Homework 7

Due: 6 March, 2026

Each problem is graded on the coarse scale of $\checkmark^+$, $\checkmark$, $\checkmark^-$ and no $\checkmark$. It is also assigned a multiplier, denoting the relative importance of the problem. Both correctness and presentation are grading criteria.

Please read and make sure you understand the collaboration policy on the course missive. Extra credit problems are clearly marked below (see course missive for the details of grade calculations).

Remember to prove all your (non-elementary and not shown in class) mathematical claims, unless stated otherwise.

Each group of students should submit only 1 pdf to the corresponding Canvas assignment.

Problem 1

(*a*) (1 $\checkmark$) Show that a k-wise independent hash family is also $k-1$-wise independent, for $k > 1$.

(*b*) (1 $\checkmark$) Given a streaming algorithm that estimates some parameter α to accuracy ϵ with probability at least $\frac{2}{3}$ using s bits of space, how would you construct a new streaming algorithm that estimates α to accuracy ϵ with probability at least $1-\delta$? What is the space complexity of your new algorithm? Credit will only be given for solutions with a reasonable dependence on δ.

Problem 2

(3 $\checkmark$s)

Complete the proof of Theorem 15.6 in class, that Algorithm 15.5 is a streaming algorithm for the Count-Distinct problem using space $O(\frac{1}{\epsilon^2}\log n)$. You should analyse Algorithm 15.5 as using a pairwise independent hash family instead of making the simplifying assumption of using the uniformly random function family.

Feel free to modify the algorithm slightly to handle edge cases, if necessary.

(Hint: the interesting general case is when $\epsilon \geq \frac{1}{\sqrt{n}}$.)

Problem 3

(3 $\checkmark$s)

Consider an adversarially ordered stream σ consisting of m distinct elements in $[n]$. Give a log-space streaming algorithm that returns an element x such that the quantile of x in the stream is in $\alpha \pm \epsilon$ with probability at least $\frac{2}{3}$, for known parameters $\epsilon > 0$ and $\alpha > 0$. You may assume that α and ϵ are constants independent of $m, n \to \infty$, and in particular that you do not need to worry about rounding/indexing issues related to taking quantiles of a finite set.

Problem 4

(3 ✓s, **Extra credit**)

Consider an arbitrary hash family $\mathcal{H} : U \to V$. In particular, we make absolutely no assumptions about $\mathcal{H}$, so no pairwise independence and no universality. Show that there must exist a pair of elements $x, y \in U$ such that

$$\underset{h \leftarrow \mathcal{H}}{\mathbb{P}}(h(x) = h(y)) \geq \frac{1}{|V|} - \frac{1}{|U|}$$

(Hint: what we learnt in the uniformity testing lecture might be helpful.)

Do 1 out of the 3 problems on this page. (This was what HW8 was going to be.)

Problem 5

(3 ✓s)

Write out and analyse the "count-median" sketch that is an adaptation of the count-min sketch, but instead can work in the turnstile model, sacrificing the one-sidedness of the approximation error.

Problem 6

(3 ✓s)

Definition 1 (ℓ_1-Heavy Hitters)**.** Given a stream σ with frequency vector $\mathbf{f}$, the ℓ_1-heavy hitters problem with parameter k is to return a list of elements L such that

- If $f_i > ||\mathbf{f}||_1/k$, then $i \in L$.
- $|L| = O(k)$.

Note that there are at most $k-1$ heavy hitters, and we require that the returned list is no more than a constant factor longer than necessary.

State and analyse a streaming algorithm, which, by maintaining the count-min sketch of a stream (possibly with sub-constant failure probability), can answer a single ℓ_1-heavy hitters query with parameter k in $\tilde{O}(k)$ time (and polylogarithmic space in n and m), with a success probability of at least 2/3.

Problem 7

(3 ✓s)

Recall that for two relations (i.e. tables in a database) $r(A, B)$ and $s(A, C)$, with a common attribute A, we define the join $r \bowtie s$ to be a relation consisting of all tuples (a, b, c) such that $(a, b) \in r$ and $(a, c) \in s$. Therefore, if $f_{r,j}$ and $f_{s,j}$ denote the frequencies of j in the common attribute (i.e. A) of r and s respectively, then the size of the join is $\sum_j f_{r,j} f_{s,j}$.

As mentioned in class, the second frequency moment F_2 of a stream can be interpreted as the size of a self-join, if the stream consists of records in a database, and the frequency f_i denotes the number of occurrences of element i in the join attribute.

Show that, using the same tug-of-war sketching algorithm, we can also estimate the size of the join of two arbitrary databases represented by streams σ_r and σ_s. For simplicity, you may assume that the streams have 2nd frequency moment (namely their squared Euclidean norms are the same). Explain how to compute such an estimate, and state and prove a theorem capturing the accuracy and space complexity guarantees of your (1-pass) streaming algorithm.

You should submit one answer to the following problem per person, in the same pdf.

Problem 8

(2 ✓s)

This is a non-technical problem. Throughout the semester, we have seen a number of models and algorithms, and even some lower bounds. What did you learn from the course? What were the biggest punchlines to you? Try to focus more on the big picture and less on each technical algorithm: how (if at all) has the course changed the way you think about computation? What kind of design/analysis/problem formulation principles have you learnt?

This problem is graded for completion (with reasonable effort) only. There is no right or wrong answer. It is a useful exercise to do though, for this course and really every course you take.

Also, feel free to discuss with your classmates (in addition to your homework partner)!