CIS6930 Network Data Streaming

Project 2 Individual Project

Implementation of Bloom Filters

1. **Description**

In this project, you will implement Bloom filter, counting Bloom filter, and coded Bloom filter.

Bloom Filter

Input: number of elements to be encoded, number of bits in the filter, number of hashes – for demo, they are 1,000, 10,000 and 7, respectively

Function: generate 1,000 elements (denoted as set A) randomly, encode them in the filter, look up them in the filter, and generate another 1.000 elements randomly (denoted as set B) and look up them in the filter.

Output: (1) the first line of the output: After lookup of elements in A, what is the number of elements you find in the filter?

(2) the second line of the output: After lookup of elements in B, what is the number of elements you find in the filter?

Counting Bloom Filter

Input: number of elements to be encoded initially, number of elements to be removed, number of elements to be added, number of counters in the filter, number of hashes – for demo, they are 1,000, 500, 500, 10,000 and 7, respectively

Function: generate 1,000 elements (denoted A) randomly, encode them in the filter, remove 500 elements in A from the filter, add other 500 randomly generated elements in the filter, and look up all original elements from A in the filter.

Output: After lookup of elements in A, what is the number of elements you find in the filter?

Coded Bloom filter

Input: number of sets, number of elements in each set, number of filters, number of bits in each filter, number of hashes – for demo, they are 7, 1000, 3, 30,000, and 7 respectively

Function: generate 7 sets of 1000 elements each, their codes are 001 through 111 respectively, encode all sets in 3 filters according to the algorithm, perform lookup on all elements in the 7 sets

Output: number of elements whose lookup results are correct

How to implement multiple hash functions?

For k hash functions, you may define an integer array s[k] and initialize the array with randomly generated numbers. When computing Hi(f), you may compute H(f XOR s[i]), where XOR is a bitwise operator with 0 XOR 0 = 0, 1 XOR 0 = 1, 0 XOR 1 = 1, and 1 XOR 1 = 0.

2. Dates

Handout: 9/19/2020
Due in Canvas: 10/19/2020

3. Programming Environment

Programming language: Java, C, C++, C#, Python
Operating System: Windows, Mac OS or Linux
Programming Tool: Eclipse, IntelliJ, Jcreator, Kawa, Netbeans, … whatever you like.

To use Eclipse, please go through the following list:

1. Download JDK from: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

2. Download Eclipse from: http://www.eclipse.org/downloads/

3. Here is a link for eclipse tutorial:
http://eclipsetutorial.sourceforge.net/totalbeginner.html

4. Here is a tutorial for socket programming in Java:
http://java.sun.com/docs/books/tutorial/networking/sockets/

4. Code Submission

You must submit the source code and one output file for each filter with the demo input parameters given earlier. Name the output file after the filter performed.

Include readme.txt to explain your files.

Submit the project through Canvas:

1) Go to https://elearning.ufl.edu/
2) Click “Login to e-Learning”
3) Login with your gator link username/password
4) Go in CIS6930 Network Data Streaming
5) Click “Assignments” and submit your project

This is an **individual** project. We will run an automatic tool to catch submissions with identical or similar code. There will be no late submissions.