CSE 638 Analysis of Algorithms	Spring 2007
Problem Set 4	
Lecturer: Michael A. Bender	*Due: Monday, 4/30/07*

Problem 1. Here is a relevant question from a previous qualifier exam that you may enjoy. You throw n balls into a bin. In each throw you decide randomly with probability 1/2 whether the ball is **blue** or **red**. Once the bin has n balls, what is the probability that:

(a) All n balls are blue.

(b) There are $n/2$ blue balls and $n/2$ red balls.

Now we have a new rule for throwing balls into bins. Let

$$B_t = \text{\# blue balls in bin,}$$
$$R_t = \text{\# red balls in bin.}$$

Initially there is **one** blue ball and **one** red ball. Thus, $B_2 = R_2 = 1$. The $t+1$st ball thrown into the bin is

$$\text{blue} \quad \text{with probability } \frac{B_t}{B_t+R_t},$$
$$\text{red} \quad \text{with probability } \frac{R_t}{B_t+R_t}.$$

That is, the probability that the next ball is blue (respectively, red) is the fraction of blue (red) balls in the bin.

(c) What is the probability that all balls thrown in the bin are blue, i.e., $B_n = n-1$? Provide provide a proof.

(d) What is the probability that exactly half of the balls thrown in the bin are blue $B_n = n/2$? Please provide a proof.

Problem 2. If you haven't already done so, please answer the following question about the marking algorithm. Given s stale and c clean requests prove that the cost of a stale request is $c/(k-s)$.

Problem 3. A *Weight Balanced Search Tree* maintains the following invariant. For each node u,

$$\text{weight(left-child)} + 1 = \Theta(\text{weight(right-child)} + 1).$$

(Recall that the weight of a node is the number of descendants.)

(a) Show that the tree can be searched in $O(\log n)$ time.

(b) How fast can you update the tree while maintaining the invariant?

(c) Suppose that on an update, whenever a node goes out of balance, you rebuild the entire subtree rooted at that node. What is the update cost then?

(d) What happens if you make a new invariant that the number of descendants of a node u is $\Theta\left(2^{\text{height}(u)}\right)$?

Problem 4. A hat contains n balls, each with a number on it. On each turn, a ball is drawn uniformly at random and without replacement, and shown to you. If you are able to identify the ball with the largest number when it is drawn, you win the game. Give a scheme that wins with probability at least $1/4$.

Problem 5. Just for practice, prove the following: If I flip a coin where the probability of heads is p, then I get heads in an expected $1/p$ rounds.

Problem 6. Hand in your scribe notes.

Problem 7. Discuss project proposal with me.