Recursion

3/22/2007

Opening Discussion

- Do you have any questions about the assignment?
- Do you have any questions about the reading?
- What types of problems are typically easier to solve with recursion than with loops?

Mazes

- Let's continue adding in the DrawMaze class and write some different algorithms that work with it.
- The normal recursive algorithms try all paths. In some mazes this can take a very long time. Let's look at why that is and how we might be able to fix it.

Divide and Conquer

- A standard approach to solving problems with recursion is divide and conquer. In this approach we break the problem up into two or more smaller problems, solve the smaller problems, them make an answer for the larger problem with the solutions of the smaller pieces.
- Of course, the key to the recursion is that the smaller problems are solved by breaking them up in the same way.
- You can use this approach for simple array functions, but there isn't much benefit there.
- For some problems though, divide and conquer can provide much simpler and/or efficient solutions to problems.

Formula Parsing

- One of my favorite divide and conquer algorithms is formula parsing.
- Given a mathematical formula, we can express the value of the whole thing as one operator working on two parts, where the operator is the lowest precedence operator.
- Viewing it this way is like converting the formula to a tree.
- Let's write code that evaluates a formula with a single variable, x, using divide and conquer. What could we put into our drawing program that would work with this?

Sorts

- The most commonly used divide and conquer algorithms are sorts. Mergesort and quicksort or both divide and conquer. They differ in how they do the division and when they do the majority of their work.
- Mergsort does virtually no work going down the call stack. It simply breaks the input array in half each time until it gets to single elements. The work happens coming back up as the two sorted pieces are merged.
- Quicksort does work on the way down. At each step a “pivot” element is picked and moved to the right spot. Recursion is done on both sides of the pivot.

Minute Essay

- Recursion and induction are very closely linked. Describe the similarity.
- Or, if you aren't in discrete, our formula parser is nice, but it isn't exactly efficient if we evaluate the same string many times with different values of x. How could we improve that?
- Mock programming competition a week from today. Participating can get you extra credit on the final and if you participate you get an extension on the assignment until the following Tuesday.