COMP4804 Assignment 2: Due Tuesday February 23rd, 23:59EDT

Print this assignment and answer all questions in the boxes provided. Any text outside of the boxes will not be considered when marking your assignment.

1 Multiplicative Hashing — The Wrong Way

This exercise studies why the choice of a in the multiplicative universal hashing algorithms is important. Recall that the multiplicative hashing scheme that takes elements from $\{0,\ldots,2^k-1\}$ onto $\{0,\ldots,2^\ell-1\}$ using the hash function $h_a(x) = (ax \bmod 2^k) \operatorname{div} 2^{k-\ell}$.

1. Suppose k/ℓ is an integer and consider the set of keys

$$S = \{j(2^k/2^\ell) : j \in \{0,\ldots,2^\ell - 1\}\} \ .$$

   Suppose a is of the form $2^i t$ where t is an odd integer. How many distinct values are in the set $h_a(S) = \{h_a(x) : x \in S\}$?

2. Suppose we choose a uniformly at random from $\{0,\ldots,2^k-1\}$. What is the probability (as a function of i) that a is of the form $2^i t$ where t is an odd integer?

3. Suppose we choose a uniformly at random from $\{0, \dots, 2^k - 1\}$ and use this to store the set S described in Part 1. What is the expected time to search for a value $x \in S$ in the resulting hash table?

2 Matrix Equality Testing

Let $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ be two real-valued vectors of length n and let $r = (r_1, \ldots, r_n)$ be a random binary vector of length n. That is, the r_i's are chosen independently and uniformly at random to be either 0 or 1. For a vector x, let $r \cdot x$ denote the sum $r_1x_1 + r_2x_2 + \cdots + r_nx_n$.

1. It is clear that, if $a = b$ then $r \cdot a = r \cdot b$. Show that if $a \neq b$ then $\Pr\{r \cdot a = r \cdot b\} \leq 1/2$. (Hint: Consider what has to happen at a particular index i such that $a_i \neq b_i$. What is the probability that this happens?)

2. Using the above fact (whether you could provide a proof or not), show that for two $n \times n$ matrices A and B that are not equal, $\Pr\{r \times A = r \times B\} \leq 1/2$. (Note that computing $r \times A$ is an n vector that takes $O(n^2)$ time to compute.)

3. Let A, B and C be three $n \times n$ matrices. Describe an algorithm that runs in $O(n^2)$ time and

   (a) If $A = B \times C$ then the algorithm always outputs yes.

   (b) If $A \neq B \times C$ then the algorithm will output no with probability at least 1/2.

   (Hint: Matrix multiplication, i.e., computing $B \times C$ explicitly takes too long. You'll have to find some other way.)

3 A Monte-Carlo Min-Tricut Algorithm

A *tricut* of an undirected graph $G = (V, E)$ is a subset of E whose removal separates G into at least 3 connected components. A min-tricut of G is a tricut of minimum size (over all possible tricuts of G). This question studies a problem of computing the min-tricut.

1. Let C be a min-tricut of G. Prove the best upper bound you can on the size of C in terms of $|V|$ and $|E|$. (Hint one possible tricut can be obtained by separating two vertices from the rest of G by deleting all their incident edges.)

2. If we pick a random edge $e \in E$ give an upper bound on the probability that $e \in C$.

3. Suppose we repeat the following $|V| - 4$ times: Select a random edge e of G, contract e (identify the two endpoints of e) and eliminate any loops (edges with both endpoints at the same vertex). Give a lower bound on the probability that all $n - 4$ edge contractions avoid the edges of C.

4. In a graph with 4 vertices, give an upper bound on the probability that a randomly selected edge is part of the Min-Tricut. (Hint: Your bound in part 1 may not be strong enough to give a non-trivial upper bound. You will really have to see what a tricut in a graph with 4 vertices looks like.)

5. Give a lower bound on the probability that a sequence of $n-3$ edge contractions of randomly chosen edges do not contract any of the edges in a min-tricup C. (Hint: You get this from the last two questions.)

6. The previous question gives a lower bound on the probability that a monte-carlo algorithm finds a min-tricut C. Unfortunately, the probability is very small. How many times would we have to run the algorithm so that the probability of finding C is at least

   (a) $1 - 1/e$

   (b) $1 - 1/1\,000$

   (c) $1 - 1/1\,000\,000\,000$

4 Monte-Carlo Landslide Finding

We are given an array $A_1, \ldots, A_n$ and we are told that some element x occurs $2n/3$ times in the array, but we are not told the value of x. Our goal is to use a fast Monte-Carlo algorithm (that may report the incorrect value) to find x.

1. Describe an $O(1)$-time Monte-Carlo algorithm to find x that is correct with probability $2/3$.

2. Suppose we sample k elements at random (with replacement) from the array to obtain k sample values $S_1, \ldots, S_k$. Give a good upper-bound on the probability that x occurs less than $(1-\epsilon)2k/3$ times in this sample.

3. Give a good upper bound on the probability that x occurs less than $k/2$ time in the sample. (Hint: This is the same as the previous question except we are using a specific value of ϵ.)

4. Describe a Monte-Carlo algorithm that runs in $O(k)$ time and reports x with probability at least $1 - 1/e^{\Omega(k)}$. (Just describe the algorithm. The error probability follows from the previous questions.)

5 McDiarmid's Inequality

Chernoff's bounds is only one of many *concentration inequalities* that probability theory offers to us. In this question we explore an extremely powerful and general inequality due to McDiarmid.

Theorem 1 (McDiarmid's Inequality)**.** *Let A be some set of values and let $f : A^n \to \mathbb{R}$ be a function that satisfies*

$$|f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n)| \le c_i$$

for all $x_1, \ldots, x_n, x'_i \in A^{n+1}$ and all $1 \le i \le n$. Then, if $X_1, \ldots, X_n$ are independent random variables that only take on values in A then

$$\Pr\{|f(X_1, \ldots, X_n) - \mathbf{E}[f(X_1, \ldots, X_n)]| \ge t\} \le \frac{2}{e^{2t^2/\sum_{i=1}^{n} c_i^2}} .$$

In words, McDiarmid's Inquality says that if we have a function f that doesn't change too much if we only change one of f's inputs then $f(X_1, \ldots, X_n)$ is stronly concentrated around its expected value.

1. A Bernoulli(p) random variable takes on values in the set $A = \{0, 1\}$. If $X_1, \ldots, X_n$ are independent Bernoulli(p) random variables and $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$ then what does McDiarmid's Inequality tell us about $f(X_1, \ldots, X_n)$? Does this remind you of anything?

2. Let $X_1, \dots, X_n$ be independent random variables that are uniformly distributed in the unit interval $A = [0, 1]$. Let $f(x_1, \dots, x_n)$ be the function that counts the number of inversions in $x_1, \dots, x_n$ (an inversion is a pair (x_i, x_j) with $i < j$ and $x_i > x_j$. What is $\mathbf{E}[f(X_1, \dots, X_n)]$?

3. Using the same setup as the previous question. If we change one value x_i to x_i', what is the maximum value of $|f(x_1,\ldots,x_n) - f(x_1,\ldots,x_i',\ldots,x_n)|$.

4. What does McDiarmid's inequality tell us about $\Pr\{|f(X_1,\ldots,X_n) - \binom{n}{2}/2| \geq \epsilon n^2\}$

5. Suppose we run the insertion sort algorithm on $X_1,\ldots,X_n$ independently and uniformly distributed in $[0,1]$. Then what does the above imply about (a) the number of swaps performed and (b) the number of comparisons performed.