

## CSE 2320-001 Lab Assignment 3

Due April 17, 2012

### Goals:

1. Understanding of binary search trees, especially ranking.
2. Experience with recursion.

### Requirements:

1. Extend the C code `topdownRB.c` from the directory <http://ranger.uta.edu/~weems/NOTES2320/LAB001SPR12/> to implement batched queries. The driver `lab3spr12.driver.c` will handle most of the needed input and output. The header file `topdownRB.h` includes prototypes for the following functions that you must write:

1. Batched version of search for the num keys in the *ordered* array `in[ ]`.
2. Batched version of finding the ranks for the num keys in the *ordered* array `in[ ]`.
3. Batched version of finding the keys for the num ranks in the *ordered* array `in[ ]`.

Unlike the simple versions of these functions, these should process all inputs *at the same time* to decrease the number of comparisons being performed. In particular, *binary searches* may be used to split the input array based on the available stored key or subtree size information. For each of the three functions, your program should print the number of comparisons used by the batched and unbatched versions. For each of the three functions, the output is to be stored in the array `out[ ]` which will be printed by the driver.

2. Email your program to `adnan.khan@mavs.uta.edu` by 1:45 p.m. on April 17, 2012. The subject should be your name as recorded by the University and you should `cc:` yourself to verify that you sent the message correctly. You only need to send your modified version of `topdownRB.c`.

### Getting Started:

1. The provided input ranks will be in the range  $0 \dots n-1$ , where  $n$  is the number of keys in the red-black tree. The provided input keys *are not* necessarily in the tree, so observe what the driver is expecting for these.
2. When splitting an array, use *pointer arithmetic* rather than copying to a new array.
3. Do not modify the elements of the `in[ ]` array during the processing for any of the three functions.