CS 70 Discrete Mathematics and Probability Theory
Summer 2015 Chung-Wei Lin Discussion 6B

1. **Balls in Bins: Independent?**

   You have k balls and n bins labelled $1, 2, \ldots, n$, where $n \geq 2$. You drop each ball uniformly at random into the bins.

   (a) What is the probability that bin n is empty?

   (b) What is the probability that bin 1 is non-empty? Argue this both by counting, and by independence.

   (c) What is the probability that both bin 1 and bin n are empty?

   (d) What is the probability that bin 1 is non-empty and bin n is empty?

   (e) What is the probability that bin 1 is non-empty given that bin n is empty?

2. **Hash Families**

   In the analysis for hashing in lecture, the hash function was presented as a completely random map to any of the locations in the table.

   (a) Let us model such a function as the function $ax + b \pmod{T}$, where prime T is the size of the table, and $x \in U$ is the element to be hashed. For a given x, is such a function uniformly random over the table?

   (b) Let's say I try a different strategy: I pick k equations of the form $a_k x + b_k \pmod{T}$, where T is the size of the table. I hash the k-th element using the k-th equation. I am so confident in my scheme of increased randomness that I publish my equations for adversaries to admire. Explain how to break (send many many elements to the same bin) my scheme if you know the equations.

   (c) Now, instead of using k equations, I use T^2 equations. What are these equations?

   (d) I then randomly pick one of the equations to use without telling adversaries. Is this a scheme that works? Why or why not?

3. Coupon Collector

Given a size k hash table, approximately how many keys will have to be added until there are no non-nempty spaces? (Hint: define X_i as the number of keys needed to add the i-th distinct value after having $(i-1)$-th distinct value.)

4. Throwing Balls into a Depth-Limited Bin

Say you want to throw n balls into n bins with depth $k-1$ (they can fit $k-1$ balls, after that the bins overflow). Suppose that n is a large number and $k = 0.1n$. You throw the balls randomly into the bins, but you would like it if they don't overflow. You feel that you might expect not too many balls to land in each bin, but you're not sure, so you decide to investigate the probability of a bin overflowing.

(a) Focus on the first bin. Get an upper bound the number of ways that you can throw the balls into the bins such that this bin overflows. Try giving an argument about the following strategy: select k balls to put in the first bin, and then throw the remaining balls randomly.

(b) Calculate an upper bound on the probability that the first bin will overflow.

(c) Upper bound the probability that some bin will overflow.

(d) How does the above probability scale as n gets really large?