CSE 326: Data Structures
Assignment #5
November 9, 1998
due: Monday, November 16

1. Page 254, exercise 26. Show the splay tree at least after each insertion and deletion.

2. (a) Suppose n is odd. You are given a splay tree on n nodes such that the path from the root to the key 1 passes through nodes with keys in the order

$$n, n-1, n-2, \ldots, 4, 3, 2, 1.$$

   Show the splay trees before and after splaying on the key 1.

   (b) How many dollars must be paid just for rotations in part (a)? (I.e., for this part ignore dollars spent to maintain the Money Invariant.)

   (c) The Investment Lemma says we are paid only $3\lfloor \log_2 n \rfloor + 1$ new dollars for the splay of part (a), so anything in excess of this in part (b) must come from money stored in the splay tree before the splay. Answer the following questions precisely. For $1 \leq i \leq n$, what was the rank of the node with key i before the splay? For $1 \leq i \leq n$, what is the rank of the node with key i after the splay? From which nodes can we take money to pay for the rotations? Show that these nodes have enough excess money to pay for all the rotations.

3. In this question you will explore what happens if you implement a dictionary as a splay tree, and do m dictionary operations starting from an arbitrary z-node splay tree T rather than the empty tree.

   (a) Suppose that someone invested the minimum number of dollars dictated by the Money Invariant in each node of T. Show that the total number of dollars invested in T would be $O(z \log z)$.

   (b) Show carefully that any m dictionary operations starting from T take a total time of $O(z \log z + m \log n)$, where n is the maximum size the dictionary attains during these operations.

   Hint: Note that in the proof given in lecture and the text, money invested in the tree represents unused time saved from earlier operations. With this interpretation, you wouldn't be able to assume that there is any money invested in T initially, since there haven't been any earlier operations. (What makes the empty tree a good starting place is that no initial investment is needed in order for the Money Invariant to hold immediately.) Instead, think of the initial investment in T as a credit line, and think about when you would be spending cash and when you would be using your credit card during the m operations. For instance, would you ever use credit from one node to reinstate the Money Invariant at some other node?

4. Prove that any Delete in a splay tree can be accomplished (maintaining the Money Invariant and paying \$1 per rotation, of course) if the client pays only $5\lfloor \log n \rfloor + 2$ dollars. (Hint: prove that $3\lfloor \log n \rfloor + 1$ dollars is sufficient for the Concat, and then use the idea from the midterm exam solutions to save another $\lfloor \log n \rfloor$ dollars.)

5. Insert the integers 87, 19, 25, 55, 36, 46, 88, 7, 67, 21 (in this order) into an initially empty hash table of size 11 using the hash function $h(x) = x \bmod 11$,

   (a) using separate chaining.

   (b) using open addressing with linear probing.

   (c) using open addressing with double hashing, where $h_2(x) = 1 + (x \bmod 10)$.

   (d) If a hash table is going to be this full (i.e., $n \approx m$) most of the time, and you have a hash function that spreads the entries over the hash table reasonably well, which of these methods is likely to be fastest, and why?