

## NEW DRIVER PROGRAM FOR 340 ASSIGNMENT 9 - BINARY SEARCH TREE

This is a new driver program that gives more information. The basic idea is the same, because I don't want to make them radically change anything while the plane is in flight, but they will get more won't be left in the dark as to what is happening when something is wrong.

I made sure to print out what was happening at each step, and I changed the output of the tree contents so that it shows a \* next to any leaf nodes, which should help clarify what's going. This does make it a bit more verbose, but it also makes the files larger, so hopefully we don't run into too many issues with student's quotas.

When I wrote this, I was doing my testing lazily, so I used `a.out`, and `test.sh` reflects that. If you replace the old `bst.cc` with `driver2.cc` and use the Makefile to compile to an executable named `bst`, you will want to change line 3 of `test.sh` from

```
PROG=./a.out
```

to

```
PROG=./bst
```

(or whatever the name of the executable ends up being)

The test file will automatically run several test cases and dump the output to predefined filenames. I've included the reference output in this tarball, which you can diff with the students' output.

### NOTE FOR STUDENTS

If you download this, do not commit the new `driver2.cc` to your repository. The TA will put it into place when grading. Also, do not commit any changes to `bst.cc`. You can, however, use it to test your version of `BinarySearchTree` locally and get a little more feedback as to what is happening, and the TAs will use the new version to test things.

### NOTE FOR TAs

TAs: Feel free to distribute this tarball to students that are having trouble with the original version. I may end up just posting it. There is nothing in this tarball that they can't see. You can add extra tests to `test.sh` with different parameters for the actual grading. I gave you working implementations of `binarysearchtree.h` and `btree.h` in the grading tarball (which obviously should not be shared with students) and you can use those to generate new, known good, reference outputs for any test cases you create.

-Lehuta