CMPUT 675: Randomized Algorithms **Fall 2005**

Problem Set 3:

Due Date: Nov 29, 2005

You are allowed to discuss the problems with each other, however you must not take any notes from your discussions. You must list the names of other students that you have discussed the problems with, or any other resources (other than the text book) that you have used. These information will not affect your mark. The questions are in arbitrary order of difficulty.

1. Consider the following random walk on a circle. There are n equidistant points numbered $0, \dots n-1$ (clockwise). We start at 0 and every step we move with probability 1/2 to one of the two neighbouring points. Which point is most likely to be the last point visited for the first time?

2. In hashing with open addressing, the hash table is just an array and there are no linked lists or chaining. For each given key i, the hash function defines a sequence $h(i,1)$, $h(i,2), \dots$ of table locations. To insert key i, we check the locations given the probe sequence one by one and insert the key in the first empty location we find. For searching, we look at the prob locations in that order, until we either find the key or find the first empty location (which means the key does not exist). Suppose we have a hash table of size $2n$ to store n items and assume that each $h(i,j)$ location is uniformly random and independent of all other locations.
   (a) Prove that probability of a single insertion requiring more than k probes is at most 2^{-k}.
   (b) Prove that for a sequence of n insertion, the maximum number of probes required for any insertion is $O(\log n)$ w.h.p. (Hint: What is the probability of any single insertion requiring more than $2\log n$ probes?)

3. Problem 7.4 from the text.

4. Consider the following scheme to build a perfect family of hash functions using the idea of "power of two choices" in the balls and bins problem. We have n items to be mapped; keep two arrays A_1 and A_2 of size $O(n^{3/2})$ each. We have two (distinct and randomly chosen) 2-universal hash functions h_1 and h_2 for mapping items into these two arrays (h_i for A_i, $i = 1, 2$). Given a sequence of n insertions, for each item find the mappings by h_1 and h_2 and place the item in the location which is empty (if any).
   (a) Show that for each item and its location in A_1, the expected number of items in that location is $O(1/\sqrt{n})$ (similarly for A_2).
   (b) Prove that the probability that both locations for item i are full is $O(1/n)$; conclude that with probability at least 1/2 we can find an empty location for each item.