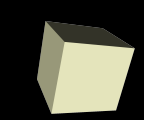
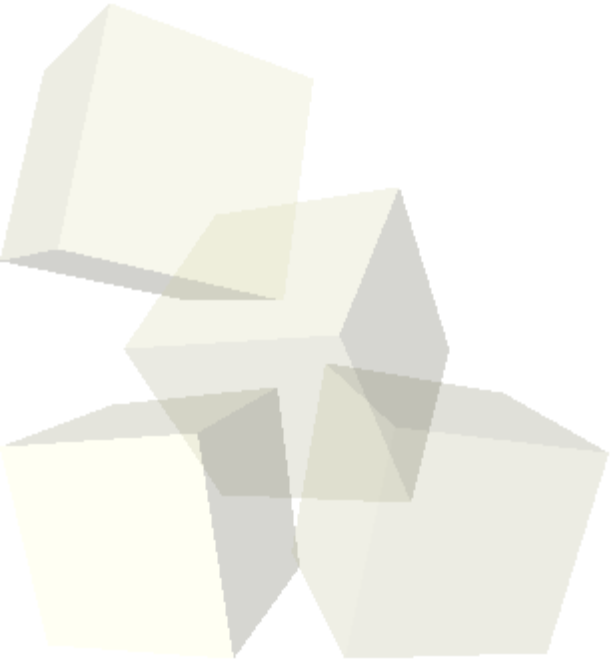


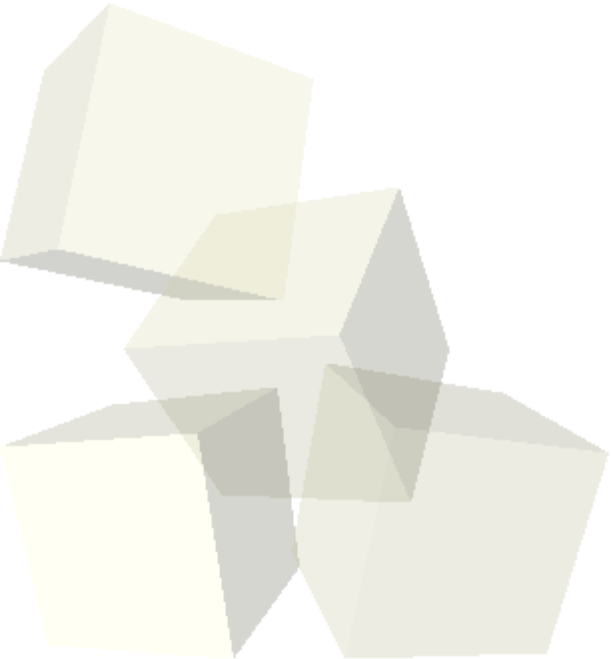
Graphs

3-7-2006



Opening Discussion

- What did we talk about last class?
- Tell me everything you remember about graphs.



Representations of Graphs

- We can represent a graph as either an adjacency matrix or adjacency lists. The latter is much better if the number of edges is significantly less than $O(n^2)$.
- We can traverse a graph in either breadth first or depth first manner. The trick is that you have to mark things to prevent from having infinite loops. CLR use a 3 color scheme for marking nodes.
- Since graphs don't have to be connected these traversals produce forests. Typically BFS is done on connected graphs. DFS is more useful when things aren't connected.

More on DFS

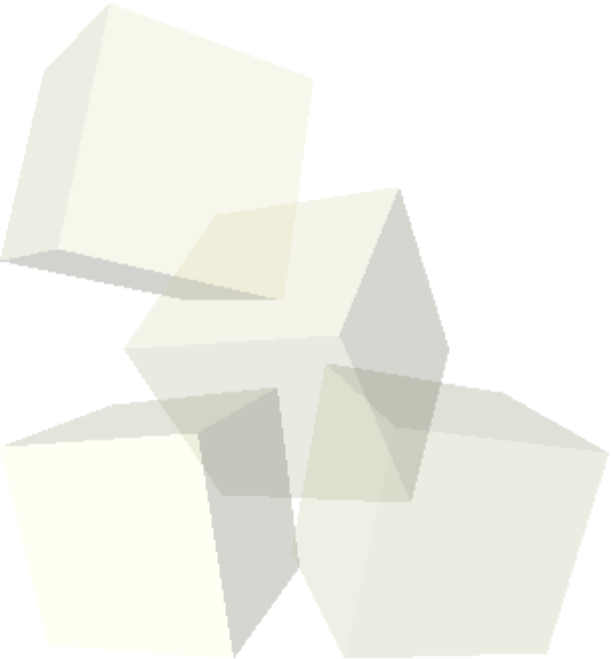
- In the case of DFS we can also keep a time stamp on elements that tells when we find them and when we are done processing them. These are called the discovery time and the finishing time.
- As our first algorithm on graphs we will discuss a topological sort. This can be done simply by doing a DFS and then sorting the nodes by inverse finishing time. Alternately you can place items at the head of a list each time they are finished. That makes sure that each element comes before every element that it can link to.

Strongly Connected Components

- A strongly connected component of a DG is a maximal set such that for any two elements of the set there is a path from the first to the second and from the second back to the first.
- First call DFS on the graph to get the finishing times.
- The find the transpose of the graph (reverse all the edges) and call DFS on the transpose but now visit vertices in order of decreasing finishing time from the first DFS.
- Each tree that you get from the forest made by that DFS is a strongly connected component.

Minimum Spanning Trees

- This is a nice problem for applying the types of algorithm design methods that we have talked about previously. In particular, this problem can be solved in a greedy manner.



Reminders

- Your next assignment will be on strongly connected components. I will get a data set out there as quickly as I can. I might also throw in something about a max clique.