Bachelor Thesis Review

Faculty of Mathematics and Physics, Charles University

Thesis author	Vojtěch Gaďurek
Thesis title	Data Structures for Sketching Dynamic Sets
Year submitted	2024
Study program	Computer Science

Review author	Pavel Veselý	Advisor
Department	Computer Science Institute of Charles University	

The thesis focuses on one-pass streaming algorithms for processing large sets so that one can extract the symmetric difference between two sets with memory proportional to the number of elements in the symmetric difference, without remembering the original sets. It describes the well-known Invertible Bloom Lookup Table (IBLT, Goodrich & Mitzenmacher '11) and its new variant, abbreviated HPW (Houen, Pagh, & Walzer '23). IBLTs are basically hash tables where, to save memory, collisions are handled by taking sums but every element is hashed into several buckets.

Implementation. These two algorithms are implemented in C#, which brings modularity such as a seamless change of the hash function. Implementing it naively would be way too slow, but Vojtěch worked hard on optimizing the code while retaining modularity. The results are quite impressive: 50-200 million processed elements per second on a 4-core processor. The main advanced C# tool is Expression trees, for which he wrote a wrapper library LittleSharp that deals with some deficiencies of Expression trees by providing a certain type safety and making the code easier to read.

On top of LittleSharp, Vojtěch created three more libraries, all available at GitHub:

- FlashHash providing fast implementation of standard hash functions used in data structures, including IBLTs,
- RedaFasta for reading files with so-called masked superstrings of k-mers in the FASTA format used in bioinformatics, and
- SymmetricDifferenceFinder for experiments with the IBLT and HPW data structures.

Overall, four C# libraries are a great outcome of the project, despite that they still need some improvements as described in the thesis. Namely, FlashHash and LittleSharp may be of interest beyond their application to implement IBLTs. The code is well-organized and includes unit tests. Finally, I emphasize that the implementation was done by Vojtěch with almost no supervision from me.

Experimental evaluation. The experimental results provide a useful insight into IBLTs. First, four standard hash functions are compared with respect to the probability of recovering the symmetric difference of a given size. Tabulation and MultiplyShift are the best, achieving high probability recovery even for 80% full table.

Another substantial contribution is a so-called recovery with an oracle, which is useful for structured data such as a collection of many short integer sequences or k-mers of genomes in bioninformatics. For such datasets, it is possible to guess the next element of the encoded set given some already recovered elements. Vojtěch utilized this observation and designed a recovery procedure which, surprisingly, recovers a structured set even if its size is by 50% larger than the number of buckets of the IBLT! That is, for structured data, the thesis demonstrates the possibility to go significantly beyond the worst-case lower bound of 100%, meaning that, in general, the IBLTs need to be larger than the set size.

Finally, the performance of the four hash functions and of the IBLTs is analyzed, showing that one can perform up to 50-200 millions of updates per second (depending on the IBLT size), which is a nice result given that it runs on the .NET framework.

Thesis text. The thesis is clearly organized into eight chapters, which demonstrates the breadth of the work, spanning both advanced randomized algorithms (IBLTs) and optimizing performance of C# code. The text would still benefit from some improvements as there remains a number of typos and grammar mistakes, and a couple of imprecise statements (e.g., in Section 1.5). The parts about algorithms (Chapters 1 and 2) clearly introduce streaming algorithms and IBLTs, though there are a few digressions, such as an introduction to the Morris counter, that are interesting but not the main focus of the thesis. The parts about C# and the four libraries (Chapters 3-7) are accompanied with a lot of examples, which makes the introduced concepts more understandable.

Overall, the contributions of this thesis are interesting, and have a potential to be extended into a paper for a conference/workshop on experimental evaluation of algorithms. Therefore, I suggest grade 1.

Date Signature