CS 561: Homework 2

Prof. Jared Saia, University of New Mexico

Remember: You are encouraged to work on the homework in groups, but please observe the "Star Trek" rule from the syllabus.

1. You are stress-testing a model of smartphone. You have a ladder with n rungs and a collection of identical phones. There is an unknown threshold t, with $0 \leq t \leq n$, such that a phone survives a drop from any rung at most t and breaks from any rung greater than t. A broken phone cannot be used again. Your goal is to determine t while minimizing the worst-case number of drops.

   (a) Suppose that you have exactly two phones. Design an algorithm that determines t using $o(n)$ drops in the worst case. Analyze the number of drops used by your algorithm.

   (b) Now suppose that you have k phones, where $k \geq 2$ is a fixed constant. Design a recursive algorithm for this problem. If $f_k(n)$ is its worst-case number of drops, give a tight asymptotic bound on $f_k(n)$ for your algorithm. Your bound should satisfy

   $$f_k(n) = o\big(f_{k-1}(n)\big).$$

   Explain how you choose the block sizes at each recursive level.

2. The game of *Match* is played with a deck of 27 cards. Each card has three attributes: color, shape, and number. Each attribute has three possible values, and every one of the $3^3 = 27$ combinations appears exactly once. A *match* is a set of three cards such that, for each attribute, the three values are either all equal or all different. For example,

   $$(3,\ \text{red},\ \text{square}),\quad (2,\ \text{blue},\ \text{square}),\quad (1,\ \text{green},\ \text{square})$$

   is a match.

   (a) The deck is shuffled and three cards are turned over. What is the probability that they form a match? Justify your answer by conditioning on the first two cards.

   (b) The deck is shuffled and the first s cards are turned over, where $3 \leq s \leq 27$. Let X be the number of matches among these cards; matches are counted separately even when they overlap. Use indicator random variables and linearity of expectation to compute $\mathbb{E}[X]$. Check your expression when $s = 27$.

3. A hash table has m slots. Each of n distinct keys is hashed independently and uniformly to one of the slots. For every unordered pair of keys $\{i, j\}$, let $X_{i,j}$ be the indicator random variable for the event that i and j hash to the same slot, and let

$$X = \sum_{1 \le i < j \le n} X_{i,j}.$$

   (a) Compute $\Pr(X_{i,j} = 1)$ and $\mathbb{E}[X]$.

   (b) Use Markov's inequality to upper-bound the probability that the table has at least one collision.

   (c) Let $0 < \delta < 1$. How large should m be, as a function of n and δ, for your bound to guarantee that the probability of any collision is at most δ?

   (d) Let Y be the number of unordered triples of keys whose three members all hash to the same slot. Compute $\mathbb{E}[Y]$, and use it to upper-bound the probability that some slot receives at least three keys.

4. You are given three $n \times n$ matrices A, B, and C with integer entries, and you want to check whether $AB = C$ without explicitly computing the matrix product AB. Consider the following randomized test:

   1. Choose a column vector $r \in \{0, 1\}^n$ by choosing each entry independently and uniformly at random.
   2. Compute $y = A(Br)$ and $z = Cr$.
   3. Reject if $y \neq z$; otherwise, accept.

   For the questions below, assume that all arithmetic is exact and count each arithmetic operation as taking constant time.

   (a) Prove that if $AB = C$, then the test always accepts.

   (b) Suppose that $AB \neq C$, and let $D = AB - C$. Prove that

$$\Pr(Dr = 0) \le \frac{1}{2}.$$

   Hint: Choose a nonzero row of D, then choose a nonzero entry in that row. Condition on all coordinates of r except the coordinate corresponding to that entry.

   (c) Conclude that if $AB \neq C$, a single execution accepts with probability at most $1/2$. Can the test make an error when $AB = C$?

   (d) Now run the test independently k times, accepting only if every run accepts. Bound the probability of accepting when $AB \neq C$.

   (e) Analyze the runtime of the repeated test. For any fixed constant $c > 0$, how should you choose k so that the error probability is at most $1/n^c$? What is the resulting runtime?