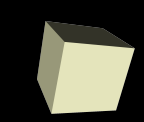
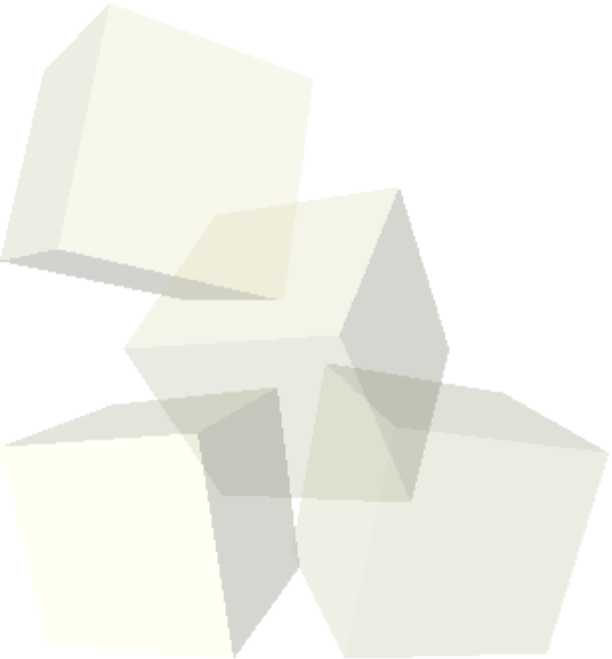


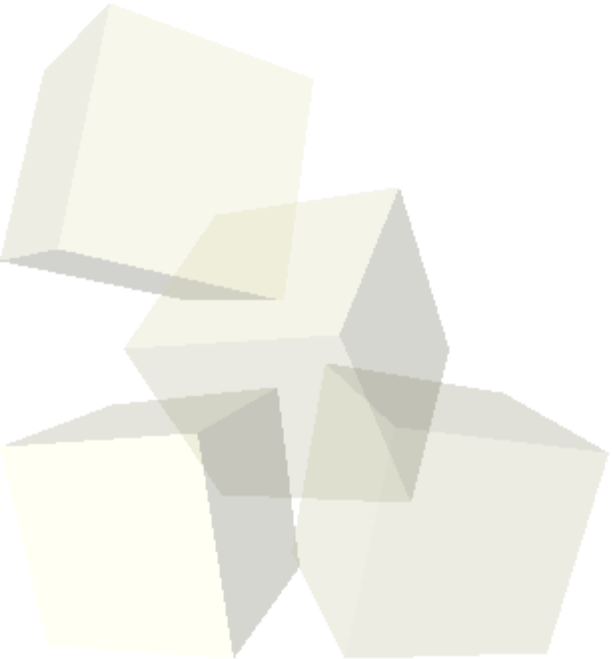
Dynamic Programming

11-16-2005



Opening Discussion

- Do you have any questions about the quiz?
- What did we talk about last class?



Longest Common Subsequence

- Last time we built a recursive function to solve the longest common subsequence problem. While the function works, it has the downfall of scaling exponentially with string size. Because of this, our method isn't really practical to use.
- Today we will talk about an alternate method of solving this problem and other optimization problems that will make them tractable.

Dynamic Programming (DP)

- The problem with our recursive solution is that it winds up solving the same problem over and over again. This is fairly common in optimization problems.
- When things do this they exhibit optimal substructure. That implies that the optimal solution for a problem is built from optimal solutions to smaller problems. Only when this is true you can use dynamic programming.
- The idea of dynamic programming is to fill in solutions starting with very small problems and build up instead of recursing down.

Applying DP

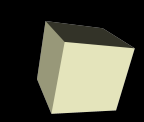
- To build a DP solution we must first find the recursive solution, then devise a way to fill in solutions from the bottom-up.
- All recursive solutions have termination conditions. In a DP solution that is where you start. Often you fill in an array with these values. Then you run through the rest of the array filling in values. The trick is that you compute each new value by looking up solutions in the parts of the array you are already filled in.
- Using our recurance relation from last class we can do this for LCS.

Steps for DP

- To apply DP we first need to develop the recurance relationship. Figuring out the best arguments for this can be a challenge.
- Next we write a solution that fills in an array with the answers looking into the array for earlier solutions.
- If you need to you can also reconstruct the optimal solution by walking back down the array and taking the optimal path.

Memoization

- An alternative to DP that can be almost as fast is memoization. In this approach we write the recursive solution, but pass in an extra argument that stores solutions we have already found.
- When the function runs it checks the stored values before doing a recursive call so it won't solve subproblems that it has solved previously.
- For some problems this method is a lot easier to think about. Memoization can also be used for problems that don't have optimal substructure so DP can't be applied.



Reminders

- I'll be putting up a description of your next assignment soon. I also need to talk to Dr. Livingstone about your project for the end of the semester.