EECS 70 Discrete Mathematics and Probability Theory
Fall 2014 Anant Sahai

Discussion 11W

1. Hashing

(a) Given k items, how big a hashtable do we need so that when the k items are hashed, there are no collisions with at least constant probability, say p? Assume the hash function assigns items to buckets independently and uniformly at random.

For this part, take p to be large enough to force n to be much larger than k.

You might need to use $\ln(1-x) \approx -x$ for small x.

(b) Given k items, how big a hashtable do we need so that when the k items are hashed, there are no collisions with *a particular item*, call it x, with constant probability? Same assumptions on the hash function as above.

2. 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?

3. String approximation

How many strings of length $4n$ have exactly n letter ‘a’s, n letter ‘b’s, n letter ‘c’s, and n letter ‘d’s? Use Stirling’s approximation to estimate how fast this grows as a function of n.