CS 170 DIS 12

Released on 2018-11-19

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 $\{0, 1, \ldots, m-1\}$. Recall that h is *universal* if for any $x \neq y \in \mathcal{U}$, $\Pr_{h \in \mathcal{H}}[h(x) = h(y)] \leq 1/m$.

We say that $\mathcal{H}$ is 2-universal if, for every fixed pair (x, y) of keys where $x \neq y$, and for any h chosen uniformly at random from $\mathcal{H}$, the pair $(h(x), h(y))$ is equally likely to be any of the m^2 pairs of elements from $\{0, 1, \ldots, m-1\}$. (The probability is taken only over the random choice of the hash function.)

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

(b) Suppose that you choose a hash function $h \in \mathcal{H}$ uniformly at random. Your friend, who does not know which hash function you picked, tells you a key x, and you tell her $h(x)$. Can your friend tell you $y \neq x$ such that $h(x) = h(y)$ with probability greater than $1/m$ (over your choice of h) if:

(i) $\mathcal{H}$ is universal?

(ii) $\mathcal{H}$ is 2-universal?

In each case, either give a choice of $\mathcal{H}$ which allows your friend to find a collision, or prove that they cannot for any choice of $\mathcal{H}$.

2 Markov Bound Review

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

3 Document Comparison with Streams

You are given a document A and then a document B, both as streams of words. Find a streaming algorithm that returns the degree of similarity between the words in the documents, given by $\frac{|I|}{|U|}$, where I is the set of words that occur in both A and B, and U is the set of words that occur in at least one of A and B.

Clearly explain your algorithm and briefly justify its correctness and memory usage (at most $\log(|A| + |B|)$. Can we achieve accuracy to an arbitrary degree of precision? That is, given any $\epsilon > 0$ can we guarantee that the solution will always be within a factor of $1 \pm \epsilon$ with high probability?

4 Lower Bounds for Streaming

(a) Consider the following simple 'sketching' problem. Preprocess a sequence of bits $b_1, \ldots, b_n$ so that, given an integer i, we can return b_i. How many bits of memory are required to solve this problem exactly?

(b) Given a stream of integers $x_1, x_2, \ldots$, the *majority element* problem is to output the integer which appears most frequently of all of the integers seen so far. Prove that any algorithm which solves the majority element problem exactly must use $\Omega(n)$ bits of memory, where n is the number of elements seen so far.