# 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?
$(\frac{n-1}{n})^k$.

(b) What is the probability that bin 1 is non-empty? Argue this both by counting, and by independence.
$1 - (\frac{n-1}{n})^k$.

(c) What is the probability that both bin 1 and bin $n$ are empty?
$(\frac{n-2}{n})^k$.

(d) What is the probability that bin 1 is non-empty and bin $n$ is empty?
$(\frac{n-1}{n})^k - (\frac{n-2}{n})^k$.

(e) What is the probability that bin 1 is non-empty given that bin $n$ is empty?
$$\frac{\left(\frac{n-1}{n}\right)^k - \left(\frac{n-2}{n}\right)^k}{\left(\frac{n-1}{n}\right)^k} = 1 - \left(\frac{n-2}{n-1}\right)^k.$$

**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?

Yes, because the field is prime, every element in the table can be mapped to with equal probability with the particular set of variables $a$ and $b$ that reach that position.

(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.

Because the enemy knows the equation, he can create messages that go to the same bin for each equation by solving each equation for a specific value (there are inverses in a prime field).

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

All different options for $a$, $b$ out of $GF(T)$ where $T$ is prime.

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

Because the constants are picked at random, even though I know the entire set of equations it could be, each different equation will randomly map the input, so my guess is as good as random.

## 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.)

Following the hint, $X = \sum_{i=1}^{k} X_i$. So $\mathbf{E}[X] = \sum_{i=1}^{k} \mathbf{E}[X_i]$. The probability that a random key goes into the $i$-th distinct location is $p_i = \frac{k-i+1}{k}$, so $\mathbf{E}[X_i] = p_i \sum_{j=1}^{\infty} j(1-p_i)^{j-1} = \frac{1}{p_i}$. So $\mathbf{E}[X] = \sum_{i=1}^{k} \frac{1}{p_i} = k \sum_{i=1}^{k} \frac{1}{i}$.

## 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.

We choose $k$ of the balls to throw in the first bin and then throw the remaining $n-k$, giving us $\binom{n}{k}n^{n-k}$. Certainly any outcome of the ball-throwing that overflows the first bin is accounted for – we can simply choose the first $k$ balls that land in the first bin and then simulate the rest of the outcome via random throwing. However, we are potentially overcounting: if $k+1$ balls go in the first bin, we have many choices for which $k$ of them that could have been the "chosen" ones, and we count each one of these choices as distinct. However, they correspond to the same configuration, namely the one where $k+1$ balls are in the first bin. Hence we get an upper bound.

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

We divide by the total number of ways the balls could have fallen into the bins, with order, so we get $\frac{\binom{n}{k}n^{n-k}}{n^n} = \frac{\binom{n}{k}}{n^k}$.

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

By symmetry, we can just upper bound this probability by $n$ multiplied by the probability that a single bin (wlog, the first bin) overflows. This gives about $n \cdot \frac{\binom{n}{k}}{n^k}$. This technique is called a *union bound*, where we upper bound the probability of the union of a bunch of events by the sums of the probabilities of the events.

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

We get $n \cdot \frac{\binom{n}{k}}{n^k} = n \cdot \frac{n\cdot(n-1)\cdot\cdots\cdot(n-k+1)}{k!n^k} \leq n \cdot \frac{n^k}{k!n^k} = \frac{n}{k!} = \frac{n}{(0.1n)\cdot(k-1)!} = \frac{10}{(0.1n-1)!}$. Clearly, as $n$ gets large this probability is going to 0. Note that this same analysis would work with $k = cn$ for any constant $0 < c < 1$. Hence, using some very coarse upper bounds we can see that as the number of balls and bins grows we have that it is very unlikely that we get a constant fraction of the balls in any single bin.