Randomized Algorithms (RA-MIRI): Assignment #3

1 Statement

In this programming assignment, you will have to study experimentally the performance of two different cardinality estimation algorithms.

You should write programs for HyperLogLog (HLL) and for KMV (K Minimum Values). You should also write routines which allow you to generate synthetic data streams $Z = z_1, z_2, \ldots, z_N$, for instance, following a Zipfian law of parameter $\alpha \geq 0$ for n distinct elements $\{x_1, \ldots, x_n\}$ in which

$$\mathbb{P}\{z_j = x_i\} = \frac{c_n}{i^\alpha}, \qquad 1 \leq j \leq N, \quad 1 \leq i \leq n,$$

and

$$c_n = \frac{1}{\sum_{1 \leq i \leq n} i^{-\alpha}}.$$

You will try your programs with the given datasets (https://mydisk.cs.upc.edu/s/fDZanwDA8So8My8) as well as with synthetic data.

For the datasets, prepare a table that compares HLL, KMV and the true cardinalities. Remember that these cardinality estimation algorithms are randomized, hence you should run each several times to get estimates of their expected performance. Both algorithms rely upon using good hash functions. This implementation of a family of random hash functions by J. Lumbroso can be of help for your work (https://github.com/jlumbroso/python-random-hash); it's in Python (also in Java), and you can easily adapt to C++ if you prefer. Other sites worth looking at might be https://xxhash.com/ or https://github.com/lemire/StronglyUniversalStringHashing/.

You should also study how the amount of memory (number of counters m in HyperLogLog, number of elements K kept in KMV) impacts the quality of the estimations; do this for dataset `D1.dat` at least, maybe for others. Check that the standard error behaves as predicted by the theory.

The same kind of experiments can be conducted using synthetic data streams. They allow you to do the same type of experiments, but now with total control on your hands of the length N and number of distinct elements n. You will notice that the value of α makes little difference—but you can conduct some small experiment just to confirm this hypothesis!

Do not forget to include in your report some details about your choice of hash functions and all references that you have used for this assignment. Make sure to include proper citation to the references in the main body of your report: the bibliography section is not just an isolated collection of references that appear at the end of your report.

Bonus: Adding more cardinality estimation algorithms (e.g., Probabilistic Counting-(PCSA), Recordinality, MinCount, Adaptive Sampling, variants of HLL, ...) to the comparative study.

2 Instructions to deliver your work

Submit your work using the FIB-Racó. The deadline for submission is December 10th, 2023 at 23:59. It must consist of a zip or tar file containing all your source code, auxiliary files and your report in PDF format. Include a README file that briefly describes the contents of the zip/tar file and gives instructions on how to produce the executable program(s) used and how to reproduce the experiments. The PDF file with your report must be called `YourLastName_YourFirstName-3.pdf`,

N.B. I encourage you to use LaTeX to prepare your report. For the plots, you can use any of the multiple packages that LaTeX has (in particular, the bundle TikZ+PGF) or use independent software such as matplotlib and then include the images/PDF plots thus generated into your document.