15-251

Some ~~Great~~ Theoretical Ideas ~~in~~ for Computer Science

Graphs II

Lecture 21 (April 1, 2008)

Recap

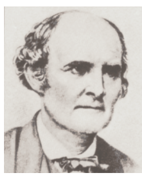
Theorem: Let G be a graph with n nodes and e edges

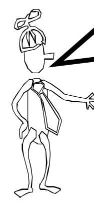
The following are equivalent:

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 a line, the resulting graph has exactly one cycle

Cayley's Formula

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

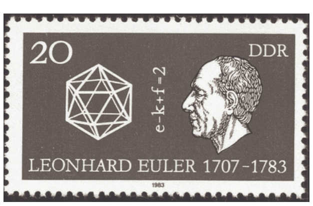




A graph is planar if it can be drawn in the plane without crossing edges

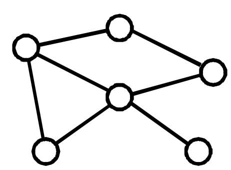
Euler's Formula

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



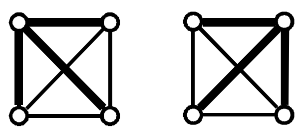
Graph Coloring

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



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

Finding Optimal Trees

Trees have many nice properties (uniqueness of paths, no cycles, etc.)

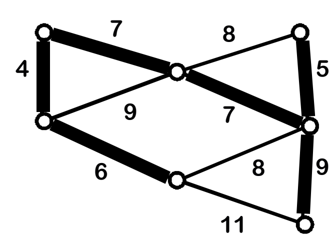
We may want to compute the "best" tree approximation to a graph

If all we care about is communication, then a tree may be enough. We want a tree with smallest communication link costs

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

Tree Approximations



Finding an MST: Kruskal's Algorithm

Create a forest where each node is a separate tree

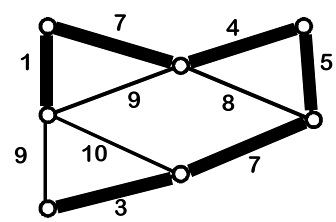
Make a sorted list of edges S

While S is non-empty:

Remove an edge with minimal weight

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

Applying the Algorithm



Analyzing the Algorithm

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.

Analyzing the Algorithm

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 exactly 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 $d(x,y) \le d(x,z) + d(z,y)$

Tours from Trees

Shortcuts only decrease the cost, so

$$\begin{aligned} \text{Cost(Greedy Tour)} &\le 2\ \text{Cost(MST)} \\ &\le 2\ \text{Cost(Optimal Tour)} \end{aligned}$$

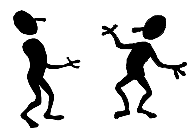
This is a 2-competitive algorithm

Bipartite Graph

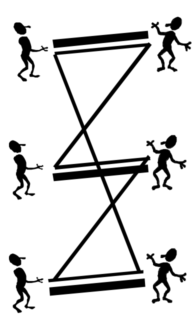
A graph is bipartite if the nodes can be partitioned into two sets A and B such that all edges go only between A and B (no edges go from A to A or from B to B)

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

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

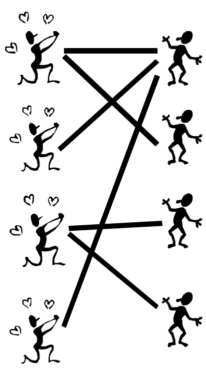
The Marriage Theorem

Theorem: A bipartite graph has a perfect matching if and only if |A| = |B| = n and for all k ∈ [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.

↑

Call a graph with this property NoShrinkFrom(A)

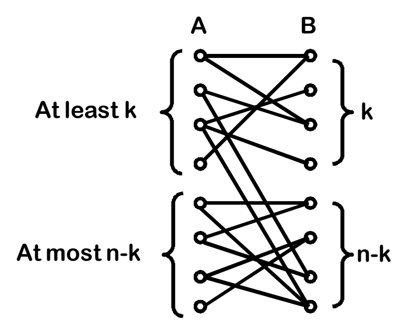
NoShrinkFrom(A)?



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



If it's NoSrhinkFrom(A), it's also NoShrinkFrom(B)

Proof of Marriage Theorem

Strategy: Break up the graph into two parts that are NoShrinkFrom(A) and recursively partition each of these into two NoSrhinkFrom(A) parts, etc., until each part has only two nodes

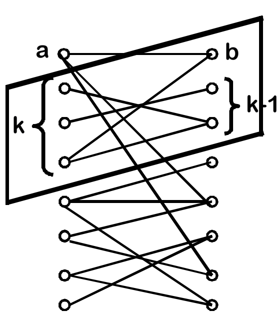
Proof of Marriage Theorem

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

Problem: G_2 need not be NoShrinkFrom(A). There could be a set of k nodes 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!

Here's What You Need to Know…

Minimum Spanning Tree
- Definition

Kruskal's Algorithm
- Definition
- Proof of Correctness

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

The Marriage Theorem