Asmt 5: Frequent Items

Turn in through **Gradescope** by Wednesday, March 1 at 2:45pm, then come to class:
100 points

Overview

In this assignment, you will explore finding frequent items in data sets, with emphasis on streaming techniques designed to work at an enormous scale. For simplicity, you will work on more manageably sized data sets, and simulate the stream by just processing with a for loop.

You will use two data sets for this assignment, `S1.txt` and `S2.txt` available on Canvas. The first data set S1 has a set of $m = 3{,}000{,}000$ characters, and the second one S2 has $m = 4{,}000{,}000$ characters. The order of the file represents the order of the stream.

Note: Homework assignments are intended to help you learn the course material, and successfully solve mid-term and final exams that will be done on paper in person.

As usual, it is recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. The latex source for this homework is available on Canvas. I recommend that you drop the two latex files (.tex and .sty) in a new Overleaf project, and compile/edit the .tex file there.

Submissions that are not uploaded on Gradescope will get 10% penalty.

1 Streaming Algorithms

A (40 points): Run the Misra-Gries Algorithm (see **L11.3.1**) with $(k-1) = 11$ counters on streams S1 and S2. Report the output of the counters at the end of the stream. In addition to each counter report the estimated ratio of each label using the estimated count/m.

In each stream, use just the counters to report which characters *might* occur at least 25% of the time (if any), and which must occur at least 25% of the time (if any).

B (40 points): Build a Count-Min Sketch (see **L12.1.1**) with $k = 12$ counters using $t = 6$ hash functions. Run it on streams S1 and S2.

For both streams, report the estimated counts for characters `a`, `b`, and `c`. In addition to each counter report the estimated ratio of each of these labels using the estimated count/m. Just from the output of the sketch, with probably $1 - \delta = 63/64$ (that is assuming the randomness in the algorithm does not do something bad), which objects *might* occur at least 25% of the time (if any), and which objects *must* occur at least 25% of the time (if any).

C (10 points): How would your implementation of these algorithms need to change (to answer the same questions) if each object of the stream was a "word" seen on Twitter, and the stream contained all tweets concatenated together?

D (10 points): Describe one advantage of the Count-Min Sketch over the Misra-Gries Algorithm.

2 BONUS

The exact heavy-hitter problem is as follows: return *all* objects that occur more than ϕ percent of the time. It cannot return any false positives or any false negatives. In the streaming setting, this requires $\Omega(\min\{m, n\})$ space if there are n objects that can occur and the stream is of length m.

A: (1 point) A 2-Pass streaming algorithm is one that is able to read all of the data in-order exactly twice, but still only has limited memory. Describe a small space algorithm to solve the exact heavy hitter problem, i.e., with $\varepsilon = 0$, (say for $\phi = 15\%$ threshold).

B: (2 points) Provide an upper bound for the probability that at least one student gets a really bad estimate for the CountMin Sketch where the count estimate for `a`, `b`, and `c` are all the same since they always collide. As in the question 1.B use $k = 12$ and $t = 6$, and assume perfect hash functions. And take into account that there are $n = 80$ students in the class, and this is a problem if it happens to any one of them.