**Problem Set 5** Andrew Koulogeorge, Eric Richardson

COSC 34: Randomized Algorithms Date: December 20, 2024

Collaboration Statement: Andrew Koulogeorge and Eric Richardson

**Problem 2: Comparing Multisets.**

*Solution.*

```
procedure COMPARE($S_1, S_2$)
    compute $n_1 = |S_1|$
    sample $h \in H$ such that $h : N \to [n_1]$
    for $x \in S_1$ do
        $j = h(x)$
        Search for $x \in T[j]$:
        if found then
            increment data to $(x, \text{freq}(x) + 1)$
        else
            store $(x, 1)$
    for $y \in S_2$ do
        $k = h(y)$
        Search for $y \in T[k]$:
        if found then
            decrement tuple to $(y, \text{freq}(y) - 1)$
            if freq$(y) - 1 = 0$ then
                delete from list
        else
            return False
    return $True$ if $T$ empty else $False$
```

**Correctness:** (1): **Suppose** $S_1$ **and** $S_2$ **are equal multisets** $Compare(S_1, S_2)$ we will hash each distinct element of $S_1$ along with its frequency, $(x, freq(x))$, into $T$. As we loop over each element of $S_2$, we will decrement the frequency of each value we see. Since the two multisets are identical, after the second loop terminates $T$ will be empty and we will return True. (2): **Suppose** $S_1 \neq S_2$. If $S_1$ contains an element that $S_2$ does not then $T$ will not be empty after the 2nd loop and $Compare(S_1, S_2)$ will return False. If $S_2$ contains an element that $S_1$ does not then we will not find it in $T$ during the second loop and $Compare(S_1, S_2)$ will return $False$.

**Runtime:** Since we have a single table $T$ where each element of $T$ stores a linked-list of ordered pairs $(x, freq(x)$, it suffices to show that $\forall\, x \in S_1\ \mathbf{E}\left[|T(h(x))|\right]$ is a constant after hashing all the elements from $S_1$ into $T$. This will ensure that each of our searches for

$x \in T[j]$ and $y \in T[k]$ will be done in constant amount of time. This analysis is nearly identical to the analysis done in the previous problem. Let $X_{x,y} = 1$ if $h(x) = h(y)$ and $0$ otherwise. It follows that:

$$|T(h(x))| = \sum_{y \in S_1} X_{x,y} = 1 + \sum_{y \in S_1 \ y \neq x} X_{x,y}$$

$$\mathbf{E}\left[|T(h(x))|\right] = 1 + \sum_{y \in S_1 \ y \neq x} \mathbb{P}\left[h(x) = h(y)\right] \leq 1 + \frac{n_1 - 1}{n_1} \leq 2$$

The first line follows from the definition of $|T(h(x))|$: elements $y$ are stored in the linked list at index $h(x)$ if $h(y) = h(x)$. When applying the expectation, there are at most $n_1$ distinct elements in $S_1$ and each of them has at most a probability of $\frac{1}{n_1}$ by the properties of the UHF. Note that this analysis upper bounds the expected size of $|T(h(x))|$ after we hashed all of $S_1$, which acts as an upper bound for each iteration of both the first loop over $S_1$ and the second loop over $S_2$. This is because in between the two loops the size of the table is maximized, as we are increasing the size of $T$ during the first loop and decreasing the size of the $T$ during the second loop. Thus, we know the expected size of each linked list of $T$ during our search is constant. Applying this upper bound to the loops over $S_1$ and $S_2$, we see that the total time taken is expected $O(n_1 + n_2)$. Computing $n_1 = |S_1|$ at the start of the algorithm is a deterministic computation and takes $O(n_1)$. At the end of the algorithm we loop over each element of $T$ and ensure that it contains all 0s which takes $O(n_1)$. Thus, the expected runtime is $O(3n_1 + n_2) = O(n)$ where $n = n_1 + n_2$

**Problem 4: Pairwise Independent Hash Family.**

*Solution.* Let $D \subset U$ be a subset of $n$ distinct elements of $U$ and consider a hash table $T[1:n]$ of lists where we put $e \in D$ in the list $T[h(e)]$ for $j \in [n]$ and $e \in D$, let $X_{e,j} = 1$ be the indicator variable for the event $h(e) = j$

(a) We will apply the law of total probability. Fix an $\hat{e} \in U$ and define the collection of events $\Omega$ such that $\omega_i = \{h(\hat{e}) = i\}$. Note that $\Omega$ is mutually exclusive and collectively exhaustive. Thus, we can apply the law of total probability on the event that $h(e) = j$:

$$\mathbb{P}\left[h(e) = j\right] = \sum_{i=1}^{n} \mathbb{P}\left[h(e) = j \cap h(\hat{e}) = i\right] = \sum_{i=1}^{n} \frac{1}{n^2} = \frac{1}{n} \quad \square$$

(b) Fix $e, \hat{e} \in U$. To show that $X_{e,j}$ and $X_{\hat{e},j}$ we need to show that:

$$\forall\, a, b\, \mathbb{P}\left[X_{e,j} = a \cap X_{\hat{e},j} = b\right] = \mathbb{P}\left[X_{e,j} = a\right] \mathbb{P}\left[X_{\hat{e},j} = b\right]$$

Note that $X_{e,j}$ and $X_{\hat{e},j}$ only take on values $0$ and $1$ so we need to show this holds for each of the $4$ cases. Let $A = X_{e,j} = 1$, $B = X_{e,j} = 1$. We know from the previous part that $\mathbb{P}[A] = \mathbb{P}[B] = \frac{1}{n} \implies \mathbb{P}[\neg A] = \mathbb{P}[\neg B] = \frac{n-1}{n}$

(1) From the definition of a pairwise independent hash family, we know that:

$$\mathbb{P}[A \cap B] = \frac{1}{n^2} = \mathbb{P}[A]\,\mathbb{P}[B]$$

(2) From inclusion exclusion, we know that:

$$\mathbb{P}[\neg A \cup \neg B] = \mathbb{P}[\neg A] + \mathbb{P}[\neg B] - \mathbb{P}[\neg A \cap \neg B]$$

$$\mathbb{P}[\neg A \cap \neg B] = \mathbb{P}[\neg A] + \mathbb{P}[\neg B] - \mathbb{P}[\neg A \cup \neg B]$$

$$\mathbb{P}[\neg A \cap \neg B] = \frac{2n(n-1)}{n^2} - \frac{n^2-1}{n^2} = \frac{n^2-2n+1}{n^2} = \frac{(n-1)^2}{n^2}$$

$$\implies \mathbb{P}[\neg A \cap \neg B] = \frac{(n-1)^2}{n^2} = \mathbb{P}[\neg A]\,\mathbb{P}[\neg B]$$

(3) Observe that $\mathbb{P}[\neg A \cap B] = \mathbb{P}[A \cap \neg B]$ since the random variables are identically distributed. We can use the fact that the sum of these $4$ events must equal $1$ to solve for $X = \mathbb{P}[\neg A \cap B] = \mathbb{P}[A \cap \neg B]$:

$$\mathbb{P}[A \cap B] + \mathbb{P}[\neg A \cap \neg B] + 2X = 1$$

$$\frac{1}{n^2} + \frac{(n-1)^2}{n^2} + 2X = 1$$

$$X = \frac{n-1}{n^2} = \mathbb{P}[\neg A]\,\mathbb{P}[B] = \mathbb{P}[A]\,\mathbb{P}[\neg B] \quad \square$$

(c) Just like the balls and bins case, we can express the load of a bin as a sum of indicators for each of the elements in $U$ which hash to the $j$th bin in $T$

$$Z_j = \sum_{e \in U} X_{e,j}$$

$$\mathbf{E}[Z_j] = \sum_{e \in U} \mathbf{E}[X_{e,j}] = \sum_{e \in U} \mathbb{P}[h(e) = j] = 1$$

where the last equality follows since there are $n$ elements and the probability any given element hashes to $j$ is $\frac{1}{n}$

$$\mathbf{Var}[Z_j] = \mathbf{E}\left[Z_j^2\right] - \mathbf{E}[Z_j]^2$$

$$Z_j^2 = \big(\sum_{e \in U} X_{e,j}\big)^2 = \sum_{i=1}^{n} X_i^2 + \sum_{i \neq j} X_i X_j$$

$$\mathbf{E}\left[Z_j^2\right] = \sum_{i=1}^{n} \mathbf{E}\left[X_i^2\right] + \sum_{i \neq j} \mathbf{E}\left[X_i\right] \mathbf{E}\left[X_j\right] = 1 + \frac{n(n-1)}{n^2} = 2 - \frac{1}{n}$$

The first expression is the definition of variance. The second is expanding out the square of the random load $Z_j$ into the squares and the cross terms. The third is applying linearity of expectation and noting that each of the cross term random variables are independent so their expectations factor. The final equality comes from the fact the the square of a Bernoulli random variable is itself and from the fact that that there a $n(n-1)$ cross terms in the second summation where each term is $\frac{1}{n^2}$.

$$\implies \mathbf{Var}\left[Z_j\right] = \mathbf{E}\left[Z_j^2\right] - \mathbf{E}\left[Z_j\right]^2 = 2 - \frac{1}{n} - 1 = 1 - \frac{1}{n}$$

(d) Let $Z = \max_j Z_j$ where $Z_j$ is the load on $T[j]$ after hashing all $n$ element from $D$ using a random $h \in H$. We want to show that:

$$\mathbb{P}\left[Z \leq \Theta_n\right] = \mathbb{P}\left[\bigcap_{i=1}^{n} Z_i \leq \Theta_n\right] \geq \frac{2}{3}$$

We will do this by upper bounding $\mathbb{P}\left[Z > \Theta_n\right]$:

$$\mathbb{P}\left[Z > \Theta_n\right] = \mathbb{P}\left[\bigcup_{i=1}^{n} Z_i > \Theta_n\right] \leq \sum_{i=1}^{n} \mathbb{P}\left[Z_i > \Theta_n\right] = n\mathbb{P}\left[Z_j > \Theta_n\right]$$

Where the second to last inequality follows from union bound and the final equality follows since each of the load random variables are identically distributed, where $j$ can be the load of any of the slots. Now we want to use the fact from part b that $Z_j$ can be expressed as the sum of pairwise independent indicators and we can apply a theorem from week $4$ which gives us a Chebyshev like bound:

$$Z_j = \sum_{e \in U} X_{e,j} \implies \mathbf{E}\left[Z_j\right] = 1$$

$$\mathbb{P}\left[|Z_j - 1| \geq c\right] \leq \frac{1}{c^2}$$

Note that since $Z_j$ only takes on integer values larger than $0$, if $c > 1$ then we can drop the absolute value signs:

$$\mathbb{P}\left[Z_j \geq c + 1\right] \leq \frac{1}{c^2}$$

Note that $Z_j$ only takes on positive integer values so the following equality holds:

$$\mathbb{P}\left[Z_j \geq c + 1\right] = \mathbb{P}\left[Z_j > c\right]$$

$$\implies \mathbb{P}\left[Z_j > c\right] \leq \frac{1}{c^2}$$

Let $c = \Theta_n$

$$\mathbb{P}[Z > \Theta_n] = n\mathbb{P}[Z_j > \Theta_n] \leq \frac{n}{\Theta_n^2} \leq \frac{1}{3}$$

$$\mathbb{P}[Z > \Theta_n] \leq \frac{1}{3} \implies \mathbb{P}[Z \leq \Theta_n] \geq \frac{2}{3}$$

where the last inequality holds if $\Theta_n \geq \sqrt{3n}$. □

**Problem 8: Another Finger Printing Approach.**

*Solution.*

(a) Alice sends a triplet $(p, r, \overline{a})$ over to bob where $p < 4n$, $r < p$, $\overline{a} < p \implies p, r, \overline{a} < 4n \implies 3\log_2 4n$ bits are needed since we need $\log_2 n$ bits to store the value $n \implies O(\log_2 n)$

(b) Assume that $\mathbf{a} \neq \mathbf{b}$.

$$\mathbb{P}[Error] = \mathbb{P}\left[\overline{a} = \overline{b}\right] = \mathbb{P}\left[\overline{a} - \overline{b} = 0\right]$$

$$\overline{a} = \sum_{i=1}^{n} a_i r^i \; mod \; p$$

$$\overline{b} = \sum_{i=1}^{n} b_i r^i \; mod \; p$$

$$\overline{a} - \overline{b} = \sum_{i=1}^{n} a_i r^i - b_i r^i \; mod \; p = \sum_{i=1}^{n} r^i(a_i - b_i) \; mod \; p = 0$$

Let $c_i = a_i - b_i$ so $\mathbf{c} = \mathbf{a} - \mathbf{b}$:

$$\overline{a} - \overline{b} = \sum_{i=1}^{n} c_i r^i \; mod \; p = 0$$

Thus, we have reduced the problem to showing that

$$\mathbb{P}\left[c_1 r + c_2 r^2 + ... + c_n r^n = 0 \; mod \; p\right]$$

where the randomness is taken from sampling $r \in \{0, 1, ..., p-1\}$ uar. From here, we can apply a theorem from algebra. We note that $\overline{a} - \overline{b}$ is a nonzero polynomial with degree at most $n$ since we assumed that $\mathbf{a} \neq \mathbf{b} \iff \exists\, a_k \neq b_k \implies c_k \neq 0$. We know that a polynomial of degree $n$ over a finite field has **at most** $n$ roots $\implies$ the above polynomial has at most $n$ values of $r \in \{0, 1, ..., p-1\}$ which satisfy the equation. Since we sample $r$ uar $\implies \mathbb{P}[\text{sample a solution}] \leq \frac{n}{p}$. Since $p > 2n \implies \frac{1}{p} < \frac{1}{2n}$:

$$\mathbb{P}[Error] = \mathbb{P}[\text{sample a solution}] \leq \frac{n}{p} \leq \frac{1}{2} \quad \Box$$