**Recitation 2: Spark (and Hadoop)**

10-26-18

**File Sources:**

Please download from sakai/Recitation/Rec2 to your pyspark local directory the text and csv related files

**TASK1: SPARK Code**

Go to $SPARK\_HOME/core folder and navigate the core code of Spark (schedulers, etc)

**TASK2: SPARK Data Types**

In this part you will learn how to work with spark datatypes. Open the ipython notebook and follow along:

./part3/spark-notebook.ipynb

Now we will use the sparse matrix part-00000

1- Read the matrix in coordinate format. Handin the output that shows you correctly created the RDD in coordinate format.

2- Convert the matrix from coordinate format to row format and show the output that demonstrates that you have correctly done the conversion.

**Hints**:

Guidelines to working with datatypes in Spark can be found at:

https://spark.apache.org/docs/2.1.0/mllib-data-types.html

If you need to read the source code visit here:

https://github.com/apache/spark/tree/v2.2.0/mllib/src/main/scala/org/apache/spark/mllib

**Task3: Jupyter and Python or Mappers and Reducers adjusted for Spark**

Criminal agency stole Facebook's friendships database and wants to analyze new data. Friendships are stored in form key/value pairs, each friendship corresponds to two key/value pairs:

Friends:

key: first friend name

value: second friend name

key: second friend name

value: first friend name

The agency owns also criminal records of all citizens:

Criminal record:

key: citizen name

value: when, where, accomplices, description

Find at risk youths. A person is considered at risk youth if more than half of his friends have criminal record.

**Solution:**

**Task 4: Hadoop-Wordcount Example in Java and Python:**

a) Compile and execute the wordcount code in the wordcount/java folder using the Makefile. Show the screenshot of the output.

b) Go to the wordcount/streaming directory. Write the wordcount map and reduce codes in python (may.py and reduce.py). Also edit the Makefile in the same directory to compile your code. Hand-in the code, makefile, and a screenshot of the output.

**Task 5: Hadoop-Sparse Matrix-Matrix Multiply in Python:**

Open the matmult directory and:

a) Write the hadoop map and reduce files in python to read the input files for matrices A and B from the input folder and print their multiplication. The rank of each matrix is 100 and they are both sparse square matrices.

b) Create a makefile similar to the one in the wordcount example to run your code.

c) Hand-in the code, makefile, and a screenshot of the output.