Homework 5

CSCI 4030 / CSCI 6220: Randomized Algorithms — Fall 2026

Rensselaer Polytechnic Institute — 100 points (CSCI 6220: 115 points)

Justify every claim. Acknowledge any collaborators and any external sources you consult, including AI tools, and describe clearly what role they played in your submission.

Throughout the homework, log denotes the natural logarithm and $\log_2$ denotes the base-2 logarithm. Ignore integer rounding when it does not affect the substance of an argument.

[30 points] Problem 1 (Growing Bloom filters).

A browser vendor distributes a compact local list of unsafe URLs. A URL on the list must never be declared safe, while a safe URL may occasionally be answered "maybe unsafe" and sent to a server for a definitive check.

The list grows over time, and its eventual size is not known when the service is deployed. Rebuilding one larger Bloom filter would require every browser to re-download the entire structure. Instead, the vendor uses a sequence of filters. Filter F_j has capacity

$$C_j = 2^j.$$

Keys are inserted into F_0 until it is full; it is then frozen, F_1 is created, and so on. A query checks every filter that has been created and answers "maybe unsafe" if any filter answers maybe.

For this problem, you may use the following asymptotic sizing rule for a tuned Bloom filter.

> **Bloom-filter sizing rule.** There are constants $0 < K_1 \le K_2$ such that a filter designed for capacity C and false-positive target $\eta \in (0, 1/2)$ uses between
>
> $$K_1 C \log \frac{1}{\eta} \qquad \text{and} \qquad K_2 C \log \frac{1}{\eta}$$
>
> bits. After at most C insertions it has no false negatives, and every fixed nonmember is a false positive with probability at most η.

Assume the inserted URLs are fixed independently of the hash functions. Bloom filters never clear bits, so once a nonmember becomes a false positive in a particular filter, it remains a false positive as that filter receives more insertions. Let δ_j be the target false-positive probability for F_j, and let $\delta \in (0, 1/2)$ be the desired overall error probability.

(a) (6 points) Prove that the cascade has no false negatives. For a fixed URL x that is never inserted, prove that over the entire lifetime of the data structure,

$$\mathbb{P}[x \text{ is ever answered "maybe unsafe"}] \le \sum_{j \ge 0} \delta_j.$$

(b) (12 points) Consider the geometric allocation

$$\delta_j = \frac{\delta}{2^{j+1}}.$$

Prove that its overall false-positive probability is at most δ. If $n \ge 1$ keys have been inserted, prove that the total space is

$$\Theta\left(n \left(\log \frac{1}{\delta} + \log(n+1) \right) \right).$$

You may use that if L is the index of the newest filter, then $2^L = \Theta(n)$ and $\sum_{j=0}^{L} 2^j = \Theta(n)$.

(c) (12 points) Consider instead

$$\delta_j = \frac{6\delta}{\pi^2(j+1)^2}, \qquad \sum_{r=1}^{\infty} \frac{1}{r^2} = \frac{\pi^2}{6}.$$

Prove that the overall false-positive probability is still at most δ and that after $n \geq 1$ insertions the total space is

$$O\left(n\left(\log\frac{1}{\delta} + \log\log(n+2)\right)\right).$$

Then, in one or two sentences, explain why the polynomial allocation uses less space than the geometric one. Your answer should refer to where most of the n keys are stored and how much of the error budget each allocation leaves for those filters.

Remark. Pagh, Segev, and Wieder (FOCS 2013) showed that, in the relevant regime, dynamic approximate membership without prior knowledge of n requires an additional $\Omega(n \log\log n)$ bits, and gave a matching construction.

CSCI 6220 extension (15 points): Bloom filters with universal hashing.

Lecture 9 analyzed Bloom filters under ideal independent uniform hashing, which leads to the familiar optimized space of approximately

$$1.44\log_2(1/\eta)$$

bits per stored key for false-positive probability η. Here we ask what can be guaranteed using only universal hash families.

Consider a *partitioned Bloom filter*. Divide the m bits into k disjoint arrays of length

$$p = \frac{m}{k}.$$

Hash function $h_i : U \to [p]$ writes only to array i. A query returns "maybe present" exactly when all k queried positions are set. The stored set S, with $|S| = n$, and the nonmember query $x \notin S$ are fixed independently of the hash functions.

(i) (7 points) Let $h_1, \ldots, h_k$ be chosen independently, with each h_i drawn from a universal family:

$$\mathbb{P}[h_i(x) = h_i(y)] \leq \frac{1}{p} \qquad (x \neq y).$$

Prove

$$\mathbb{P}[\text{false positive}] \leq \left(\frac{n}{p}\right)^k = \left(\frac{kn}{m}\right)^k.$$

Use only universality of each h_i and independence across the k hash functions.

(ii) (4 points) Write $m = cn$ and optimize the bound from part (i) over k, treating k as a positive real and ignoring rounding. What space per stored key is sufficient to achieve false-positive probability η? Compare the resulting constant with the ideal Bloom-filter constant

$$\frac{1}{\log 2}\log_2\frac{1}{\eta} \approx 1.44\log_2\frac{1}{\eta}.$$

(iii) (4 points) Is the loss in part (ii) an artifact of a loose analysis? Let $U = \mathbb{Z}_p \times \{0, 1\}$, and consider the family of functions

$$h_{s,t}(a, 0) = a + s \bmod p, \qquad h_{s,t}(a, 1) = a + t \bmod p,$$

with s, t independent and uniform on $\mathbb{Z}_p$. Show that this family is universal. Take $S \subseteq \mathbb{Z}_p \times \{0\}$ with $|S| = n \leq p$ and $x \in \mathbb{Z}_p \times \{1\}$, and draw each h_i independently from this family. Show that the false-positive probability equals $(n/p)^k$ exactly, so the bound in part (i) cannot be improved using universality alone.

Under ideal hashing, the probability that x's position in one array is set would be $1 - (1 - 1/p)^n < n/p$. In one sentence, explain why this universal family produces more false positives than ideal hashing in this example.

[34 points] Problem 2 (Detecting concentrated repetition from hash collisions).

A web service receives a stream of requests tagged by identifiers such as clients, resources, or sessions. We want a small summary that can flag traffic concentrated on a few identifiers without storing every identifier seen.

Let the insertion-only stream be

$$x_1, x_2, \ldots, x_N \in U,$$

where $N \geq 2$, and let $\delta \in (0, 1/2)$ be a target error probability. Assume the stream is fixed independently of the hash function. Let f_x be the number of occurrences of key x. Define

$$D = \sum_{x \in U} \binom{f_x}{2},$$

the number of unordered pairs of stream positions containing the same key. Normalized, $D/\binom{N}{2}$ is the probability that two uniformly chosen distinct stream positions contain the same key.

Choose a random hash function $h : U \to [m]$ from a universal family and maintain only the bucket counts

$$B_b = |\{t : h(x_t) = b\}|.$$

At the end compute

$$C = \sum_{b=1}^{m} \binom{B_b}{2}.$$

Thus C counts all pairs of stream positions whose keys land in the same bucket.

For example, for the stream

$$a, a, a, b, c, c, d,$$

we have $D = 4$. If $h(a) = 1$, $h(b) = 2$, and $h(c) = h(d) = 3$, then the bucket loads are $(3, 1, 3)$ and $C = 6$: four collisions are genuine repetitions and two come from the distinct keys c and d hashing together.

(a) (8 points) Decompose the statistic into

$$C = D + X,$$

where X counts collisions between unequal keys. Use universality and linearity of expectation to prove

$$\mathbb{E}[X] \leq \frac{1}{m}\left(\binom{N}{2} - D\right) \leq \frac{1}{m}\binom{N}{2}.$$

(b) (16 points) Fix $\alpha \in (0, 1/2)$ and suppose the stream satisfies one of the promises

$$\text{LOW: } D \leq \alpha\binom{N}{2}, \qquad \text{HIGH: } D \geq 2\alpha\binom{N}{2}.$$

Run r independent copies of the summary, each with

$$m = \left\lceil \frac{2}{\alpha} \right\rceil$$

buckets, and declare HIGH exactly when

$$\min_{1\le i\le r} C^{(i)} \ge 2\alpha\binom{N}{2}.$$

Prove that HIGH streams are always accepted. Show that choosing

$$r = \left\lceil \log_2 \frac{1}{\delta} \right\rceil$$

makes the error probability on LOW streams at most δ, using $O(\alpha^{-1}\log(1/\delta))$ counters.

(c) (10 points) Relate D to concentration on the most frequent key. Let $f_{\max} = \max_x f_x$. Prove

$$\frac{D}{\binom{N}{2}} \le \frac{f_{\max} - 1}{N - 1},$$

and use this to show that a HIGH stream contains a key occurring at least $1 + 2\alpha(N-1)$ times.

Now let N be even and consider a stream in which $N/2$ distinct keys each occur exactly twice. Compute D and show that, for every fixed α, this stream satisfies the LOW promise once $N > 1 + 1/\alpha$. Explain what this example shows about the kind of repetition detected by $D/\binom{N}{2}$.

[36 points] Problem 3 (Heavy hitters with Misra–Gries and Count–Min).

A web service sees a stream of requests for URLs from a large universe U. Let f_x be the number of requests for URL x and

$$N = \sum_x f_x.$$

Given $0 < \varepsilon < \phi < 1$, we want to report every URL with $f_x > \phi N$, report no URL with $f_x \leq (\phi - \varepsilon)N$, and give an additive-$\varepsilon N$ estimate for every reported URL. We call this the *gray-zone guarantee*: URLs with $(\phi - \varepsilon)N < f_x \leq \phi N$ may or may not be reported.

Count–Min gives accurate estimates for keys we already know to query, but U is far too large to query every possible URL. Misra–Gries instead keeps a small set of candidate keys. Run the two summaries on the same insertion-only stream: Misra–Gries supplies the candidates, and Count–Min estimates their frequencies.

1. Misra–Gries with $k = \lceil 1/\phi \rceil$ slots. Let S be the set of keys remaining at the end.
2. Count–Min with width $w = \lceil 3/\varepsilon \rceil$ and depth d.
3. Query Count–Min only on S. Report each $x \in S$ with $\widehat{f}_x > \phi N$, together with its estimate $\widehat{f}_x$.

You may use the following guarantees proved in Lecture 10. For Misra–Gries with k slots, let $\widetilde{f}_x$ be the counter stored for x, with the convention $\widetilde{f}_x = 0$ if x is not stored at the end. Then, simultaneously for all keys,

$$f_x - \frac{N}{k+1} \leq \widetilde{f}_x \leq f_x.$$

For Count–Min with width $w = \lceil 3/\varepsilon \rceil$ and depth d,

$$\mathbb{P}[f_x \leq \widehat{f}_x \leq f_x + \varepsilon N] \geq 1 - 3^{-d}$$

for each fixed key x. Assume the stream is fixed independently of the Count–Min hash functions.

(a) (8 points) Prove that every key with $f_x > \phi N$ belongs to S, and that $|S| \leq k$.

(b) (16 points) Choose d so that, with probability at least $1-\delta$, the algorithm satisfies the gray-zone guarantee, and prove that it does.

Your proof will union-bound over the keys in S, a set that was not known before the stream began. Explain why this is nevertheless legitimate.

(c) (12 points) Misra–Gries alone can also satisfy the full gray-zone and estimation guarantee deterministically. Run it with $k' = \lceil 1/\varepsilon \rceil$ slots and report the stored keys whose final Misra–Gries counts exceed $(\phi - \varepsilon)N$. First verify that this gives additive-εN estimates and the same gray-zone guarantee.

Now compare its memory with the hybrid for

$$\phi = 10^{-2}, \qquad \varepsilon = 10^{-4}, \qquad \delta = 10^{-3}.$$

Assume that storing a URL identifier costs 2048 bits and each counter costs 32 bits; ignore the space needed to describe the hash functions. Using your value of d from part (b), compute the

total number of bits used by each approach and determine which is smaller. Explain why the hybrid can save space when $\varepsilon \ll \phi$ and item identifiers are much more expensive than counters.