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
Planar Graphs

http://www.planarity.net

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.

Implementing Graphs

Adjacency Matrix

Suppose we have a graph $G$ with $n$ vertices. The adjacency matrix is the $n \times n$ matrix $A=[a_{ij}]$ with:

- $a_{ij} = 1$ if $(i,j)$ is an edge
- $a_{ij} = 0$ if $(i,j)$ is not an edge

Good for dense graphs!
Counting Paths

The number of paths of length $k$ from node $i$ to node $j$ is the entry in position $(i,j)$ in the matrix $A^k$.

$$A^2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 3 & 2 & 2 & 2 \\ 2 & 3 & 2 & 2 \\ 2 & 2 & 3 & 2 \\ 2 & 2 & 2 & 3 \end{pmatrix}$$

Adjacency List

Suppose we have a graph $G$ with $n$ vertices. The adjacency list is the list that contains all the nodes that each node is adjacent to.

Good for sparse graphs!

Example

Graphical Muzak

“Can you hear the shape of a graph?”

http://www.math.ucsd.edu/~fan/hear/

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.

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.

Finding an MST: Kruskal’s Algorithm

A simple algorithm for finding a minimum spanning tree.

Let M be a minimum spanning tree.

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

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 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

$\text{Cost(Greedy Tour)} \leq 2 \text{Cost(MST)} \leq 2 \text{Cost(Optimal Tour)}$

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?

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”

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.

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 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:

\[
\begin{align*}
A &< k \\
B &> n-k
\end{align*}
\]

**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.

**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, and thus has a perfect matching. (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.)
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 \( U = \text{the union of } S_t \text{ for } t \in T, \) 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…

- Adjacency matrix
- 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