1. Given the sfpd RDD, to create a pair RDD consisting of tuples consisting of the form, (Category, 1)

a. pairs = sfpd.parallelize()

b. pairs = sfpd.map(x=>(x(Category),1))

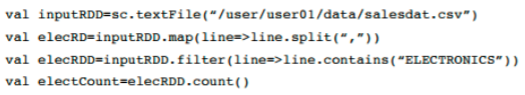
c. pairs = sfpd.map(x=>x.parallelize))

2. Which of the following will give the top 10 resolutions to the console assuming that sfpdDF is the DataFrame registered as a table - sfpd?

a. sqlContext.sql("SELECT resolution, count (incidentnum) AS inccount FROM sfpd GROUP BY resolution ORDER BY inccount DESC LIMIT 10")

b. sfpdDF.select("resolution").count.sort($"count".desc).show(10)

c. sfpdDF.groupBy("resolution").count.sort($"count".desc).show(10)

3.Given the following lines of code in Scala, identify at which step the input RDD is actually computed. 

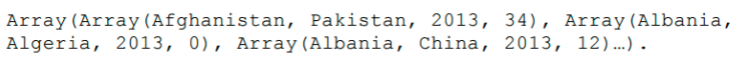
a. The inputRDD is computed and the data loaded as soon as it is defined

b. The inputRDD is computed and data loaded when count() is applied

c. The inputRDD is computed and data loaded when filter() is applied

d. The inputRDD is computed and data loaded when map() is applied

4. An existing RDD, unhcrRDD contains refugee data from the UNHCR. It contains the following fields: (Country of residence, Country of origin, Year, Number of refugees). Sample data is shown below. Assume that number of refugees is of type Int and all other values are of type String. To get the count of all refugees by country of residence, use which of the following?



a. Country = unhcrRDD.map(lambda x:(x(0),x(3))).reduceByKey(lambda(a,b):a+b)

b. country = unhcrRDD.map(lambda x:(x(0),1)).reduceByKey(lambda(a,b):a+b)

c. country = unhcrRDD.map(lambda x:x.parallelize())

5. There are two datasets on online auctions. D1 has the number of the bids for an auction item. D2 contains the seller rating for an auction item. Not every seller has a seller rating. What would you use to get all auction items with the numbe rof bids count and the seller rating (if the data exists) in Scala? \*

1 point

a. D2.join(D1)

b. D1.join(D2)

c. D1.leftOuterJoin(D2)

d. D2.leftOuterJoin(D1)

6. Which of the following 3 DataFrame operations are classified as a wide transformation (that is, they result in a shuffle)?

a. filter()

b. orderBy()

c. cache()

d. distinct()

e. repartition()

f. drop()

7. Which of the following 3 DataFrame operations are NOT classified as an action?

a. printSchema()

b. cache()

c. first()

d. show()

e. limit()

f. foreach()

8. Given an instance of SparkSession named spark, which one of the following code fragments will execute most quickly and produce a DataFrame with the schema specified below?

Sample data:

id:firstName:middleName:lastName:gender:birthDate:ssn:salary

1:Pennie:Carry:Hirschmann:F:1955-07-02:981-43-9345:56172

Schema:

id: integer

firstName: string

middleName: string

lastName: string

gender: string

birthDate: timestamp

ssn: string

salary: integer

File sizes:

CSV File: 608,145,966 bytes

Gzipped CSV File: 273,242,367 bytes

Snappy CSV File: 463,057,399 bytes

Assume that all of the files are stored as a single file (no partitioning) and contain the same data.

Which one of the following code fragments will execute most quickly and produce a DataFrame with the specified schema?

a. csvDF = spark.read

.option("header", "true")

.option("sep", ":")

.option("inferSchema", "true")

.csv("mnt/training/dataframes/people-with-header-10m.txt")

b. csvDF = spark.read

.option("header", "true")

.option("sep", ":")

.csv("mnt/training/dataframes/people-with-header-10m.txt")

c. csvDFsnappy = spark.read

.option("header", "true")

.option("sep", ":")

.option("inferSchema", "true")

.csv("mnt/training/dataframes/people-with-header-10m.txt.snappy")

d. csvDFgz = spark.read

.option("header", "true")

.option("sep", ":")

.option("inferSchema", "true")

.csv("mnt/training/dataframes/people-with-header-10m.txt.gz")

9. tableA is a DataFrame consisting of 20 fields and 40 billion rows of data with a surrogate key field.

tableB is a DataFrame functioning as a lookup table for the surrogate key consisting of 2 fields and 5,000 rows.

If the in-memory size of tableB is 22MB, what occurs when the following code is executed:?

df = tableA.join(tableB, "primary\_key")

a. The contents of tableB will be replicated and sent to each executor to eliminate the need for a shuffle stage during the join.

b. The contents of tableB will be partitioned so that each of the keys that need to be joined on in tableA partitions on each executor will match.

c. An exception will be thrown due to tableB being greater than the 10MB default threshold for a broadcast join.

d. A non-broadcast join will be executed with a shuffle phase since the broadcast table is greater than the 10MB default threshold and the broadcast hint was not specified.

10. Given an instance of SparkSession named spark, review the following code:  
  
a = [1002, 3001, 4002, 2003, 2002, 3004, 1003, 4006]  
  
b = spark  
  .createDataset(a)  
  .withColumn("x", col("value") % 1000)  
  
c = b  
  .groupBy(col("x"))  
  .agg(count("x"), sum("value"))  
  .drop("x")  
  .toDF("count", "total")  
  .orderBy(col("count").desc, col("total"))  
  .limit(1)  
  .show()

Which of the following results is correct?

 a.+-----+-----+  
    |count|total|  
    +-----+-----+  
    |    2| 8008|  
    +-----+-----+  
  b.+-----+-----+  
    |count|total|  
    +-----+-----+  
    |    8|20023|  
    +-----+-----+  
 c.+-----+-----+  
    |count|total|  
    +-----+-----+  
    |    1| 3001|  
    +-----+-----+  
 d.+-----+-----+  
    |count|total|  
    +-----+-----+  
    |    3| 7006|  
    +-----+-----+