CS 161 Summer 1997

Handout #13

Problem Set 3 Due 3:05p, Tuesday, July 15, 1997

Readings

1. Handouts 9, 10, 11, 12.
2. Chapters 7, 14, 20.
3. Sections 13.1–13.3.
4. We recommend that you visit the WWW site

   `http://langevin.usc.edu/BST/`

   for interactive animations of tree algorithms, including red/black trees, splay trees, and treaps.

Exercises

Exercises are for you to work out, with the assistance of classmates and the course staff. Please don't hand in your solutions.

1. Exercise 7.1-2.
2. Exercise 7.2-5.
3. Exercise 7.3-3.
4. Exercise 20.2-1.
5. Exercise 20.2-3.
6. Show that at most $2n$ nodes are used in a deterministic skip list containing n elements. A "node" is a single box in figure 12.22 of handout 10.
7. Exercise 13.2-6.
8. Exercise 13.3-6.
9. Problem 13-4.
10. Exercise 14.3-4.
11. Exercise 14.4-3. If the sentinel were modified, then the whole red/black tree might get corrupted. Why does this never happen? *Tip:* examine the conditions under which RB-DELETE-FIXUP is invoked.
12. Where does the *Remove* procedure for treaps (figure 12.41 in handout 11) fail when the input contains elements with duplicate keys?

Problems

Your problem set grade depends only on the solutions you hand in for the problems below.

1. **[5 points]** Exercise 7.5-5.
2. **[15 points]** Problem 7-2.
3. **[10 points]** Exercise 20.1-3.
4. **[5 points]** Exercise 20.2-4.
5. **[10 points]** Show that the expected memory requirement of a randomized skip list (with $p = 0.5$) for a specific set S of size n is $O(n)$; use each of the boxes in figure 12.22 of handout 10 as your "unit" or memory. What is the worst-case, best-case, and average expected memory consumption of a randomized skip list (with $p = 0.5$) for an arbitrary set S of size n?
6. **[10 points]** Problem 13-2.
7. **[5 points]** Exercise 14.1-3.
8. **[5 points]** Exercise 14.4-6.
9. **[15 points]** Problem 14-2.
10. **[10 points]** Assume you are given an array of n pairs, each containing a key and a priority. The keys are distinct; the priority of each key was randomly determined (and cannot be modified); finally, the pairs are sorted by key.

    (a) **[6 points]** Devise an algorithm that can construct a treap for these n pairs, using the given priorities, in $O(n)$ time. You need not write pseudocode; a clear and concise description of your algorithm is sufficient.

    (b) **[4 points]** Give a clear and intuitive argument for the linear running time of your algorithm; a formal and rigorous proof is not necessary.

11. **[10 points]** The *internal path length* of a tree is the sum of the depths of all tree nodes. Consider the leftmost tree of figure 4.46 in handout 12; this is an example of a *splay tree of left children*. Consider a splay tree of 1024 left children with keys $1, 2, \ldots, 1024$: key 1 is at the leaf while key 1024 is at the root.

    (a) **[2 points]** What is the precise internal path length of this tree?

    (b) **[4 points]** Calculate the internal path length after each of the following operations, executed in sequence: $\mathit{Find}(1)$, $\mathit{Find}(2)$, $\mathit{Find}(3)$, and $\mathit{Find}(4)$.

(c) **[4 points]** If the sequence of successive *Find* operations is continued, after which call to *Find* is the internal path length minimized? What is this minimal length? Justify your answers.