

# **Recursion 2**

**4-6-2004**

# Opening Discussion

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

# Two Maze Problems

- Counting the number of paths. This was your minute essay from last time. It is like a floodFill, but it returns the number of paths. That number is 1 for an exit, 0 for a wall, and the sum of all the neighbors for an open spot. Remember bread crumbs.
- Shortest path. This is a slightly harder problem which can be done efficiently or not. Not is almost like the one above.

# Bin Packing

- Recursion is also extremely handy for solving problems where you want to “test all possible options”. The bin packing problem I gave you as extra credit for the first test is an example of this.
- We need to test all possible ways of putting the items in the bins. We can do this with a function that tries all the items in the current bin and recurses with each attempt.

# Divide and Conquer

- Another common usage of recursive functions is in divide and conquer algorithms. In these, we repeatedly break a problem down into smaller pieces until we get to a point where we can solve it easily. We then combine the partial solutions to find a full solution.
- Some extremely simple examples could be finding the sum of the elements of an array or the min of them.

# Equation Parsing

- Another fun application of divide and conquer is string parsing for things like equations. Here we break the problem up around the lowest priority operator and recursively parse smaller sections of the formula, then deal with them as we move back up the stack.
- With just a little more work, this type of method can be turned into a very powerful tool.

# Faster Sorts

- Divide and Conquer can also be used to come up with faster sorting algorithms.
  - Mergesort - divide directly in two and recurse, then merge the two sorted halves. Does the work on the way back up.
  - Quicksort - pick a pivot, and move everything lower than the pivot to one side and everything higher to the other then recurse on the two sides. Does the work on the way down.

# Object Oriented “Recursion”

- In an object-oriented language we can find another type of recursive function. In this case we are looking at a method that calls the same method, but on a different object. The draw function in our semester example is like this.
- Because the functions can be virtual, it isn't always the same method that we are calling.



# Minute Essay

- Trace a quicksort that uses the first element as a pivot for the array  $\{5,8,2,4,7\}$ .
- Remember that the design for assignment #6 is due on Thursday.