CES Data Scientist

Mauro Sozio
sozio@telecom-paristech.fr

Exercise 1. (PageRank, 5 pts) Write down a system of linear equations in order to compute the PageRank score of the graph G shown in Figure 1 (without introducing random jumps). Give the solution for the system of linear equations. You can find the solution to the system of linear equations by either solving it or by running the PageRank algorithm with the graph G in input.

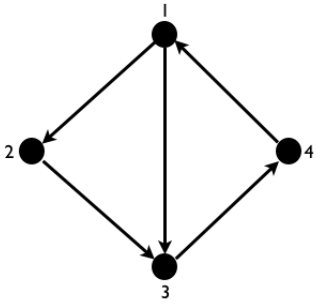

Figure 1: Graph G.

Exercise 2. (MapReduce, 5 pts) Give a pseudocode for one map function and one reduce function in MapReduce so as to find the maximum integer in a list of positive integers given in input. You can assume that the integers are stored in a file as a list of lines of the kind "i n" where i is the line number and n is an integer. You can also assume that the input key and the input value of the map function are a line number and an integer, respectively. Your pseudocode should specify output key and output value for both the map and reduce function.

Exercise 3. (MapReduce, 5 pts) The *yield* keyword in Python returns a generator which allows you to iterate over all the values in the *generator*. Are all the values accessed through the generator stored in main memory? Explain why yield is useful (or not) in the MapReduce framework and in particular why it is useful not to keep all those values in main memory.

Exercise 4. (PageRank in MapReduce, 5 pts). During our class we argue that the PageRank algorithm can be implemented efficiently in MapReduce. For each of the following statements specify whether it is true or false and write one or two sentences motivating your answer.

1. Matrix-vector multiplications can be implemented efficiently in MapReduce while balancing the computational load across different machines.
2. All algorithms consisting of a sequence of matrix-vector multiplications can be implemented efficiently in MapReduce.
3. PageRank can be implemented efficiently in MapReduce because it consists of a sequence of matrix-vector multiplications, while the number of iterations needed to get meaningful results is relatively small (≈ 100).
4. All iterative algorithms requiring at most 100 iterations can be implemented efficiently in MapReduce.
5. MapReduce can handle efficiently frequent updates in a large collection of text documents.

The solutions to the exercises are due by **Tuesday October 4th**. Write down a short report (max 2 pages, in English or French) with all the answers to the previous exercises and send it to `sozio@telecom-paristech.fr` before the deadline. Discussions with other students and the professor is allowed and encouraged, however, everybody should submit his/her own solutions. Students submitting the same solutions (or very similar) for one or more exercises will be penalized. The maximum number of points is 20.