Problem #1:

Generalize the lower bound argument for the complexity of property testing for element distinctness to show that any algorithm that always accepts distinct inputs, but with probability at least 1/2 rejects all inputs with \(< (1 - \varepsilon)n\) distinct values requires \(\Omega(\sqrt{n/\varepsilon})\) samples.

Problem #2:

Consider the following deterministic algorithm which, unlike the Misra-Gries algorithm, computes an over-estimate, rather than an under-estimate of the heavy hitter frequencies.

Space-Saving Algorithm

1: Initialize: \(k \leftarrow \lceil 1/\varepsilon \rceil\)
2: \(A \leftarrow \emptyset\), \(A\) is a set of up to \(k\) pairs \((j, \tilde{f}_j)\).
3: Process:
4: for each \(i\) do
5: if \(x_i \in A\) then
6: \(\tilde{f}_{x_i} \leftarrow \tilde{f}_{x_i} + 1\)
7: else if \(|A| < k\) then
8: Add \(x_i\) to \(A\)
9: \(\tilde{f}_{x_i} \leftarrow 1\)
10: else
11: \(j' \leftarrow \arg\min_{j \in A} \tilde{f}_j\)
12: \(\tilde{f}_{x_i} \leftarrow \tilde{f}_{j'} + 1\)
13: Replace \((j', \tilde{f}_{j'})\) in \(A\) with \((x_i, \tilde{f}_{x_i})\)
14: end if
15: end for
16: Output: \(\tilde{f} \leftarrow A\)
17: \(\tilde{f}_j\) is as given for \(j \in A\), \(\tilde{f}_j = 0\) if \(j \notin A\).

Show that:

(a) For every \(j \in A\), \(\tilde{f}_j \geq f_j\).
(b) For every \(j \in [M]\), \(\tilde{f}_j \leq f_j + \tilde{f}_{\min}\) where \(\tilde{f}_{\min} = \{f_j : j \in A\}\).
(c) \(\sum_{j \in A} \tilde{f}_j = n\)
(d) $\tilde{f}_{\text{min}} \leq \lfloor n/k \rfloor$ and hence $f_j \leq \tilde{f}_j \leq f_j + \lfloor n/k \rfloor$ for every $j \in A$.

(e) For $i \leq k$, the $i$-th largest $\tilde{f}_j$ value is an upper bound on the $i$-th largest $f_j$ value (even though they might be for different values of $j$).

Problem #3:

This problem shows a tight relationship between the approximations of the Space-Saving algorithm above and the Misra-Gries algorithm when run using the same value of $k$:

Let $A^{MG}$ be the set of size at most $k - 1$ maintained during the execution of the Misra-Gries algorithm and $f^{MG}$ be the frequency values maintained by that algorithm, where $\tilde{f}_j^{MG} = 0$ for $j \notin A^{MG}$.

Similarly, define $A^{SS}$, the set of size up to $k$ maintained by the Space-Saving algorithm. Let $\text{min}^{SS}$ be the $k$-th largest frequency in $A^{SS}$ where $\text{min}^{SS} = 0$ if $|A^{SS}| < k$. Let $\tilde{f}^{SS}$ be the vector of frequency estimates maintained by the Space-Saving algorithm during its execution, where we consider $\tilde{f}_j^{SS} = \text{min}^{SS}$ if $j \notin A^{SS}$.

Finally, let $\text{sum}^{MG}$ be $\sum_{j=1}^{M} \tilde{f}_j^{MG}$.

Prove by induction on $n$ that after processing the same sequence of $n$ inputs,

$$\text{min}^{SS} = (n - \text{sum}^{MG})/k$$

and for every $j \in [M]$,

$$\tilde{f}_j^{SS} = \tilde{f}_j^{MG} + \text{min}^{SS}.$$

Problem #4:

Given two relations $R$ and $S$ with a common attribute, for query optimization it is useful to estimate the size of the join $R \bowtie S$ without actually executing it. If the frequency vectors for the attribute in the two relations are $f = (f_1, \ldots, f_M)$ and $g = (g_1, \ldots, g_M)$ then the number of tuples in that join is precisely $\sum_{j=1}^{M} f_j g_j = \langle f, g \rangle$. Design and analyze an algorithm based on the Tug-of-War Sketch for $F_2$ that maintains sketches for both $f$ and $g$ that provides a $1 \pm \epsilon$ factor approximation of the join size with probability at least $1 - \delta$.

Hint: Replace the use of $y^2$ in the Tug-of-War Sketch with a product of two different values.