CSE 326: Data Structures
Assignment #5
November 13, 2009
due: Friday, November 20, 11:30 a.m.

1. (a) Consider the following splay tree T:

```
10
 \
  20
   \
    30
     \
      40
       \
        50
         \
          60
           \
            70
             \
              80
               \
                90
```

Someone in class pointed out that, in a Case II rotation, you could rotate first around the parent and then around the grandparent, and that would still bring the splayed node up to the root. Supposing that this is how Case II rotations were done, in the splay tree T show what the result of splaying 90 would have been.

   (b) Explain now why it would be a terrible idea to do Case II rotations as suggested in part (a). Consider what series of operations on T would perform very poorly.

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 0 passes through nodes with keys in the order

$$-\frac{n-1}{2}, \frac{n-1}{2}, -\frac{n-3}{2}, \frac{n-3}{2}, \ldots, -2, 2, -1, 1, 0.$$

Show the splay trees before and after splaying on the key 0.

   (b) How many RA (Rotation Allowances) must you pay just for rotations in part (a)? (That is, for this part ignore RA used to maintain the RA Invariant.)

   (c) The Investment Lemma says we are given only an additional $3\lfloor\log_2 n\rfloor + 1$ RA for the splay of part (a), so anything in excess of this in part (b) must come from RA stored in the splay tree before the splay. Answer the following questions precisely. For $-\frac{n-1}{2} \le i \le \frac{n-1}{2}$, what was the rank of the node with key i before the splay? For $-\frac{n-1}{2} \le i \le \frac{n-1}{2}$, what is the rank of the node with key i after the splay? From which nodes can we take RA to pay for the rotations? Show that these nodes have enough excess RA to pay for all the rotations.

3. We showed that you can delete key K from a splay tree if you are given an additional $7\lfloor \log n \rfloor + 2$ RA (in addition to the RA that the RA Invariant states are already in the tree). Recall that the breakdown for this figure was $3\lfloor \log n \rfloor + 1$ RA to splay on K, another $3\lfloor \log n \rfloor + 1$ RA to splay the left subtree on $+\infty$, and $\lfloor \log n \rfloor$ extra RA to invest in the new root because it takes on new descendents.

   Prove that any Delete in a splay tree can be accomplished (maintaining the RA Invariant and paying 1 RA per rotation, of course) if the client supplies only $5\lfloor \log n \rfloor + 2$ additional RA. (Hint: prove that $3\lfloor \log n \rfloor + 1$ RA is sufficient for the Concat and that another $\lfloor \log n \rfloor$ RA can be saved in the other part of Delete. In both cases the trick is to be careful about the investment at the root.)

4. 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?

5. Consider the universal class consisting of the hash functions

$$h_{a,b}(x) = ((ax + b) \bmod 71) \bmod 11$$

   for $0 < a < 71$ and $0 \leq b < 71$. Let x_1 and x_2 be any two distinct keys in $\{0, 1, \ldots, 70\}$. Justify your answers to the following. (Hint: you definitely don't want to start listing all the (a, b) pairs, as there are 4970 of them. Instead, use what you know from the proof of the Universal Classes of Hash Functions Theorem.)

   (a) Exactly how many of the $70 \cdot 71$ (a, b) pairs hash both x_1 and x_2 into bucket 4 (where the buckets are numbered $0, 1, \ldots, 10$ according to the value of $h_{a,b}(x)$)? Exactly how many hash both x_1 and x_2 into bucket 5?

   (b) Exactly how many ordered pairs of distinct numbers (q, r) are there such that $0 \leq q, r < 71$ and $q \equiv r \pmod{11}$? Compare your answer to the upper bound $N(N-1)/m$ proved in the Theorem.

   (c) What is the exact probability that a randomly chosen hash function $h_{a,b}$ will cause x_1 and x_2 to collide? Compare your answer to the upper bound $1/m$ proved in the Theorem.