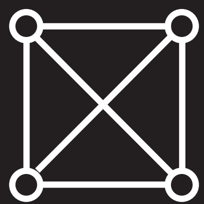
Suppose we have a graph G with n vertices. The adjacency matrix is the n x n matrix $A=[a_{ij}]$ with:

$a_{ij} = 1$ if (i,j) is an edge

$a_{ij} = 0$ if (i,j) is not an edge

Good for dense graphs!

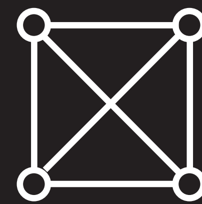
Example



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Counting Paths

The number of paths of length k from node i to node j is the entry in position (i,j) in the matrix A^k



$$A^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

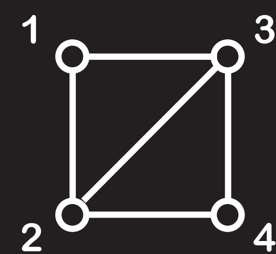
$$= \begin{pmatrix} 3 & 2 & 2 & 2 \\ 2 & 3 & 2 & 2 \\ 2 & 2 & 3 & 2 \\ 2 & 2 & 2 & 3 \end{pmatrix}$$

Adjacency List

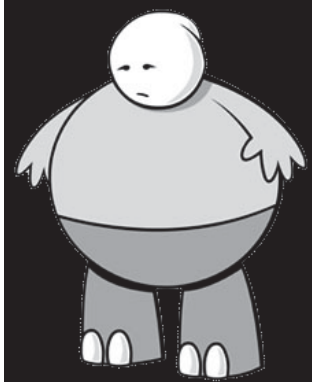
Suppose we have a graph G with n vertices. The adjacency list is the list that contains all the nodes that each node is adjacent to

Good for sparse graphs!

Example



1: 2,3
2: 1,3,4
3: 1,2,4
4: 2,3



Here's What You Need to Know…

Trees

- Counting Trees
- Different Characterizations

Planar Graphs

- Definition
- Euler's Theorem
- Coloring Planar Graphs

Adjacency Matrix and List

- Definition
- Useful for counting