Assignment 1: Basics and Map-Reduce

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

Due date: 11:59pm, 28 March, 2018, Weight: 10.0 % of the course

1 Overview

Assignments should be done in groups consisting of two students. If you have problems finding a group partner use the forum to search for group partners or contact the lecturer.

2 Assignment

Exercise 1 *Suspected Pairs (10 points)*

Using the information from the first lecture (or Section 1.2.3 in the textbook), what would be the number of suspected pairs if the following changes were made to the data (all changes should be applied at once).

1. The number of days of observation was raised to 5000.
2. The number of people observed was raised to 5 billion (and there were therefore 500,000 hotels).
3. We only reported a pair as suspect if they were at the same hotel at the same time on four different days.

Exercise 2 *Hadoop (10+10 points)*

For this exercise, you have to set up and configure your system to use Hadoop. Follow the instructions in Stanford document at http://snap.stanford.edu/class/cs246-2017/homeworks/hw0/tutorialv3.pdf and set up the virtual machine as described in Section 1. Run the example program of Section 2 and carry out the different steps given in that section.

- Write your own Hadoop Map-Reduce job that outputs the number of words that start with each letter (see Sections 2.5 and 3 of the Stanford document).
- Run your job on the file http://www.gutenberg.org/files/100/100-0.txt in *standalone mode* and *pseudo-distributed mode* and record the output.

Exercise 3 *Friend Recommendation System (Stanford) (35 points)*

Write a MapReduce program in Hadoop that implements a simple People You Might Know social network friendship recommendation algorithm. The key idea is that if two people have a lot of mutual friends, then the system should recommend that they connect with each other. You have to run the program on the system setup in Exercise 2 in order to receive points for this exercise.

Input: Download the input file from the link: http://snap.stanford.edu/class/cs246-data/hw1q1.zip. The input file contains the adjacency list and has multiple lines in the following format:

```
<User><TAB><Friends>
```

Here, `<User>` is a unique integer ID corresponding to a unique user and `<Friends>` is a comma separated list of unique IDs corresponding to the friends of the user with the unique ID `<User>`. Note that the friendships are mutual (i.e., edges are undirected): if A is friend with B then B is also friend with A. Algorithm: Let us use a simple algorithm such that, for each user U, the algorithm recommends N = 10 users who are not already friends with U, but have the most number of mutual friends in common with U.

Output: The output should contain one line per user in the following format:

```
<User><TAB><Recommendations>
```

where `<User>` is a unique ID corresponding to a user and `<Recommendations>` is a comma separated list of unique IDs corresponding to the algorithms recommendation of people that `<User>` might know, ordered in decreasing number of mutual friends. Even if a user has less than 10 second-degree friends, output all of them in decreasing order of the number of mutual friends. If there are recommended users with the same number of mutual friends, then output those user IDs in numerically ascending order. Also, please provide a description of how you are going to use MapReduce jobs to solve this problem. Do not write more than 3 to 4 sentences for this: we only want a very high-level description of your strategy to tackle this problem. Note: It is possible to solve this question with a single MapReduce job. But if your solution requires multiple map reduce jobs, then that is fine too.

For your submission

- Include your source code
- Include in your writeup a short paragraph describing your algorithm to tackle this problem.
- Include in your writeup the recommendations for the users with following user IDs: 924, 8941, 8942, 9019, 9020, 9021, 9022, 9990, 9992, 9993.

Exercise 4 *MapReduce (15 points)*

This exercise has 4 parts. In this exercise, you will be writing and implementing two MapReduce programs. Both are a bit challenging, but they will help you to have a better understanding about the MapReduce implementation. After you write the programs, you will need to answer some questions about them.

Remember that neither problem is case sensitive, so transform words to lowercase or uppercase. Also remember to use the StringTokenizer to find the correct answers.

Part 1: Write a program that processes the `FirstInputFile` http://www.gutenberg.org/cache/epub/100/pg100.txt and the `SecondInputFile` http://www.gutenberg.org/files/3399/3399.txt. This program should count the number of words with a specific amount of letters in these files - for example, the number of words with 4 letters, 5 letters and so on. If one word is repeated 20 times in the text, count it individually 20 times.

Part 2: Answer Questions 1-6.

- Q1: How many words are there with length 10 in `FirstInputFile`?
- Q2: How many words are there with length 4 in `FirstInputFile`?
- Q3: What is the longest length between words and what is its frequency in `FirstInputFile`?
- Q4: How many words are there with length 2 in `SecondInputFile`?
- Q5: How many words are there with length 5 in `SecondInputFile`?
- Q6: What is the most frequent length and what is its frequency in `SecondInputFile`?

Part 3: Write a second program that again processes the `FirstInputFile` http://www.gutenberg.org/cache/epub/100/pg100.txt and the `SecondInputFile` http://www.gutenberg.org/files/3399/3399.txt. However, in addition to counting the number of words with a specific amount of letters, if one word is repeated several times, count it only once. So, your output should be the frequency of words with same length, but count a repeated word only once. Note: You may need to use 2 MapReduce jobs.

Part 4: Answer Questions 7-12.

- Q7: How many words are there with length 10 in `FirstInputFile`?
- Q8: How many words are there with length 4 in `FirstInputFile`?
- Q9: What is the most frequent length and what is its frequency in `FirstInputFile`?
- Q10: How many words are there with length 5 in `SecondInputFile`?
- Q11: How many words are there with length 2 in `SecondInputFile`?
- Q12: What is the second-most frequent length and what is its frequency in `SecondInputFile`?

Exercise 5 *Summary of 2.4 and 2.5 (10 +10 points) (**Postgraduate Students (COMP SCI 7306) only**)*

For this exercise you have to read Section 2.3.9-2.3.11, 2.4, and 2.5 in Leskovec, Rajaraman, Ullman (second edition, 2014).

- Summarize the content of 2.4 in your own words (600 words).
- Summarize the content of 2.5 in your own words (600 words).

3 Procedure for handing in the assignment

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

- pdf file of your solutions for theoretical assignments.
- all source files
- descriptions as required in the statement of the exercises
- Hadoop outputs for the exercises
- a README.txt file containing instructions to run the code, the names, student numbers, and email addresses of the group members, only one submission per group.

In addition, there will be a discussion session where you have to explain your solutions.