Practice 1

Word Count

Problem

- ➢ Separate sentences into words and count how many each words are, using Multi-threading method.

"A long time ago in a galaxy far far away"

A	: 1
long	: 1
time	: 1
ago	: 1
in	: 1
a	: 1
galaxy	: 1
far	: 2
away	: 1

- ➢ Data("testfile1.txt", "testfile2.txt", "LargeTextfile.txt") are provided on I-Campus
- ➢ You should submit the results of applying word count results to "LargeTextfile.txt" and report the time between when you use only 1 core and when you use maximum core.

Datatset

1. **The datasets named with "testfile1.txt" and "testfile2.txt" have short simple sentences(you can check whether your code is able to run without problem.**

   Testfile1.txt: "A long time ago in a galaxy far far away"

   Testfile2.txt: "Another episode of Star Wars"

2. **The "LargeTextfile.txt" data is the novel "Animal Farm". The capacity of this data is 1.15Gb, which is very heavy.**

   (We made this data bigger replicating the novel "Animal Farm" 10 times)

3. **So if you don't use maximum cores/threads then, you may run the Hadoop File System for a long time.**

Practice 1

1. Make Python code ‘mapper.py’ and ‘reducer.py’ and save in your directory

   mapper.py : Separate sentences into words

   reducer.py : Count the number of words

   NOTE: There must be NO SPACE in your directory

2. Use “sys.stdin” for processing input sentence

   Import sys

3. Input and output file should be processed in HDFS

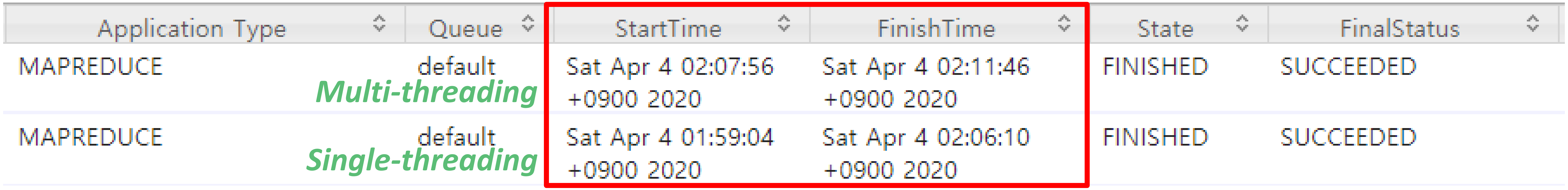
Practice 1

4. You can use Multi-threading method if data is very large.
   - Since, mapper in hdfs uses full cores but reducer uses only one core automatically.
   - So, *if you want to use full cores during reduce process*, use ***-numReduceTasks*** argument in your command line. This argument has value 1 as default, it means you will use only one core.
   - You can set this number as your maximum number of core.
   - *So test running time when you use single core or maximum number of cores(In our case we have the 8 cores).*

Please refer to our practice1_solution pdf file

Submission

- *If you run word count example without problem, you can get result file in hdfs.*
- *Go localhost:50070, then click “utilities” and “Browse the file system”*
- *Click “output” and download “part-00000” .*
- *And you need to submit screenshot of time difference, after you use different number of cores.*
- *For example,*



Application Type	Queue	StartTime	FinishTime	State	FinalStatus
MAPREDUCE	default *Multi-threading*	Sat Apr 4 02:07:56 +0900 2020	Sat Apr 4 02:11:46 +0900 2020	FINISHED	SUCCEEDED
MAPREDUCE	default *Single-threading*	Sat Apr 4 01:59:04 +0900 2020	Sat Apr 4 02:06:10 +0900 2020	FINISHED	SUCCEEDED

- *Submit YOUR_STUDENT_ID.zip file which includes part-00000 and your screenshot on I-Campus*
- *Submission deadline: April 16 23:59*

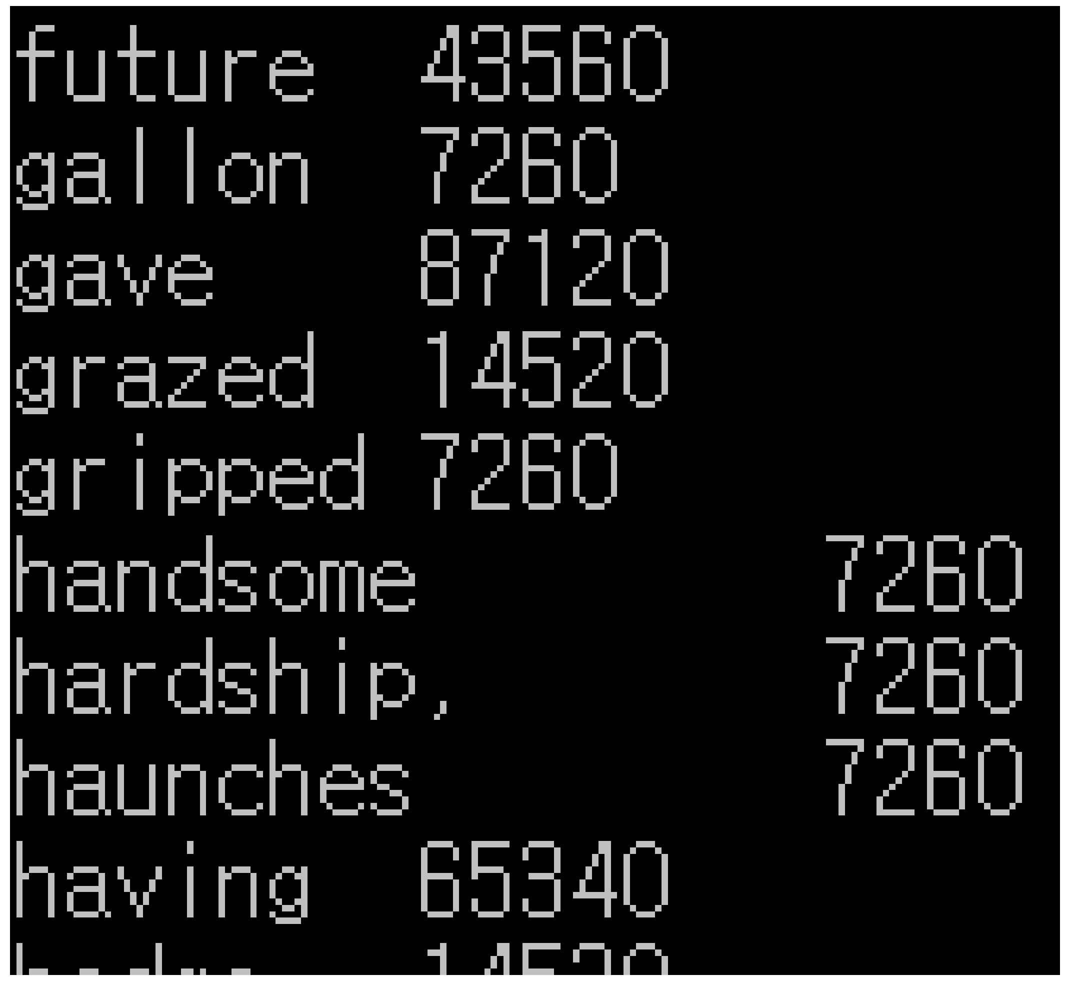
Submission

- *You can see your* ***part-00000*** *file with the following command*

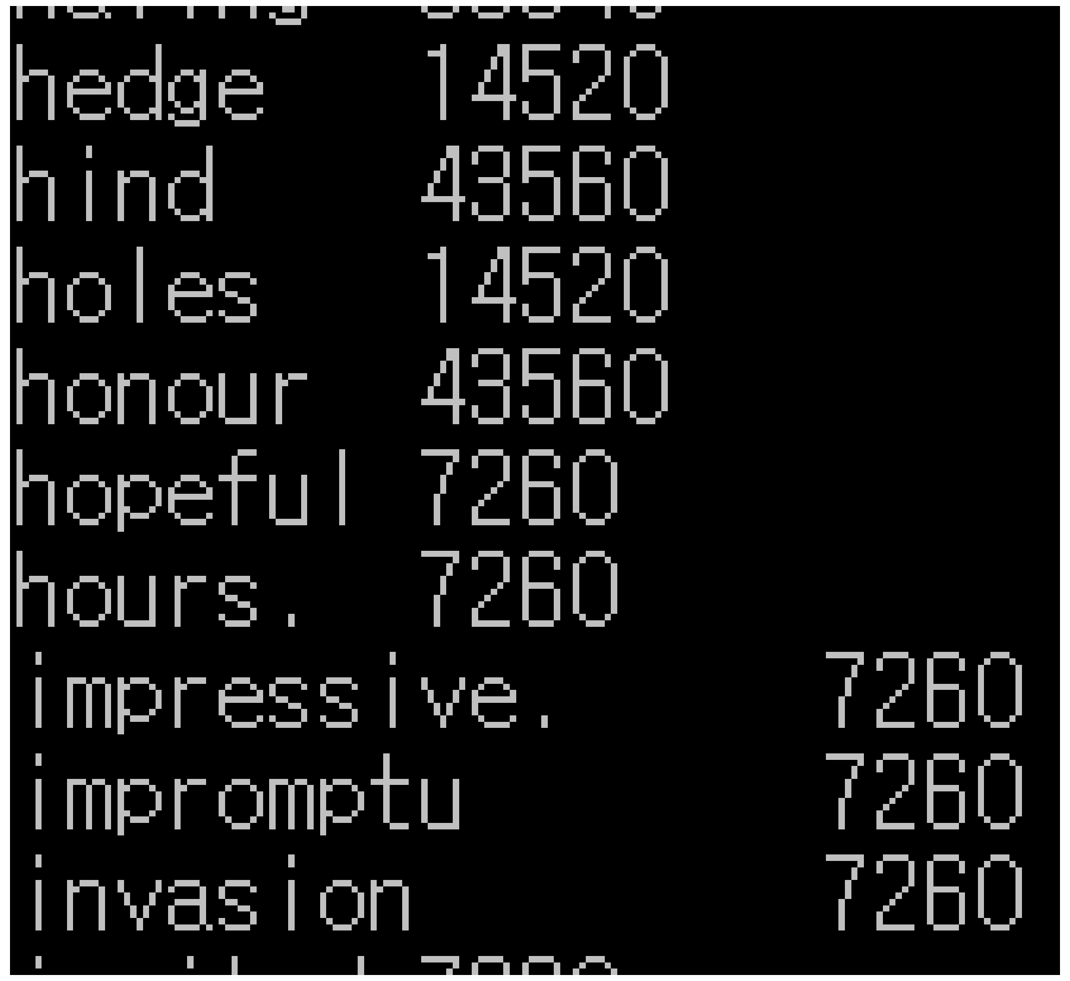
  hdfs dfs -cat part-00000 (Windows)

  sudo $HADOOP_HOME/bin/hdfs dfs -cat /output/part-00000 (Linux)

- *Your result file "part-00000" should be as follows.*



```
future  43560
gallon  7260
gave    87120
grazed  14520
gripped 7260
handsome        7260
hardship,       7260
haunches        7260
having  65340
```



```
hedge   14520
hind    43560
holes   14520
honour  43560
hopeful 7260
hours.  7260
impressive.     7260
impromptu       7260
invasion        7260
```

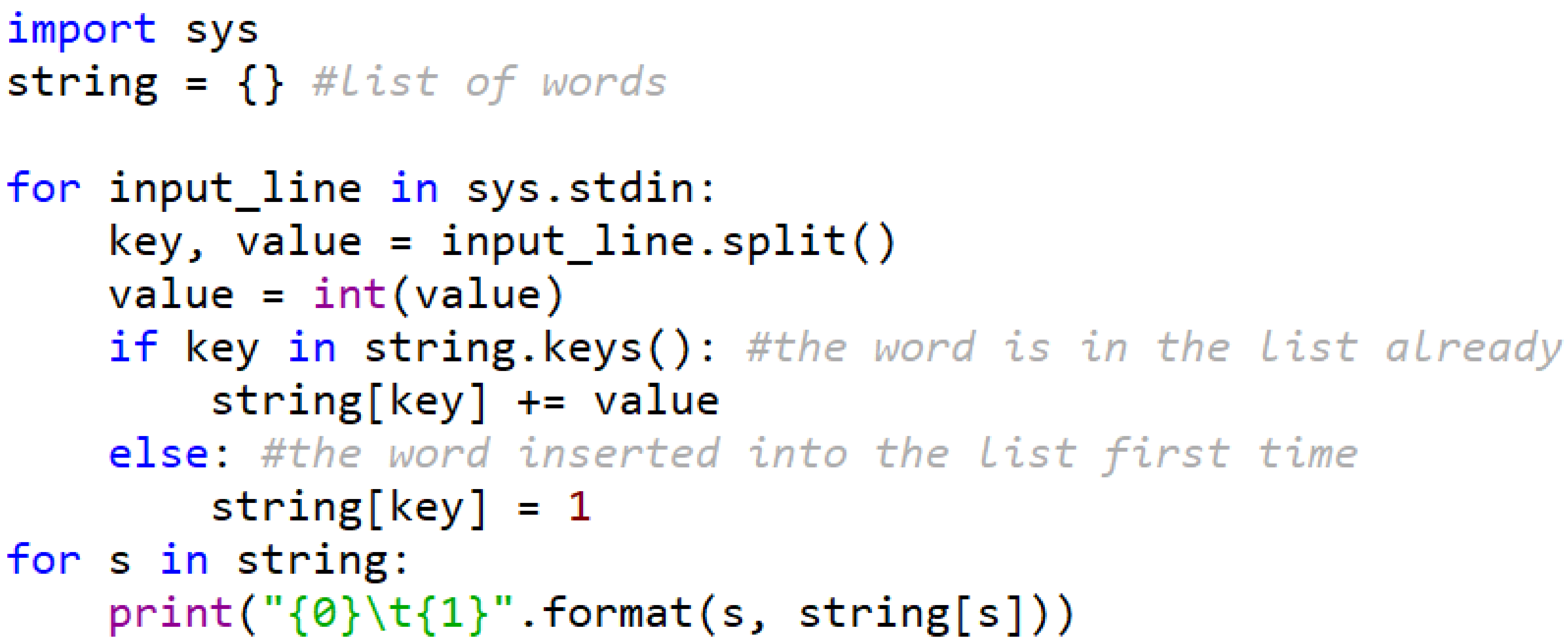
Solution - Python code

- **mapper.py**

```
import sys
for line in sys.stdin:
    line = line.strip() #strip the carrage return (by default)
    keys = line.split() #split line at blanks (by default)
    for key in keys:
        value = 1
        print('{0}\t{1}'.format(key, value))
        #note that the Hadoop default is 'tab' separates key from the value
```

Solution - Python code

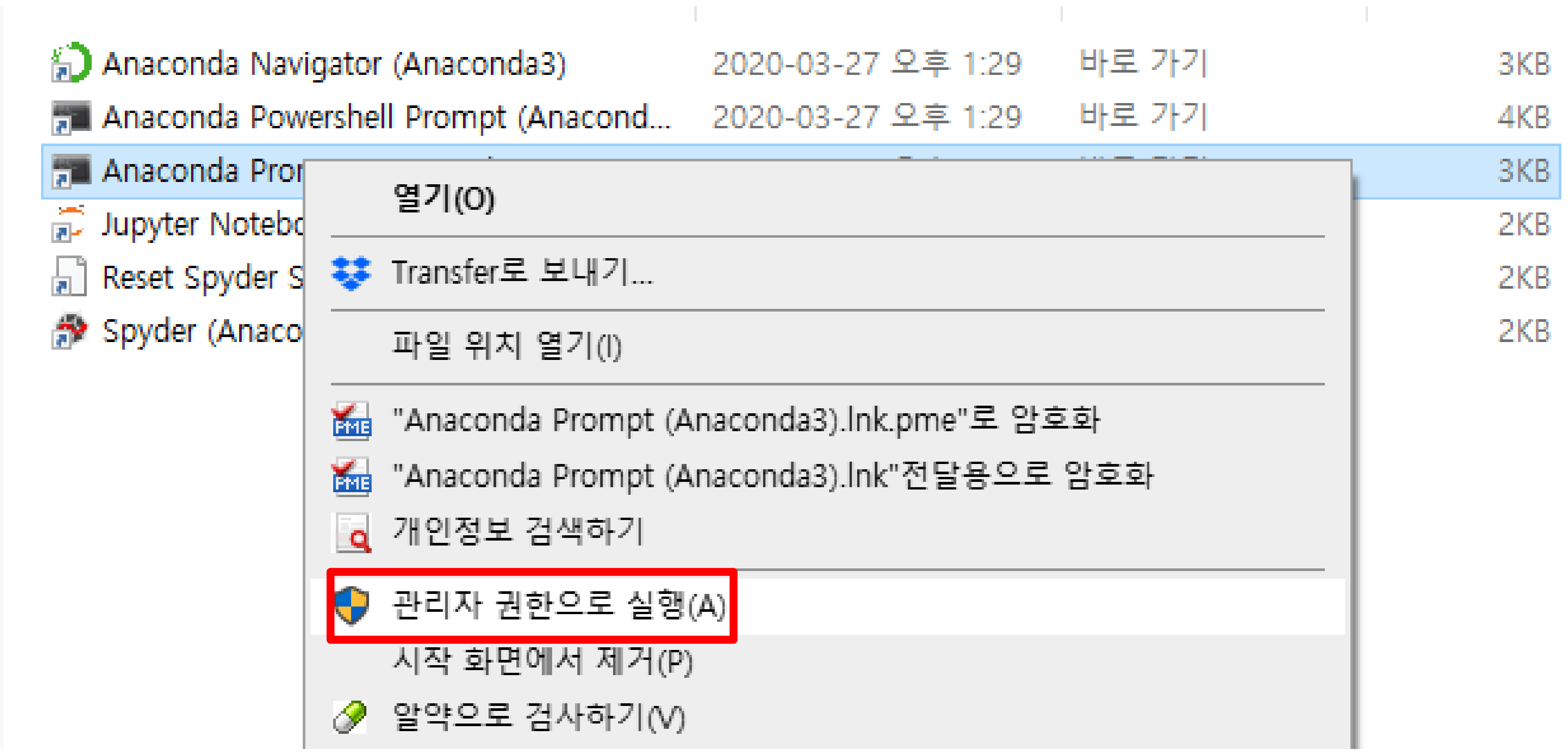
- **reducer.py**



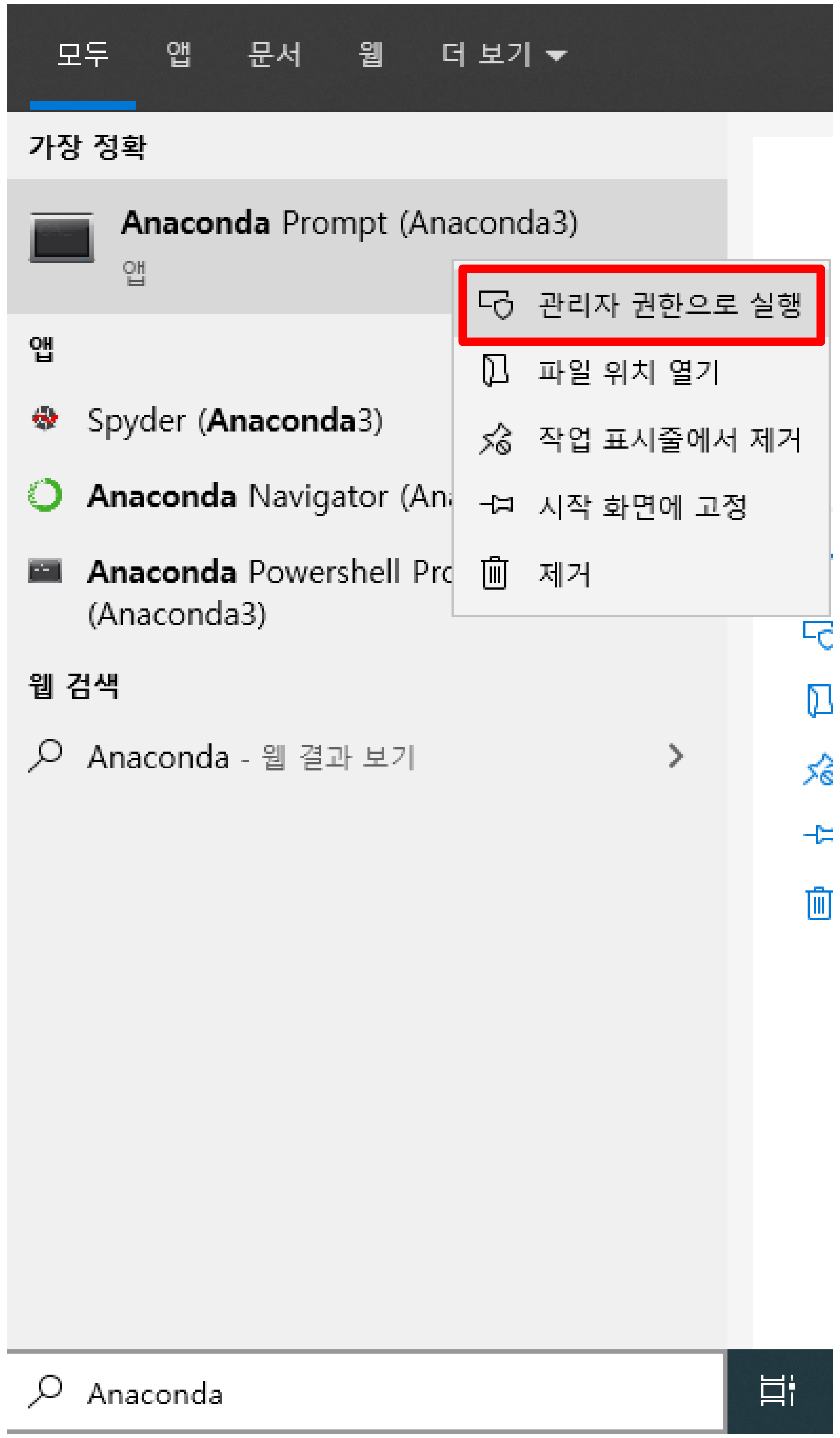
```python
import sys
string = {} #list of words

for input_line in sys.stdin:
    key, value = input_line.split()
    value = int(value)
    if key in string.keys(): #the word is in the list already
        string[key] += value
    else: #the word inserted into the list first time
        string[key] = 1
for s in string:
    print("{0}\t{1}".format(s, string[s]))
```

Solution - Mapreduce (Windows)

- Open Anaconda Prompt as admin (click right button)
- We recommend to work in other folder rather than system32



Anaconda Navigator (Anaconda3) 2020-03-27 오후 1:29 바로 가기 3KB
Anaconda Powershell Prompt (Anacond... 2020-03-27 오후 1:29 바로 가기 4KB
Anaconda Pror 3KB
Jupyter Notebo 2KB
Reset Spyder S 2KB
Spyder (Anaco 2KB
열기(O)
Transfer로 보내기...
파일 위치 열기(I)
"Anaconda Prompt (Anaconda3).lnk.pme"로 암호화
"Anaconda Prompt (Anaconda3).lnk"전달용으로 암호화
개인정보 검색하기
관리자 권한으로 실행(A)
시작 화면에서 제거(P)
알약으로 검사하기(V)



Solution - Mapreduce (Windows)

You must keep executing yarn and dfs windows during this tutorial

- **In your directory where mapper.py & reducer.py are located, create text files for word counting test with the following command**

echo A long time ago in a galaxy far far away > testfile1.txt

echo Another episode of Star Wars > testfile2.txt

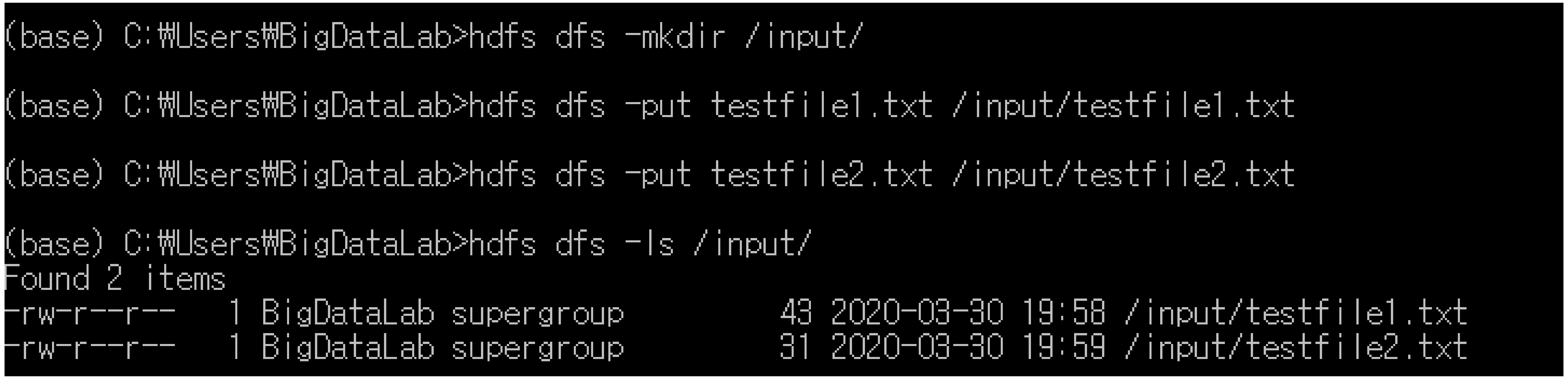
- **Move text files to HDFS from local**

hdfs dfs -mkdir /input/

hdfs dfs -put testfile1.txt /input/testfile1.txt

hdfs dfs -put testfile2.txt /input/testfile2.txt

hdfs dfs -ls /input/



```
(base) C:₩Users₩BigDataLab>hdfs dfs -mkdir /input/

(base) C:₩Users₩BigDataLab>hdfs dfs -put testfile1.txt /input/testfile1.txt

(base) C:₩Users₩BigDataLab>hdfs dfs -put testfile2.txt /input/testfile2.txt

(base) C:₩Users₩BigDataLab>hdfs dfs -ls /input/
Found 2 items
-rw-r--r--   1 BigDataLab supergroup         43 2020-03-30 19:58 /input/testfile1.txt
-rw-r--r--   1 BigDataLab supergroup         31 2020-03-30 19:59 /input/testfile2.txt
```

Solution - Mapreduce (Windows)

- Change text file to execute mode

```
hdfs dfs -chmod +x /input/testfile1.txt
hdfs dfs -chmod +x /input/testfile2.txt
```

- Execute Mapreduce code

```
hadoop jar %HADOOP_HOME%\share\hadoop\tools\lib\hadoop-streaming-2.7.1.jar -input /input
-output /output -mapper "python YOUR_DIRECTORY\mapper.py" -reducer "python
YOUR_DIRECTORY\reducer.py"
```

NOTE: execute command is only one line

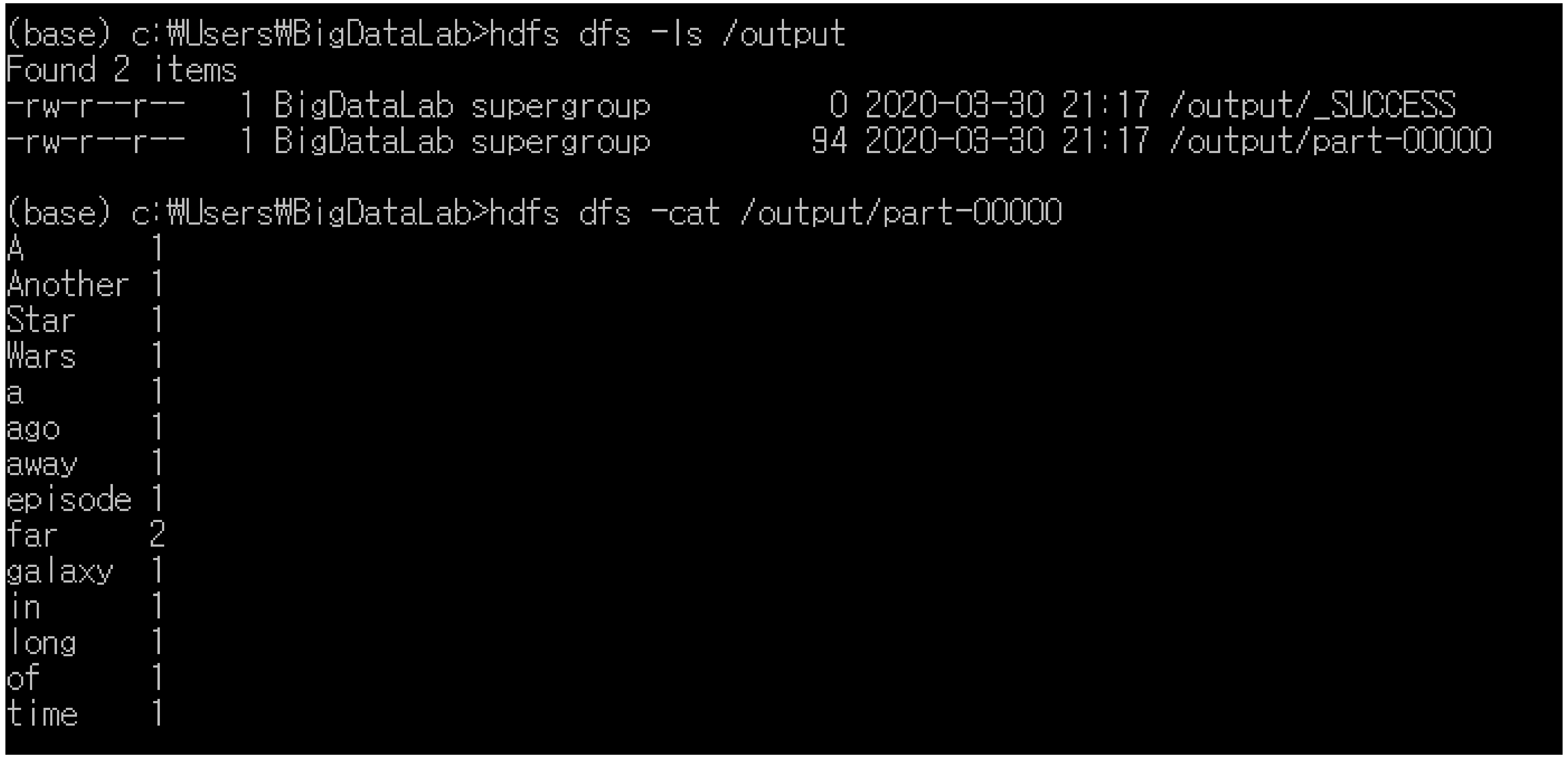
Solution - Mapreduce (Windows)

- **Check the result with the following command**

hdfs dfs -ls /output

hdfs dfs -cat /output/part-00000

- **Then you can see the following results.**



```
(base) c:₩Users₩BigDataLab>hdfs dfs -ls /output
Found 2 items
-rw-r--r--   1 BigDataLab supergroup          0 2020-03-30 21:17 /output/_SUCCESS
-rw-r--r--   1 BigDataLab supergroup         94 2020-03-30 21:17 /output/part-00000

(base) c:₩Users₩BigDataLab>hdfs dfs -cat /output/part-00000
A       1
Another 1
Star    1
Wars    1
a       1
ago     1
away    1
episode 1
far     2
galaxy  1
in      1
long    1
of      1
time    1
```

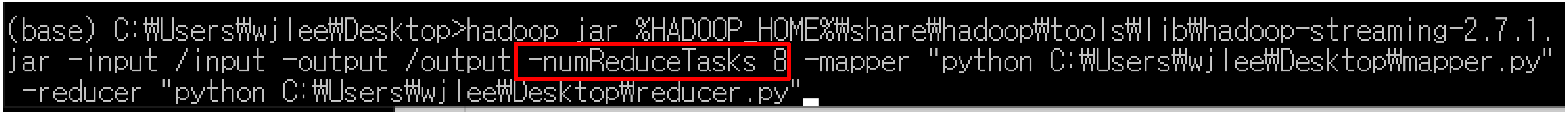
Solution - Mapreduce (Windows)

➢ Multi-threading

- You can use Multi-threading method only add *-numReduceTasks* argument in your command line like following.

```
hadoop jar %HADOOP_HOME%\share\hadoop\tools\lib\hadoop-streaming-2.7.1.jar -input /input
-output /output –numReduceTasks NUMBER OF CORES -mapper "python
YOUR_DIRECTORY\mapper.py" -reducer "python YOUR_DIRECTORY\reducer.py"
```

NOTE: execute command is only one line



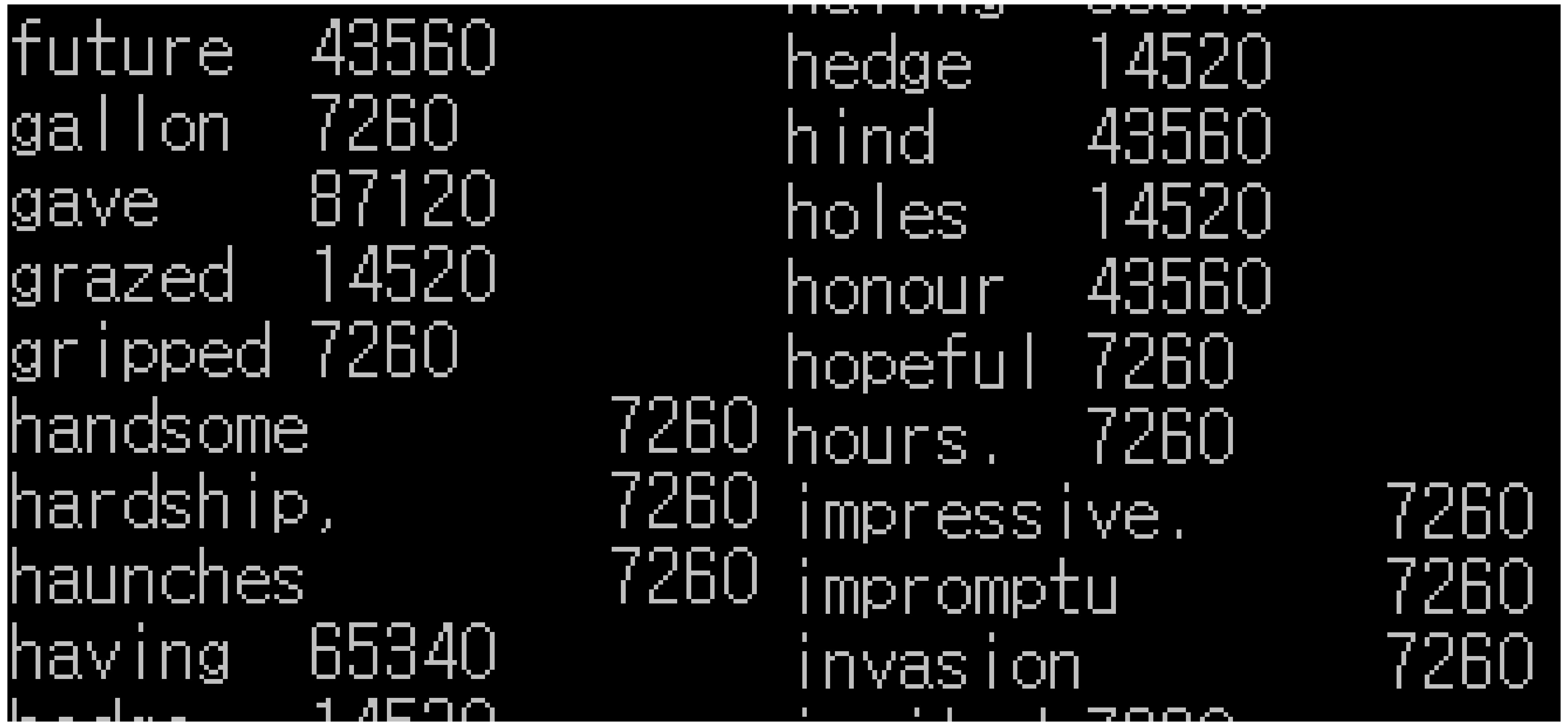
```
(base) C:₩Users₩wjlee₩Desktop>hadoop jar %HADOOP_HOME%₩share₩hadoop₩tools₩lib₩hadoop-streaming-2.7.1.
jar -input /input -output /output -numReduceTasks 8 -mapper "python C:₩Users₩wjlee₩Desktop₩mapper.py"
 -reducer "python C:₩Users₩wjlee₩Desktop₩reducer.py"
```

Solution - Mapreduce (Windows)

➢ Submission

- You need to submit the *YOUR_STUDENT_ID.zip* file include “*part-00000*” and *screenshot* of time difference, after you use different number of cores.
- For example,

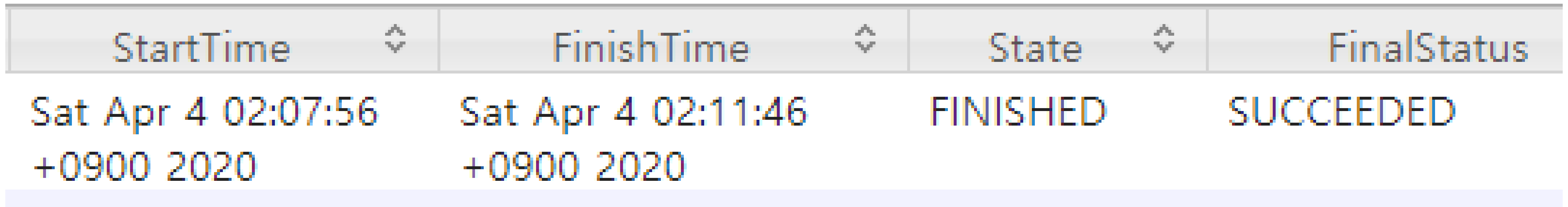
part-00000



```
future   43560              hedge    14520
gallon   7260               hind     43560
gave     87120              holes    14520
grazed   14520              honour   43560
gripped  7260               hopeful  7260
handsome          7260      hours.   7260
hardship,         7260      impressive.       7260
haunches          7260      impromptu         7260
having   65340              invasion          7260
```

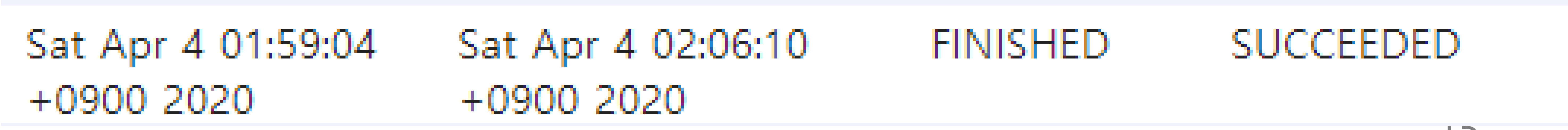
screenshot

Multi-threading



StartTime	FinishTime	State	FinalStatus
Sat Apr 4 02:07:56 +0900 2020	Sat Apr 4 02:11:46 +0900 2020	FINISHED	SUCCEEDED

Single-threading



Sat Apr 4 01:59:04 +0900 2020	Sat Apr 4 02:06:10 +0900 2020	FINISHED	SUCCEEDED

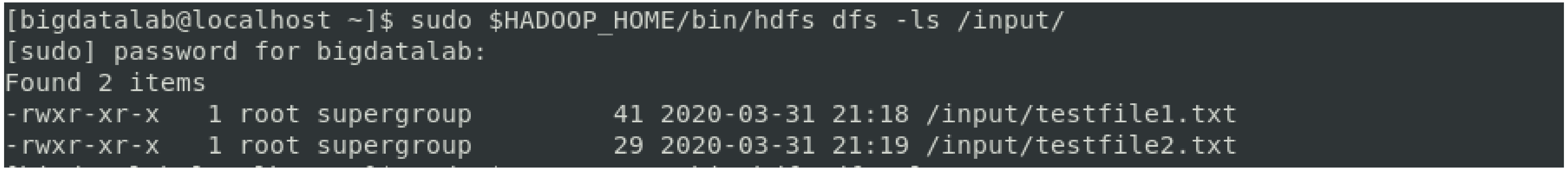
Solution - Mapreduce (Linux)

- In your directory where mapper.py & reducer.py are located, create text files for word counting test with the following command

```
echo “A long time ago in a galaxy far far away”>> testfile1.txt
echo “Another episode of Star Wars”>> testfile2.txt
```

- Move text files to HDFS from local

```
sudo $HADOOP_HOME/bin/hdfs dfs -mkdir /input/
sudo $HADOOP_HOME/bin/hdfs dfs -put testfile1.txt /input/testfile1.txt
sudo $HADOOP_HOME/bin/hdfs dfs -put testfile2.txt /input/testfile2.txt
sudo $HADOOP_HOME/bin/hdfs dfs -ls /input/
```



```
[bigdatalab@localhost ~]$ sudo $HADOOP_HOME/bin/hdfs dfs -ls /input/
[sudo] password for bigdatalab:
Found 2 items
-rwxr-xr-x   1 root supergroup         41 2020-03-31 21:18 /input/testfile1.txt
-rwxr-xr-x   1 root supergroup         29 2020-03-31 21:19 /input/testfile2.txt
```

Solution - Mapreduce (Linux)

- **Change text file to execute mode**

```
sudo $HADOOP_HOME/bin/hdfs dfs -chmod +x /input/testfile1.txt
sudo $HADOOP_HOME/bin/hdfs dfs -chmod +x /input/testfile2.txt
```

- **Execute Mapreduce code**

```
sudo $HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-2.7.7.jar -input /input -output /output -mapper “python YOUR_DIRECTORY/mapper.py” -reducer “python YOUR_DIRECTORY/ reducer.py”
```

NOTE: execute command is only one line

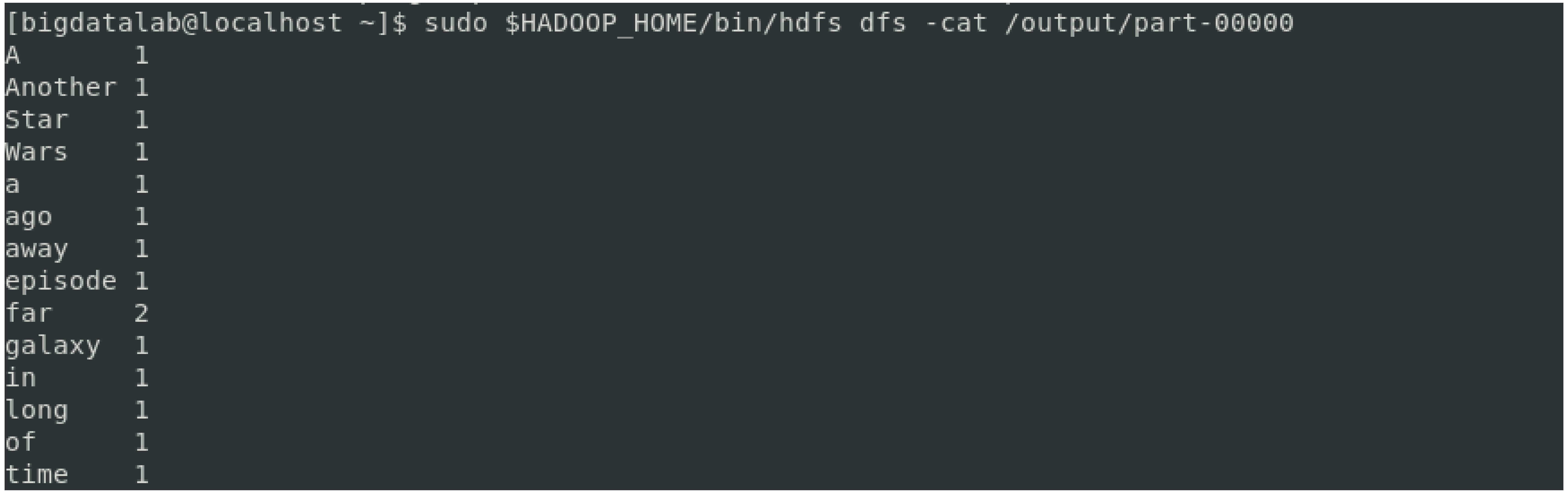
Solution - Mapreduce (Linux)

- ➢ Check the result with the following command

sudo $HADOOP_HOME/bin/hdfs dfs -ls /output

sudo $HADOOP_HOME/bin/hdfs dfs -cat /output/part-00000

- ➢ Then you can see the following results.



```
[bigdatalab@localhost ~]$ sudo $HADOOP_HOME/bin/hdfs dfs -cat /output/part-00000
A       1
Another 1
Star    1
Wars    1
a       1
ago     1
away    1
episode 1
far     2
galaxy  1
in      1
long    1
of      1
time    1
```

Solution - Mapreduce (Linux)

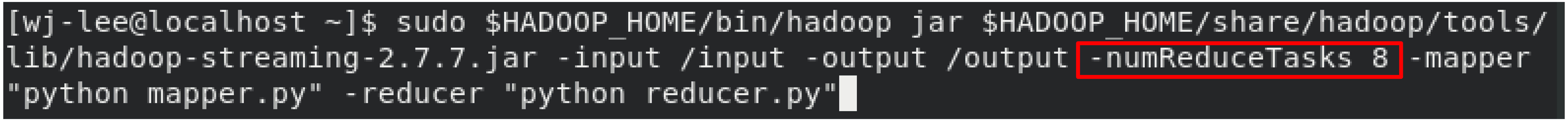
Multi-threading

- You can use Multi-threading method only add ***-numReduceTasks*** argument in your command line like following.

If you use virtual machine, we recommend you reduce size of LargeTextfile.txt to 100Mbs.

```
sudo $HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/share/Hadoop/tools/lib/hadoop-streaming-2.7.7.jar -input /input
-output /output –numReduceTasks NUMBER OF CORES -mapper "python
YOUR_DIRECTORY\mapper.py" -reducer "python YOUR_DIRECTORY\reducer.py"
```

NOTE: execute command is only one line



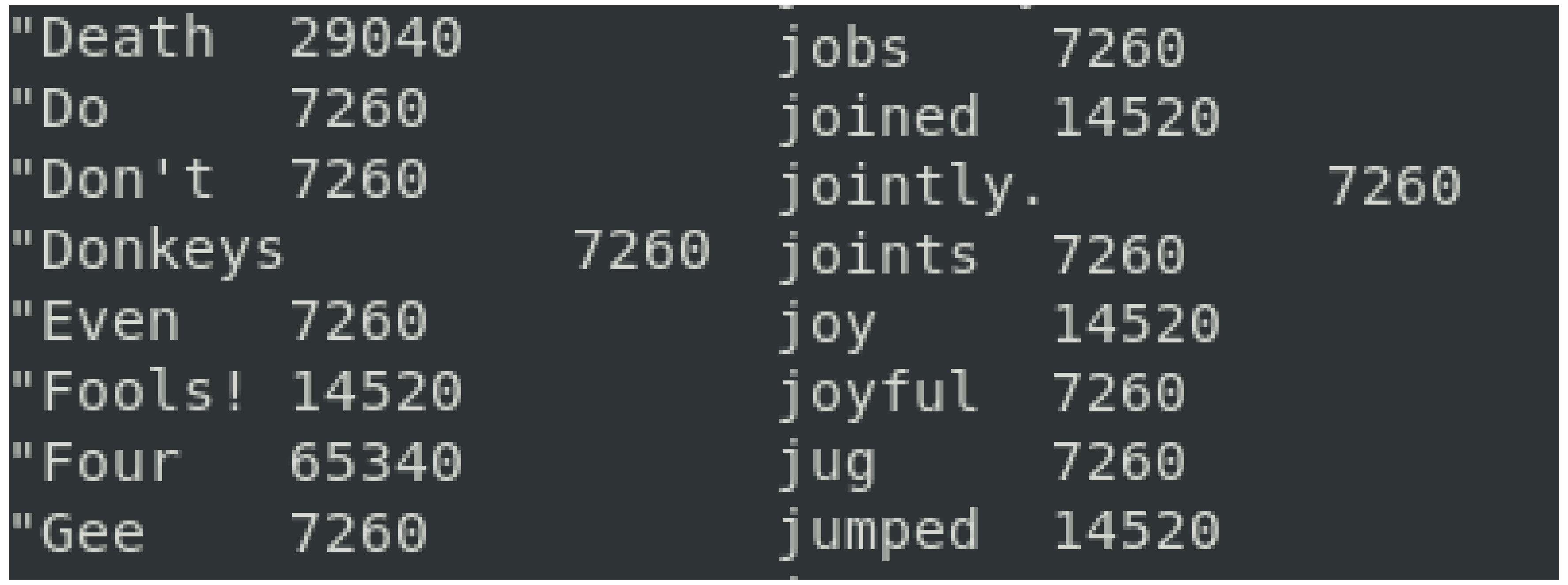
```
[wj-lee@localhost ~]$ sudo $HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/share/hadoop/tools/
lib/hadoop-streaming-2.7.7.jar -input /input -output /output -numReduceTasks 8 -mapper
"python mapper.py" -reducer "python reducer.py"
```

Solution - Mapreduce (Linux)

➢ **Submission**

- You need to submit the *YOUR_STUDENT_ID.zip* file include "*part-00000*" and *screenshot* of time difference, after you use different number of cores.
- For example,

part-00000

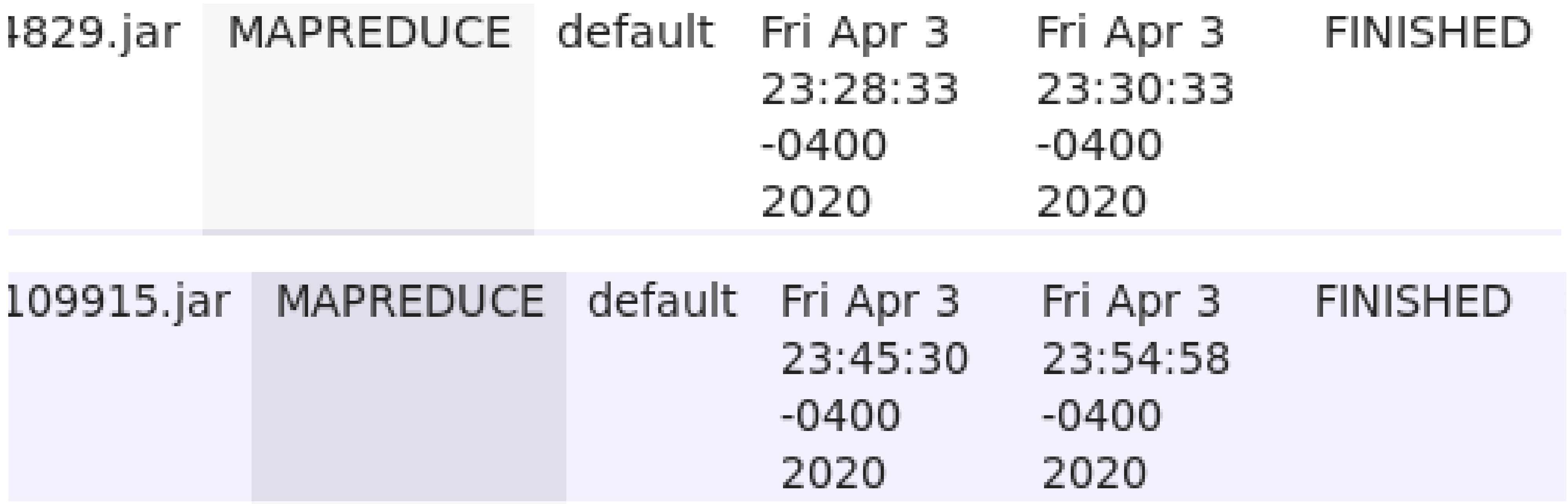


```
"Death   29040          jobs    7260
"Do      7260           joined  14520
"Don't   7260           jointly.        7260
"Donkeys         7260   joints  7260
"Even    7260           joy     14520
"Fools!  14520          joyful  7260
"Four    65340          jug     7260
"Gee     7260           jumped  14520
```

screenshot

Multi-threading

Single-threading



4829.jar	MAPREDUCE	default	Fri Apr 3 23:28:33 -0400 2020	Fri Apr 3 23:30:33 -0400 2020	FINISHED
109915.jar	MAPREDUCE	default	Fri Apr 3 23:45:30 -0400 2020	Fri Apr 3 23:54:58 -0400 2020	FINISHED