Database Project (SWE3033) (Fall 2023)

Homework #4 (50pts, Due date: 10/11)

Student ID: 2020315798

Student Name: Choi Jin Woo

Instruction: In this homework, we provide you with a jupyter notebook file (DBP_Homework4.ipynb). You should follow the instructions in these documents carefully.

Submit two files as follows:

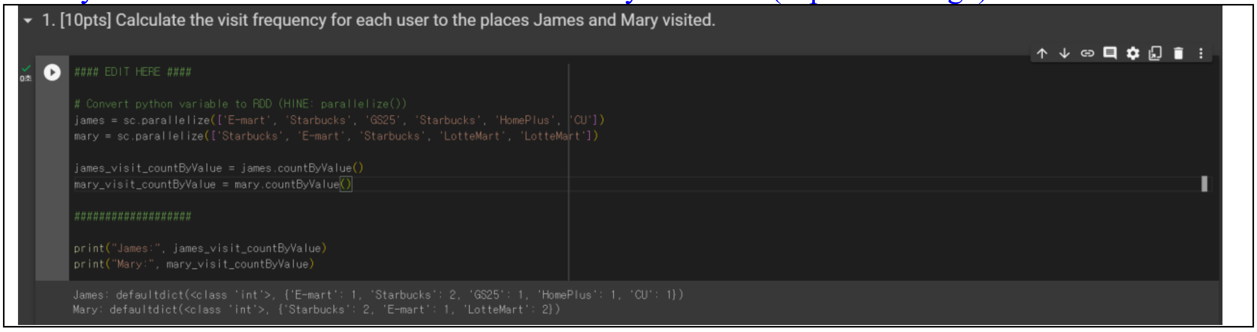
- DBP_Homework4_StudentID.zip
    - DBP_Homework4_StudentID.ipynb
    - DBP_Homework4_StudentID.pdf

1. [10pts] Calculate the visit frequency for each user to the places James and Mary visited.

a. Places that James visited:
- [‘E-mart’, ‘Starbucks’, ‘GS25’, ‘Starbucks’, ‘HomePlus’, ‘CU’]

b. Places that Mary visited:
- [‘Starbucks’, ‘E-mart’, ‘Starbucks’, ‘LotteMart’, ‘LotteMart’]

[Answer]
Enter your code and result here. You must show your result (captured image).



1. [10pts] Calculate the visit frequency for each user to the places James and Mary visited.

```
#### EDIT HERE ####

# Convert python variable to RDD (HINE: parallelize())
james = sc.parallelize(['E-mart', 'Starbucks', 'GS25', 'Starbucks', 'HomePlus', 'CU'])
mary = sc.parallelize(['Starbucks', 'E-mart', 'Starbucks', 'LotteMart', 'LotteMart'])

james_visit_countByValue = james.countByValue()
mary_visit_countByValue = mary.countByValue()

###################

print("James:", james_visit_countByValue)
print("Mary:", mary_visit_countByValue)
```

```
James: defaultdict(<class 'int'>, {'E-mart': 1, 'Starbucks': 2, 'GS25': 1, 'HomePlus': 1, 'CU': 1})
Mary: defaultdict(<class 'int'>, {'Starbucks': 2, 'E-mart': 1, 'LotteMart': 2})
```

2. [20pts] Count the number of words in the given data using the following two operations and explain the difference between the two operations.

Data:

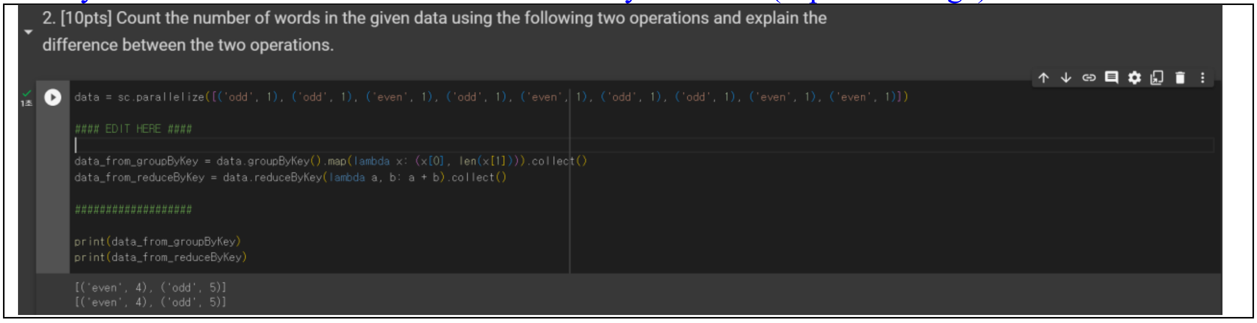
[(‘odd’, 1), (‘odd’, 1), (‘even’, 1), (‘odd’, 1), (‘even’, 1), (‘odd’, 1), (‘odd’, 1), (‘even’, 1), (‘even’, 1)]

a. groupByKey()
b. reduceByKey()
c. Explain the difference between the two operations.

[Answer]

a)	`[('even', 4), ('odd', 5)]`
b)	`[('even', 4), ('odd', 5)]`
c)	[groupByKey()] This operation groups the data by key and produces an iterable of values For the intermediate step, it reaches [('odd', [1, 1, 1, 1, 1, 1]), ('even', [1, 1, 1, 1])] and then gets the final result to [('odd', 5), ('even', 4)] [reduceByKey()] This operation combines the values for each key using a reduction function and provides the word count as the result. For the intermediate step, it reaches directly to [('odd', 5), ('even', 4)]

Enter your code and result here. You must show your result (captured image).



2. [10pts] Count the number of words in the given data using the following two operations and explain the difference between the two operations.

```
data = sc.parallelize([('odd', 1), ('odd', 1), ('even', 1), ('odd', 1), ('even', 1), ('odd', 1), ('odd', 1), ('even', 1), ('even', 1)])

#### EDIT HERE ####

data_from_groupByKey = data.groupByKey().map(lambda x: (x[0], len(x[1]))).collect()
data_from_reduceByKey = data.reduceByKey(lambda a, b: a + b).collect()

###################

print(data_from_groupByKey)
print(data_from_reduceByKey)
```

```
[('even', 4), ('odd', 5)]
[('even', 4), ('odd', 5)]
```

3. **[20pts]** The following data represents the songs Mary and James have listened to and the play counts. Answer the following three questions.

Data: key-value data in (music, # of plays) format

- James: [('Thriller', 30), ('Everybody', 34), ('Everybody', 30), ('Billie_Jean', 2)]
- Mary: [('Thriller', 20), ('Sorry', 23), ('Sorry', 3), ('Billie_Jean', 5)]

a. For each user, calculate the number of times each song has been listened to, s tore it in a new RDD. (HINT: reduceByKey())

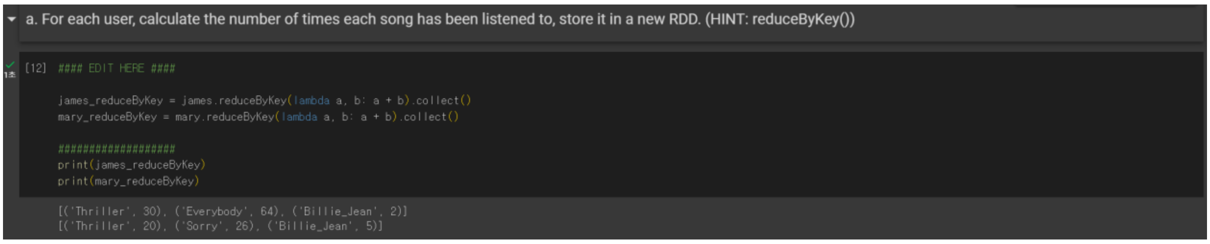
b. Create a new RDD containing songs that both users have listened to and their respectiv e play counts. (HINT: join())

c. Calculate the total number of music plays that James and Mary have played in common.

[Answer]

Enter your code and result here. You must show your result (captured image).

a.



a. For each user, calculate the number of times each song has been listened to, store it in a new RDD. (HINT: reduceByKey())

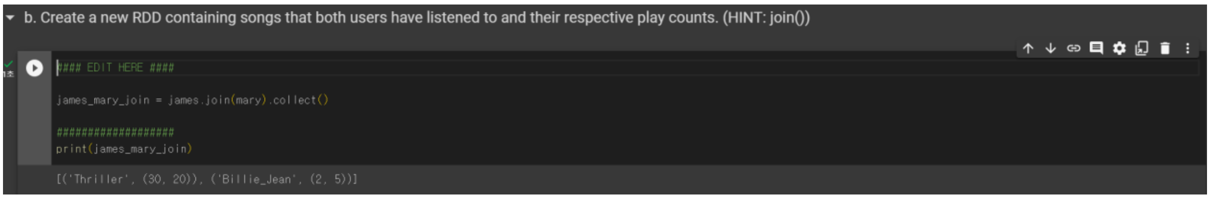
```
#### EDIT HERE ####

james_reduceByKey = james.reduceByKey(lambda a, b: a + b).collect()
mary_reduceByKey = mary.reduceByKey(lambda a, b: a + b).collect()

###################
print(james_reduceByKey)
print(mary_reduceByKey)
```

```
[('Thriller', 30), ('Everybody', 64), ('Billie_Jean', 2)]
[('Thriller', 20), ('Sorry', 26), ('Billie_Jean', 5)]
```

b.



b. Create a new RDD containing songs that both users have listened to and their respective play counts. (HINT: join())

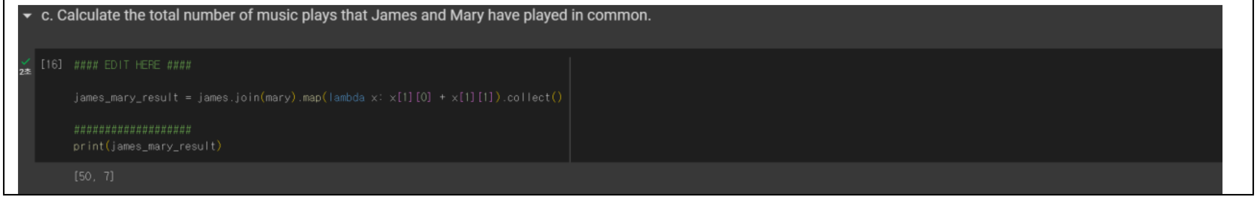
```
#### EDIT HERE ####

james_mary_join = james.join(mary).collect()

###################
print(james_mary_join)
```

```
[('Thriller', (30, 20)), ('Billie_Jean', (2, 5))]
```

c.



c. Calculate the total number of music plays that James and Mary have played in common.

```
#### EDIT HERE ####

james_mary_result = james.join(mary).map(lambda x: x[1][0] + x[1][1]).collect()

###################
print(james_mary_result)
```

```
[50, 7]
```