CS 577: Introduction to Algorithms — Homework 7

Out: 03/27/18 — **Due: 04/12/18**

Ground Rules

- The homework is worth 5 points (out of a total of 100 points you can accumulate in this course).
- The homework is to be done and submitted in pairs. You can partner with someone from either section.
- The homework is due at the beginning of either lecture on the due date.
- No extensions to the due date will be given under any circumstances.
- Turn in your solution to each problem on a **separate sheet** of paper (or sheets stapled together), with your names clearly written on top.

Problems

1. **(Worth: 2 points. Page limit: 1 sheet; 2 sides)**

   (a) We have n users on a P2P network who want to pick *distinct* IDs in a distributed fashion. Each user independently picks one uniformly random b bit number. Show that when $b \geq 6 + 2\log n$, the probability of the users picking distinct IDs is at least 0.99.

   (b) We have n users with distinct IDs on a P2P network, who want to elect a leader in a distributed fashion. Each user independently picks a uniformly random b-bit number, and the leader is determined to be the user with the smallest number. Show that when $b \geq 8+\log n$, the probability that a unique leader is chosen is at least 0.99.

   (*Hint: To obtain a simple bound on the sum of a series, approximate it by an integral. In particular, for a positive increasing function h, $\sum_{i=1}^{k} h(i) \geq \int_{i=0}^{k} h(i)\, di$.*)

2. **(Worth: 3 points. Page limit: 1 sheet; 2 sides)**

   (a) You are given a circle of unit circumference. You pick k points on the circle independently and uniformly at random and snip the circle at those points, obtaining k different arcs. Determine the expected length of any single arc.

   (Hint: Note that the length of each arc is identically distributed, so each has the same expectation. What is the sum of their expectations?)

   

   (b) You are given a sorted circular linked list containing n integers, where every element has a "next" pointer to the next larger element. (The largest element's "next" pointer points to the smallest element.) You are asked to determine whether a given target element belongs to the list. There are only two ways you can access an element of the list: (1) to follow the next pointer from a previously accessed element, or (2) via a given function RAND that returns a pointer to a uniformly random element of the list.

   Develop a randomized algorithm for finding the target that makes at most $O(\sqrt{n})$ comparisons in expectation and always returns the correct answer.

   (Hint: Your algorithm will perform some random accesses and some amount of linear search. Use part (a) to analyze the number of steps in the linear search.)