CS 170 DIS 12

Released on 2018-04-17

1 2-Universal Hashing

Let $\mathcal{H}$ be a class of hash functions in which each $h \in \mathcal{H}$ maps the universe $\mathcal{U}$ of keys to $\{0, 1, \ldots, m-1\}$. We say that $\mathcal{H}$ is 2-universal if, for every fixed pair, $\langle x, y \rangle$ of keys where $x \neq y$, and for any h chosen uniformly at random from $\mathcal{H}$, the fair $\langle h(x), h(y) \rangle$ is equally likely to be any of the m^2 pairs of elements from $\{0, 1, \ldots, m-1\}$. (The probability it taken only over the random choise of the hash function.)

(a) Show that, if $\mathcal{H}$ is 2-universal, then it is universal.

(a) Suppose that an adversary knows the hash family $\mathcal{H}$ and controls the keys we hash, and the adversary wants to force a collision. In this problem part, suppose that $\mathcal{H}$ is universal. The following scenario takes place: we choose a hash function h randomly from $\mathcal{H}$, keeping it secret from the adversary, and then the adversary chooses a key x and learns the value $h(x)$. Can the adversary now force a collision? In other words, can it find a $y \neq x$ such that $h(x) = h(y)$ with probability greater than $1/m$?

2 Markov Bound Review

Recall Markov's Inequality from CS 70. That is, for any non-negative random variable X, $Pr(X \geq a) \leq \frac{E[X]}{a}$. Provide a simple proof for Markov's bound.

3 Streaming for Voting

Consider the following scenario. Votes are being cast for a major election, but due to a lack of resources, only one computer is available to count the votes. Furthermore, this computer only has enough space to store one vote at a time, plus a single extra integer. Each vote is a single integer 0 or 1, denoting a vote for Candidate A and Candidate B respectively.

(a) Come up with an algorithm to determine whether candidate A or B won, or if there was a tie.

(b) Consider now an election with 3 candidates. Say there is a winner only if a candidate recieves more than 50 percent of the vote, otherwise there is no winner. If we're given another integer's worth of storage, come up with an algorithm to determine the winner if there is one. For simplicity, your algorithm can output any of the candidates in the case that there is no winner (not necessarily the one with the most votes). Votes are now numbered 0, 1, 2.