Written #2: Sorting and Trees
Due: see Canvas

For this assignment, you'll submit both a PDF and a Java file (Cuckoo.java).

Problem 1. Quicksort Space. A criticism of Quicksort is that in the worst case, it uses $O(N)$ extra space (space beyond that used for the input). This is because the recursive calls can have depth up to N. See share/writ2/Quick.java for a modified version of Quicksort: in this version, the recursive `sort` method uses recursion on the left subproblem, but it uses iteration (no recursion!) on the right subproblem.

1(a). Argue that this modified version of Quicksort can still use $O(N)$ extra space, in the worst case.

1(b). Write out a revised version of the recursive `sort` method (either in Java, or just in pseudocode) so that it uses recursion on the `smaller` subproblem (left or right), and then it uses iteration on the other subproblem.

1(c). Argue that your revised version of Quicksort needs only $O(\lg N)$ extra space.

Problem 2. Cuckoo UF Experiment. Suppose we have N isolated vertices. We add random edges, until some component of the graph has more edges than vertices. Let M be the number of edges added. We are interested in estimating $E[M]$, the expected value of M. We are also interested in the ratio $E[M]/N$. (This is important to "Cuckoo hashing", a later course topic.)

Copy and edit share/writ2/Cuckoo.java, as directed in its comments. You'll be modifying a copy of share/writ2/ErdosRenyi.java, which does a similar experiment (adding edges until the graph is connected). In particular, you will need to modify its `UF` subclass.

Using your program, estimate the ratio $E[M]/N$ for $N = 10^1$, $N = 10^3$, and $N = 10^5$. For each estimate, report an average of at least $T = 1000$ experiments. (You should see the ratios approaching 1/2.)

Submit a copy of your `Cuckoo.java` program with this assignment. It should exactly reproduce the "test output" described in the comments.

Problem 3. LLRB Tree Insertion. Consider the left-leaning-red-black BST (in our textbook). Find a specific example where a single insertion requires at least four rotations, and the inserted key ends up at the root of the tree. Draw the initial tree (just after the inserted key has been attached as a leaf with a red edge) and also draw the final tree (after all the rotations and color flips are done). Be sure to indicate which edges are red (you may use "thick" edges to represent red).

Problem 4. Red-Black Tree. Look up the "Red-Black Tree" on wikipedia, and compare it to the LLRB tree described in our textbook. In particular:

4(a). How does its definition differ from the LLRB tree?

4(b). Give an example of a red-black tree that is not an LLRB tree.

4(c). According to wikipedia, what is the maximum number of rotations needed for a single red-black tree insertion?