

Lab Exercises. (Due: Tues., 6 November 2001 in lab)

Copy the directory `$master/hw/lab8` to a directory of your own, using commands such as

```
cp -r $master/hw/lab8 .
```

In this directory, you should find a file called `README.lab8`, with directions about what to do during the scheduled lab. We strongly suggest that you read over this file *before* going to your lab.

We intend that you finish the lab exercises *in lab* and have your TA check them off. You can have any TA in any lab section check off your lab.

Homework Exercises. (Due: Tues., 6 November 2001 at midnight) You have more than the usual time for this homework, due to the project. You will find it useful to at least figure out the answers to these questions before the test on 8 November, so we are assigning them now. Create a directory to hold your answers to this homework set. Copy the files from `$master/hw/hw8` into this directory. Put non-program answers into a file `hw8`. Use the command `submit hw8` to submit your solutions to the problems below.

1. Suppose that we have an array, D , of N records. Without modifying this array, I would like to compute an N -element array, P , containing a permutation of the integers 0 to $N - 1$ such that the sequence $D[P[0]], D[P[1]], \dots, D[P[N - 1]]$ is sorted *stably*. Give a general method that works with any sorting algorithm (stable or not) and doesn't require any additional storage (other than that normally used by the sorting algorithm).
2. I am given a list of ranges of numbers, $[x_i, x'_i]$, each with $0 \leq x_i < x'_i \leq 1$. I want to know all the ranges of values between 0 and 1 that are *not* covered by one of these ranges of numbers. So, if the only input is $[0.25, 0.5]$, then the output would be $[0.0, 0.25]$ and $[0.5, 1.0]$ (never mind the end points). See the template `~cs61b/hw/hw8/Ranges.java`.
3. Fill in the following to agree with its comments (see `Rotate.java`) in the `hw8` directory):

```
/** Return a modified version of T containing the same nodes
 * with the same inorder traversal, but with the node containing
 * label X at the root. Does not create any new Tree nodes. */
static Tree rotateUp (Tree T, Object X) {
    // FILL IN
}
```

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