

Burt Rosenberg

Problem Set 4

OUT: 28 SEPTEMBER, 1995

DUE: 10 OCTOBER, 1995

Announcements

The machine ocala.cs.miami.edu is running, and you can log in using your id and password. Your home directory will appear thanks to the magic of NFS — the Network File System, which allows sharing of file systems across networks. There is a bit of trouble with getting the terminal setup correctly. Most of the equipment at school is DEC, and DEC runs best with DEC. Ocala is an Silicon Graphics machine, a server based on the Indy architecture. It's really cool.

I have modified your .login files to sense the current architecture and vector out to .login.pmax (for DEC) or .login.indy (for SGI). I hope this works. Furthermore, to insure that mail sent to you will end up on cs, I have created a .forward file in your home directory, the contents of this file is the single line:

```
myname@cs.cs.miami.edu
```

(but your actual username appears instead of `myname`). If you would like to receive all mail on ocala, so you will never need to log into cs any more, change the contents of this file to:

```
myname@ocala.cs.miami.edu
```

Problems are to be expected. Let me know by email. For the remainder of this course, you may either use Ocala or continue to work on Cs.

Programming Assignment

Write a program to insert, print and delete items in a binary tree. The supplied template was a working program before I took out code and left comments in the empty spaces. You fill in the code. *Write ASSERTIONS where appropriate.* The correct use of assertions will be part of your grade.

Deletion is the most difficult, and you may not care for the manner in which I broke the problem down. Feel free to change it, if it bothers you greatly. My code breaks the problem down into NULL and non-NULL pointers. In the condition that I have an actual node (not NULL) to look at, I first count the number of children. The three cases: no children (easy), one child (not too bad), two children (ouch!), are handled by separate code blocks.

The program is to work as follows - there is a working copy in the pub directory for you to try. If the input is 0, the program exits. If the input is a positive integer, the integer is inserted into the tree. If the integer is already in the tree, the tree is unchanged. If the input is a negative integer, the sign is reversed, making the integer positive, and that integer is deleted from the tree. If the integer does not appear in the tree, the tree is unchanged.