Homework #2: Trees and Hashing  
Due Date: program 5pm Friday 2/11, written 5pm Wednesday 2/16

Program Part

The program was released February 2 (and discussed February 4). Briefly, you should finish implementing a persistent left-leaning red-black tree (just the put method, no deletion). You should also try to implement a “stack-based” iterator, to replace the given iterator. For more details, including the turnin procedure, see share/hw2/Notes.txt.

Written Part

Try to solve these problems, and turn in your solutions on paper.

Problem 1. Think. Consider the left-leaning red-black BST, as described by 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 2. Read. Find and skim the paper “Planar point location using persistent search trees” by Sarnak and Tarjan (1986). (In Emory’s network, you can get the full text via dl.acm.org.) In this paper, they describe a persistent red-black search tree (but fancier than ours), using a combination of path-copying and “fat nodes”. This paper is also discussed on wikipedia, so you can look there (under “persistent data structures”).

2(a). They do not implement “full persistence”, but a more limited kind (that suffices for their “point location” problem). Describe the limited kind of persistence that they implement.

2(b). What are their big-Oh bounds on the time and space used, per update (insertion or deletion)? Also for each bound, state whether it is a worst-case bound or an amortized bound.

2(c). For their amortized analysis (of time or space, you decide), they used a “potential function”. What is that function? (You may assume $k = 1$, in their description.)

Problem 3. Experiment. For this problem I want you to write a little program that does a random experiment, to estimate some random quantities. Leave your program in your cs323/hw2/ directory (no other turnin required). By repeating the experiment sufficiently, try to get numbers within 5% of the correct value.

3(a). Suppose we insert $N = 200$ random items into a separate-chaining hash table with $M = 100$ lists, and no resizing. Estimate the expected number of empty lists, and also the expected length of the longest list.

3(b). Repeat the above, for $N = 20000$ and $M = 10000$.

3(c). Repeat the above, for $N = 20000$ and $M = 5000$. 

1
Reminder

Work in this course is governed by the Emory Honor Code[^1] and program assignments are governed by the Math/CS SPCA[^2]. In particular, you should take care to protect the confidentiality of your homework files. Apparent honor code violations will be referred to the Emory Honor Council. If you have questions about what is allowed under the policy, please ask.

[^1]: http://catalog.college.emory.edu/academic/policy/honor_code.html
[^2]: http://mathcs.emory.edu/spca.php