course: Big Data Computing	Master's Degree in Computer Science
Homework	
instructor: Prof. Irene Finocchi	April 7, 2017

Please, read carefully the following instructions:

- *You should hand in: (1) a report – as a pdf file – summarizing your solutions; (2) your code in a compressed archive. Besides a description of the solution, the report must contain instructions to run the code.*
- *You are allowed to work in groups up to three persons: write in the report your names, student ID numbers, and the name of your Master's degree.*
- *Write legibly: English is preferable, but Italian is equally fine. The description and analysis of your algorithms and implementation details should be as clear as possible (which does not imply long).*
- *If you group is formed, e.g., by two students named "StudentA" and "StudentB", send me an email with subject: "BDC 2017 Homework: StudentA Student B". Similarly, your report must be named "BDC17-Report-StudentA-StudentB.pdf".*

Analysis of the Semantic Web

In this assignment you will perform some basic analyses over a large graph obtained from the Billion Triples Dataset, an RDF (Resource Description Framework) dataset that contains about a billion triples from the Semantic Web:

`http://km.aifb.kit.edu/projects/btc-2010/`

The dataset describes an RDF graph, as detailed below.

RDF graph. This is a directed, labeled graph that extends the linking structure of the Web, where nodes represent Web pages, with a semantics for arcs. The idea is to make statements about Web resources in the form of

$$subject - predicate - object$$

expressions, also known as *triples*. Subject and object are two graph nodes, while the predicate is a labeled arc that denotes aspects of a resource and expresses a relationship between subject and object. In mode details:

- the subject is either a Uniform Resource Identifier (URI) or a blank node (anonymous resource);
- the object is a URI, a blank node, or a Unicode string literal;
- the predicate is a URI specifying a resource.

Each triple can have an *optional context* after the object: this is sometimes added to tell where the data is coming from. For example, file `btc-2010-chunk-200` contains the two triples:

`<http://www.last.fm/user/ForgottenSound>` `<http://xmlns.com/foaf/0.1/nick>` "ForgottenSound" `<http://rdf.opiumfield.com/lastfm/friends/life-exe>` .

`<http://dblp.l3s.de/d2r/resource/publications/journals/cg/WestermannH96>` `<http://xmlns.com/foaf/0.1/maker>` `<http://dblp.l3s.de/d2r/resource/authors/Birgit_Westermann>` `<http://dblp.l3s.de/d2r/data/publications/journals/cg/WestermannH96>` .

Both of them have a context. The first triple says that the Webpage <`http://www.last.fm/user/ForgottenSound`> has the nickname “ForgottenSound”. The second describes the maker of another Webpage.

The Billion Triples Dataset. You can find the entire dataset here:

`http://km.aifb.kit.edu/projects/btc-2010/000-CONTENTS`

Notice that the size of each file is about 2GB. Standard editors might experience problems and be unable to open/visualize the file content. The Unix command `more` seems a safe option.

Analyses to be performed. Write Hadoop code to answer the following questions on the first 3 files of the Billion Triples Dataset.

1. Compute the number of distinct nodes and edges in the corresponding RDF graph.
2. Compute the outdegree distribution: does it follow a power law? Plot the result in a figure.
3. Compute the indegree distribution: does it follow a power law? Plot the result in a figure.
4. Which are the 10 nodes with maximum outdegree, and what are their respective degrees?
5. Compute the percentage of triples with empty context, the percentage of triples whose subject is a blank node, and the percentage of triples whose object is a blank node.
6. Each triple can appear with different contexts in the dataset. For each triple, compute the number of distinct contexts in which the triple appears (the empty context counts as 1). Report the 10 triples with the largest number of distinct contexts (break ties arbitrarily).
7. Remove duplicate triples (i.e., produce one or more output files in which triples have no context and each triple appears only once). How much does the dataset shrink? Consider the three input files as a unique dataset by summing up their sizes. Similarly for the output files.

Report the answers in appropriate tables and figures in your report, and explain the results when needed. If convenient, you can think of the computation as being divided into separate jobs (but try to limit the number of jobs).

Running time analysis. Report in a table the running time obtained by your implementation to perform the analyses. If you used different jobs, describe carefully what the goal of each job is (for instance, you might have a job that answers points 1, 2, and 4, another job that takes care of point 3, and so on) and show the running time of each job.

Scalability analysis (optional). How does the running time of your implementation scale on a larger number of input files? Consider a subset of the first k files in the Billion Triples Dataset, and run your jobs. How does the running time increase as k gets larger and larger? Plot the result in a figure.

Report. Besides the results of your analyses, the running times, and instructions for running your code, describe in the report the MapReduce algorithmic strategy behind each analysis: what do the map and reduce functions do? Did you optimize the computation using combiners or other means? Give also details on the platform used to perform the experiments (e.g., your laptop, a cluster, or AWS) and its hardware configuration (e.g., number of nodes, processor clock rate, available memory).