Assignment 2: Similar Items, Data Streams, PageRank

Formative, Weight (10%), Learning objectives $(1, 2, 3)$,
Abstraction (4), Design (4), Communication (4), Data (5), Programming (5)

Due date: $11 : 59$ **pm,** 30 **April,** 2021

1 Overview

This assignment should be done in groups consisting of **TWO** students. Please use *A2-groups* on MyUni to organise yourselves into groups. The group members can be the same as assignment 1 or can be different; you MUST organise a group in *A2-groups* regardless. If you have problems/questions regarding grouping or require assistance, please contact the teaching assistant Mahdi (mahdi.kazemimoghaddam@adelaide.edu.au).

2 Assignment

Exercise 1 *S-curve (exercise 3.4.1 in Leskovec, Rajaraman and Ullman) (5+5+5 points)*

Evaluate the S-curve $1 - (1 - s^r)^b$ for $s = 0.1, 0.2, ...0.9$, for the following values of r and b;

1. r=3 and b=10.
2. r=6 and b=20.
3. r=5 and b=50.

Exercise 2 *Filtering Streams (similar to Exercises of 4.3 in Leskovec, Rajaraman and Ullman) (8 + 8 points)*

1. For the situation of our running example of Section 4.3.1 with changed conditions (10 billion bits, 2 billion members of the set S), calculate the

false-positive rate when using three hash functions. Do the same for four hash functions.

2. As a function of n, the number of bits and m the number of members in the set S, what number of hash functions minimizes the false-positive rate?

Exercise 3 *PageRank (22+13 points)*

1. Implement the PageRank Algorithm as discussed in Section 5.1 and 5.2 (Leskovec, Rajaraman and Ullman) in JAVA, Python or C++. Your implementation should make use of the improvements regarding efficiency and the methods of dealing with dead-ends and spider traps. There are several PageRank implementations available on the web. You have to do your own implementation without using any code from other sources.

2. Run your algorithm on the Google Web Graph 2002 available at

   http://snap.stanford.edu/data/web-Google.html

   and provide a file listing the PageRank for each node. Report separately, the ordered list of the ten nodes having the largest PageRank

Your approach should be efficient as possible in terms of runtime and memory requirements.

Exercise 4 *Data streams (7 + 7 points)*

Follow the scenario 1 and 2 below and answer the related questions regarding the Flajolet-Martin Algorithm. The hash functions are of the form $h(x) = ax+b$ mod 32 for some a and b. You should treat the result as a 5-bit binary integer

1. Scenario 1: Suppose a data stream consists of the integers 3, 1, 4, 6, 5, 9.

   Determine (a) the maximum tail length for each stream element and (b) the resulting estimate of the number of distinct elements for the hash functions in Question 1-3 below.

   – Question 1: Hash function: $h(x) = (2x + 1) \bmod 32$
   – Question 2: Hash function: $h(x) = (3x + 7) \bmod 32$
   – Question 3: Hash function: $h(x) = 4x \bmod 32$

2. Scenario 2: Suppose a data stream consists of the integers 4, 5, 6, 7, 10, 15.

   Determine (a) the maximum tail length for each stream element and (b) the resulting estimate of the number of distinct elements for the hash functions in Question 4-6 below.

   - Question 4: Hash function: $h(x) = (6x + 2) \bmod 32$
   - Question 5: Hash function: $h(x) = (2x + 5) \bmod 32$
   - Question 6: Hash function: $h(x) = 2x \bmod 32$

Exercise 5 *Summary of 3.6 and 3.7 (10 +10 points)* ***(Postgraduate Students (COMP SCI 7306) only)***

For this exercise you have to read Section 3.6 and 3.7 in Leskovec, Rajaraman, Ullman (third edition, 2020).

1. Summarize the content of 3.6 in your own words (600 words).

2. Summarize the content of 3.7 in your own words (600 words).

3 Procedure for handing in the assignment

Work should be handed in using MyUni. The submission should include:

- PDF file of your solutions for theoretical assignments. The solutions should contain detailed description of how to obtain the result.
- All source files, all the project files.
- PDF or txt file with descriptions of your implementations to understand your code.
- Files containing the results of your algorithms on the benchmark sets.
- PDF or txt file of your computation times of the algorithms on benchmark sets.
- For Exercise 4: input and output files, PDF or txt file with the calculations used to obtain your answers.
- A README.txt file containing instructions to run the code, the names, student numbers, and email addresses of the group members.
- The names, student numbers, and email addresses of the group members