

INFO-F-413 Data structures & Algorithms - Bloom filters

Contents

1. [Introduction](#)
2. [Technologies](#)
3. [Content](#)
4. [Program usage](#)
5. [Contributor](#)

Introduction

This project is about solving the problem of the membership of an element in a set. To achieve such a task, a well-known, space and time efficient data structure exists. This structure is called "Bloom filter". Unfortunately, this structure has a false positive rate. This project covers the implementation of the structure as well as the theoretic concepts behind the false positive rate and the practical experimentations that confirms those theoretic aspects.

This project has been done within the framework of the "INFO-F-413 - Data structures & Algorithms" course of the ULB (Université Libre de Bruxelles) Computer Science Master's Degree.

Technologies

The technologies used to implement those algorithms are all .NET based. C# (.NET 6) has been used for implementation and a Powershell script has been created for testing purposes.

Content

This repository contains all the files of a traditional C# project as well as a PDF report for the theory about bloom filters as well as the analysis of the false positive rate. For details, please refer to that report. All the code is documented with conventional XML "summary" notation.

The program runs tests on to understand the evolution of false positive rate. For a program run, a bloom filter is instanciated and filled with n random values. After that, 150 values which are all negative are tested. The false positives are added to a counter. This test is realized 500 times. At the end of the 500 tests, the rate of false positives is computed (see PDF report for more details).

The goal of this project is simple and purely educative to illustrate theoretical concepts seen during my lectures and paper readings.

Program usage

The program can be compiled with a traditional C#.NET compiler. In order to use the program, it must be followed by at least 3 arguments, which are respectively :

- The number of hash functions (k)
- The size of the bit array (m)
- The number of elements to insert into the structure.

A last, optional, argument is "--verbose" to display all the details about the program run.

Example of command to execute a false positive rate test on a bloom filter with 10 hash functions, a bit array of size 1000 and 100 elements inserted in the structure, with all the logs enabled :

```
./Assignment2.exe 10 1000 100 --verbose
```

At the end, the program outputs the number of false positives, the false positive rate as well as the expected (theoretical) false positive rate.

Note that a Powershell test script is available in the Assignment2/Tests folder.

Contributor

This project has been made by Dudziak Thomas, ULB student (ULBId : 542286).