Exercise 2

Wordcount on the cluster

Getting ready!

Creating hdfs files and dirs:

We first ran `hdfs dfs -mkdir -p wordcount/input` to create an input directory for the distributed file system.

If we run `hdfs dfs -ls /user/user_lsc_3/wordcount` we get the following output

```
drwxr-xr-x   - user_lsc_3 hadoop          0 2021-11-24 20:04 /user/user_lsc_3/wordcount/input
drwxr-xr-x   - user_lsc_3 hadoop          0 2021-11-24 20:07 /user/user_lsc_3/wordcount/output
```

If we run `hdfs dfs -ls /user/user_lsc_3/` we get the following output:

```
drwxr-xr-x   - user_lsc_3 hadoop          0 2021-11-24 20:07 /user/user_lsc_3/wordcount
```

Loading data into the distributed file system

we ran `hdfs dfs -put data/* /user/user_lsc_3/wordcount/input`

```
put: `/user/user_lsc_3/wordcount/input/hp1.txt': File exists
put: `/user/user_lsc_3/wordcount/input/hp2.txt': File exists
put: `/user/user_lsc_3/wordcount/input/hp3.txt': File exists
put: `/user/user_lsc_3/wordcount/input/hp4.txt': File exists
```

and then check `hdfs dfs -ls /user/<user lsc i>/wordcount/input`

```
Found 4 items
-rw-r--r--   1 user_lsc_3 hadoop     448798 2021-11-24 20:04 /user/user_lsc_3/wordcount/input/hp1.txt
-rw-r--r--   1 user_lsc_3 hadoop     498919 2021-11-24 20:04 /user/user_lsc_3/wordcount/input/hp2.txt
-rw-r--r--   1 user_lsc_3 hadoop     622055 2021-11-24 20:04 /user/user_lsc_3/wordcount/input/hp3.txt
-rw-r--r--   1 user_lsc_3 hadoop    1120392 2021-11-24 20:04 /user/user_lsc_3/wordcount/input/hp4.txt
```

Ok everything is ready to do some data computations!

runnning tasks

```
hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-*streaming*.jar -mapper mapper.py -file ./mapper.py -r
```

At this point we got some java map runner exceptions so we fell back to the python 2 reducer and mapper. Using those files provided with the lab we got the following statistics:

```
File System Counters
        FILE: Number of bytes read=4503824
        FILE: Number of bytes written=10348456
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=2690708
        HDFS: Number of bytes written=393347
        HDFS: Number of read operations=17
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0
Job Counters
        Launched map tasks=4
        Launched reduce tasks=1
        Rack-local map tasks=4
        Total time spent by all maps in occupied slots (ms)=6237
        Total time spent by all reduces in occupied slots (ms)=3070
        Total time spent by all map tasks (ms)=6237
        Total time spent by all reduce tasks (ms)=3070
        Total vcore-milliseconds taken by all map tasks=6237
        Total vcore-milliseconds taken by all reduce tasks=3070
        Total megabyte-milliseconds taken by all map tasks=6386688
        Total megabyte-milliseconds taken by all reduce tasks=3143680
Map-Reduce Framework
        Map input records=38481
        Map output records=467511
        Map output bytes=3568796
        Map output materialized bytes=4503842
        Input split bytes=544
        Combine input records=0
        Combine output records=0
        Reduce input groups=37329
        Reduce shuffle bytes=4503842
        Reduce input records=467511
        Reduce output records=37329
        Spilled Records=935022
        Shuffled Maps =4
        Failed Shuffles=0
        Merged Map outputs=4
        GC time elapsed (ms)=120
        CPU time spent (ms)=4220
        Physical memory (bytes) snapshot=1390817280
        Virtual memory (bytes) snapshot=14012624896
        Total committed heap usage (bytes)=1452802048
        Peak Map Physical memory (bytes)=296632320
        Peak Map Virtual memory (bytes)=2806906880
        Peak Reduce Physical memory (bytes)=209199104
        Peak Reduce Virtual memory (bytes)=2806427648
Shuffle Errors
```

```
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2690164
        File Output Format Counters
                Bytes Written=393347
```

we then wanted to see some results from the reduce function so we ran `hdfs dfs -text wordcount/output/* | tail` to have a taste:

```
  you've 1
  ??      767
  ??!"    2
  ??"     339
  ??'     1
  ??."    1
  ???"    14
  ???"    1
  ??????????      1
  ??????????BY      1
```

not very satisfying as some characters may have not be in the right encoding. We guess UTF-8 encoding is needed there.
Also the python programs we had from the lab do not take account for the encoding. so we tryied to see the head:

```
  !       1
  !"      1
  "       101
  ""You're        1
  ""as    1
  "'      2
  "'.     1
  "'A     2
  "'Anything      1
  "'Arry!"        1
```

this characters are recognized but are no really words, but still looks promising.

to get the full output file and sort it out we have to do a merger among the dfs output:

```
hdfs dfs -getmerge wordcount/output/ ./output.txt
cat output.txt | sort -k2,2 -nr | head

the     20211
to      11034
and     10741
of      8683
a       8671
was     6655
his     5875
he      5790
Harry   5244
said    5203
```

In the harry potter books the word Harry appears `5244` times!

Better way of splitting words

we found many words with puntuation attached, so we wanted to clean it a bit and used a regex matcher in the mapper instead of the one gave in the zip file, we also converted all words to lower case :

```
#!/usr/bin/env python
"""mapper.py"""

import sys
import re

# input comes from STDIN (standard input)
for line in sys.stdin:
    # # remove leading and trailing whitespace
    # line = line.strip()
    # split the line into words
    words = re.split(r"[^A-Za-z]+", line.strip().lower())
    # increase counters
    for word in words:
        if len(word) > 0:
            # write the results to STDOUT (standard output);
            # what we output here will be the input for the
            # Reduce step, i.e. the input for reducer.py
            #
            # tab-delimited; the trivial word count is 1
            print '%s\t%s' % (word, 1)
```

in this way we get rid of the dirt and count is more precise:

```
[user_lsc_3@it EX2]$ head output.txt
a       9229
aaaaaaaaaargh   1
aaaaaaaarrrrrgh 1
aaaaaaand       1
aaaaaaarrrgh    1
aaaaaand        1
aaaaah  1
aaaaahed        1
aaaah   1
aaaargh 1
```