TASK

Load the complete Shakespeare writings, strip the header and search for the #24 most used word in his writings. Provide your code in one pdf, txt or by one link!

THE CODE & THE ANSWER

Is public on kaggle: https://www.kaggle.com/kewagbln/shakespeare-word-count-with-spark-python

```
1 : the : 27572
2 : and : 26752
3 : i : 20191
4 : to : 19338
5 : of : 18135
6 : a : 14520
7 : you : 12991
8 : my : 12468
9 : that : 10964
10 : in : 10914
11 : is : 9503
12 : not : 8453
```

```
13 : for : 8215
14 : with : 7973
15 : it : 7224
16 : be : 6979
17 : me : 6962
18 : your : 6875
19 : his : 6825
20 : this : 6299
21 : but : 6272
22 : he : 6102
23 : as : 5934
24 : have : 5845
```

Future improvement: reading the text from line 245, not from 1.

HISTORICAL SETUP DOCUMENATION

SETUP SPARK

TRY: INSTALL ON WINDOWS

https://blog.sicara.com/get-started-pyspark-jupyter-guide-tutorial-ae2fe84f594f

https://medium.com/@GalarnykMichael/install-spark-on-windows-pyspark-4498a5d8d66c

Doesn't work:

```
In [11]: import findspark
         findspark.init()

         import pyspark
         import random

         sc = pyspark.SparkContext(appName="Pi")

         num_samples = 100

         def inside(p):
           x, y = random.random(), random.random()
           return x*x + y*y < 1

         count = sc.parallelize(range(0, num_samples)).filter(inside).count()

         pi = 4 * count / num_samples
         print(pi)

         sc.stop()
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
<ipython-input-11-dda98519d6e9> in <module>
     13   return x*x + y*y < 1
     14
---> 15 count = sc.parallelize(range(0, num_samples)).filter(inside).count()
     16
     17 pi = 4 * count / num_samples

~\Spark\spark-2.4.0-bin-hadoop2.7\python\pyspark\rdd.py in count(self)
   1053         3
   1054         """
-> 1055         return self.mapPartitions(lambda i: [sum(1 for _ in i)]).sum()
   1056
   1057     def stats(self):
```

https://stackoverflow.com/questions/47761758/pyspark-python-issue-py4jjavaerror-an-error-occurred-while-calling-o48-showstr

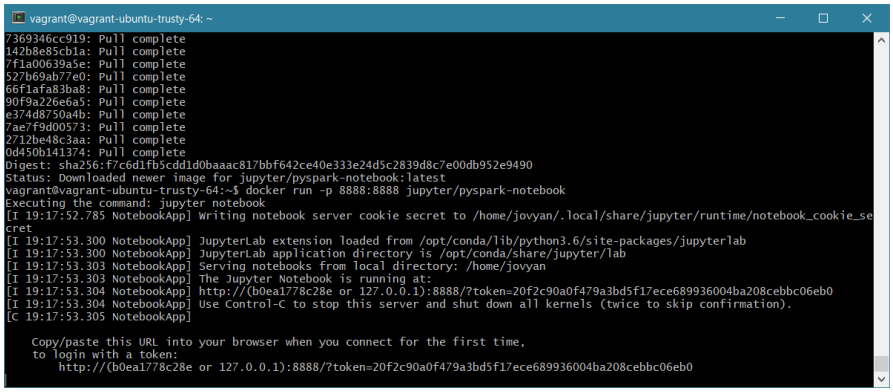
The error is

> Caused by: java.lang.OutOfMemoryError: Java heap space

You need more memory to perform the operations and avoid the OOM error.

TRY: DOCKER – PYSPARK

1. Setup a Vagrant VM: vagrant init
2. Setup a docker in the VM: Refresh the vagrantfile / `vagrant up` / `vagrant ssh`
3. Get an image: `docker pull jupyter/pyspark-notebook`
   https://hub.docker.com/r/jupyter/pyspark-notebook
   More information about the images: https://jupyter-docker-stacks.readthedocs.io/en/latest/using/selecting.html
4. Run docker: `docker run -p 4040:4040 jupyter/pyspark-notebook`
   Installation takes some time with the first run



5. Open notebook in browser:

TRY: DOCKER – GETTYIMAGES/SPARK

1. Setup a Vagrant VM: vagrant init
2. Setup a docker in the VM: Refresh the vagrantfile

   If you just need to access a vagrant machine from only the host machine, setting up a "private network" is all you need. Either uncomment the appropriate line in a default `Vagrantfile`, or add this snippet. If you want your VM to appear at `172.30.1.5` it would be the following:

   ```
   config.vm.network "private_network", ip: "172.30.1.5"
   ```

   192.168.3.10

   https://stackoverflow.com/questions/14870900/how-to-find-the-vagrant-ip

   / `vagrant up`

   ```
   $ vagrant up
   Bringing machine 'default' up with 'virtualbox' provider...
   ==> default: Checking if box 'ubuntu/trusty64' is up to date...
   ==> default: Resuming suspended VM...
   ==> default: Booting VM...
   ==> default: Waiting for machine to boot. This may take a few minutes...
       default: SSH address: 127.0.0.1:2222
       default: SSH username: vagrant
       default: SSH auth method: private key
   ==> default: Machine booted and ready!
   ==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
   ==> default: flag to force provisioning. Provisioners marked to run always will still run.
   ```

   / `vagrant ssh`

   ```
   Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-164-generic x86_64)

    * Documentation:  https://help.ubuntu.com/

     System information as of Mon Feb  4 09:42:40 UTC 2019

     System load:  0.02              Processes:              77
     Usage of /:   6.7% of 39.34GB   Users logged in:        0
     Memory usage: 36%               IP address for eth0:    10.0.2.15
     Swap usage:   0%                IP address for docker0: 172.17.0.1

     Graph this data and manage this system at:
       https://landscape.canonical.com/

     Get cloud support with Ubuntu Advantage Cloud Guest:
       http://www.ubuntu.com/business/services/cloud

   New release '16.04.5 LTS' available.
   Run 'do-release-upgrade' to upgrade to it.


   Last login: Mon Feb  4 09:38:19 2019 from 10.0.2.2
   vagrant@vagrant-ubuntu-trusty-64:~$ docker --version
   Docker version 18.06.1-ce, build e68fc7a
   ```

3. Get an image: `docker pull gettyimages/spark`

   https://hub.docker.com/r/gettyimages/spark/

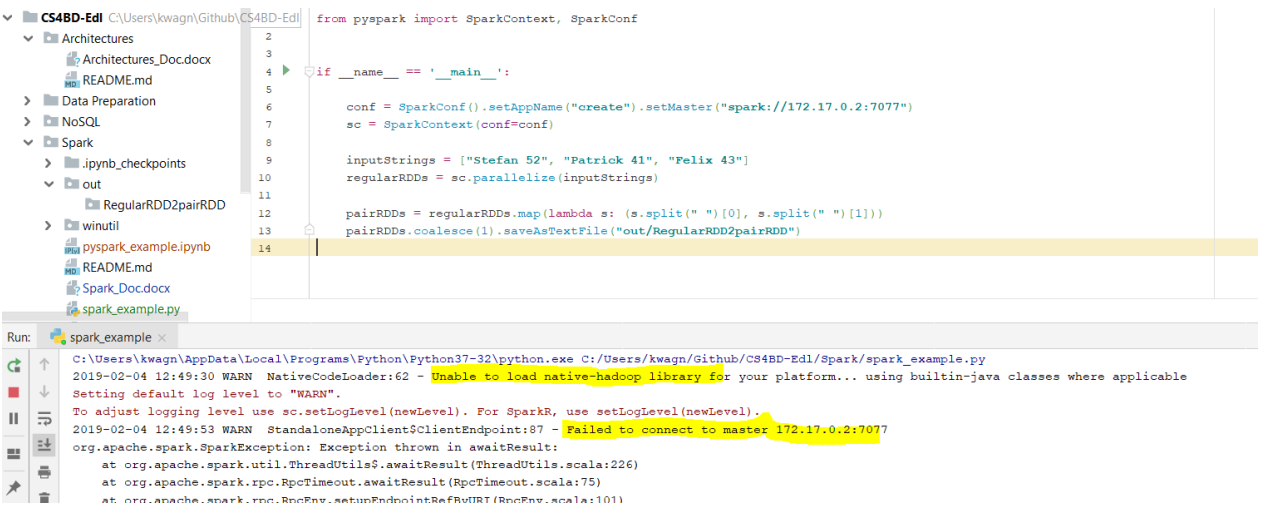
4. **Run an example:** `docker run --rm -it -p 4040:4040 gettyimages/spark bin/run-example SparkPi 10`

```
vagrant@vagrant-ubuntu-trusty-64:~$ docker run --rm -it -p 4040:4040 gettyimages/spark bin/run-example SparkPi 10
2019-02-04 05:29:57 WARN  NativeCodeLoader:60 - Unable to load native-hadoop library for your platform... using builtin-java clas
ses where applicable
2019-02-04 05:30:00 INFO  SparkContext:54 - Running Spark version 2.4.0
2019-02-04 05:30:00 INFO  SparkContext:54 - Submitted application: Spark Pi
2019-02-04 05:30:00 INFO  SecurityManager:54 - Changing view acls to: root
2019-02-04 05:30:00 INFO  SecurityManager:54 - Changing modify acls to: root
```

Run spark

```
vagrant@vagrant-ubuntu-trusty-64:~$ docker run --rm -it -p 4040:4040 gettyimages/spark
2019-02-04 10:26:54 INFO  Master:2566 - Started daemon with process name: 1@4d3607c8a93e
2019-02-04 10:26:54 INFO  SignalUtils:54 - Registered signal handler for TERM
2019-02-04 10:26:54 INFO  SignalUtils:54 - Registered signal handler for HUP
2019-02-04 10:26:54 INFO  SignalUtils:54 - Registered signal handler for INT
2019-02-04 10:26:55 WARN  NativeCodeLoader:60 - Unable to load native-hadoop library for
our platform... using builtin-java classes where applicable
2019-02-04 10:26:55 INFO  SecurityManager:54 - Changing view acls to: root
2019-02-04 10:26:55 INFO  SecurityManager:54 - Changing modify acls to: root
```

Run code – doesn't work:



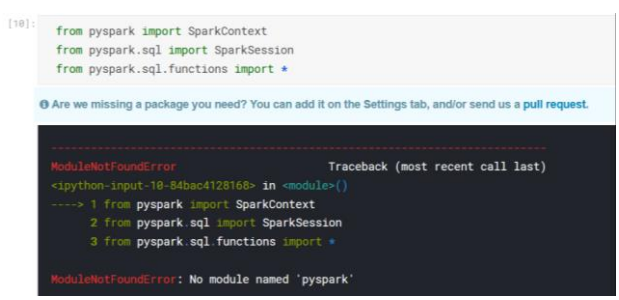
https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-tips-and-tricks-running-spark-windows.html

TRY: KAGGLE

Works in https://www.kaggle.com/kernels/notebooks/new?forkParentScriptVersionId=5495607

```
[8]:
from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
```

But package missed in my notebook:



Add custom package:

Add a custom package

Kaggle Kernels come preloaded with the most popular Python and R packages. If there's a package you'd like to use which isn't preinstalled, you can add it here.

pyspark | apache/spark

In progress...

Custom packages

No custom packages installed

Failed:

Add a custom package

Kaggle Kernels come preloaded with the most popular Python and R packages. If there's a package you'd like to use which isn't preinstalled, you can add it here.

pyspark | apache/spark

Failed. ✖

Custom packages

No custom packages installed

Next try – pyspark installation:

```
[*]: !pip install pyspark

Collecting pyspark
  Downloading https://files.pythonhosted.org/packages/88/01/a37e827c2d80c6a754e40e99b9826d978b55254cc6c6672b5b08f2e18a7f/pyspark-2.4.0.tar.gz (213.4MB)
    100% |[redacted]| 213.4MB 93kB/s  eta 0:00:011B 53.7MB/s eta 0:00:02
Collecting py4j==0.10.7 (from pyspark)
  Downloading https://files.pythonhosted.org/packages/e3/53/c737818eb9a7dc32a7cd4f1396e787bd94200c3997c72c1dbe028587bd76/py4j-0.10.7-py2.py3-none-any.whl (197kB)
    100% |[redacted]| 204kB 30.8MB/s ta 0:00:01
Building wheels for collected packages: pyspark
  Running setup.py bdist_wheel for pyspark ... /
```

```
[ ]: # This Python 3 environment comes with many helpful analytics libraries installed
     # It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
     # For example, here's several helpful packages to load in

     import numpy as np # linear algebra
     import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

draft based on V1

1 committed version

V1 19m +25 -0

Draft Environment

+ Add Data

input (read-only)

shakespeare-online

t8.shakespeare.txt

Settings

Sharing: Private, 0 collaborators

Language: Python

Docker: Latest available

GPU BETA: GPU off

Internet BETA: Internet connected

Packages: No custom packages