به‌نام خداوند بخشنده و مهربان

تمرین اول: مقدمه‌ای بر اسپارک
درس: پایگاه‌داده پیشرفته

استاد: محمدعلی نعمت‌بخش
دستیاران: فاطمه ابراهیمی، پریسا لطیفی، امیر سرتیپی

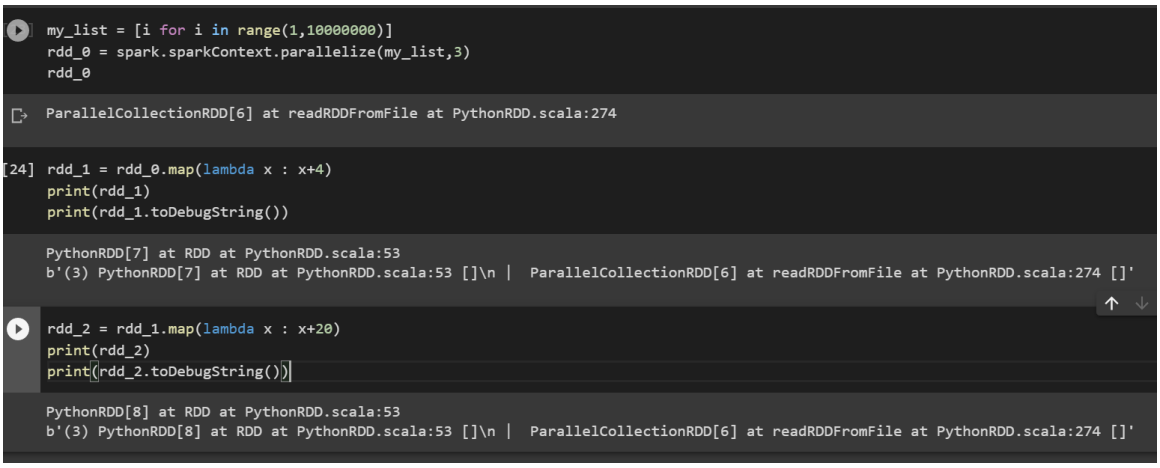
نام و نام‌خانوادگی: زلفا شفرئی
آدرس گیت: https://github.com/zolfaShefreie/spark_basics

- **لطفا پاسخ تمارین حتما در سامانه‌ی کوئرا ارسال شود.**
- **لطفا پاسخ‌های خود را در خود سند سوال نوشته و در قالب یک فایل PDF ارسال کنید.**
- **نام سند ارسالی HW-{homework number}-{Name Family}-{student number}**
- **تمامی فایل‌های مورد نیاز این تمرین در این لینک قابل دسترس است.**
- **خروجی از هر مرحله‌ی تمرین را در سند خود بارگذاری کنید.**
- **کد + سند را در گیت بارگذاری کرده و لینک آن را در سند قرار دهید.**

در این تمرین، هدف ما آشنایی Action و Transformation در موتور تحلیلی Spark است.

۱. منظور از Lazy Evaluation در Spark چیست؟ این مفهوم را همراه با یک مثال توضیح دهید.

منظور از Lazy Evalution این است تا زمانی که یک عمل راه‌اندازی نشود آن عمل اجرا نخواهد شد. به‌بیان ساده‌تر عملیات به‌صورت کد به Spark گفته می‌شود و هنگامی که نیاز به نتایج آن باشد Spark بهترین مسیر برای اجرای کم‌هزینه‌تر پیدا می‌کند و از طریق آن اجرا می‌کند. عملیات یاد شده همان مجموعه تغییرات در یک مجموعه‌داده می‌باشد. برای مثال از یک لیست یک RDD با سه پارتیشن می‌سازیم. با استفاده از تابع map به هرکدام از آن‌ها a را اضافه می‌کنیم. این عملیات را با عدد b تکرار می‌کنیم. خروجی هر مرحله در شکل ۱ مشاهده می‌شود. rdd_1 و rdd_2 هر دو عملیات خود را از rdd_0 انجام می‌دهد یعنی بهترین مسیر انجام عملیات اضافه کردن مجموع دو عدد a و b به rdd_0 می‌باشد [۱].



شکل ۱: نمونه‌ای از lazy evaluation

۲. منظور از Narrow Transmitaion (NT) و Wide Transmitation (WT) را در Spark همراه با یک مثال بیان کنید. تفاوت اصلی این دو مفهوم چیست؟

Transformation عملیاتی هستند که برای پردازش داده برروی مجموعه‌داده اعمال می‌شوند. دو نوع Transformation برروی Spark وجود دارد که توضیحات آن‌ها به‌شرح زیر است:

- Narrow Transformation: در این تبدیل تمامی المان‌ها و رکوردهای موردنیاز برای انجام یک عمل، در یک پارتیشن از RDD در دسترس می‌باشند. از این تبدیل‌ها می‌توان به map() و filter() اشاره کرد.
- Wide Transformation: در این تبدیل المان و رکوردهای موردنیاز برای انجام یک عملیات در بسیاری از پارتیشن‌های یک RDD در دسترس هستند. groupbyKey و reducebyKey از مثال‌های این تبدیل هستند [۲].

۳. با توجه به سوال پیشین، ۴ مورد از NT، WT و Action هایی که در اسپارک وجود دارند نام ببرید.

نمونه‌ی هر یک از موارد ذکر شده به‌شرح زیر است [۳]:

- Narrow Transformation: map، flatmap، filter، sample، union
- Wide Transformation: groupByKey، reduceByKey، join، intersection
- Action: count، collect، first، take

۴. برای آ شنایی بی‌شتر با مفاهیم بیان شده و مقدمه‌ای بر توابع عملیات‌های زیر را انجام داده و خروجی هریک به همراه بلاک کد آن را گزارش دهید. مثالی از خروجی برای هر بخش نمایش داده شده است.

- برای کار با اسپارک، کتاب‌خانه‌ای با نام pyspark وجود دارد.

- نوت‌بوکی بر روی گوگل کولب ایجاد کرده و این کتاب‌خانه را فراخوانی کنید.

به‌علت وجود نداشتن کتابخانه‌ی pyspark ابتدا این کتابخانه را دانلود کرده و سپس آن را import می‌کنیم. برای استفاده از pyspark نیاز به یک جلسه است که با استفاده از کد نشان داده شده در شکل ۲ تنظیمات مربوطه را انجام می‌دهیم.

```
!pip install pyspark

Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
     |████████████████████████████████| 281.4 MB 31 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
     |████████████████████████████████| 198 kB 59.7 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl
  Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd8017084dce4e93
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.3 pyspark-3.2.1
```

```
[2] import pyspark
    from pyspark.sql import SparkSession
```

```
[3] spark = SparkSession.builder.appName('train').getOrCreate()
```

```
[4] sparkContent = spark.sparkContext
```

شکل ۲: نصب و تنظیمات اولیه‌ی جلسه

- سپس یک لیست ۵۰ تایی از یک موضوع را برای خود در ست کنید. برای مثال لیستی از (کتاب‌ها، نرم‌افزارها و ...)

این لیست از یک مجموعه‌داده مربوط به کتاب تهیه شده است. به‌همین علت کدهایی شامل دانلود و خواندن فایل و تبدیل dataframe به لیست در آن مشاهده می‌شود.

```
import requests

def download_file(url, file_name):
  file_content = requests.get(url).text
  file = open(file_name, 'w')
  file.write(file_content)
  file.close()
```

شکل ۳: تابع دانلود فایل

```
[6]  import pandas as pd

     url = "https://gist.githubusercontent.com/jaidevd/23aef12e9bf56c618c41/raw/c05e98672b8d52fa0cb94aad80f75eb78342e5d4/books.csv"
     file_name = "books.csv"

[7]  download_file(url, file_name)

[8]  book_names = pd.read_csv(file_name)['Title'].head(50).to_list()
     book_names
```

شکل ۴: دانلود مجموعه‌داده و ایجاد لیستی از نام کتاب‌ها

```
['Fundamentals of Wavelets',
 'Data Smart',
 'God Created the Integers',
 'Superfreakonomics',
 'Orientalism',
 'Nature of Statistical Learning Theory, The',
 'Integration of the Indian States',
 "Drunkard's Walk, The",
 'Image Processing & Mathematical Morphology',
 'How to Think Like Sherlock Holmes',
 'Data Scientists at Work',
 'Slaughterhouse Five',
 'Birth of a Theorem',
 'Structure & Interpretation of Computer Programs',
 'Age of Wrath, The',
 'Trial, The',
 "Statistical Decision Theory'",
 'Data Mining Handbook',
 'New Machiavelli, The',
 'Physics & Philosophy',
 'Making Software',
 'Analysis, Vol I',
 'Machine Learning for Hackers',
 'Signal and the Noise, The',
 'Python for Data Analysis',
 'Introduction to Algorithms',
 'Beautiful and the Damned, The',
 'Outsider, The',
 'Complete Sherlock Holmes, The - Vol I',
 'Complete Sherlock Holmes, The - Vol II',
 'Wealth of Nations, The',
 'Pillars of the Earth, The',
 'Mein Kampf',
 'Tao of Physics, The',
 "Surely You're Joking Mr Feynman",
 'Farewell to Arms, A',
 'Veteran, The',
 'False Impressions',
 'Last Lecture, The',
 'Return of the Primitive',
 'Jurassic Park',
 'Russian Journal, A',
 'Tales of Mystery and Imagination',
 'Freakonomics',
 'Hidden Connections, The',
 'Story of Philosophy, The',
 'Asami Asami',
 'Journal of a Novel',
 'Once There Was a War',
 'Moon is Down, The']
```

شکل ۵: لیست نام کتاب‌ها

- لیست خود را به RDD تبدیل کنید.

```
[9]  books_rdd = sparkContent.parallelize(book_names)

     books_rdd

     ParallelCollectionRDD[0] at readRDDFromFile at PythonRDD.scala:274
```

شکل ۶: تبدیل به RDD

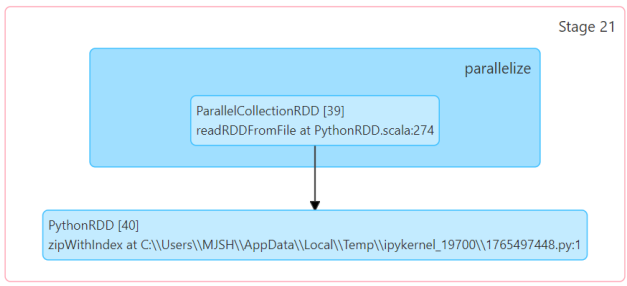
- با کمک دستور filter بر روی RDD، از آن برای بازیابی عنصر ۲۰ام لیست خود استفاده کنید. (برابر با عنصر ۲۰ام باشد)

```
['Amir Sartipi']
```

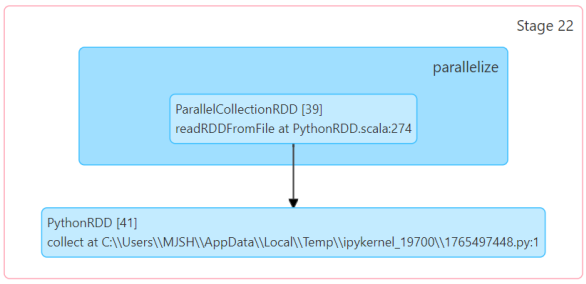
```
[10] books_rdd.zipWithIndex().filter(lambda x: x[1]==19).map(lambda x: x[0]).collect()

     ['Physics & Philosophy']
```

شکل ۷: بازیابی عنصر ۲۰ام



شکل ۹: ترتیب انجام عملیات در job15



شکل ۸: ترتیب انجام عملیات در job16

این عملیات در دو job و stage متفاوت اجرا می‌شود. در یک job در ابتدا zipWithIndex انجام می‌گیرد و در job دیگر عملیات تغییرات و collect.

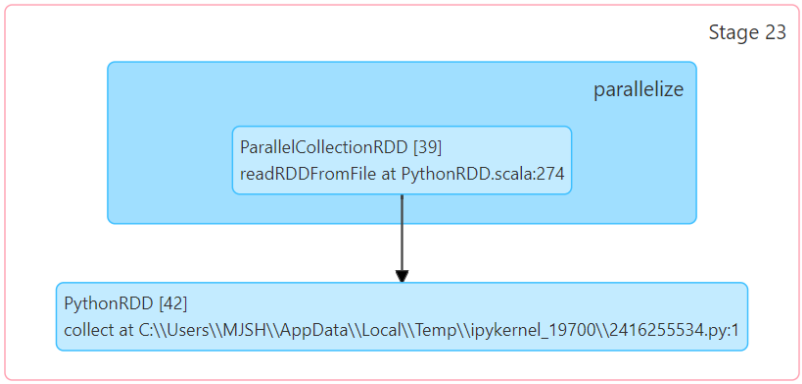
- با کمک map تمامی عناصر لیست خود را به حروف بزرگ تبدیل و آن را بازیابی کنید.

```
['AMIR SARTIPI',
 'SOBHAN',
 'REZA',
 'AMIR REZA',
 'SADEGH',
 'AHMAD AHMIDAN',
 'MOHAMMAD',
 'MOHAMMAD REZA',
 'SIRVAN',
 'SHIMA',
 'AREZOO',
 'MARYAM',
 'FARIMA']
```

```
[11] books_rdd.map(lambda x: x.upper()).collect()

    ['FUNDAMENTALS OF WAVELETS',
     'DATA SMART',
     'GOD CREATED THE INTEGERS',
     'SUPERFREAKONOMICS',
     'ORIENTALISM',
     'NATURE OF STATISTICAL LEARNING THEORY, THE',
     'INTEGRATION OF THE INDIAN STATES',
     "DRUNKARD'S WALK, THE",
     'IMAGE PROCESSING & MATHEMATICAL MORPHOLOGY',
     'HOW TO THINK LIKE SHERLOCK HOLMES',
     'DATA SCIENTISTS AT WORK',
     'SLAUGHTERHOUSE FIVE',
     'BIRTH OF A THEOREM',
     'STRUCTURE & INTERPRETATION OF COMPUTER PROGRAMS',
     'AGE OF WRATH, THE',
     'TRIAL, THE',
     "STATISTICAL DECISION THEORY'",
     'DATA MINING HANDBOOK',
     'NEW MACHIAVELLI, THE',
     'PHYSICS & PHILOSOPHY',
     'MAKING SOFTWARE',
     'ANALYSIS, VOL I',
     'MACHINE LEARNING FOR HACKERS',
     'SIGNAL AND THE NOISE, THE'
```

شکل ۱۰: تبدیل عناصر به حروف بزرگ



شکل ۱۱: مراحل انجام

- با کمک دستور groupby و map، لیست خود را بر اساس اولین کاراکتر آن دسته بندی کنید.

```
[('A',
  ['Amir Sartipi',
   'Amir Sartipi',
   'Amir Reza',
   'Ahmad Ahmidan',
   'Ava',
   'Arezoo']),
 ('M', ['Mohammad', 'Mohammad Reza', 'Maryam']),
 ('F', ['Farima']),
 ('S', ['Sobhan', 'Sadegh', 'Sanaz', 'Sirvan', 'Shima']),
 ('R', ['Reza'])]
```

ابتدا یک نگاشت انجام می‌شود تا یک زوج حرف اول نام و نام تشکیل شود. سپس براساس اولین المان هر زوج دسته‌بندی انجام می‌شود و در آخر یک نگاشت برای تبدیل به لیست مشابه خواسته شده انجام می‌گردد.

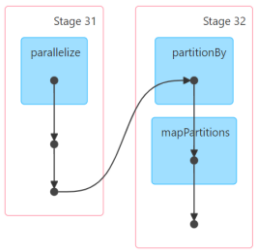
```
books_rdd.map(lambda x: (x[0].upper(), x)).\
groupBy(lambda x:x[0]).\
map(lambda group: (group[0], [x[1] for x in group[1]])).\
sortBy(lambda x: x[0]).collect()
```

شکل ۱۲: دسته‌بندی لیست

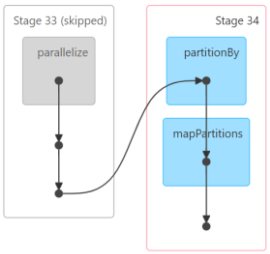
```
[('A', ['Age of Wrath, The', 'Analysis, Vol I', 'Asami Asami']),
 ('B', ['Birth of a Theorem', 'Beautiful and the Damned, The']),
 ('C',
  ['Complete Sherlock Holmes, The - Vol I',
   'Complete Sherlock Holmes, The - Vol II']),
 ('D',
  ['Data Smart',
   "Drunkard's Walk, The",
   'Data Scientists at Work',
   'Data Mining Handbook']),
 ('F',
  ['Fundamentals of Wavelets',
   'Farewell to Arms, A',
   'False Impressions',
   'Freakonomics']),
 ('G', ['God Created the Integers']),
 ('H', ['How to Think Like Sherlock Holmes', 'Hidden Connections, The']),
 ('I',
  ['Integration of the Indian States',
   'Image Processing & Mathematical Morphology',
   'Introduction to Algorithms']),
 ('J', ['Jurassic Park', 'Journal of a Novel']),
 ('L', ['Last Lecture, The']),
 ('M',
  ['Making Software',
   'Machine Learning for Hackers',
   'Mein Kampf',
   'Moon is Down, The']),
 ('N', ['Nature of Statistical Learning Theory, The', 'New Machiavelli, The']),
 ('O', ['Orientalism', 'Outsider, The', 'Once There Was a War']),
 ('P',
  ['Physics & Philosophy',
   'Python for Data Analysis',
   'Pillars of the Earth, The']),
 ('R', ['Return of the Primitive', 'Russian Journal, A']),
 ('S',
  ['Superfreakonomics',
   'Slaughterhouse Five',
   'Structure & Interpretation of Computer Programs',
   "Statistical Decision Theory'",
   'Signal and the Noise, The',
   "Surely You're Joking Mr Feynman",
   'Story of Philosophy, The']),
 ('T',
  ['Trial, The', 'Tao of Physics, The', 'Tales of Mystery and Imagination']),
 ('V', ['Veteran, The']),
 ('W', ['Wealth of Nations, The'])]
```

شکل ۱۳: نتیجه‌ی دسته‌بندی بر اساس حروف اول

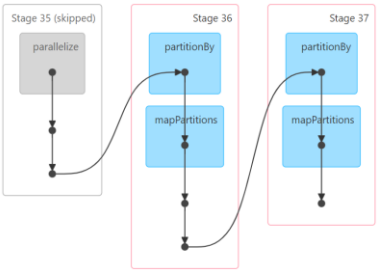
این عملیات در سه job متفاوت انجام شد که جزئیات آن را در شکل‌های زیر مشاهده می‌کنید. در ابتدا groupBy انجام می‌شود و با استفاده از نتایج آن map و سپس sort انجام می‌گردد.



شکل ۱۶: **job21**

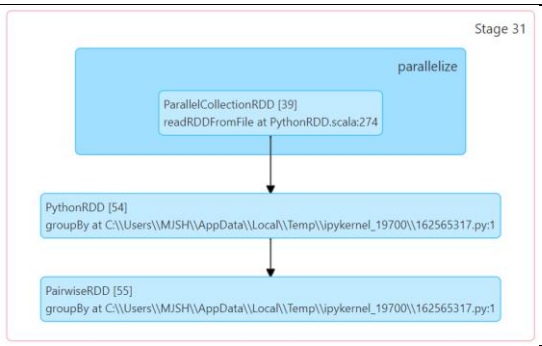
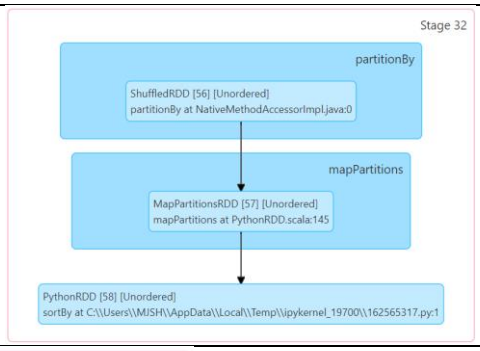
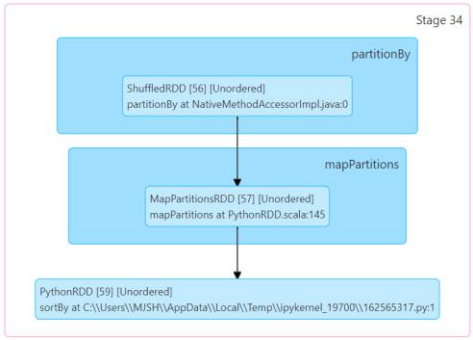
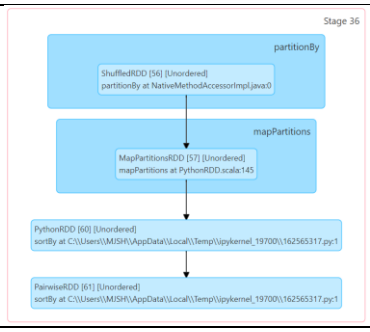
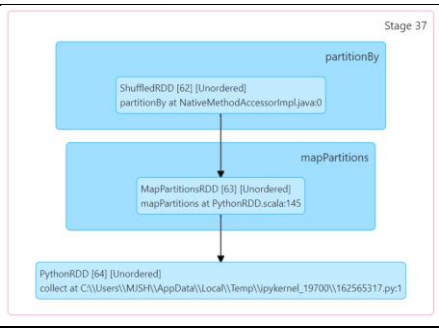


شکل ۱۵: **job22**



شکل ۱۴: **job32**

جدول ۱: جزئیات اجرا در هر **stage** و **job**

		21	jobs
		22	
		23	

- عملیات map و reduce را بر روی یک متن نســـبتا بلند پس از تبدیل توکن‌های آن به rdd، انجام دهید.

```
[('January', 1),
 ('2001,', 1),
 ('Jimmy', 1),
 ('Wales[6]', 1),
 ('Larry', 1),
 ('name', 1),
 ('as', 2),
 ('of', 4),
 ('was', 1),
 ('spontaneous', 1),
 ('Austrian', 2),
 ('after', 1),
 ('exposed', 1),
 ('these', 1),
 ('Institute', 1),
 ('Fellow', 1),
 ('Mark', 1),
 ('Initially', 1),
 ('only', 1),
 ('in', 3),
 ('English,', 1),
 ('versions', 1),
 ('other', 1),
 ('quickly', 1),
```

```
txt_url = "https://raw.githubusercontent.com/brunoklein99/deep-learning-notes/master/shakespeare.txt"
txt_file_name = "shakespeare.txt"
```

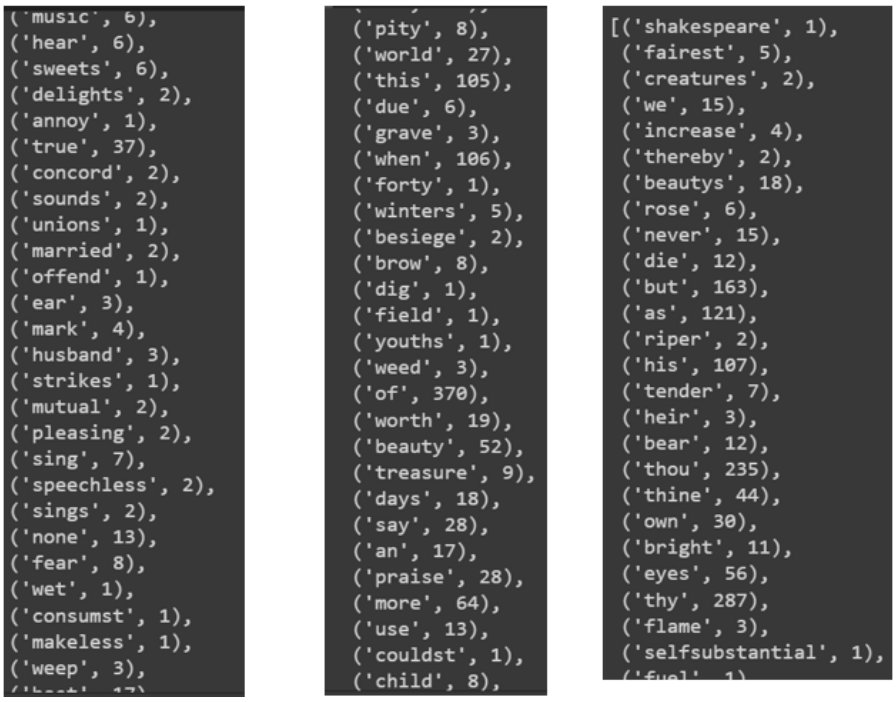
```
[16] download_file(txt_url, txt_file_name)
```

preprocessing and map-reduce

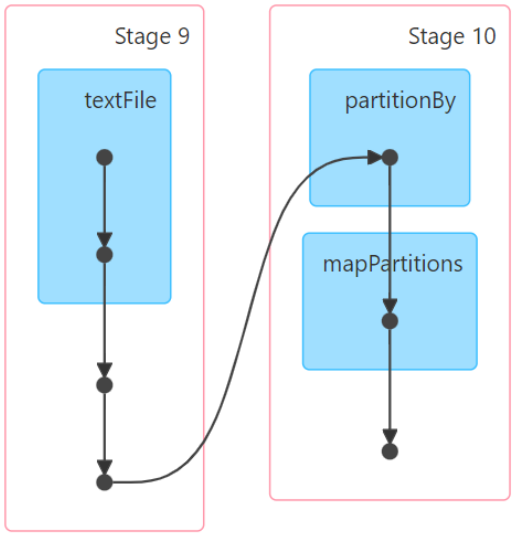
```
[17] def delete_puctuation(x):
       new_str = str() + x
       for each in '''!()-[]{.};:'"\,<>/?@#$%^&*_~`|’“”…—–''':
         new_str = new_str.replace(each, "")
       return new_str.strip()
```

```
[18] text_rdd = sparkContent.textFile(txt_file_name)
     words = text_rdd.flatMap(lambda line: line.split(" ")).filter(lambda x: x.strip())
     words = words.map(delete_puctuation).map(lambda x: x.lower())
     words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b).collect()
```

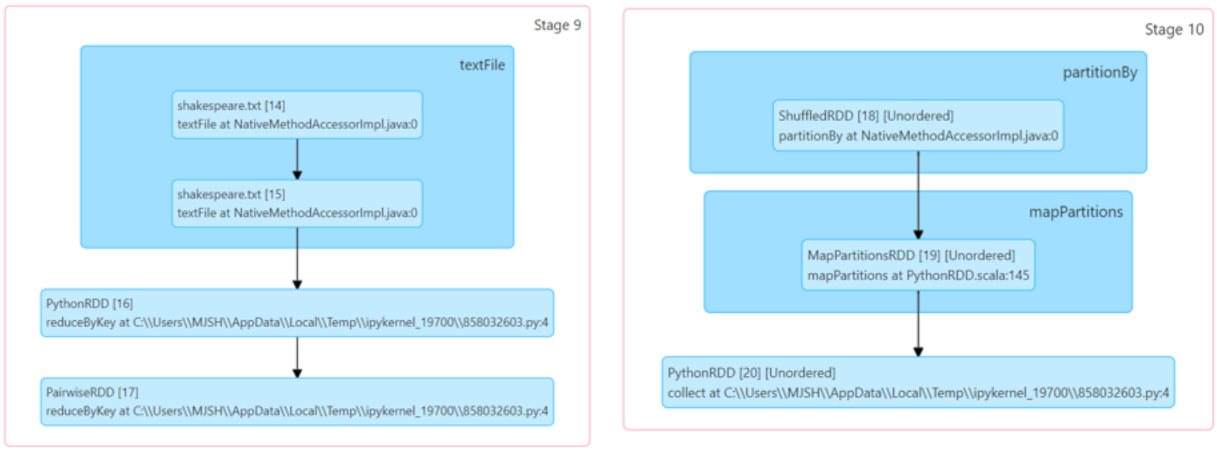
شکل ۱۷: اجرای word counting با استفاده از spark بر روی متن مجموعه‌داده شکسپیر



شکل ۱۸: قسمت‌هایی از نتایج



شکل ۱۹: مراحل اجرا در job



شکل ۲۰: جزئیات اجرایی در هر stage

- چه تفاوتی بین Actionهای take و collect وجود دارد؟

take تعدادی از عناصر یک RDD را برمیگرداند که مقدار آن توسط کاربر به‌عنوان ورودی ارسال می‌شود و سعی می‌کند دسترسی پارتیشن‌ها را به حداقل برساند اما collect تمامی عناصر موجود در RDD را برمی‌گرداند.

- در صورتی که بتوانید توالی انجام هریک از عملیات‌ها در ا سپارک که برای هر د ستور انجام می‌دهد را برای هریک از د ستورات بالا نمایش دهید و باتوجه به مفاهیم سوالات قبل آن را ت صویر سازی کنید، نمره اضافه‌ای دریافت خواهید کرد. (به کمک ngrok و UI Spark)

برای دسترسی به UI Spark کدها در سیستم لوکال اجرا شده‌اند. با اجرای کدهای شکل فلان می‌توان به UI Spark با آدرس http://localhost:4050 دسترسی داشت.

```
37 import pyspark
   from pyspark.sql import SparkSession

38 import findspark
   findspark.init()

39 spark = SparkSession.builder.appName('train').config('spark.ui.port', '4050').getOrCreate()
   spark
```

شکل ۲۱: تیکه کد برای دسترسی به UI Spark

شایان ذکر است شکل‌های DAG مربوط به سوالات بالا با ترتیب اجرای پی‌درپی نمی‌باشد به همین دلیل شناسه‌ها تمامی مراحل سوال به‌ترتیب و پشت سرهم نمی‌باشد.

مراجع

[1] L. Arora, "Being Lazy is Useful — Lazy Evaluation in Spark," 28 October 2019. [Online]. Available: https://medium.com/analytics-vidhya/being-lazy-is-useful-lazy-evaluation-in-spark-1f04072a3648.

[2] A. Anthony, "How Apache Spark's Transformations And Action works…," medium, 12 July 2017. [Online]. Available: https://medium.com/@aristo_alex/how-apache-sparks-transformations-and-action-works-ceb0d03b00d0.

[3] "Spark RDD Operations-Transformation & Action with Example," data-flair, [Online]. Available: https://data-flair.training/blogs/spark-rdd-operations-transformations-actions/.