Homework 2, Basic Algorithms, Spring 2014.
Due date: Tuesday, February 11.

Problems 2.1, 2.10, 2.19, 2.25c, 2.26a, 2.11 from the text.
Honors Section. Problem 1.20 and 1.18c from the text in addition.

Comments on the problems

For all these problems, your procedures should have the form SuitableName(v, other variables as needed); the initial call, to process a tree T, will be to SuitableName(T, suitable initial values). You will need to replace “suitable ...” with something appropriate.

2.10. Interpret this as follows: if v is a leaf then v.1dis = v.2dis = 0; if v has one child then v.2dis = 0.

2.19. For parts a and b, if your answer is “yes” try to provide brief justifications, and if it is “no” give an example demonstrating this.

When processing S, assume that the two pointer fields for a node v are named v.child and v.sibling.

2.25c. Return nil if k is not in the tree.

2.26. Hint. First write a recursive procedure to solve the following BackThread(v, pred) problem. For each node v it creates a pointer v.bthread which points to v’s predecessor in the binary search tree. For the leftmost node u in the tree, u.bthread will be set to nil.

Try to connect this to the solution to problem 2.4b.

Now modify one line of code (twice) to essentially solve the thread problem. You will also need to correctly handle the leftmost and rightmost nodes; this will need some extra code, and possibly a driver routine.

2.11. Give a formula expressing w.up, where w is a child of v, in terms of v.up and 1dis and 2dis values for u and w. Now write a recursive procedure to calculate this formula for all the nodes in the tree. You will want a driver procedure Driver(T) and a recursive procedure UP(v). The Driver procedure will make two passes over the tree, the first one being to calculate the solution to problem 2.10 (there is no need to copy over your solution to this problem).