CS 498ABD Spring 2019
Midterm Practice Problems

Disclaimer: These problems are not meant to be representative of the exam. It is just to give you more opportunities to practice concepts that are likely to be relevant.

1. Consider a biased coin that flips heads with probability p.
   (a) What is the expected number of times you need to flip the coin before seeing heads?
   (b) What is the variance?
2. Consider a biased coin that flips heads with unknown probability $p \geq \alpha$ where α is a known fixed constant. Suppose we flip the coin n times and record the number of heads observed.
   (a) Derive an unbiased estimator $\tilde{p}$ of p from the information recorded.
   (b) How large must n be so that $\Pr[|\tilde{p} - p| \geq \varepsilon p] \leq \delta$, in terms of $\varepsilon, \delta, \alpha$?
3. In the selection problem we are given an array A of n numbers (not necessarily sorted) and an integer k and the goal is output the rank k element of A. Consider a randomized version where we pick a number x uniformly at random from A and use it as a pivot as in quick sort to partition A into numbers less than equal to x and numbers greater than x. The algorithm recurses on *one* of these arrays depending on k and the size of the two arrays.
   (a) Write down a formal description of randomized quick selection.
   (b) Prove that the expected run time of this algorithm is $O(n)$ and that the expected depth of recursion is $O(\log n)$.
   (c) Prove that the depth of recursion is $O(\log n)$ with high probability. Does this imply that the algorithm terminates in $O(n)$ time with high probability?
4. *Tabulation hashing* is a scheme that uses tables of random numbers to compute hash values. Suppose $U = \{0,1\}^n \times \{0,1\}^n$, i.e., pairs of n-bit strings, and $m = 2^\ell$.

   Let $A[0 \mathinner{..} 2^n - 1]$ and $B[0 \mathinner{..} 2^n - 1]$ be arrays of independent ℓ-bit strings and define the hash function $h_{A,B} : U \to [m]$ by
   $$h_{A,B}(x, y) = A[x] \oplus B[y]$$
   where $\oplus$ denotes bitwise-xor. Here we obtain $A[x]$ and $B[y]$ by interpreting x and y as numbers in $[2^n]$. Filling the arrays A and B with independent uniform random bit-strings is equivalent to choosing a hash function uniformly at random.
   (a) Prove that h is pairwise independent.
   (b) (Extra Credit) Prove that h is 3-wise independent.
   (c) Prove that h is *not* 4-wise independent.
5. Suppose we independently throw n balls into m bins uniformly at random.
   (a) For a given bin, what is the probability that it contains more than one ball?
   (b) What is the expected number of bins that contain more than one ball?