Today.

Types of graphs.
Complete Graphs.
Trees.
Hypercubes.

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 $n(n - 1)$.
$⇒$ Number of edges is $n(n - 1)/2$.
Remember sum of degree is $2|E|$.

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.

Trees: Definitions

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.

Trees: Equivalence of Definitions

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

Proof of $⇒$ (only if): By induction on $|V|$.
Base Case: $|V| = 1$. 0 = $|V| - 1$ edges and has no cycles.
Induction Step: Assume for $G$ with up to $k$ vertices. Prove for $k + 1$ 
Consider some vertex $v$ in $G$. How is it connected to the rest of $G$?
Might it be connected by just 1 edge?
Is there a Degree 1 vertex?
Is the rest of $G$ connected?

$K_4$ and $K_5$

$K_5$ is not planar.
Cannot be drawn in the plane without an edge crossing!
Prove it! Read Note 5!!
**Equivalence of Definitions: Useful Lemma**

**Theorem:**
“G connected and has \( |V|−1 \) edges” = “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.

**Proof of only if.**

**Thm:**
“G connected and has \( |V|−1 \) edges” = “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: Assume for \( G \) up to \( k \) vertices. Prove for \( k + 1 \)
Claim: There is a degree 1 node.
Proof: First, connected \( \implies \) every vertex degree \( \geq 1 \).
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 part”**

**Thm:**
“G is connected and has no cycles” \( \implies \) “G connected and has \( |V|−1 \) edges”

**Proof:**
Can we use the “degree 1” idea again?
Walk from a vertex using untraversed edges and vertices.
Until get stuck. Why? Finitely-many vertices, no cycle!
Claim: Degree 1 vertex.
Proof of Claim:
Can’t visit more than once since no cycle.
New graph is connected. (from our Degree 1 lemma).
By induction \( G−v \) has \( |V|−2 \) edges.
\( G \) has one more or \( |V|−1 \) edges.

**Hypercubes.**

Complete graphs, really well connected! Lots of edges.
\(|V|(|V|−1)/2\)
Trees, connected, few edges.
\(|V|−1\)

Hypercubes. Well connected. \(|V|\log|V| \) edges!
Also represents bit-strings nicely.
\( G = (V,E) \)
\(|V| = (0,1)^n\)
\(|E| = \{(x,y) | x \text{ and } y \text{ differ in one bit position.}\}

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).

**Recursive Definition.**

*Thm:* Any subset \( S \) of the hypercube where \( |S| \leq |V|/2 \) has \( \geq |S| \) edges connecting it to \( V−S \):
\(|E \cap S \times (V−S)| \geq |S|\)

**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: \(n = 1\) \(V = \{0, 1\}\).
\(S = \emptyset\) has one edge leaving.
\(S = \emptyset\) has 0.

**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. 

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 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, \(|S|\). 

**Proof: Induction Step.**
Recursive definition:

- \(H_0 = (V_0, E_0)\), \(H_1 = (V_1, E_1)\), edges \(E_1\) that connect them.
- \(S = S_0 \cup S_1\), 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

**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.**

| \(|S_0| \geq |V_0|/2, \text{ since } |S| \leq |V|/2\) |
|---|
|\(|S_1| \geq |V_1|/2\) |
|\(|S_0| \geq |V_0|/2\) |
|\(|V_0| - |S_0| \leq |V_0|/2\) |
|\(\Rightarrow |V_0| \geq |S_0|\) |
|\(\text{edges cut in } E_0\) |
|Edgess in \(E_1\) connect corresponding nodes.
\(\Rightarrow |S_1| = |S_1|\) |
|\text{edges cut in } E_1\) |
|Total edges cut: \(|S_1| + |V_0| - |S_0| + |S_0| - |S_1| = |V_0|\) |
|\(|V_0| = |V|/2 \geq |S|\) |

Also, case 3 where \(|S_1| \geq |V_0|/2\) 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.