**Problem 1)** Lecture notes contain script network-count.py in both Spark Streaming APIand Spark Structured Streaming API. Use Linux nc (NetCat) utility to demonstrate that scripts work. Run both scripts on your own VM with Spark 2.2 installation. Cloudera VM with Spark 1.6 does not have Spark Structured Streaming API.

[rsha@localhost ~]$ nc -lk 9999

sakshaksh

sahskhskaS

SHAKSHKAsh

[rsha@localhost hw6]$ spark-submit --master local[4] network-count.py localhost 9999

-------------------------------------------

Time: 2017-10-09 22:31:09

-------------------------------------------

-------------------------------------------

Time: 2017-10-09 22:31:12

-------------------------------------------

-------------------------------------------

Time: 2017-10-09 22:31:15

-------------------------------------------

-------------------------------------------

Time: 2017-10-09 22:31:18

-------------------------------------------

(u'sakshaksh', 1)

-------------------------------------------

Time: 2017-10-09 22:31:21

-------------------------------------------

(u'sahskhskaS', 1)

(u'SHAKSHKAsh', 1)

-------------------------------------------

Time: 2017-10-09 22:31:24

-------------------------------------------

-------------------------------------------

Time: 2017-10-09 22:31:27

-------------------------------------------

-------------------------------------------

Time: 2017-10-09 22:31:30

-------------------------------------------

[rsha@localhost ~]$ nc -lk 9999

rea is a

grild

little gitl

[rsha@localhost hw6]$ spark-submit --master local[4] ss-network-count.py localhost 9999

-------------------------------------------

Batch: 0

-------------------------------------------

+----+-----+

|word|count|

+----+-----+

| is| 1|

| a| 1|

| rea| 1|

+----+-----+

-------------------------------------------

Batch: 1

-------------------------------------------

+------+-----+

| word|count|

+------+-----+

| is| 1|

| a| 1|

| gitl| 1|

| rea| 1|

| grild| 1|

|little| 1|

+------+-----+

**Problem 2)** Expand provide orders.tar.gz file. Also, download shell scrips splitAndSend.original.sh and splitAndSend.sh and the Python script count-buys.py. First run splitAndSend.original.sh and count-buys.py. Record the failure mode of count-buys.py. Simply read the error message produced and tell us what is happening. Then run script splitAndSend.original.sh and Python program count-buys.py and tell us what the results are. In both cases show use contents of your HDFS directories input, output and staging. The second script splitAndSend.sh is supposed to reduce or eliminate the race condition. You might want to rename HDFS directory output from the first run in order to preserve it’s content. In both cases, show the partial contents of your HDFS directories input, output and staging. In the second run, locate an output file named part-00000 that is not empty and show its content to us. Run these experiments on Cloudera VM. You need HDFS for these programs to run.

**Problem 3).** In the second run of the previous problem you will notice that many of part-00000 files in your output directory are empty. Could you explain why.

(10%)

**Problem 4)** Could you rewrite count-buys.sh in Spark Structured Streaming API. If you do that change script splitAndSend.sh to move generated chunks from the local files system directory staging to local file system directory input. Run this experiment on your VM with Spark 2.2.

(20%)

**Problem 5)** Examine provided Python program stateful\_wordcount.py. Make it work as is. If there are errors on the code, fix them. Modify the code so that it outputs the number of words starting with letters a and b. Demonstrate that modified program work. You should provide several both positive and negative examples.

(20%)