Dynamic Programming

4-10-2002

Opening Discussion

- Do you have any questions about the quiz?
- What did we talk about last class?
- Do you have any questions about the assignments? What elements of the assignments are causing you problems at this point?

Minute Essay Comments

- Multithreading
  - Modern operating systems have the ability to act like they are running many things at once. When they share the same memory and are part of the same program they are called threads. Do man pthread for more in C/C++.
- Graphics libraries
  - OpenGL for 3D-graphics. Mesa is free.
- Exciting projects/teaching Java

Repeating Work with Recursion

- When we first discussed recursion we talked about how it wasn't ideal for all problems. The Fibonacci numbers were a key example of this.
- The problem is that it doesn't remember what work it has done and can often repeat large amounts of work computing the same value over and over again.

Dynamic Programming

- Dynamic programming is a way of constructing solutions to these problems that doesn't repeat the work. It can be quite efficient. It is also best when we get rid of the recursion completely.
- Like D&C we are interested in problems that can be broken into pieces. In this case though we want common subproblems that we only solve once.

Steps in DP (Form CLR)

- Characterize the structure of an optimal solution.
- Recursively define the structure of an optimal solution.
- Computer the value of an optimal solution in a bottom-up fashion.
- Construct an optimal solution from computed information (optional).

Principle of Optimality

- For dynamic programming to work the problem has to follow the principle of optimality. This implies that "no matter what the first decision, the remaining decisions must be optimal with respect to the state that results from the first decision".
- This is much nicer than greedy algorithms where proving they are actually optimal can be quite painful.

Memoization

- The real key to DP is that we want to keep track of what we have already solved. This is the real difference between it and D&C. With D&C we always recurse to get solutions. With DP we look for earlier solutions if they are available.

Change Maker Problem

- Your book presents a change maker problem as a good DP problem. The objective is to make change with as few coins as possible. In some cases this can be done with greedy algorithms but they give a nice example where it can't.
- The recursive solution they present is to test all possibilities where we break it into 2 pieces using a divide and conquer type method.
- You could also loop through all denominations and recurse assuming each one was used at that point.

Change Maker Continued

- That method is very inefficient however, because it recalculates many values. Instead, we can work up from 1 cent storing off the minimum number of coins it takes to get that value.
- At each value we loop through all denominations and use the one where we can get to that value with the fewest coins using previously calculated values.

Minute Essay

- Trace through the DP solution to the change maker using denominations of 1, 4, and 5 cents for making change on 13 cents. Just write the number of coins used to make each amount of change between 1 and 13 cents.