CS206 Introduction to Data Structures

Lab 7

Tracing Recursion

Friday Oct 30

In this lab you will write no code. Rather you will run some recursive code that I wrote for solving a maze. The code is available from

/home/gtowell/Public/206/lab07/*.java

From this directory pull of the java code as well as the three files of the form maze?.txt into a VSC folder. (All of these files are also on the class website.)

To run the maze solver, execute the main method of Maze.java, then when asked for a maze file respond with one of: maze1.txt, maze2.txt or maze3.txt.

For maze1.txt only, write a full listing of every method call (Something like the stepped through Fibonacci calculation on Tuesday). Specifically, show for each method call:

- the arguments to a recursive method,
- the Java call stack
- the current state of the maze. (What is the character in every position of the maze holding array?)
- the return value from the recursive method

It is possible to do this using only pen and paper. However, it is much more easily done using the VSC debugger. (See my description of using the debugger at XXXXXXX)

For maze2.txt and maze3.txt, have the program solve them. Be sure you understand exactly why the program explored the parts of the maze that it did explore. Then answer the following question.

How could you adjust Maze.java so that more of these mazes are explored before reaching the end? Less? (Hint, if you change more than 5 characters for these answers you are working too hard).

Finally, write your own maze file. Have the program solve it.

What to hand in:

1. The stack trace for the solution to the first maze.

2. The adjustment(s) you would make to the program to have it search more on mazes 2 and 3 and less on mazes 2 and 3.

3. Your own maze.