15-451 — Algorithms — Spring 2006

Sleator, Golovin, Kissner

Homework #3

Due: Tuesday, February 28, 2006, in class.

Some Reminders:

- You may discuss these problems with others, in small groups. However we strongly recommend that you think for a while about them yourself before starting such discussions.
- The work that you turn in must be your own, written by you in your own words. We are allowing handwritten solutions, although typeset ones are preferred. If you handwrite, WRITE CLEARLY, or we will revert to the old system of requiring you to typeset solutions.
- The cover page of your submission must clearly display the assignment number, your name, your recitation section and your Andrew ID.

1 Splaying

Suppose that a set of n keys $\{1, 2, \ldots, n\}$ are stored in a splay tree. The keys are then accessed (i.e. splayed) at random according to the following distribution: Key j is accessed with probability $C(9/10)^j$. The sequence of accesses is infinite.

(a) What is the value of the normalization constant C which makes the probabilities add up to 1?

(b) Give a bound (the tightest you can, and not a big-oh bound) on the expected cost (number of splay steps) of an access.

(c) Suppose the same sequence of accesses are done in a Red-Black tree. Give a bound (the tightest you can, and not a big-oh bound) on the expected cost (number of nodes touched) of an access.

2 Universal Hashing

(a) Describe an explicit universal hash function family from $U = \{0, 1, 2, 3, 4, 5, 6, 7\}$ to $\{0, 1\}$. Hint: you can do this with a set of 4 functions.

(b) Let H be a universal family of hash functions from some universe U into a table of size n. Let $S \subseteq U$ be some set we wish to hash. Prove that if we choose h from H uniformly at random, the expected number of pairs (x, y) in S that collide is at most $\frac{|S|^2}{2n}$.

(c) (Extra Credit) Prove that for some constants b and c, with probability at least $\frac{3}{4}$, no bin gets more than $b + \frac{c|S|}{\sqrt{n}}$ elements. (So, if $|S| = n$, you are showing that with probability $\frac{3}{4}$ no bin gets more than $b + c\sqrt{n}$ elements.) Hint: use part (b). To solve this question, you should use Markov's inequality. Markov's inequality is the following pretty obvious fact: if you have a non-negative random variable X with expectation $E[X]$, then for any $k > 0$, $Pr(X > kE[X]) \leq \frac{1}{k}$. For instance, the chance that X is more that 100 times its expectation is at most $1/100$. You can see that this has to be true just from the definition of expectation.

3 Pound the Monster

In the carnival game of *Pound the Monster*, there are n monsters numbered $1, 2, \dots, n$ all in a line. Each monster sits in a hole. Each second one of the monsters sticks its head out of its hole. You then have 1 second to pound the monster with a mallet. If you are successful you get some points.

Let's say you know in advance the sequence of monsters that will be popping up. So at time t $(1 \leq t \leq T)$ let m_t denote the monster that will pop up at that time. If you successfully pound that monster you get p_t points. Also, the mallet is heavy, so that in 1 second you can only move the mallet a distance of 1. So that the sequence of choices of what to pound must have the property that two pounds that are d apart must occur at times that differ by at least d. You are allowed to pick any desired starting point for the mallet.

Give an algorithm to efficiently compute the optimal sequence of monsters to hit. Actually, instead, compute the position to be at at time t which we'll call x_t. And if $x_t = m_t$ then this means that we hit the monster at time t picking up p_t points. Otherwise we pick up 0 points at time t.

Express your algorithm as formally as you can. And express its running time as a function of n and T. (Obviously, you should try to find as efficient an algorithm as possible.)