Guangda Ouyang, Adrien Xie guangda 467428, 451852

Problem 1

(a)

```
sc.textFile("//localhost/loudacre/weblogs/").filter(lambda x: x.find("html") != -1 ).map(lambda line: line.split(' ')).map(lambda x: x[0] + "/" + x[2]).take(10)
```

```
[u'142.19.184.108/187', u'66.72.212.11/91', u'194.91.6.192/18641', u'252.232.52.250/67809', u'61.211.36.7/123', u'8.135.236.171/10022', u'138.229.167.20/34', u'153.100.5.159/71', u'46.250.3.164/70', u'16.52.46.209/1055']
```

(b)

There are 1079891 records in total.

Command:

```
sc.textFile("//localhost/loudacre/weblogs/").count()
```

There are 474360 html requests.

Command:

```
sc.textFile("//localhost/loudacre/weblogs/").filter(lambda x: (x.find("html") != -1)).count()
```

Problem 2

(a)

There are 1079891 records in total.

Command:

```
sc.textFile("//localhost/loudacre/weblogs/").count()
```

There are 474360 html requests.

Command:

```
sc.textFile("//localhost/loudacre/weblogs/").filter(lambda x: (x.find("html") != -1)).count()
```

(b)

```
>>> data = sc.textFile("//loudacre/weblogs/")
>>> pair = data.filter(lambda x: x.find("html")!= -1).map(lambda y: y.split(" ")).map(lambda z: (z[2],1)).reduceByKey(lambda v1,v2: v1+v2)
```

(c)

There are 5831 users visited once, 1460 users visited 7 times, and 635 users visited 12 times.

Command:

```
>>> pair = data.filter(lambda x: x.find("html")!= -1).map(lambda y: y.split(" ")).map(lambda z: (z[2],1)).reduceByKey(lambda v1,v2: v1+v2)
>>> pair.filter(lambda x: x[1] == 1).count()
5831
>>> pair.filter(lambda x: x[1] == 7).count()
1460
>>> pair.filter(lambda x: x[1] == 12).count()
635
```

(d)

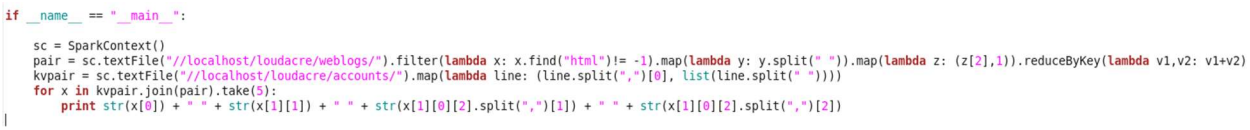
data = sc.textFile("//localhost/loudacre/accounts/")

kvpair = data.map(lambda line: (line.split(",")[0], list(line.split(" "))))

(e)

Command:

```
sc = SparkContext()
pair = sc.textFile("//localhost/loudacre/weblogs/").filter(lambda x: x.find("html")!= -1).map(lambda y: y.split(" ")).map(lambda z: (z[2],1)).reduceByKey(lambda v1,v2: v1+v2)
kvpair = sc.textFile("//localhost/loudacre/accounts/").map(lambda line: (line.split(",")[0], list(line.split(" "))))
for x in kvpair.join(pair).take(5):
    print str(x[0]) + " " + str(x[1][1]) + " " + str(x[1][0][2].split(",")[1]) + " " + str(x[1][0][2].split(",")[2])
```



Result:

102667 7 Louise Johnson

41471 16 Debbie Patterson

28305 4 William Hughes

43572 1 Philip McHale

102180 35 Susan Brown

Problem 3

(b)

64978

spark-submit CountJPGs.py /loudacre/weblogs/

Driver program is executed locally; processing happens locally; result is stored locally.

(c)

64978

spark-submit --master yarn-client CountJPGs.py /loudacre/weblogs/

Driver program is executed locally; processing happens on node manager; result is stored on HDFS.

(d)

Stage 1

Tasks 311

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	count at /home/cloudera/training_materials/dev1/exercises/spark-application/CountJPGs.py:8	2018/11/11 04:51:06	1.4 min	1/1	311/311

(e)

64978

spark-submit --master yarn-cluster CountJPGs.py /loudacre/weblogs/

Driver program is executed on application master; processing happens on node manager; result is stored on HDFS.

Guangda Ouyang, Adrien Xie guangda 467428, 451852

Problem 4

(a)

Raw:

(a-a)

(a-b)

(a-c)

(b-a)

(b-c)

(c-b)

(c-c)

Links:

(a-[a, b, c])

(b-[a, c])

(c-[b, c])

(b)

For n pages and m links, with raw, we need to store m piece of data and with links, we need to store n piece of data. So, when m > n, i.e., more links than pages, links representation is more storage efficient; when m < n, i.e., more pages than links, raw representation is more storage efficient. When n = m, they are equally efficient.