Spark Training questions

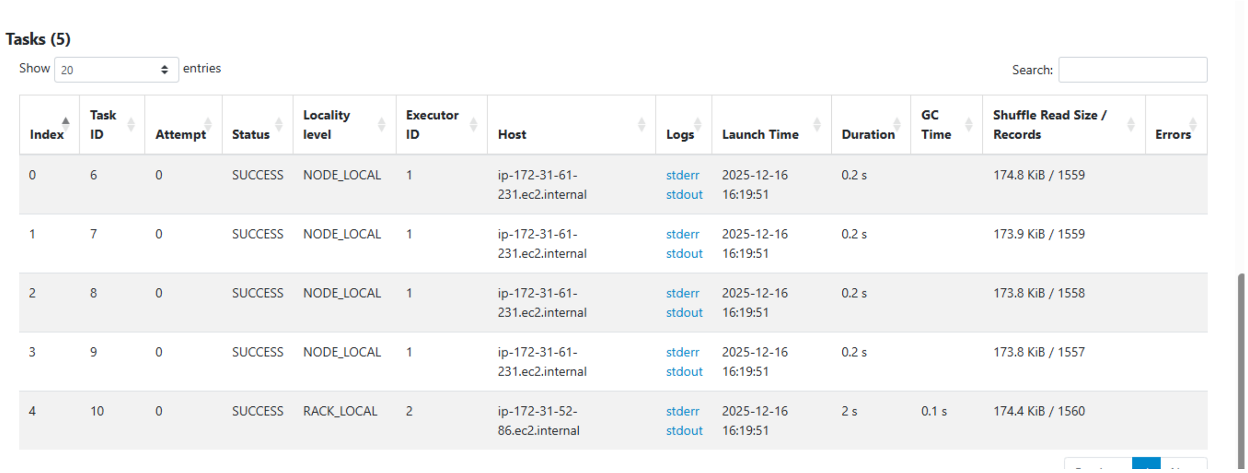
Please answer the questions below

Exercise 1

Q1. Please put your code here:

```
import sys
from pyspark import SparkContext, SparkConf

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: wordcount <input_folder>", file=sys.stderr)
        sys.exit(-1)

    conf = SparkConf().setAppName("python-word-count")
    sc = SparkContext(conf=conf)

    text_file = sc.textFile("hdfs://" + sys.argv[1])
    counts = text_file.flatMap(lambda line: line.split(" ")) \
            .map(lambda word: (word, 1)) \
            .reduceByKey(lambda a, b: a + b) \
                            .repartition(5) \
            .filter(lambda x: len(x[0])>5)
    # "takeOrdered" is an action.
    list = counts.takeOrdered(40, key = lambda x: -x[1])
    print("--------------------------------------------")
    # print (repr(list)[1:-1])
    print(*list, sep="\n")
    print("--------------------------------------------")
```

Q2. Add print-screen of the stage proving you have 5 tasks



Tasks (5)

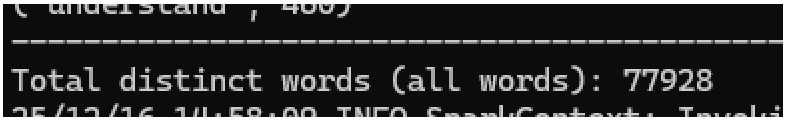
Index	Task ID	Attempt	Status	Locality level	Executor ID	Host	Logs	Launch Time	Duration	GC Time	Shuffle Read Size / Records	Errors
0	6	0	SUCCESS	NODE_LOCAL	1	ip-172-31-61-231.ec2.internal	stderr stdout	2025-12-16 16:19:51	0.2 s		174.8 KiB / 1559	
1	7	0	SUCCESS	NODE_LOCAL	1	ip-172-31-61-231.ec2.internal	stderr stdout	2025-12-16 16:19:51	0.2 s		173.9 KiB / 1559	
2	8	0	SUCCESS	NODE_LOCAL	1	ip-172-31-61-231.ec2.internal	stderr stdout	2025-12-16 16:19:51	0.2 s		173.8 KiB / 1558	
3	9	0	SUCCESS	NODE_LOCAL	1	ip-172-31-61-231.ec2.internal	stderr stdout	2025-12-16 16:19:51	0.2 s		173.8 KiB / 1557	
4	10	0	SUCCESS	RACK_LOCAL	2	ip-172-31-52-86.ec2.internal	stderr stdout	2025-12-16 16:19:51	2 s	0.1 s	174.4 KiB / 1560	

Exercise 2

Q1. Please put your code here:

```
import sys
from pyspark import SparkContext, SparkConf

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: wordcount <input_folder>", file=sys.stderr)
        sys.exit(-1)

    conf = SparkConf().setAppName("python-word-count")
    sc = SparkContext(conf=conf)

    text_file = sc.textFile("hdfs://" + sys.argv[1])

    raw_words = text_file.flatMap(lambda line: line.split(" ")).cache()
    distinct_words_count = raw_words.distinct().count()

    counts = raw_words.map(lambda word: (word, 1)) \
        .reduceByKey(lambda a, b: a + b) \
        .repartition(5) \
        .filter(lambda x: len(x[0]) > 5)

    list = counts.takeOrdered(40, key=lambda x: -x[1])

    print("--------------------------------------------")
    print(*list, sep="\n")
    print("--------------------------------------------")
    print("Total distinct words (all words):", distinct_words_count)
```

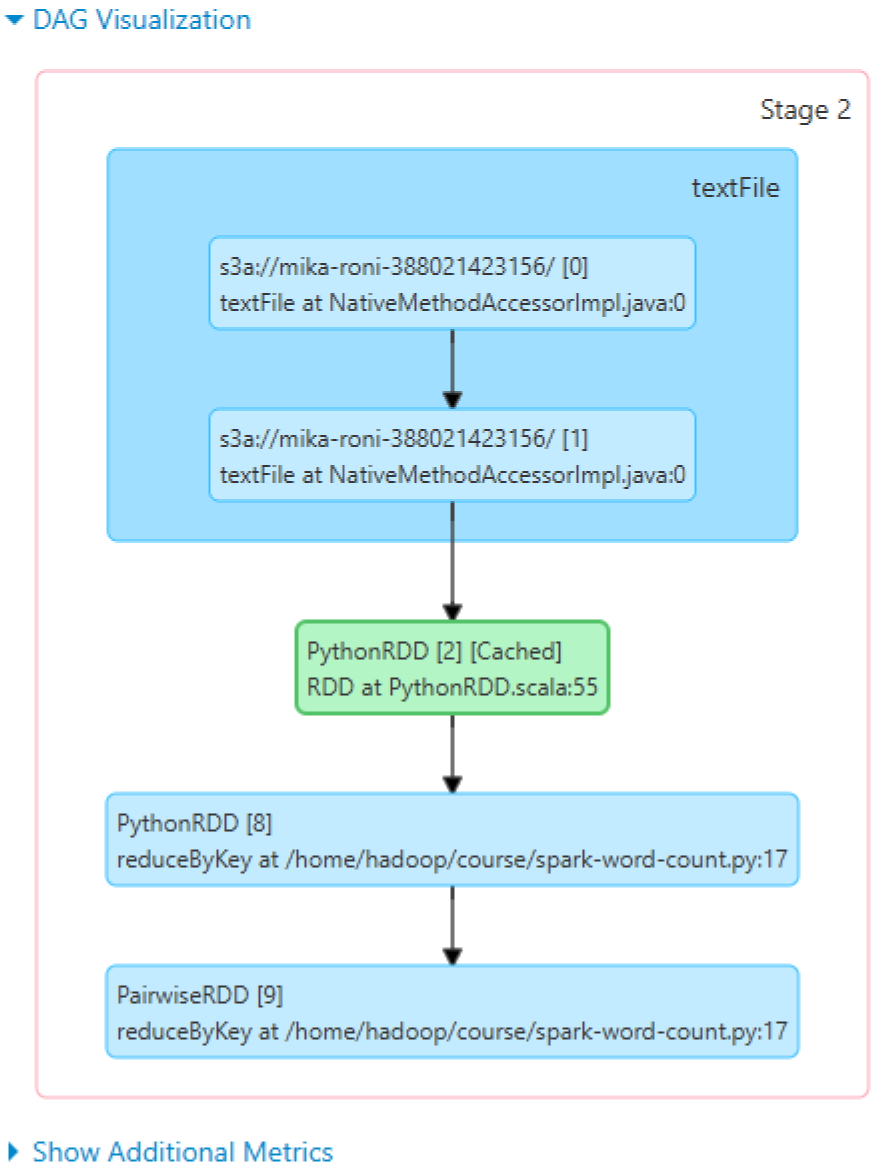
Q2. Write the number of words found



```
('understand', 460)
------------------------------------------------
Total distinct words (all words): 77928
25/12/16 14:58:09 INFO SparkContext: Invoki
```

Exercise 3

Put a print-scrin with the DAG of the first stage, which shows it reads the files from s3a://<your_bucket_name>

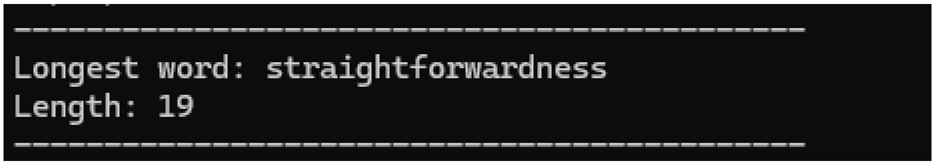

DAG Visualization
Stage 2
textFile
s3a://mika-roni-388021423156/ [0]
textFile at NativeMethodAccessorImpl.java:0
s3a://mika-roni-388021423156/ [1]
textFile at NativeMethodAccessorImpl.java:0
PythonRDD [2] [Cached]
RDD at PythonRDD.scala:55
PythonRDD [8]
reduceByKey at /home/hadoop/course/spark-word-count.py:17
PairwiseRDD [9]
reduceByKey at /home/hadoop/course/spark-word-count.py:17
Show Additional Metrics

Exercise 4

Q1. Please put your code here:

```
import sys
from pyspark import SparkContext, SparkConf

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: wordcount <input_folder>", file=sys.stderr)
        sys.exit(-1)

    conf = SparkConf().setAppName("python-word-count")
    sc = SparkContext(conf=conf)

    text_file = sc.textFile("s3a://" + sys.argv[1])

    words = text_file.flatMap(lambda line: line.split(" ")) \
                     .map(lambda w: w.rstrip('.,')) \
                     .filter(lambda w: w.isalpha())

            longest_word = words.reduce(lambda w1, w2: w1 if len(w1) >= len(w2) else w2)

            print("------------------------------------------")
            print("Longest word:", longest_word)
            print("Length:", len(longest_word))
            print("------------------------------------------")
```

Q2. Put here the printout of the longest word:



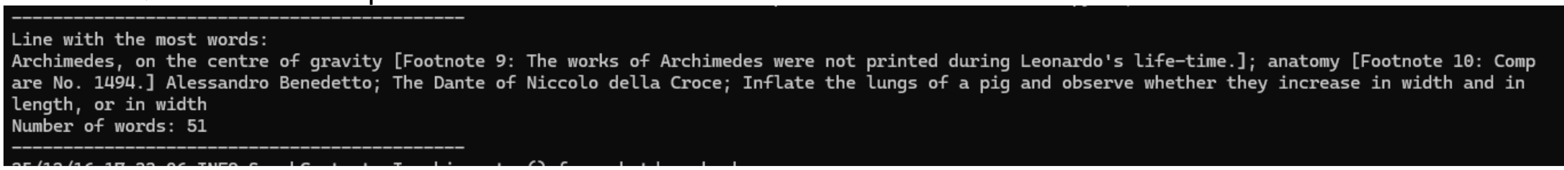
```
--------------------------------------------
Longest word: straightforwardness
Length: 19
--------------------------------------------
```

Exercise 5

Q1. Please put your code here:

```
import sys
from pyspark import SparkContext, SparkConf

def count_words_in_line(text_file):
    lines_with_counts = text_file.map(lambda line: (line, len(line.split(" "))))

    max_line = lines_with_counts.reduce(lambda l1, l2: l1 if l1[1] >= l2[1] else l2)

    return max_line

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: count_words_in_line <s3_path>", file=sys.stderr)
        sys.exit(-1)

    conf = SparkConf().setAppName("count-words-in-line")
    sc = SparkContext(conf=conf)

    text_file = sc.textFile("s3a://" + sys.argv[1])

    max_line = count_words_in_line(text_file)

    print("-------------------------------------------")
    print("Line with the most words:")
    print(max_line[0])
    print("Number of words:", max_line[1])
    print("-------------------------------------------")
```

Q2. Put here the printout of the line with the most words:



```
--------------------------------------------
Line with the most words:
Archimedes, on the centre of gravity [Footnote 9: The works of Archimedes were not printed during Leonardo's life-time.]; anatomy [Footnote 10: Comp
are No. 1494.] Alessandro Benedetto; The Dante of Niccolo della Croce; Inflate the lungs of a pig and observe whether they increase in width and in
length, or in width
Number of words: 51
--------------------------------------------
```