CS61B, Fall 2004 **HW #6** **P. N. Hilfinger**

Homework Exercises. (Due: Wed., 27 October 2004 at midnight) Create a directory to hold your answers to this homework set. Copy the files from `$master/hw/hw6` into this directory. Put non-program answers into a file `hw6`. Use the command `submit hw6` to submit your solutions to the problems below.

1. Exercise 4.2 from the *Data Structures (Into Java)* reader. See `~cs61b/hw/hw6/ConcatList.java`.

2. Using the linked representation of binary trees in `~cs61b/hw/hw6/Tree.java`, fill in the following complementary functions (see `~cs61b/hw/hw6/FlattenTree.java`).

```
/** The sequence of all labels of T, in inorder. */
static String[] flatten (Tree T) {
   // FILL IN
}

/** A binary tree of minimum depth whose labels, in inorder, are
 *  L[0], L[1], .... */
static Tree treeify (String[] L) {
   // FILL IN
}
```

3. A *threaded tree* has an extra link in each node such that all of the nodes in the tree are linked into a list using this link. For example, you might use this link to store a pointer to the next node in preorder after this one, so that a preorder traversal of the tree is just a traversal of this list (see the example below; the threads are dashed.) Implement a function `preorderThread` that computes these links in a general tree. Use the template in file `~cs61b/hw/hw6/ThreadedTree.java`.