CS61B, Fall 2004 **HW #9** **P. N. Hilfinger**

Due: Wed., 24 November 2004 at midnight

Create a directory to hold your answers to this homework set. Copy the files from `$master/hw/hw9` into this directory. Put non-program answers into a file `hw9`. Use the command `submit hw9` to submit your solutions to the problems below.

1. Write a routine *equalLabels(T1,T2)* that tells whether trees `T1` and `T2` have the same sequence of labels when traversed in preorder. Use *coroutines* implemented with threads. That is, create two threads, one that traverses `T1` in preorder and one that traverses `T2`, each transmitting the labels it encounters back to the main program, which compares the two sequences of labels. See the file `~cs61b/hw/hw9/Labels.java`.

2. What is the maximum height of an order-5 B-tree containing N nodes? What is the minimum height? What sequences of keys, when inserted, give the maximum height (that is, give a general characterization of such sequences). What sequences of keys give the minimum height?

3. Write a program to find your way out of a maze. The maze will consist of an $M \times N$ array of squares. An $M \times (N-1)$ array, `V`, of booleans will represent vertical walls: that is, `V[`r`][`c`]` will be true iff there is a wall between the squares at (r, c) and $(r, c+1)$. Likewise, an $(M-1) \times N$ array, `H`, will represent horizontal walls: `H[`r`][`c`]` will be true iff there is a wall between the squares at (r, c) and $(r+1, c)$. Your program is to print out a non-repeating path (a sequence of squares) from a designated starting square to a designated exit square. See the file `~cs61b/hw/hw9/Maze.java`.