Lecture Today.

To homework (score) or not to homework (score)
Do proofs of optimality/pessimality again.
Graphs

Job Propose and Candidate Reject is optimal!

For jobs? For candidates?

Theorem: Job Propose and Reject produces a job-optimal pairing.

Proof:
Assume not: there is a job $b$ does not get optimal candidate, $g$.
There is a stable pairing $S$ where $b$ and $g$ are paired.
Let $t$ be first day job $b$ gets rejected by its optimal candidate $g$ who it is paired with in stable pairing $S$.
$b'$ - knocks $b$ off of $g$'s string on day $t$ \(\implies g\) prefers $b'$ to $b$
By choice of $t$, $b'$ likes $g$ at least as much as optimal candidate.
\(\implies b'\) prefers $g$ to its partner $g'$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable. $(b',g')\in S$. But $(b',g)$ is rogue couple!
Used Well-Ordering principle...Induction.

How about for candidates?

Theorem: Job Propose and Reject produces candidate-pessimal pairing.

$T$ – pairing produced by JPR.
$S$ – worse stable pairing for candidate $g$.
In $T$, $(g, b)$ is pair.
In $S$, $(g, b')$ is pair.
$g$ prefers $b$ to $b'$.
$T$ is job optimal, so $b$ prefers $g$ to its partner in $S$.
$(g, b)$ is Rogue couple for $S$.
$S$ is not stable. Contradiction.

Notes: Not really induction.
Structural statement: Job optimality \(\implies\) Candidate pessimality.

Lecture 5: Graphs.

Graphs!
Definitions: model.
Fact!
Planar graphs.
Euler Again!!!

Map Coloring.

Fewer Colors?
Yes! Three colors.
Fewer Colors?

Four colors required!
Theorem: Four colors enough.

Yes! Three colors.

Scheduling: coloring.

Exam Slot 1.
Exam Slot 2.
Exam Slot 3.
**Graphs: formally.**

Graph: $G = (V, E)$.
- $V$ - set of vertices.
- $E$ - set of edges.
- $E \subseteq V \times V$ - set of edges.

For CS 70, usually simple graphs.
No parallel edges.
Multigraph above.

**Directed Graphs**

$G = (V, E)$.
- $V$ - set of vertices.
- $E$ ordered pairs of vertices.
- $(1, 2), (1, 3), (1, 4), (2, 4), (3, 4)$

One way streets.
Tournament: 1 beats 2, ...
Precedence: 1 is before 2, ...
Social Network: Directed? Undirected?
Friends. Undirected.
Likes. Directed.

**Graph Concepts and Definitions.**

- Neighbors of 10? 1, 5, 7, 8.
- $u$ is neighbor of $v$ if $(u, v) \in E$.
- Edge $(10, 5)$ is incident to vertex 10 and vertex 5.
- Edge $(u, v)$ is incident to $u$ and $v$.
- Degree of vertex 1? 2
- Degree of vertex $u$ is number of incident edges.
- Equals number of neighbors in simple graph.
- Directed graph?
  - In-degree of 10? 1
  - Out-degree of 10? 3

**Sum of degrees?**

The sum of the vertex degrees is equal to
(A) the total number of vertices, $|V|$.
(B) the total number of edges, $|E|$.
(C) What?

Not (A)! Triangle.
Not (B)! Triangle.

What? For triangle number of edges is 3, the sum of degrees is 6.
Could it always be... $2|E|$? or $2|V|$?

**Quick Proof of an Equality.**

The sum of the vertex degrees is equal to ??

Recall:
- edge, $(u, v)$, is incident to endpoints, $u$ and $v$.
- degree of $u$ number of edges incident to $u$

Let’s count incidences in two ways.

How many incidences does each edge contribute? 2.
Total Incidences? $|E|$ edges, 2 each. $\rightarrow 2|E|$.
What is degree $v$? Incidences corresponding to $v$!
Total Incidences? The sum over vertices of degrees!

Thm: Sum of vertex degrees is $2|E|$.

**Paths, walks, cycles, tour.**

A path in a graph is a sequence of edges.

Path? $(1, 10), (8, 5), (4, 5)$? No!
Path? $(1, 10), (10, 5), (5, 4), (4, 11)$? Yes!
Path: $(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)$.
Quick Check! Length of path? $k$ vertices or $k - 1$ edges.
Cycle: Path with $v_1 = v_k$. Length of cycle? $k - 1$ vertices and edges!
Path is usually simple. No repeated vertex!
Walk is sequence of edges with possible repeated vertex or edge.
Tour is walk that starts and ends at the same node.
Quick Check!
Path is to Walk as Cycle is to ?? Tour!
Directed Paths.

1. Take a walk starting from $v_1$ (on "unused" edges... till you get back to $v_1$.
2. Remove tour, $C$.
3. Let $G_1, \ldots, G_k$ be connected components. Each is touched by $C$.
4. Recurse on $G_1, \ldots, G_k$ starting from $v_i$.
5. Splice together.

$\{1, 10, 7, 8, 5, 4, 3, 11\}, \{2, 9, 6\}$.

Connected Components: Quiz.

Eulerian Tour

Konigsberg bridges problem.

Euclidian Tour is connected so graph is connected.
Uses two incident edges per visit. Tour uses all incident edges. Therefore $v$ has even degree.

Can you draw a tour in the graph where you visit each edge once? Yes? No?
We will see!

Can you make a tour visiting each bridge exactly once?

Konigsberg bridges problem.

"Konigsberg bridges" by Bogdan Giusca - License.

Connectivity: undirected graph.

$u$ and $v$ are connected if there is a path between $u$ and $v$.
A connected graph is a graph where all pairs of vertices are connected.

If one vertex $x$ is connected to every other vertex.
Is graph connected? Yes? No?
Proof: Use path from $u$ to $x$ and then from $x$ to $v$.
May not be simple:
Either modify definition to walk.
Or cut out cycles.

Quick Check: Is $\{10, 7, 5\}$ a connected component? No.

Finding a tour!

Proof of if: Even + connected $\Rightarrow$ Eulerian Tour.
We will give an algorithm. First by picture.

1. Take a walk starting from $v_1$ on "unused" edges...
2. Remove tour, $C$.
3. Let $G_1, \ldots, G_k$ be connected components. Each is touched by $C$.
4. Recurse on $G_1, \ldots, G_k$ starting from $v_i$.
5. Splice together.

1, 10, 7, 8, 5, 4, 3, 11, 4, 5, 2, 6, 9, 2 and to 1!

Eulerian Tour

An Eulerian Tour is a tour that visits each edge exactly once.

Theorem: Any undirected graph has an Eulerian tour if and only if all vertices have even degree and is connected.
Proof of only if: Eulerian $\Rightarrow$ connected and all even degree.
Eulerian Tour is connected so graph is connected.
Tour enters and leaves vertex $v$ on each visit.
Uses two incident edges per visit. Tour uses all incident edges. Therefore $v$ has even degree.

When you enter, you can leave.
For starting node, tour leaves first...then enters at end.
Not The Hotel California.

Connected Components: Quiz.

Is graph above connected? Yes!
How about now? No!
Connected component - maximal set of connected vertices.
Quick Check: $\{10, 7, 5\}$ connected component? No.
Recursive/Inductive Algorithm.

1. Take a walk from arbitrary node \( v \), until you get back to \( v \).  
   **Claim:** Do get back to \( v \)!
   **Proof of Claim:** Even degree. If enter, can leave except for \( v \). □

2. Remove cycle, \( C \), from \( G \).
   Resulting graph may be disconnected. (Removed edges!)
   Let components be \( G_1, \ldots, G_k \).
   Let \( v \) be first vertex of \( C \) that is in \( G_i \).
   Why is there a \( v \) in \( C \)?
   \( G \) was connected \( \iff \) a vertex in \( G_i \) must be incident to a removed edge in \( C \).
   **Claim:** Each vertex in each \( G_i \) has even degree and is connected.
   **Prf:** Tour \( C \) has even incidences to any vertex \( v \).

3. Find tour \( T_c \) of \( G_i \) starting/ending at \( v \). Induction.
4. Splice \( T_c \) into \( C \) where \( v \) first appears in \( C \).

Visits every edge once:
Visits edges in \( C \) exactly once.
By induction for all edges in each \( G_i \).

Equivalence of Definitions.

**Theorem:**
"\( G \) connected and has \(|V| − 1 \) edges" ∋
"\( G \) is connected and has no cycles."

**Lemma:** If \( v \) is 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)
\( \iff \ G − v \) is connected.

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.

To tree or not to tree!

Proof of only if.

**Thm:**
"\( G \) connected and has \(|V| − 1 \) edges" \( \iff \)
"\( G \) is connected and has no cycles."

**Proof of \( \Rightarrow \):** 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 \( \Rightarrow \) every vertex degree \( ≥ 1 \).
Sum of degrees is \( 2|E| = 2(|V| − 1) = 2|V| − 2 \)
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
\( \Rightarrow \) 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"
\( \iff \) "\( G \) 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.
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.
Planar graphs.

A graph that can be drawn in the plane without edge crossings.

Planar? Yes for Triangle.
Four node complete? Yes.
(complete = every edge present. \( K_n \) is \( n \)-vertex complete graph.)
Five node complete or \( K_5 \)? No! Why? Later.

Two to three nodes, bipartite? Yes.
Three to three nodes, complete/bipartite or \( K_3,3 \)? No. Why? Later.