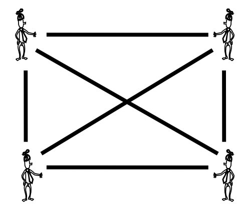
15-251

Great Theoretical Ideas in Computer Science

Anupam Gupta
Danny Sleator

Graphs II

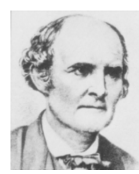
Lecture 19, October 26, 2010

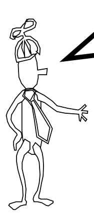


Recap

Cayley's Formula

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





A graph is planar if it can be drawn in the plane without crossing edges

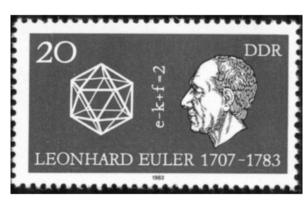
Kuratowski's Theorem

A graph G is planar if and only if $K_{3,3}$ and K_5 are not minors of G.

The 5-color theorem and the 4-color theorem

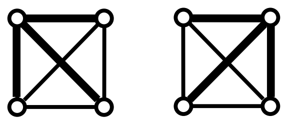
Euler's Formula

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



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

Counting Spanning Trees

How can we count the number of spanning trees in a graph?

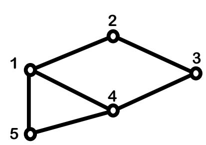
Brute force: try all $\binom{e}{n-1}$ subsets of n-1 edges.

There's a faster way.

Counting Spanning Trees Efficiently

Form the Laplacian of the graph. This is an nxn matrix L, where:

$$L_{i\,j} = \begin{cases} \text{Degree(i)} & \text{if } i=j \\ -1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$



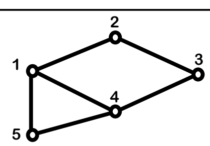
$$L = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Counting Spanning Trees Efficiently

Matrix-Tree Theorem: The number of spanning trees of a graph is the determinant of the Laplacian with one row and column removed.

Proof:

Beyond the scope of this course



$$L = \begin{bmatrix} \cancel{3} & \cancel{-1} & \cancel{0} & \cancel{-1} & \cancel{-1} \\ \cancel{-1} & 2 & -1 & 0 & 0 \\ \cancel{0} & -1 & 2 & -1 & 0 \\ \cancel{-1} & 0 & -1 & 3 & -1 \\ \cancel{-1} & 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\begin{vmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 2 \end{vmatrix} = 11$$

(do on doc cam)

Number of spanning trees = 11 (do on doc cam)

If G is the complete graph on n nodes what happens? We have this (n-1)x(n-1) det:

$$\begin{vmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & & \vdots & \vdots \\ -1 & -1 & \dots & n-1 \end{vmatrix} \xrightarrow{\text{Subtract 1st column from every other column}} \begin{vmatrix} n-1 & -n & \dots & -n \\ -1 & n & \dots & 0 \\ \vdots & & \vdots & \vdots \\ -1 & 0 & \dots & n \end{vmatrix}$$

$$\xrightarrow{\text{Add every one of the rows to the first}} \begin{vmatrix} 1 & 0 & \dots & 0 \\ -1 & n & \dots & 0 \\ \vdots & & \vdots & \vdots \\ -1 & 0 & \dots & n \end{vmatrix} = n^{n-2}$$

Cayley's Formula!

This is a beautiful example from

Spectral Graph Theory

Which involves using linear algebra to study and compute things on graphs

Finding Minimum Spanning Trees

Say that each edge in a graph has a cost.
A very natural question to ask is:

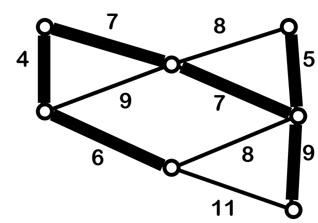
What is the cheapest subset of edges that connect the nodes?

i.e: what is the minimum spanning tree of the graph?

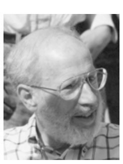
Finding Optimal Trees

Problem: Find a minimum spanning tree, that is, a tree that has a node for every node in the graph, such that the sum of the edge weights is minimum

Minimum Spanning Trees



Kruskal's Algorithm



A simple algorithm for finding a minimum spanning tree

Finding an MST: Kruskal's Algorithm

Create a forest where each node is a separate tree

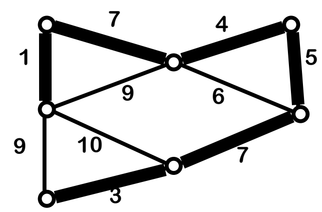
Make a sorted list of edges S

While S is non-empty:

Remove an edge from S with minimal weight

If it connects two different trees, add the edge. Otherwise discard it.

Applying the Algorithm



Proving the Algorithm Works

The algorithm outputs a spanning tree T.

Suppose that it's not minimal. (For simplicity, assume all edge weights in graph are distinct)

Let M be a minimum spanning tree.

Let e be the first edge chosen by the algorithm that is not in M.

If we add e to M, it creates a cycle. Since this cycle isn't fully contained in T, it has an edge f not in T.

N = M+e-f is another spanning tree.

Proving the Algorithm Works

N = M+e-f is another spanning tree.

Claim: e < f, and therefore N < M

Suppose not: e > f

Then f would have been visited before e by the algorithm, but not added, because adding it would have formed a cycle.

But all of these cycle edges are also edges of M, since e was the first edge not in M. This contradicts the assumption M is a tree.

Greed is Good (In this case…)

The greedy algorithm, by adding the least costly edges in each stage, succeeds in finding an MST

But — in math and life — if pushed too far, the greedy approach can lead to bad results.

TSP: Traveling Salesman Problem

Given a number of cities and the costs of traveling from any city to any other city, what is the cheapest round-trip route that visits each city at least once and then returns to the starting city?

TSP from Trees

We can use an MST to derive a TSP tour that is no more expensive than twice the optimal tour.

Idea: walk "around" the MST and take shortcuts if a node has already been visited.

We assume that all pairs of nodes are connected, and edge weights satisfy the triangle inequality $d(x,y) \leq d(x,z) + d(z,y)$

Tours from Trees

Shortcuts only decrease the cost, so
Cost(Greedy Tour) $\leq$ 2 Cost(MST)
$\leq$ 2 Cost(Optimal Tour)

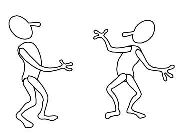
This is a 2-competitive algorithm

Bipartite Graph

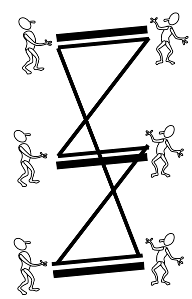
A graph is bipartite if the nodes can be partitioned into two sets V_1 and V_2 such that all edges go only between V_1 and V_2 (no edges go from V_1 to V_1 or from V_2 to V_2)

Dancing Partners

A group of 100 boys and girls attend a dance. Every boy knows 5 girls, and every girl knows 5 boys. Can they be matched into dance partners so that each pair knows each other?



Dancing Partners



Perfect Matchings

A matching is a set of edges, no two of which share a vertex. The matching is perfect if it includes every vertex.

Regular Bipartite Matching Theorem: If every node in a bipartite graph has the same degree $d \geq 1$, then the graph has a perfect matching.

Note: if degrees are the same then $|A| = |B|$, where A is the set of nodes "on the left" and B is the set of nodes "on the right"

A Matter of Degree

Claim: If degrees are the same then $|A| = |B|$

Proof:

If there are m boys, there are md edges

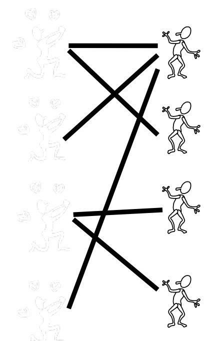
If there are n girls, there are nd edges

The Regular Bipartite Matching Theorem follows from a stronger theorem, which we now come to. (Remind me to return to the proof of the RBMT later.)

The Marriage Theorem

Theorem: A bipartite graph has a perfect matching if and only if $|A| = |B| = n$ and for all $k \in [1,n]$: for any subset of k nodes of A there are at least k nodes of B that are connected to at least one of them.

The Marriage Theorem



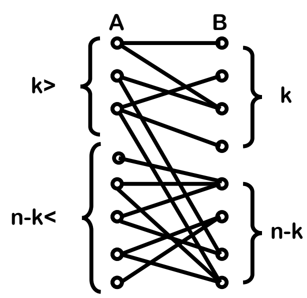
For any subset of (say) k nodes of A there are at least k nodes of B that are connected to at least one of them

The condition fails for this graph

The Feeling is Mutual

The condition of the theorem still holds if we swap the roles of A and B: If we pick any k nodes in B, they are connected to at least k nodes in A

Proof by Contradiction:



Proof of Marriage Theorem

Call a bipartite graph "matchable" if it has the same number of nodes on left and right, and any k nodes on the left are connected to at least k on the right

Strategy: Break up the graph into two matchable parts, and recursively partition each of these into two matchable parts, etc., until each part has only two nodes

Proof of Marriage Theorem

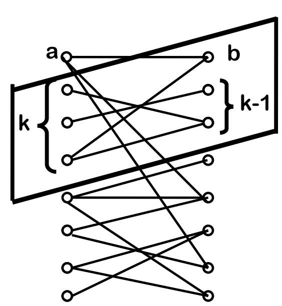
Select two nodes $a \in A$ and $b \in B$ connected by an edge

Idea: Take $G_1 = (a,b)$ and G_2 = everything else

If G_2 is matchable, we're done. So let's assume that G_2 is not matchable.

If G_2 is not matchable then there is a set of k nodes in G_2 that has only k-1 neighbors.

Proof of Marriage Theorem



The only way this could fail is if one of the missing nodes is b

Add this in to form G_1, and take G_2 to be everything else.

This is a matchable partition!*

(*Done in lecture on the document cam.)

Example

Suppose that a standard deck of cards is dealt into 13 piles of 4 cards each

Then it is possible to select a card from each pile so that the 13 chosen cards contain exactly one card of each rank

Proof: Form a bipartite graph as follows: Start with 52 cards on the left and the same 52 cards on the right, connected by 52 edges.

Now group the cards on the left into 13 sets according to the given piles. Group the cards on the right into 13 groups according to rank. Let the edges be inherited from the original ones.

This bipartite graph is matchable -- k groups on the left have to connect to 4k cards on the right, thus they connect to at least k groups on the right.

And thus has a perfect matching.

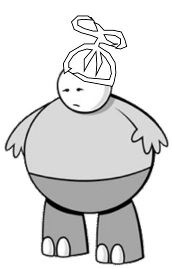
Generalized Marriage: Hall's Theorem

Let $S = \{S_1, S_2, \ldots S_n\}$ be a set of finite subsets that satisfies: For any subset T of $\{1,2,\ldots,n\}$ let U = the union of S_t for t in T, we have: $|U| \geq |T|$. I.E. any k subsets contain at least k elements

Then we can choose an element x_i from each S_i so that $\{x_1, x_2, \ldots\}$ are all distinct

The proof of Hall's Theorem is slightly more complicated (but not much) than our proof of the Marriage Theorem.

You can find the proof on Wikipedia, or on pages 218 and 219 of Mathematical Thinking by D'Angelo and West.



Here's What You Need to Know…

Minimum Spanning Tree
- Definition

Counting spanning trees
- Matrix-Tree Theorem

Kruskal's Algorithm
- Definition
- Proof of Correctness

Traveling Salesman Problem
- Definition
- Using MST to get an approximate solution

The Marriage Theorem