Final Exam

Note 1: Keep your answers brief and precise.

Note 2: For the individual parts of the exam (Part I and Part II), please refrain from discussing any parts of it with your classmates or share any material that can help answer the questions.

Part I [30%]

1. Answer the following questions.
   (a) In the Google File System paper, it is mentioned that one of the design choices that was made is co-designing the applications and the file system. Explain that choice and give examples of how this decision affected the design.
   (b) Discuss how Spark has addressed one of the shortcomings of Hadoop.
   (c) If you are analyzing a very large dataset using Spark, would you rather choose to write your code using RDDs or DataFrames. Justify your answer.
   (d) Given a 1000-machine cluster and a dataset of size 128TB, you want to count frequencies of words in this dataset using Hadoop. You decided to set the number of reducers of Hadoop to one (R=1) to create one output file (since the number of output files created by Hadoop is equal to R). Is this a good choice? Justify your answer.
2. We have discussed in class two approaches for implementing fault tolerance: replication and lineage. Briefly discuss each of one of them describing how Hadoop and Spark employed these approaches in their system. Briefly compare the pros and cons of each approach.
3. Assume that you want to build a real-time system that detects fraud credit card transactions. Note that a decision to reject or accept a card has to be made in less than a second from swiping it in a card reader machine. The way the decision is made is based on several queries such as: (1) matching the information from the card with incidents of fraud that has occurred in the past; (2) checking engagement of the customer with other transactions with the same vendor in the past periods of time: days, months, years; and (3) location of the customer and their stored home information.
   Input: streamed credit card transactions including identification for the customer using the credit card and the vendor; historical data about details of fraud occurrences; historical data about the customer previous transactions; historical data about the vendors previous transactions.
   Output: accept or reject decision for each transaction.
   (a) Sketch the architecture for a distributed computing system that you would build as a solution to this problem.
   (b) For each platform that you chose to use in your architecture, justify your choice.
   (c) Describe the data flow in your architecture and potential views of the data (intermediate outputs) that you can prepare to speed the processing.

4. A large number of temperature sensors in a nuclear reactor continuously measure temperature and stream the measured values as well as an identifier to the location of these sensor to a processing system. The streamed data is analyzed to detect patterns that trigger alarms for potential dangerous situations. Discuss whether you would prefer to use Spark Streaming or Storm to implement the processing system. Justify your answer and highlight the pros and cons of each system.

5. In the following code snippet, we load a DataFrame from kinesis: dataStream and then process this stream. Briefly describe what this code snippet does and what the purpose of each line of the code is. Note that the class DynamoDbWriter is similar to the one we studied in Exercise 13.

```
val dataStream = spark
      .readStream
      .format("kinesis")
      .... //few more options to identify the kinesis stream
      .option("startingposition", "earliest")
      .load
val query =
     .selectExpr("CAST(data AS STRING)")
     .flatMap(_.split("\\s+"))
     .filter(s => !s.isEmpty)
     .groupBy("value")
     .count()

query.writeStream
     .outputMode("complete")
     .foreach(new DynamoDbWriter(dynamoTableName,regionName))
     .trigger(Trigger.ProcessingTime("3 seconds"))
     .start()
     .awaitTermination()
```

6. In the following code snippet, first, a stackoverflow dataset that contains both questions and answers records is loaded into a DataFrame. Then, the data is split into two DataFrames, one for questions and another for answers. Finally, questions and answers are joined to map each question to its answers.
   Is the use of persist() required or not? Justify your answer.

```
val stackoverflowDF = spark.read......

val questions =
   stackoverflowDF
     .filter("postTypeId == 1")
     .toDF(stackoverflowDF.
     columns.map(_ + "_Q"):_*)
val answers =
   stackoverflowDF
     .filter("postTypeId == 2")
     .toDF(stackoverflowDF
     .columns.map(_ + "_A"):_*)

    questions
      .join(answers,questions("id_Q")===answers("parentId_A"), "inner")
      .groupBy(questions("id_Q"),questions("tag_Q"))
```

Part II [20%]

You are required to work on the following mini-assignment individually.

Code
You will need to add your code to the provided project template available at: `https://github.itu.dk/imel/BIDM_F19_Exam`.

Overview
You are required to analyze a dataset that represent library checkout records and library inventory. The analysis includes writing the code for three queries.

Dataset
You will work with a dataset that is in csv format and that represents details about Seattle Public library Inventory and the checkout records from 2005 till 2017. The dataset is available from Kaggle and can be downloaded from AWS S3 bucket. It contains the following three files:

- Checkouts_By_Title_Data_Lens_20XX: represents the records of all checkouts during the year 20XX. We are mainly interested in the columns: BibNumber and ItemType.
- Integrated_Library_System_ILS_Data_Dictionary: it includes the codes of ItemTypes and their details. We are mainly interested in the columns: Code (which could represent a code for item type item location, ...), Code Type, Format Group, and Format Subgroup.
- Library_Collection_Inventory: includes information about all the library inventory. We are mainly interested in the columns: BibNum and ItemLocation.

Required Implementation
You are required to fill the missing code for the following functions:

- $libraryItemsPerAuthor$: find the total number of items in the library inventory per author.

```
def libraryItemsPerAuthor(libraryInventoryDF:DataFrame):DataFrame = ???
```

- $numberCheckoutRecordsPerFormat$: given library checkout records and the dictionary of library codes, find the total number of checkout occurrences for each item type specified by a Format (Format Gorup + Format SubGroup).

```
def numberCheckoutRecordsPerFormat(
   checkoutDF: DataFrame, dataDictionaryDF:DataFrame) :DataFrame  = ???
```

- $topKCheckoutLocations$: given library checkout records and library inventory details, find the top k locations that have the highest numbers of checkout records. Decode the location using the library dictionary.

```
def topKCheckoutLocations(checkoutDF: DataFrame,
   libraryInventoryDF:DataFrame, dataDictionaryDF:DataFrame,
   k: Int):DataFrame = ???
```

Report
Your report should be two pages maximum and should contain the following:

- A very brief justification for your code choices.
- The execution times of your application when run locally and when run on a cluster of 3 workers (in addition to a master). In both cases, you need to mention the specs of the machines.
- Effect of scaling the data size on the execution time.
- Did you need to "persist" your data? Justify.

Part III [50%]

Submit the reports and codes for projects 1, 2, and 3. This is a group submission, therefore, only one submission is required for a group. For each report, you will need to add a section describing the modifications that you made to your code and report after getting your feedback.

Deliverable

You are required to make two submissions: individual and group submissions.

Individual Submission

This includes the answers for Parts I and II. It should be a zip file that contains the following:

- A report containing the solutions for the questions in Part I of this exam. Note that your report should not exceed three pages (without references).
- A directory containing the submission for Part II: report + LibraryCheckoutBatchProcessing.scala.

Group Submission

This includes Part III. It should be a zip file containing the following directories:

- project 1: this should contain your report for project 1+ FlattenNestedList.scala + SortingLists.scala + HuffmanCoding.scala.
- project 2: this should contain your report for project 2 + AirlineDataAnalysisRDD.scala + AirlineDataAnalysisSQL.scala + YelpAnalysis.scala.
- project 3: this should contain your report for project 3 + AmazonProductsClustering.scala.