Assignment 19: RDD's Deep Dive Assignment Problems

Problem Statement

--==========

Task 1

1.Write a program to read a text file and print the number of rows of data in the document.

Terminal Execution :

```
scala> val fileContent = sc.textFile("file:///home/acadgild/Desktop/19_Dataset.txt")
fileContent: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/Desktop/19_Dataset.txt
MapPartitionsRDD[3] at textFile at <console>:24

scala> fileContent.foreach(println)
Mathew,science,grade-3,45,12
Mathew,history,grade-2,55,13
Mark,maths,grade-2,23,13
Mark,science,grade-1,76,13
John,history,grade-1,14,12
John,maths,grade-2,74,13
Lisa,science,grade-1,24,12
Lisa,history,grade-3,86,13
Andrew,maths,grade-1,34,13
Andrew,science,grade-3,26,14
Andrew,history,grade-1,74,12
Mathew,science,grade-2,55,12
Mathew,history,grade-2,87,12
Mark,maths,grade-1,92,13
Mark,science,grade-2,12,12
John,history,grade-1,67,13
John,maths,grade-1,35,11
Lisa,science,grade-2,24,13
Lisa,history,grade-2,98,15
Andrew,maths,grade-1,23,16
Andrew,science,grade-3,44,14
Andrew,history,grade-2,77,11

scala> fileContent.count
res3: Long = 22
```

2. Write a program to read a text file and print the number of words in the document.

Terminal Execution :

```
scala> val fileContentSplit = fileContent.flatMap(x=>x.split(","))
fileContentSplit: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[4] at flatMap at
<console>:26

scala> fileContentSplit.foreach(println)
Mathew
science
grade-3
45
12
Mathew
history
grade-2
55
13
Mark
maths
grade-2
23
13
Mark
science
grade-1
76
13
John
history
grade-1
14
12
John
maths
grade-2
74
13
```

Lisa
science
grade-1
24
12
Lisa
history
grade-3
86
13
Andrew
maths
grade-1
34
13
Andrew
science
grade-3
26
14
Andrew
history
grade-1
74
12
Mathew
science
grade-2
55
12
Mathew
history
grade-2
87
12
Mark
maths
grade-1
92
13
Mark
science
grade-2
12

12
John
history
grade-1
67
13
John
maths
grade-1
35
11
Lisa
science
grade-2
24
13
Lisa
history
grade-2
98
15
Andrew
maths
grade-1
23
16
Andrew
science
grade-3
44
14
Andrew
history
grade-2
77
11

scala> fileContentSplit.count
res5: Long = 110

3. We have a document where the word separator is -, instead of space. Write a spark code, to obtain the count of the total number of words present in the document.

Terminal Execution :

```
scala> val fileContent = sc.textFile("file:///home/acadgild/Desktop/19_Dataset.txt")
18/08/30 04:16:07 WARN util.SizeEstimator: Failed to check whether UseCompressedOops is set; assuming yes
fileContent: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/Desktop/19_Dataset.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> fileContent.foreach(println)
Mathew,science,grade-3,45,12
Mathew,history,grade-2,55,13
Mark,maths,grade-2,23,13
Mark,science,grade-1,76,13
John,history,grade-1,14,12
John,maths,grade-2,74,13
Lisa,science,grade-1,24,12
Lisa,history,grade-3,86,13
Andrew,maths,grade-1,34,13
Andrew,science,grade-3,26,14
Andrew,history,grade-1,74,12
Mathew,science,grade-2,55,12
Mathew,history,grade-2,87,12
Mark,maths,grade-1,92,13
Mark,science,grade-2,12,12
John,history,grade-1,67,13
John,maths,grade-1,35,11
Lisa,science,grade-2,24,13
Lisa,history,grade-2,98,15
Andrew,maths,grade-1,23,16
Andrew,science,grade-3,44,14
Andrew,history,grade-2,77,11

scala> val fileContentSplit2 = fileContent.flatMap(x=>x.split("-"))
fileContentSplit2: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:26

scala> fileContentSplit2.foreach(println)
Mathew,science,grade
3,45,12
```

Mathew,history,grade
2,55,13
Mark,maths,grade
2,23,13
Mark,science,grade
1,76,13
John,history,grade
1,14,12
John,maths,grade
2,74,13
Lisa,science,grade
1,24,12
Lisa,history,grade
3,86,13
Andrew,maths,grade
1,34,13
Andrew,science,grade
3,26,14
Andrew,history,grade
1,74,12
Mathew,science,grade
2,55,12
Mathew,history,grade
2,87,12
Mark,maths,grade
1,92,13
Mark,science,grade
2,12,12
John,history,grade
1,67,13
John,maths,grade
1,35,11
Lisa,science,grade
2,24,13
Lisa,history,grade
2,98,15
Andrew,maths,grade
1,23,16
Andrew,science,grade
3,44,14
Andrew,history,grade
2,77,11

```
scala> fileContentSplit2.count
res2: Long = 44
```

Task 2

Problem Statement 1:

1. Read the text file, and create a tupled rdd.

Terminal Execution :

```
scala> val fileContent = sc.textFile("file:///home/acadgild/Desktop/19_Dataset.txt")
18/08/30 04:16:07 WARN util.SizeEstimator: Failed to check whether UseCompressedOops
is set; assuming yes
fileContent: org.apache.spark.rdd.RDD[String] =
file:///home/acadgild/Desktop/19_Dataset.txt MapPartitionsRDD[1] at textFile at
<console>:24
 scala> fileContent.foreach(println)
Mathew,science,grade-3,45,12
Mathew,history,grade-2,55,13
Mark,maths,grade-2,23,13
Mark,science,grade-1,76,13
John,history,grade-1,14,12
John,maths,grade-2,74,13
Lisa,science,grade-1,24,12
Lisa,history,grade-3,86,13
Andrew,maths,grade-1,34,13
Andrew,science,grade-3,26,14
Andrew,history,grade-1,74,12
Mathew,science,grade-2,55,12
Mathew,history,grade-2,87,12
Mark,maths,grade-1,92,13
Mark,science,grade-2,12,12
John,history,grade-1,67,13
John,maths,grade-1,35,11
Lisa,science,grade-2,24,13
Lisa,history,grade-2,98,15
Andrew,maths,grade-1,23,16
Andrew,science,grade-3,44,14
Andrew,history,grade-2,77,11

scala> val rdd = fileContent.map( x=> {
     | val row = x.split(",").toList
     | (row.apply(0),row.apply(1),row.apply(2),row.apply(3).toInt,row.apply(4).toInt)})
rdd: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[3] at
map at <console>:26
```

```
scala> rdd.foreach(println)
(Mathew,science,grade-3,45,12)
(Mathew,history,grade-2,55,13)
(Mark,maths,grade-2,23,13)
(Mark,science,grade-1,76,13)
(John,history,grade-1,14,12)
(John,maths,grade-2,74,13)
(Lisa,science,grade-1,24,12)
(Lisa,history,grade-3,86,13)
(Andrew,maths,grade-1,34,13)
(Andrew,science,grade-3,26,14)
(Andrew,history,grade-1,74,12)
(Mathew,science,grade-2,55,12)
(Mathew,history,grade-2,87,12)
(Mark,maths,grade-1,92,13)
(Mark,science,grade-2,12,12)
(John,history,grade-1,67,13)
(John,maths,grade-1,35,11)
(Lisa,science,grade-2,24,13)
(Lisa,history,grade-2,98,15)
(Andrew,maths,grade-1,23,16)
(Andrew,science,grade-3,44,14)
(Andrew,history,grade-2,77,11)
```

2. Find the count of total number of rows present.

Terminal Execution :

```
scala> val rdd = fileContent.map( x=> {
     | val row = x.split(",").toList
     | (row.apply(0),row.apply(1),row.apply(2),row.apply(3).toInt,row.apply(4).toInt)})
rdd: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[3] at map at
<console>:26

scala> rdd.foreach(println)
(Mathew,science,grade-3,45,12)
(Mathew,history,grade-2,55,13)
(Mark,maths,grade-2,23,13)
(Mark,science,grade-1,76,13)
(John,history,grade-1,14,12)
(John,maths,grade-2,74,13)
(Lisa,science,grade-1,24,12)
(Lisa,history,grade-3,86,13)
(Andrew,maths,grade-1,34,13)
(Andrew,science,grade-3,26,14)
```

(Andrew,history,grade-1,74,12)
(Mathew,science,grade-2,55,12)
(Mathew,history,grade-2,87,12)
(Mark,maths,grade-1,92,13)
(Mark,science,grade-2,12,12)
(John,history,grade-1,67,13)
(John,maths,grade-1,35,11)
(Lisa,science,grade-2,24,13)
(Lisa,history,grade-2,98,15)
(Andrew,maths,grade-1,23,16)
(Andrew,science,grade-3,44,14)
(Andrew,history,grade-2,77,11)

scala> rdd.count
res5: Long = 22

3. What is the distinct number of subjects present in the entire school

Terminal Execution :

scala> val fileContent = sc.textFile("file:///home/acadgild/Desktop/19_Dataset.txt")
18/09/02 15:19:53 WARN util.SizeEstimator: Failed to check whether UseCompressedOops is set; assuming yes
fileContent: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/Desktop/19_Dataset.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> val rdd = fileContent.map(x=> {
    | val row = x.split(",").toList
    | (row.apply(0),row.apply(1),row.apply(2),row.apply(3).toInt,row.apply(4).toInt)})
rdd: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[2] at map at <console>:26

scala> rdd.foreach(println)
(Mathew,science,grade-3,45,12)
(Mathew,history,grade-2,55,13)
(Mark,maths,grade-2,23,13)
(Mark,science,grade-1,76,13)
(John,history,grade-1,14,12)
(John,maths,grade-2,74,13)
(Lisa,science,grade-1,24,12)
(Lisa,history,grade-3,86,13)
(Andrew,maths,grade-1,34,13)
(Andrew,science,grade-3,26,14)
(Andrew,history,grade-1,74,12)
(Mathew,science,grade-2,55,12)
(Mathew,history,grade-2,87,12)
(Mark,maths,grade-1,92,13)
(Mark,science,grade-2,12,12)
(John,history,grade-1,67,13)

(John,maths,grade-1,35,11)
(Lisa,science,grade-2,24,13)
(Lisa,history,grade-2,98,15)
(Andrew,maths,grade-1,23,16)
(Andrew,science,grade-3,44,14)
(Andrew,history,grade-2,77,11)

```
scala> rdd.map(tup => (tup._2,tup)).reduceByKey{ case (a, b) => a}.map(_._1)
res2: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[5] at map at <console>:29

scala> val distinctRDD = rdd.map(tup => (tup._2,tup)).reduceByKey{ case (a, b) => a}.map(_._1)
distinctRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at map at <console>:28

scala> distinctRDD.collect().foreach(println)
maths
history
science

scala> distinctRDD.count
res6: Long = 3
```

4. What is the count of the number of students in the school, whose name is Mathew and marks is 55

Terminal Execution :

```
scala> val filterList = rdd.filter(x => (x._1 == "Mathew" && x._4 == 55) )

filterList: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[9] at
filter at <console>:28

scala> filterList.foreach(println)
(Mathew,history,grade-2,55,13)
(Mathew,science,grade-2,55,12)

scala> filterList.count
res8: Long = 2
```

Problem Statement 2:

1. What is the count of students per grade in the school?

Terminal Execution :

```
scala> val fileContent = sc.textFile("file:///home/acadgild/Desktop/19_Dataset.txt")
18/09/02 15:19:53 WARN util.SizeEstimator: Failed to check whether UseCompressedOops is set;
assuming yes
fileContent: org.apache.spark.rdd.RDD[String] = file:///home/acadgild/Desktop/19_Dataset.txt
MapPartitionsRDD[1] at textFile at <console>:24

scala> val studentPerGrade = fileContent.map(x => (x.split(",")
(2),1)).reduceByKey((x,y)=>x+y)
studentPerGrade: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[11] at
reduceByKey at <console>:26

scala> studentPerGrade.foreach(println)
(grade-3,4)
(grade-1,9)
(grade-2,9)
```

2. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

Terminal Execution :

```
scala> val initialRDD = fileContent.map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")
(3).toInt))
initialRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[12] at
map at <console>:26

scala> val initialRDD1 = initialRDD.mapValues(x => (x,1))
initialRDD1: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] =
MapPartitionsRDD[13] at mapValues at <console>:28

scala> val initialRDD2 = initialRDD1.reduceByKey((x,y) => (x._1 + y._1,x._2 + y._2))
initialRDD2: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = ShuffledRDD[14] at
reduceByKey at <console>:30

scala> val avgPerStudent = initialRDD2.mapValues{case(total,count) => (1.0*total)/count}
avgPerStudent: org.apache.spark.rdd.RDD[((String, String), Double)] =
```

```
MapPartitionsRDD[15] at mapValues at <console>:32

scala> avgPerStudent.foreach(println)
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Mathew,grade-3),45.0)
((Andrew,grade-2),77.0)
((Andrew,grade-1),43.666666666666664)
((Lisa,grade-3),86.0)
((John,grade-1),38.666666666666664)
((John,grade-2),74.0)
((Mark,grade-1),84.0)
((Andrew,grade-3),35.0)
((Mathew,grade-2),65.66666666666667)
```

3. What is the average score of students in each subject across all grades?

Terminal Execution :

```
scala> val initialRDD = fileContent.map(x => ((x.split(",")(0),x.split(",")(1)),x.split(",")(3).toInt))
initialRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[16] at map at
<console>:26

scala> val initialRDD1 = initialRDD.mapValues(x => (x,1))
initialRDD1: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = MapPartitionsRDD[17] at
mapValues at <console>:28

scala> val initialRDD2 = initialRDD1.reduceByKey((x,y) => (x._1 + y._1,x._2 + y._2))
initialRDD2: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = ShuffledRDD[18] at
reduceByKey at <console>:30

scala> val avgPerSubject = initialRDD2.mapValues{case(total,count) => (1.0*total)/count}
avgPerSubject: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[19] at
mapValues at <console>:32

scala> avgPerSubject.foreach(println)
((Lisa,history),92.0)
((Mark,maths),57.5)
((Andrew,science),35.0)
((Mark,science),44.0)
((Mathew,science),50.0)
```

((Andrew,maths),28.5)
((Mathew,history),71.0)
((John,maths),54.5)
((John,history),40.5)
((Lisa,science),24.0)
((Andrew,history),75.5)

4. What is the average score of students in each subject per grade?

Terminal Execution :

```
scala> val initialRDD = fileContent.map(x => ((x.split(",")(1),x.split(",")(2)),x.split(",")(3).toInt))
initialRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[20] at map at <console>:26

scala> val initialRDD1 = initialRDD.mapValues(x => (x,1))
initialRDD1: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = MapPartitionsRDD[21] at mapValues at <console>:28

scala> val initialRDD2 = initialRDD1.reduceByKey((x,y) => (x._1 + y._1,x._2 + y._2))
initialRDD2: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = ShuffledRDD[22] at reduceByKey at <console>:30

scala> val avgPerSubjectPerGrade = initialRDD2.mapValues{case(total,count) => (1.0*total)/count}
avgPerSubjectPerGrade: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[23] at mapValues at <console>:32

scala> avgPerSubjectPerGrade.foreach(println)
((history,grade-2),79.25)
((history,grade-3),86.0)
((maths,grade-1),46.0)
((science,grade-3),38.333333333333336)
((science,grade-1),50.0)
((science,grade-2),30.333333333333332)
((history,grade-1),51.666666666666664)
((maths,grade-2),48.5)
```

5. For all students in grade-2, how many have average score greater than 50?

Terminal Execution :

```
scala> val initialRDD = fileContent.map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
initialRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[24] at map at
<console>:26

scala> val initialRDD1 = initialRDD.mapValues(x => (x,1))
initialRDD1: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = MapPartitionsRDD[25] at
mapValues at <console>:28

scala> val initialRDD2 = initialRDD1.reduceByKey((x,y) => (x._1 + y._1,x._2 + y._2))
initialRDD2: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = ShuffledRDD[26] at
reduceByKey at <console>:30

scala> val avg = initialRDD2.mapValues{case(total,count) => (1.0*total)/count}
avg: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[27] at mapValues at
<console>:32

scala> avg.foreach(println)
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Mathew,grade-3),45.0)
((Andrew,grade-2),77.0)
((Andrew,grade-1),43.666666666666664)
((Lisa,grade-3),86.0)
((John,grade-1),38.666666666666664)
((John,grade-2),74.0)
((Mark,grade-1),84.0)
((Andrew,grade-3),35.0)
((Mathew,grade-2),65.66666666666667)

scala> val avgFilter = avg.filter(x => x._1._2 == "grade-2" && x._2>50)
avgFilter: org.apache.spark.rdd.RDD[((String, String), Double)] = MapPartitionsRDD[28] at filter at
<console>:34

scala> avgFilter.foreach(println)
((Lisa,grade-2),61.0)
((Andrew,grade-2),77.0)
((John,grade-2),74.0)
((Mathew,grade-2),65.66666666666667)

scala> avgFilter.count
res15: Long = 4
```

Problem Statement 3:

Are there any students in the college that satisfy the below criteria:

1. Average score per student_name across all grades is same as average score per student_name per grade

Hint - Use Intersection Property

Terminal Execution :

```
scala> val initialRDD = fileContent.map(x => ((x.split(",")(0)),x.split(",")(3).toInt))
initialRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[29] at map at
<console>:26

scala> val initialRDD1 = initialRDD.mapValues(x => (x,1))
initialRDD1: org.apache.spark.rdd.RDD[(String, (Int, Int))] = MapPartitionsRDD[30] at
mapValues at <console>:28

scala>  val initialRDD2 = initialRDD1.reduceByKey((x,y) => (x._1 + y._1,x._2 + y._2))
initialRDD2: org.apache.spark.rdd.RDD[(String, (Int, Int))] = ShuffledRDD[31] at
reduceByKey at <console>:30

scala> val AvgStudent = initialRDD2.mapValues{case(total,count) => (1.0*total)/count}
AvgStudent: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[32] at
mapValues at <console>:32

scala> AvgStudent.foreach(println)
(Mark,50.75)
(Andrew,46.333333333333336)
(Mathew,60.5)
(John,47.5)
(Lisa,58.0)


scala> AvgStudent.count
res17: Long = 5

scala> val initialRDD = fileContent.map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")
(3).toInt))
initialRDD: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[33] at
map at <console>:26
```

```
scala> val initialRDD1 = initialRDD.mapValues(x => (x,1))
initialRDD1: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] =
MapPartitionsRDD[34] at mapValues at <console>:28

scala>  val initialRDD2 = initialRDD1.reduceByKey((x,y) => (x._1 + y._1,x._2 + y._2))
initialRDD2: org.apache.spark.rdd.RDD[((String, String), (Int, Int))] = ShuffledRDD[35] at
reduceByKey at <console>:30

scala> val AvgStudentPerGrade = initialRDD2.mapValues{case(total,count) =>
(1.0*total)/count}
AvgStudentPerGrade: org.apache.spark.rdd.RDD[((String, String), Double)] =
MapPartitionsRDD[36] at mapValues at <console>:32

scala> AvgStudentPerGrade.foreach(println)
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Mathew,grade-3),45.0)
((Andrew,grade-2),77.0)
((Andrew,grade-1),43.666666666666664)
((Lisa,grade-3),86.0)
((John,grade-1),38.666666666666664)
((John,grade-2),74.0)
((Mark,grade-1),84.0)
((Andrew,grade-3),35.0)
((Mathew,grade-2),65.66666666666667)

scala> AvgStudentPerGrade.count
res19: Long = 12

scala> val AvgStudentPerGradeFormat = AvgStudentPerGrade.map(x=>x._1._1 + ","
+x._2.toDouble)
AvgStudentPerGradeFormat: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[37] at
map at <console>:34

scala> val AvgStudentFormat = AvgStudent.map(x=>x._1 + "," +x._2.toDouble)
AvgStudentFormat: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[38] at map at
<console>:34

scala> val commonCriteria = AvgStudentPerGradeFormat.intersection(AvgStudentFormat)
commonCriteria: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[44] at intersection
at <console>:40

scala> commonCriteria.foreach(println)
```

```
scala> commonCriteria.count
res21: Long = 0
```

Thus, this shows that there is no student that satisfies criteria Average score per student_name across all grades is same as average score per student_name per grade.