Lecture 6.

Finish Euler's Formula.

Planar Five Color theorem.

Types of graphs.

- Complete Graphs.
- Trees.
- Hypercubes.

Planarity and Euler

These graphs **cannot** be drawn in the plane without edge crossings.

Euler's Formula: $v + f = e + 2$ for any planar drawing.

$\implies$ for simple planar graphs: $e \leq 3v - 6$.
Idea: Face is a cycle in graph of length 3.
Count face-edge incidences.

$\implies$ for bipartite simple planar graphs: $e \leq 2v - 4$.
Idea: face is a cycle in graph of length 4.
Count face-edge incidences.

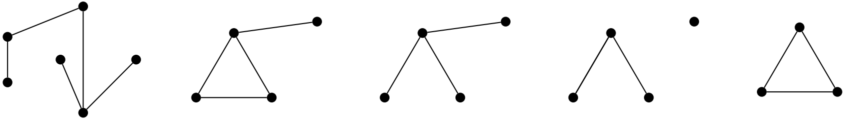
Proved absolutely no drawing can work for these graphs.

So......so ...Cool!

Tree.

A tree is a connected acyclic graph.

To tree or not to tree!



Yes. No. Yes. No. No.

Faces? 1. 2. 1. 1. 2.
Vertices/Edges. Notice: $e = v - 1$ for tree.

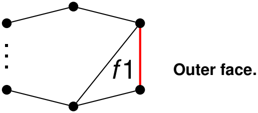
One face for trees!

Euler works for trees: $v + f = e + 2$.
$v + 1 = v - 1 + 2$

Euler's formula.

Euler: Connected planar graph has $v + f = e + 2$.

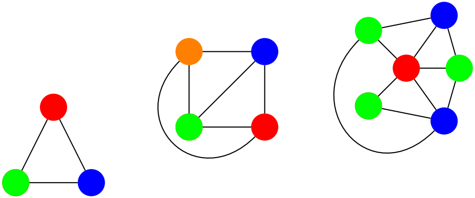
Proof: Induction on e.
Base: $e = 0$, $v = f = 1$.
Induction Step:
If it is a tree. Done.
If not a tree.
Find a cycle. Remove edge.



Joins two faces.
New graph: v-vertices. $e-1$ edges. $f-1$ faces. Planar.
$v + (f-1) = (e-1) + 2$ by induction hypothesis.
Therefore $v + f = e + 2$. $\square$

Graph Coloring.

Given $G = (V, E)$, a coloring of G assigns colors to vertices V where for each edge the endpoints have different colors.



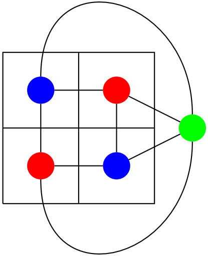
Notice that the last one, has one three colors.

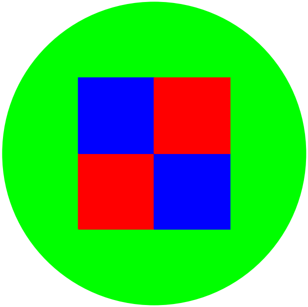
- Fewer colors than number of vertices.
- Fewer colors than max degree node.

Interesting things to do. Algorithm!

Planar graphs and maps.

Planar graph coloring $\equiv$ map coloring.





Four color theorem is about planar graphs!

Six color theorem.

Theorem: Every planar graph can be colored with six colors.

Proof:
Recall: $e \leq 3v - 6$ for any planar graph where $v > 2$.
From Euler's Formula.

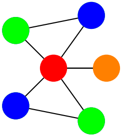
Total degree: $2e$
Average degree: $= \frac{2e}{v} \leq \frac{2(3v-6)}{v} \leq 6 - \frac{12}{v}$.

There exists a vertex with degree < 6 or at most 5.

Remove vertex v of degree at most 5.
Inductively color remaining graph.
Color is available for v since only five neighbors...
and only five colors are used. ☐

Five color theorem: prelimnary.

Preliminary Observation: Connected components of vertices with two colors in a legal coloring can switch colors.



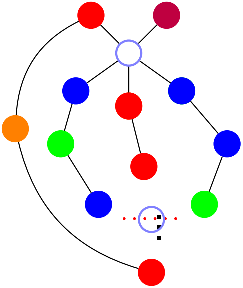
Look at only green and blue.
Connected components.
Can switch in one component.
Or the other.

Five color theorem

Theorem: Every planar graph can be colored with five colors.

Preliminary Observation: Connected components of vertices with two colors in a legal coloring can switch colors.

Proof: Again with the degree 5 vertex. Again recurse.



Assume neighbors are colored all differently.
Otherwise one of 5 colors is available. $\Longrightarrow$ Done!
Switch green and blue in green's component.
Done. Unless blue-green path to blue.
Switch orange and red in oranges component.
Done. Unless red-orange path to red.

Planar. $\Longrightarrow$ paths intersect at a vertex!

What color is it?
Must be blue or green to be on that path.
Must be red or orange to be on that path.

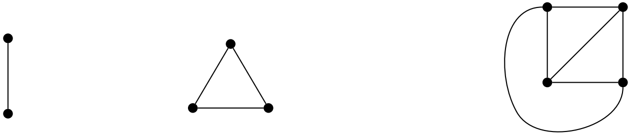
Contradiction. Can recolor one of the neighbors.
Gives an available color for center vertex! ☐

Four Color Theorem

Theorem: Any planar graph can be colored with four colors.

Proof: Not Today!

Complete Graph.



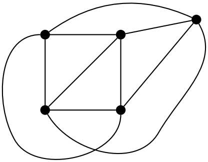
K_n complete graph on n vertices.

All edges are present.
Everyone is my neighbor.
Each vertex is adjacent to every other vertex.

How many edges?

Each vertex is incident to $n-1$ edges.
Sum of degrees is $n(n-1) = 2|E|$
$\implies$ Number of edges is $n(n-1)/2$.

K_4 and K_5



K_5 is not planar.

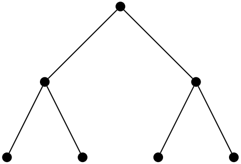
Cannot be drawn in the plane without an edge crossing!

Prove it! We did!

A Tree, a tree.

Graph $G = (V, E)$.

Binary Tree!



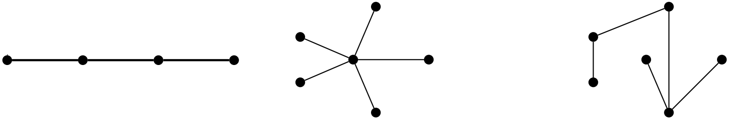
More generally.

Trees.

Definitions:

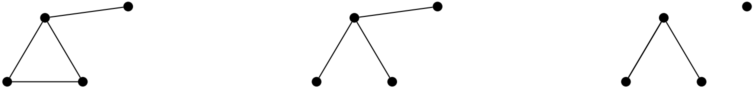
A connected graph without a cycle.
A connected graph with $|V| - 1$ edges.
A connected graph where any edge removal disconnects it.
A connected graph where any edge addition creates a cycle.

Some trees.



no cycle and connected? Yes.
$|V| - 1$ edges and connected? Yes.
removing any edge disconnects it. Harder to check. but yes.
Adding any edge creates cycle. Harder to check. but yes.

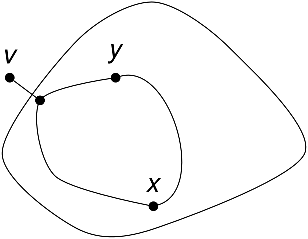
To tree or not to tree!



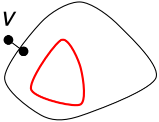
Equivalence of Definitions.

Theorem:
"G connected and has $|V|-1$ edges" $\equiv$
"G is connected and has no cycles."

Lemma: If v is a degree 1 in connected graph G, $G-v$ is connected.
Proof:
For $x \neq v, y \neq v \in V$,
there is path between x and y in G since connected.
and does not use v (degree 1)
$\implies$ $G-v$ is connected. $\square$



Proof of only if.



Thm:
"G connected and has $|V|-1$ edges" $\equiv$
"G is connected and has no cycles."

Proof of $\implies$: By induction on $|V|$.
Base Case: $|V|=1$. $0=|V|-1$ edges and has no cycles.

Induction Step:
Claim: There is a degree 1 node.
Proof: First, connected $\implies$ every vertex degree ≥ 1.
Sum of degrees is $2|V|-2$
Average degree $2-2/|V|$
Not everyone is bigger than average! ☐

By degree 1 removal lemma, $G-v$ is connected.
$G-v$ has $|V|-1$ vertices and $|V|-2$ edges so by induction
$\implies$ no cycle in $G-v$.
And no cycle in G since degree 1 cannot participate in cycle. ☐

Proof of if

Thm:
"G is connected and has no cycles"
$\implies$ "G connected and has $|V| - 1$ edges"

Proof:
Walk from a vertex using untraversed edges.
Until get stuck.

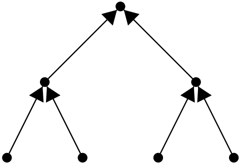
Claim: Degree 1 vertex.

Proof of Claim:
Can't visit more than once since no cycle.
Entered. Didn't leave. Only one incident edge. $\square$

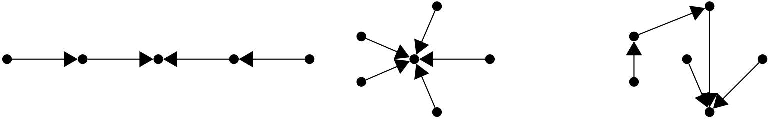
Removing node doesn't create cycle.
New graph is connected.
Removing degree 1 node doesn't disconnect from Degree 1 lemma.
By induction $G - v$ has $|V| - 2$ edges.
G has one more or $|V| - 1$ edges. $\square$

Tree's fall apart.

Thm: There is one vertex whose removal disconnects $|V|/2$ nodes from each other.



Idea of proof.
Point edge toward bigger side.
Remove center node.



Hypercubes.

Complete graphs, really connected! But lots of edges.

$|V|(|V|-1)/2$

Trees, few edges. $(|V|-1)$

but just falls apart!

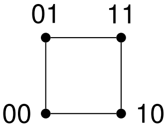
Hypercubes. Really connected. $|V|\log|V|$ edges!

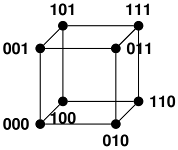
Also represents bit-strings nicely.

$G=(V,E)$

$|V|=\{0,1\}^n$,

$|E|=\{(x,y)|x$ and y differ in one bit position.$\}$





2^n vertices. number of n-bit strings!

$n2^{n-1}$ edges.

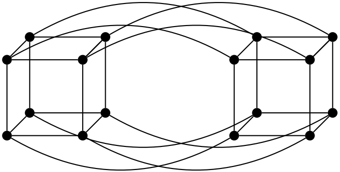
2^n vertices each of degree n

total degree is $n2^n$ and half as many edges!

Recursive Definition.

A 0-dimensional hypercube is a node labelled with the empty string of bits.

An n-dimensional hypercube consists of a 0-subcube (1-subcube) which is a $n-1$-dimensional hypercube with nodes labelled $0x$ $(1x)$ with the additional edges $(0x, 1x)$.



Hypercube: Can't cut me!

Thm: Any subset S of the hypercube where $|S| \le |V|/2$ has $\ge |S|$ edges connecting it to $V - S$; $|E \cap S \times (V - S)| \ge |S|$

Terminology:
  $(S, V - S)$ is cut.
  $(E \cap S \times (V - S))$ - cut edges.

Restatement: for any cut in the hypercube, the number of cut edges is at least the size of the small side.

Proof of Large Cuts.

Thm: For any cut $(S, V - S)$ in the hypercube, the number of cut edges is at least the size of the small side.

Proof:

Base Case: $n = 1$ V= $\{0,1\}$.

$S = \{0\}$ has one edge leaving. $|S| = \phi$ has 0.

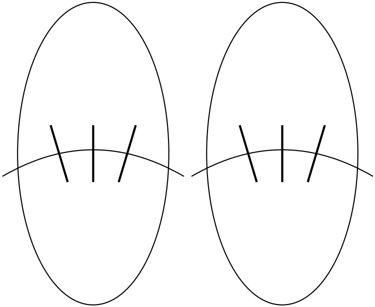
Induction Step Idea

Thm: For any cut $(S, V - S)$ in the hypercube, the number of cut edges is at least the size of the small side.

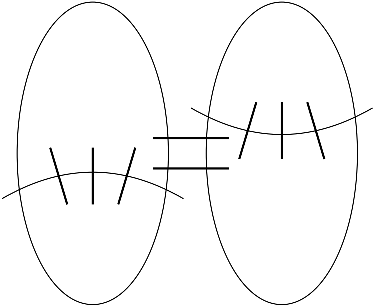
Use recursive definition into two subcubes.

Two cubes connected by edges.

Case 1: Count edges inside subcube inductively.



Case 2: Count inside and across.



Induction Step

Thm: For any cut $(S, V - S)$ in the hypercube, the number of cut edges is at least the size of the small side, $|S|$.

Proof: Induction Step.
Recursive definition:
$H_0 = (V_0, E_0)$,$H_1 = (V_1, E_1)$, edges E_x that connect them.
$H = (V_0 \cup V_1, E_0 \cup E_1 \cup E_x)$
$S = S_0 \cup S_1$ where S_0 in first, and S_1 in other.

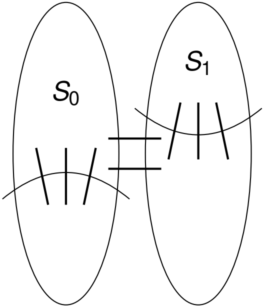
Case 1: $|S_0| \leq |V_0|/2, |S_1| \leq |V_1|/2$
Both S_0 and S_1 are small sides. So by induction.
Edges cut in $H_0 \geq |S_0|$.
Edges cut in $H_1 \geq |S_1|$.

Total cut edges $\geq |S_0| + |S_1| = |S|$. □

Induction Step. Case 2.

Thm: For any cut $(S, V-S)$ in the hypercube, the number of cut edges is at least the size of the small side, $|S|$.

Proof: Induction Step. Case 2.



$|S_0| \geq |V_0|/2$.

Recall Case 1: $|S_0|, |S_1| \leq |V|/2$

$|S_1| \leq |V_1|/2$ since $|S| \leq |V|/2$.

$\implies \geq |S_1|$ edges cut in E_1.

$|S_0| \geq |V_0|/2 \implies |V_0 - S| \leq |V_0|/2$

$\implies \geq |V_0| - |S_0|$ edges cut in E_0.

Edges in E_x connect corresponding nodes.

$\implies = |S_0| - |S_1|$ edges cut in E_x.

Total edges cut:

$\geq |S_1| + |V_0| - |S_0| + |S_0| - |S_1| = |V_0|$

$|V_0| = |V|/2 \geq |S|$. $\square$

Also, case 3 where $|S_1| \geq |V|/2$ is symmetric.

Hypercubes and Boolean Functions.

The cuts in the hypercubes are exactly the transitions from 0 sets to 1 set on boolean functions on $\{0,1\}^n$.

Central area of study in computer science!

Yes/No Computer Programs $\equiv$ Boolean function on $\{0,1\}^n$

Central object of study.

Summary.

We did lots today!

Euler, coloring, types of graphs.

And Isoperimetric inequality for Hypercubes.

Welcome to Berkeley!

Have a nice weekend!