Small Project

Question1.

a.

Java environment used by hadoop:

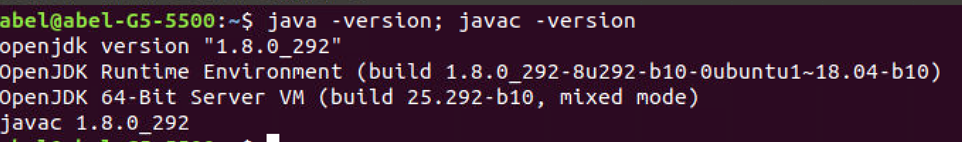


Figure 1: CMD for Java version

Configuration files that I'd modified to setup the pseudo-distributed mode:

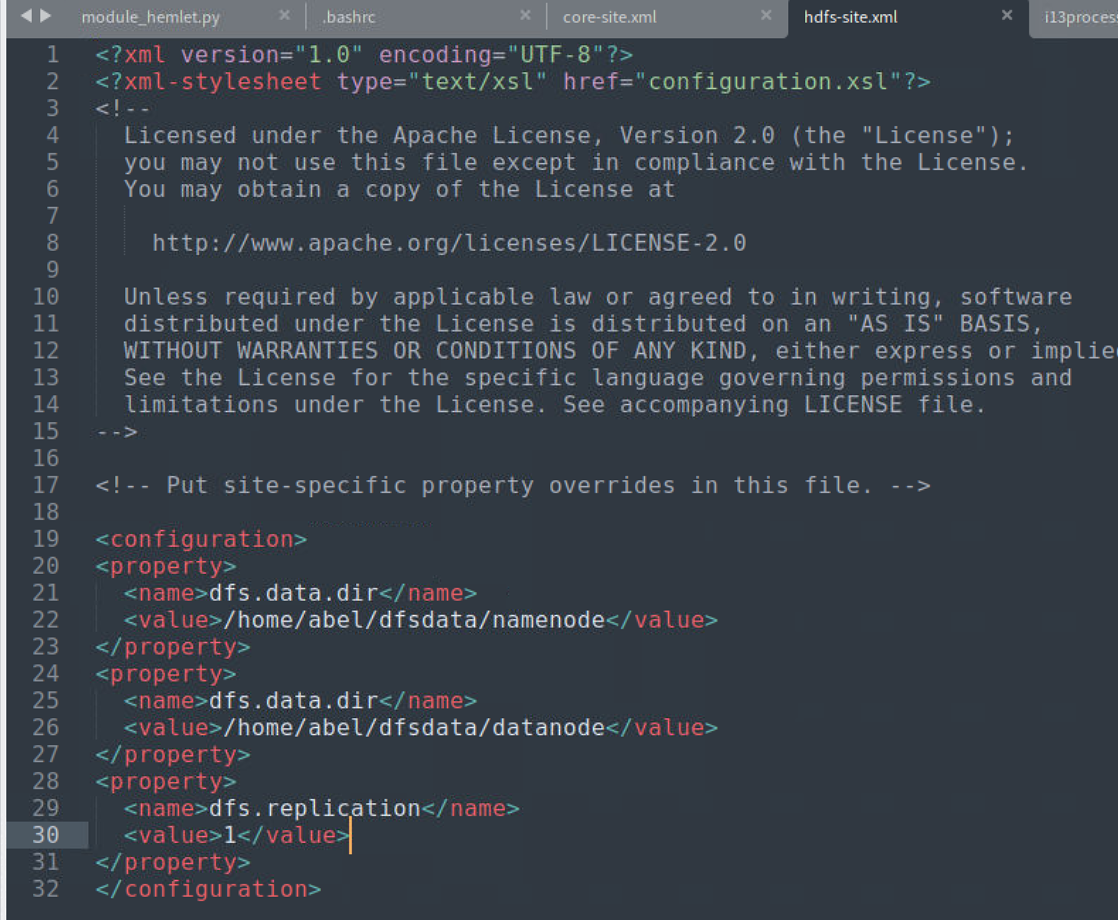


Figure 2: XML for hdfs-site

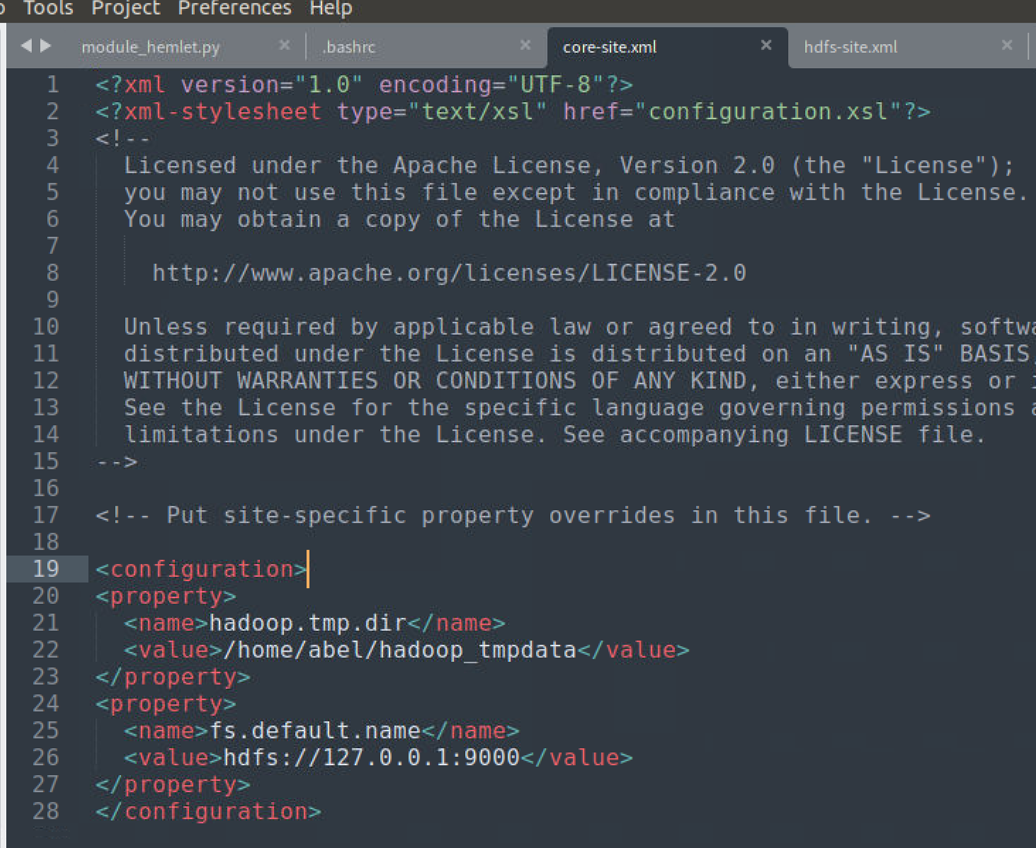


Figure 3: XML for core-site

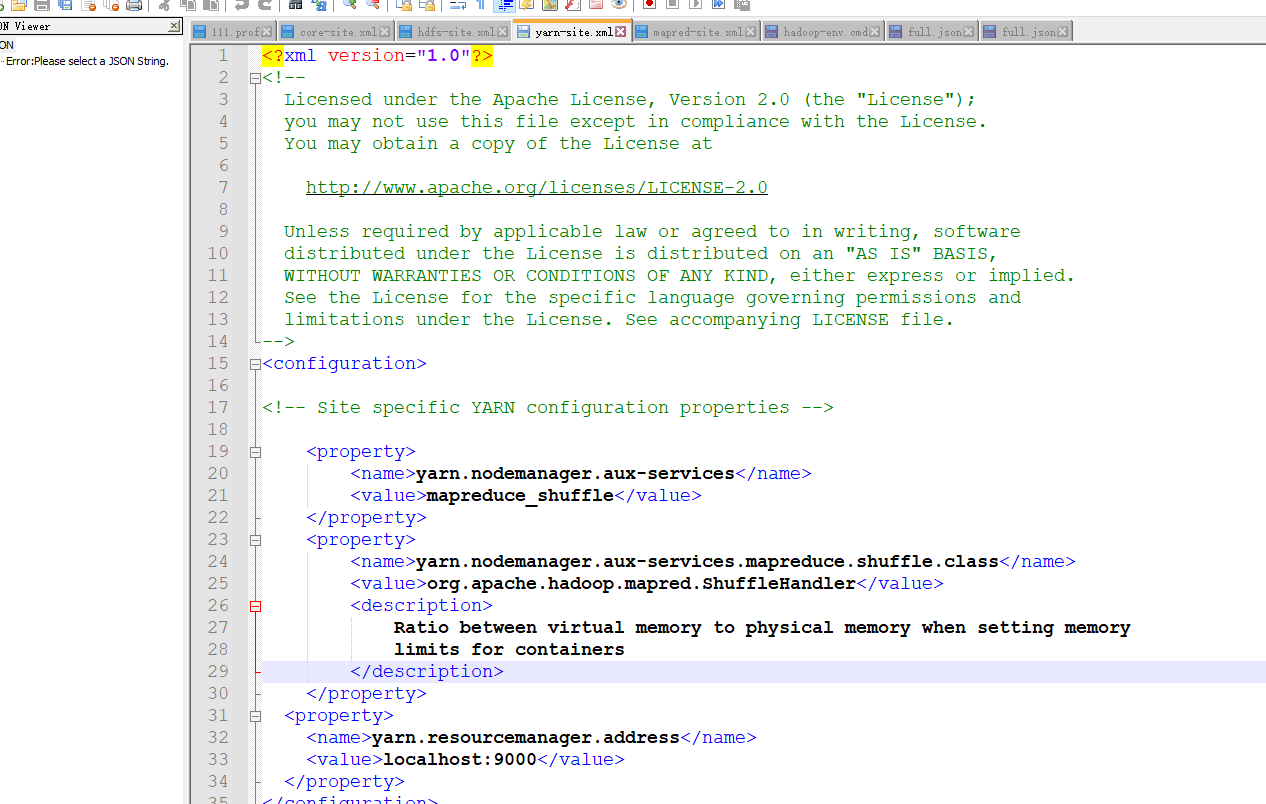


Figure 4: XML for yarn-site

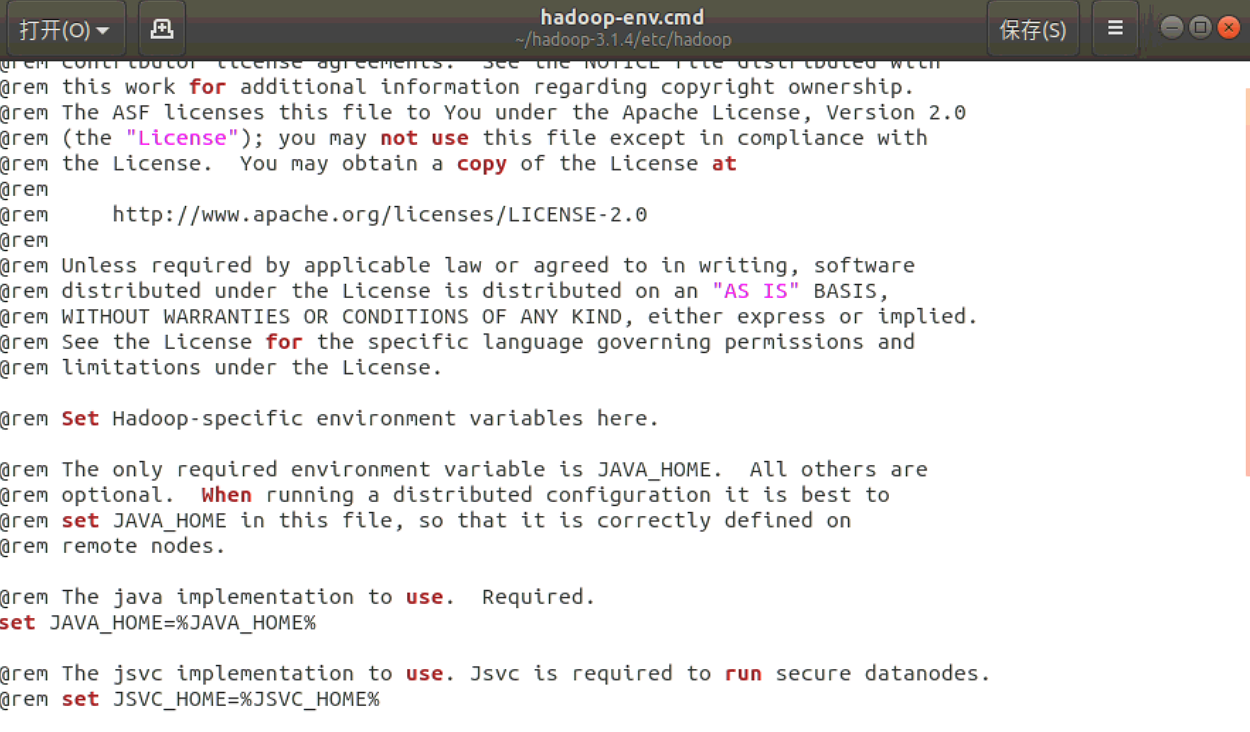


Figure 5: CMD for Hadoop-env

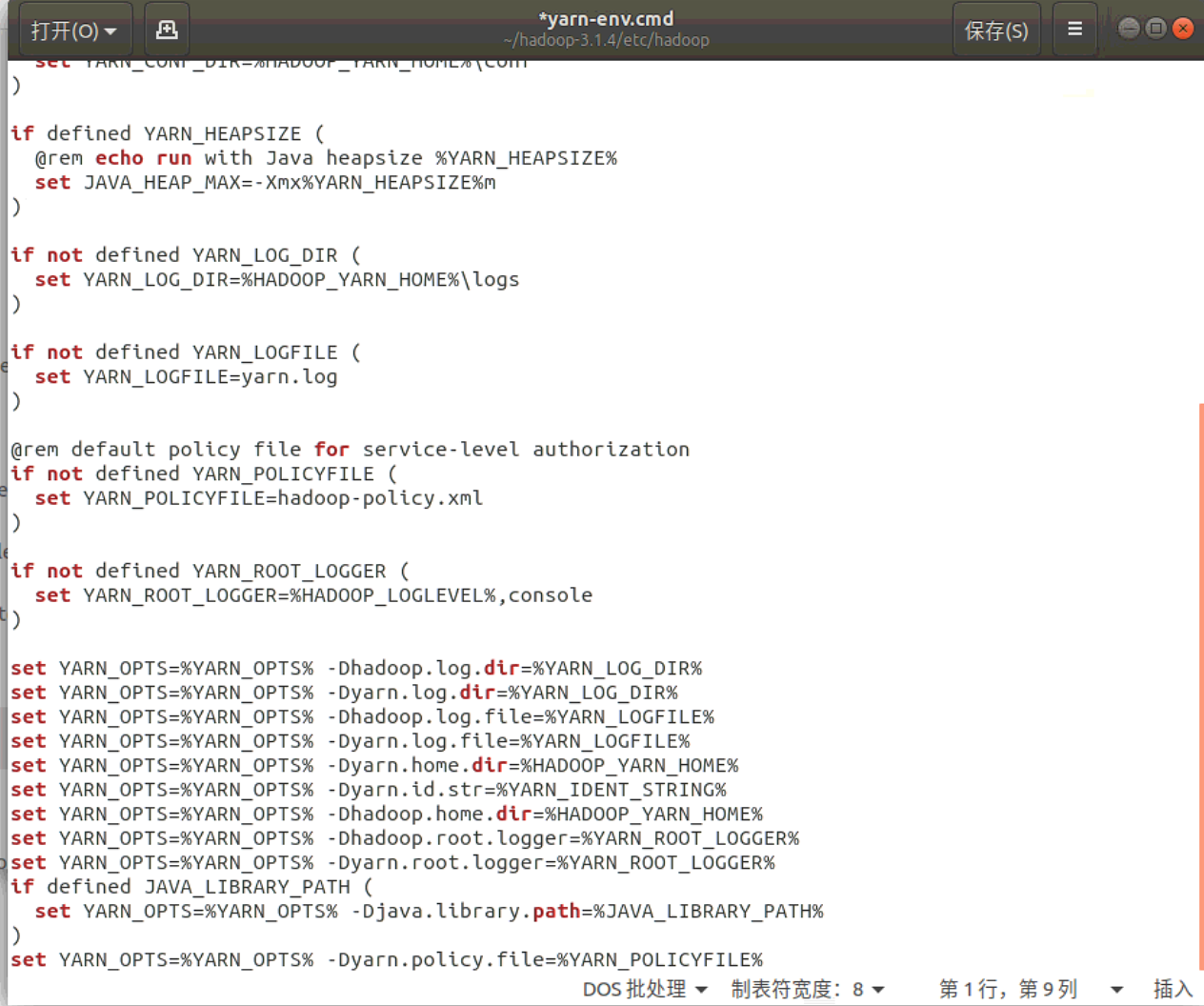


Figure 6: CMD for Yarn-env

b.

Firstly, use the command “hdfs dfs -mkdir /user/Jiang\_Ruizhao” to create a folder named Jiang-Ruizhao under the user folder. Then, write “Name: Jiang ruizhao, ID number: 510122199904210065” to the txt file called DMT1709166. Lastly, use hdfs dfs language to copy the txt file into the folder /user/Jiang\_Ruizhao.

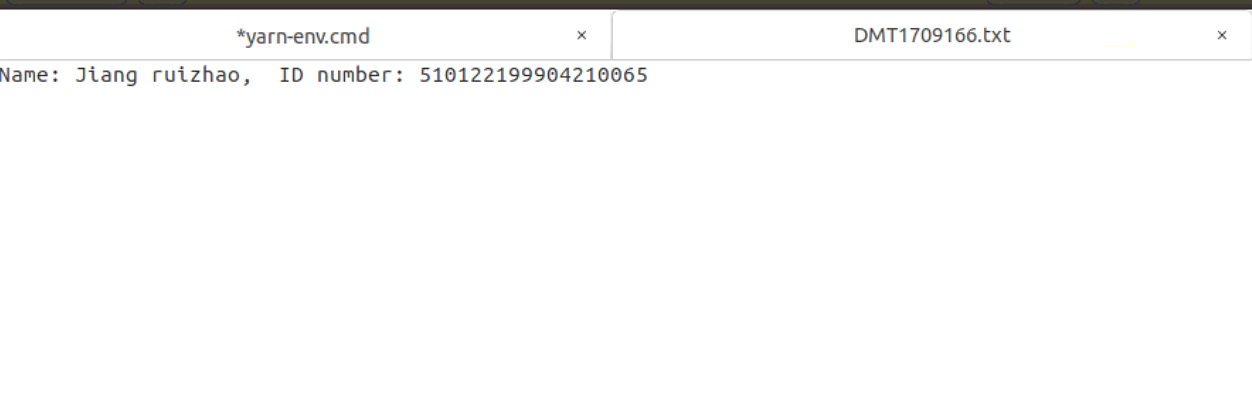


Figure 7: content of DMT1709166.txt

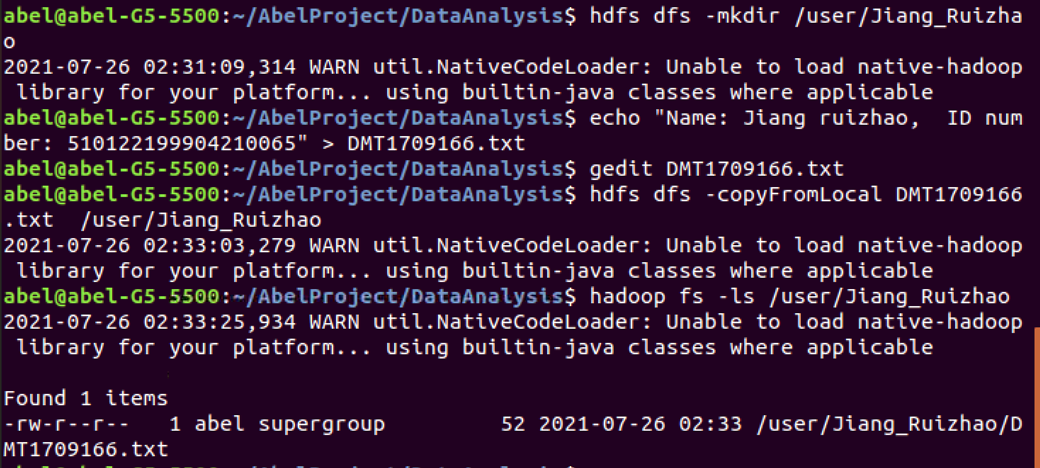


Figure 8: Implementation of writting DMT1709166.txt

c.

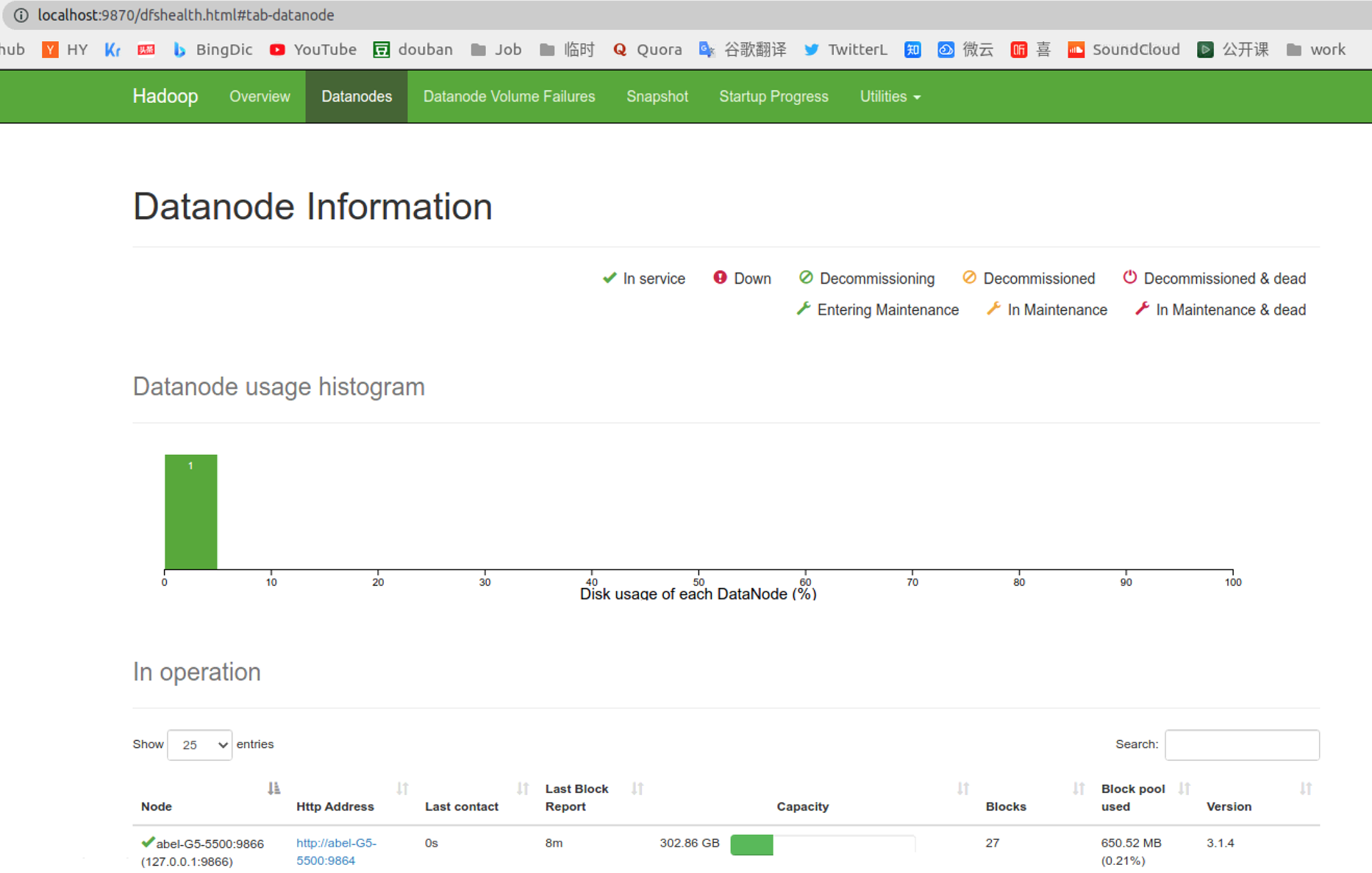


Figure 9: Datanode information web UI

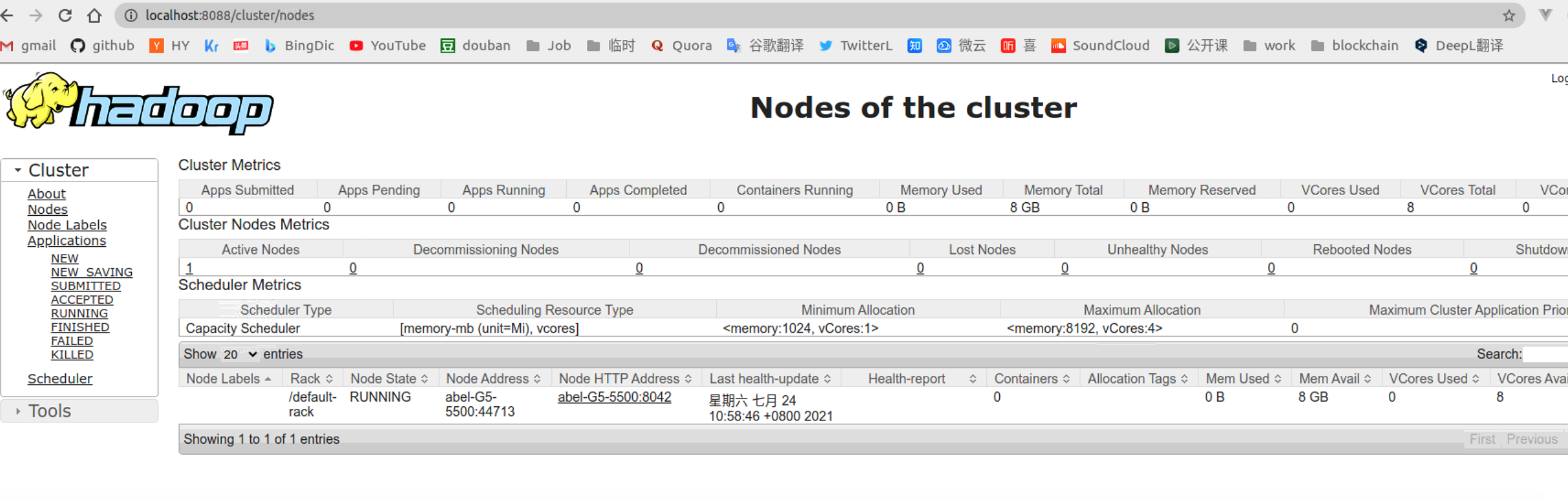


Figure 10: Nodes of the cluster in web UI

**Question 2**

2.1.a

For the first map-reduce computing of task (1):

I use mapping function like:

(itemx in x) (itemy in y) -> (itemx, itemy) in (x,y)

At the same time, the reduce function is as follows:

Many itemx+itemy -> min number of the sum(itemx+itemy)

添加的设计解释：

We frist design x, y list which store X = (x1,··· ,xn) and Y = (y1,··· ,yn).

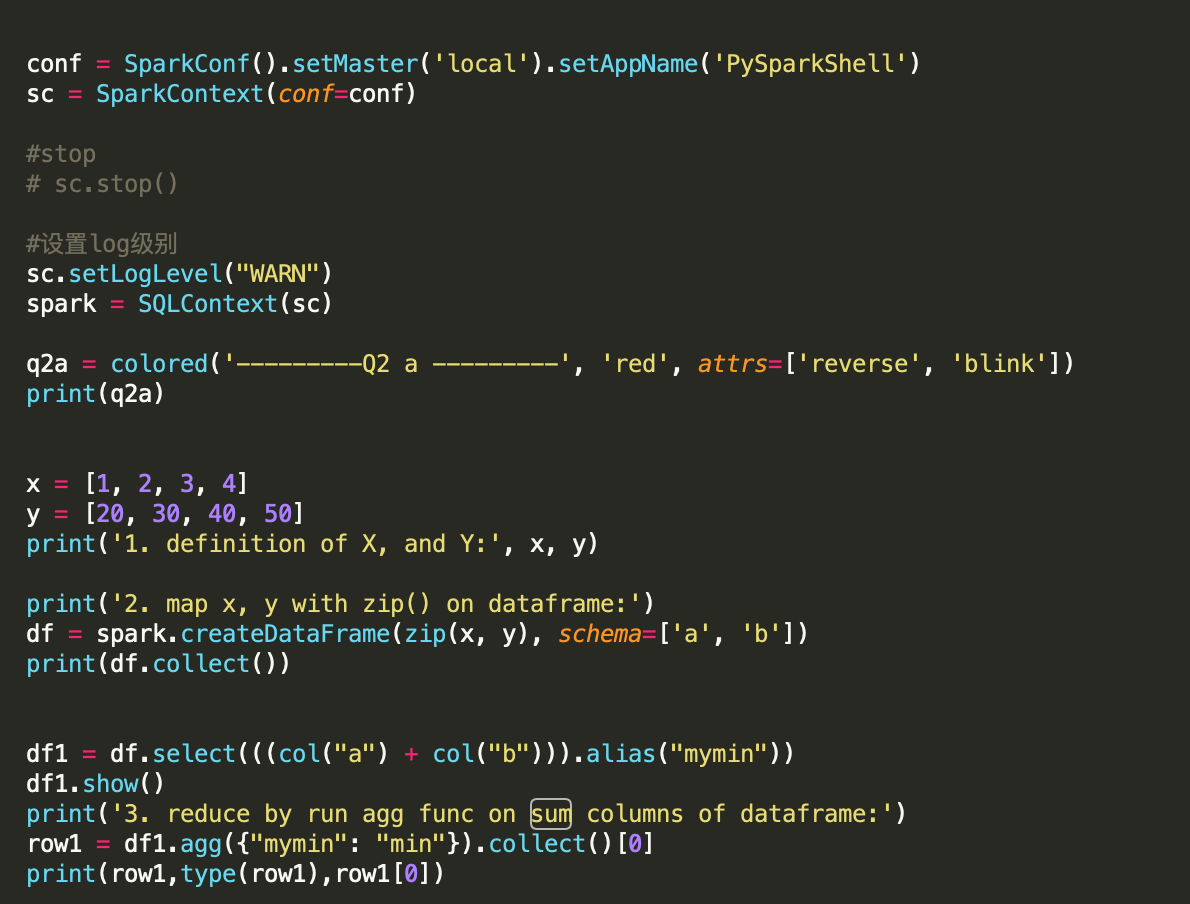
映射（Mapping）对集合里的每个目标应用同一个操作。即，我们把X,Y表单里每个单元格进行zip合并成一个原声zip(x,y) 当x属于X，y属于Y，那么把这个函数单独地应用在每个单元格上的操作就属于mapping。

接着我们对每一个原色进行x+y相加，当x属于X，y属于Y，那么把这个函数单独地应用在每个单元格上的操作就属于mapping

化简（Reducing ）遍历集合中的元素来返回一个综合的结果。即，输出表单里一列数字的agg({"mymin": "min"} 找出这一列数据中最小的值，这个任务属于reducing。

2.1.b

Figure 11: Codes to implement the task



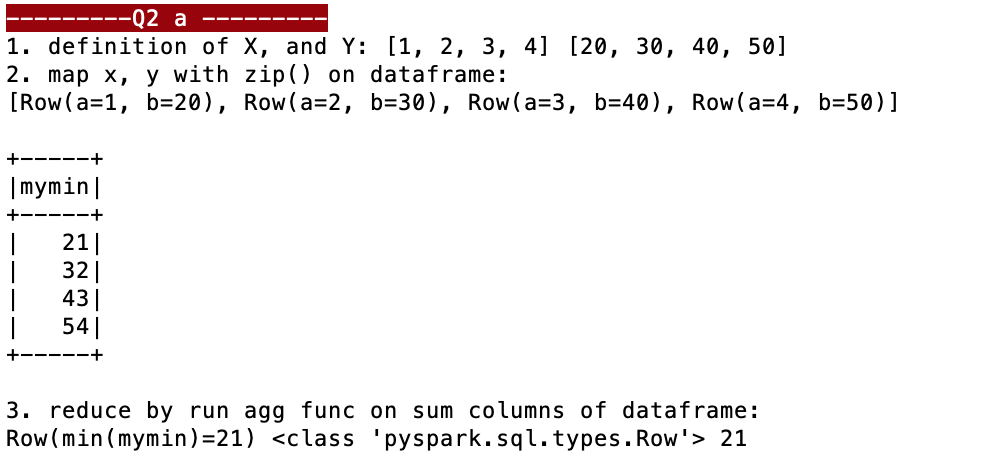


Figure 12: Results of the task

2.2.a

For the second map-reduce computing of task (2):

I use mapping function like:

(itemx in x) (columns\_of\_y in y) -> (itemx, columns\_of\_y) in (x,y)

At the same time, the reduce function is as follows:

Many itemx+columns\_of\_y -> min number of the sum(itemx+columns\_of\_y)

添加的设计解释：

We frist design x, y list which store X = (x1,··· ,xn)

Y = [] 为空，存储结算后的结果

A定义为一个numpy矩阵，存储定义中的A的

( a11 a12 … a1m

a21 a22 … a2m

…

an1 an2 … anm

)

首先我们对A进行列压缩，然后遍历每一列：

在遍历过程中：

映射（Mapping）对集合里的每个目标应用同一个操作。即，我们把X,Y表单里每个单元格进行zip合并成一个原声zip(x,y) 当x属于X，y属于Y，那么把这个函数单独地应用在每个单元格上的操作就属于mapping。

接着我们对每一个原色进行x+y相加，当x属于X，y属于Y，那么把这个函数单独地应用在每个单元格上的操作就属于mapping

化简（Reducing ）遍历集合中的元素来返回一个综合的结果。即，输出表单里一列数字的agg({"mymin": "min"} 找出这一列数据中最小的值，这个任务属于reducing。

然后我们对Y 存储添加对应每一列的 ymin值

最后返回Y的向量，就是我们循环执行map-reduce后的结果

2.2.b



Figure 13: Codes to implement the task

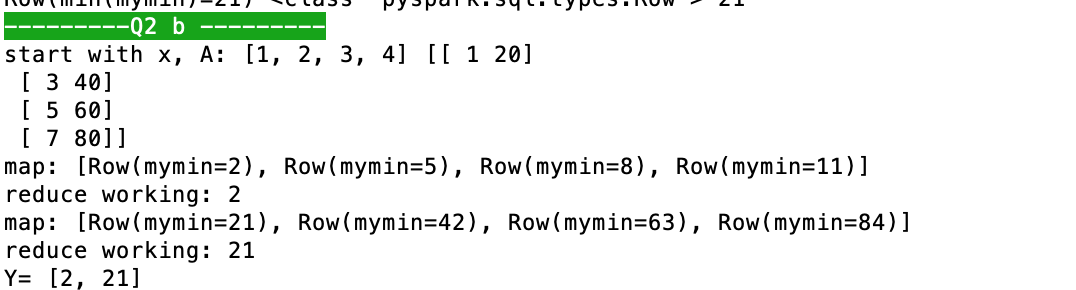


Figure 14: Results of the task

**Question 3**



**(a)**

print('1. column=',len(myair\_transits.columns))

myair\_transits = hiveCtx.sql("SELECT count(\*) FROM air\_transit ")

print('1. rows : ', myair\_transits.collect())

There are 7,009,728 rows and 13 columns in the “flights”.

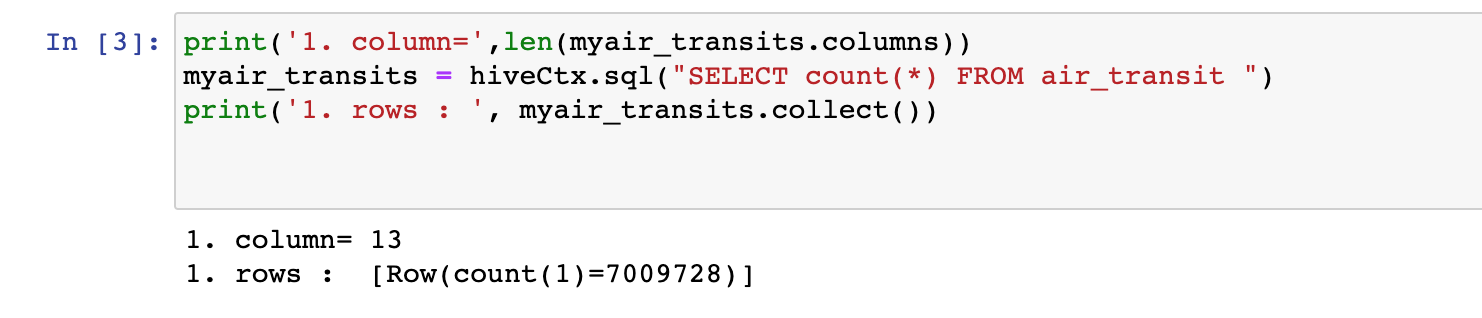


Figure 15: Results of the task (a)

There are 7,009,728 rows and 13 columns in the “flights”.

**(b)**

print('2. print out the schema of this dataframe')

input.printSchema()

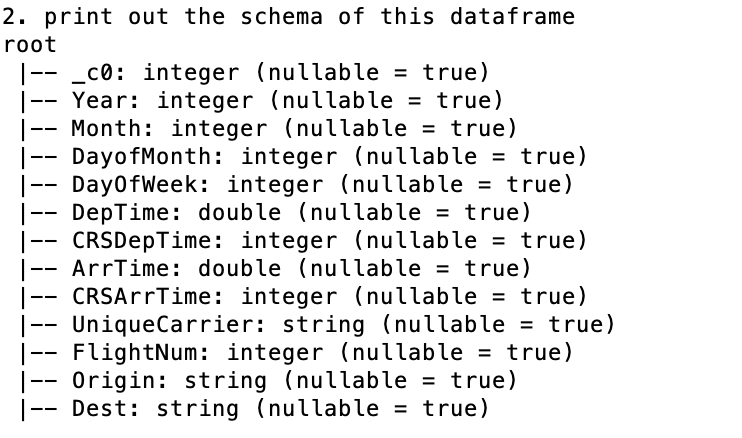


Figure 16: Results of the task (b)

The schema of the dataframe “flight” is shown above.

**(c)**

print('3. List the distinct carriers of in the dataframe:')

myair\_transits = hiveCtx.sql("SELECT distinct UniqueCarrier FROM air\_transit ")

print(myair\_transits.collect())

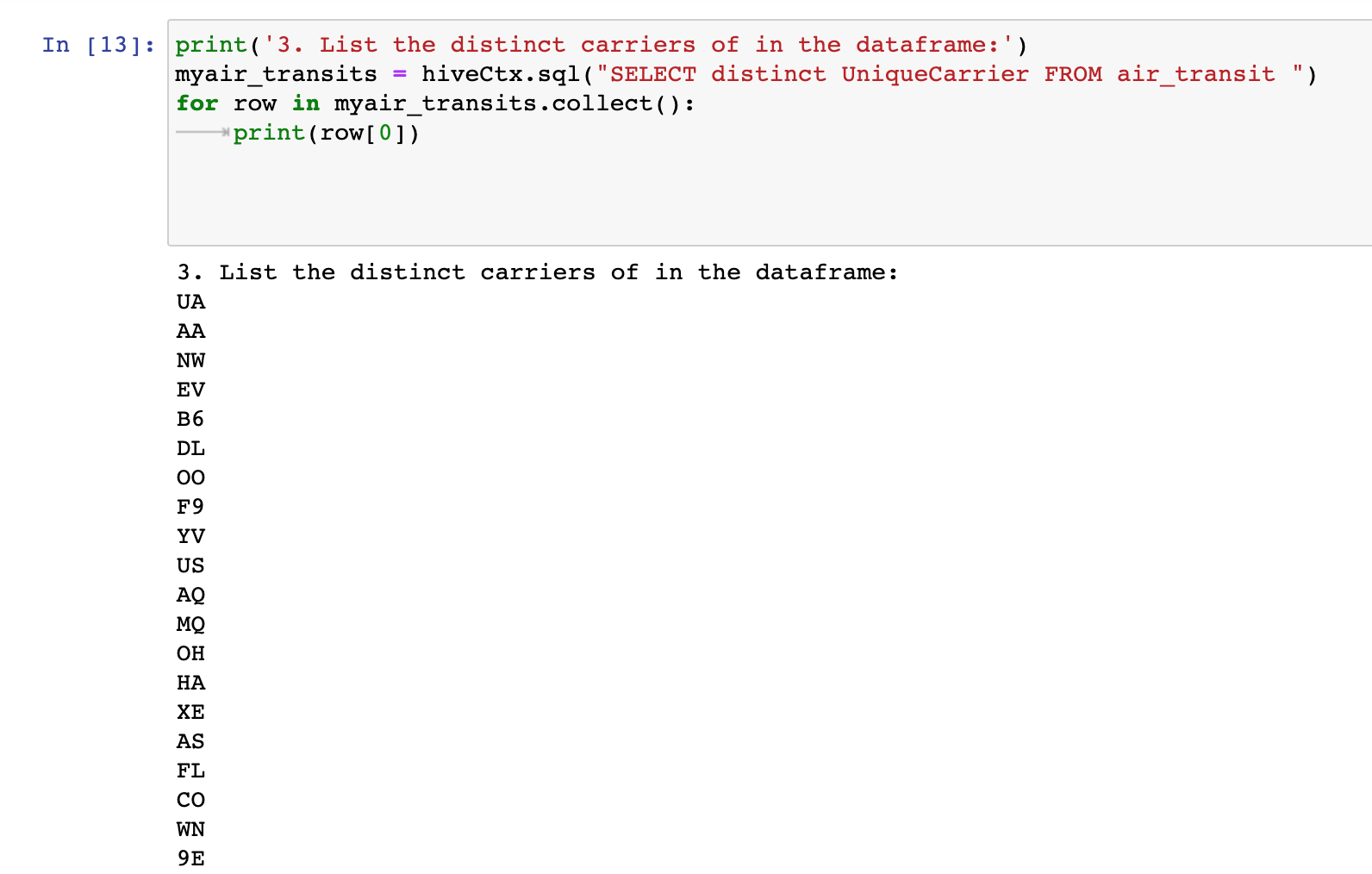
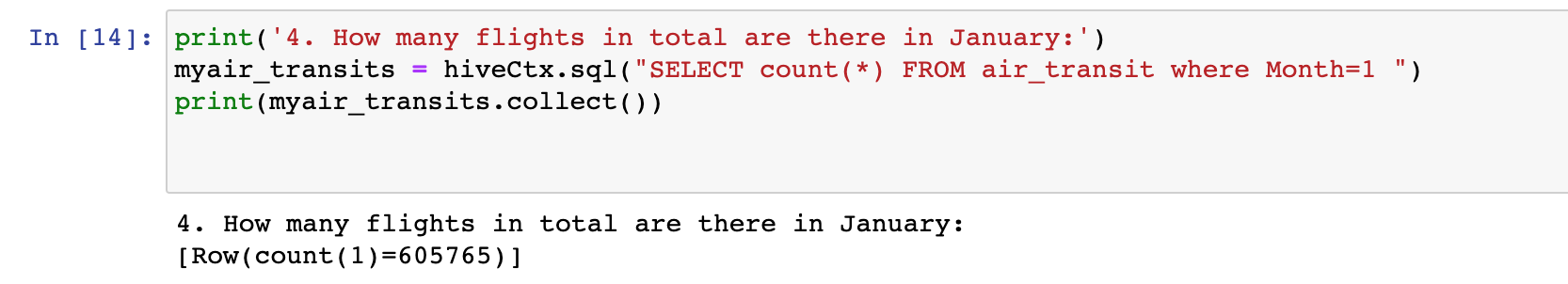


Figure 17: Results of the task (c)

Select the column of “UniqueCarrier”, and then list the distinct carriers in the dataframe.

**(d)**



print('4. How many flights in total are there in January:')

myair\_transits = hiveCtx

.sql("SELECT count(\*) FROM air\_transit where Month=1 ")

print(myair\_transits.collect())

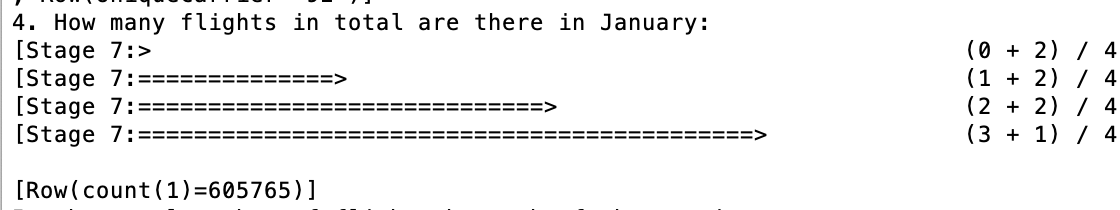
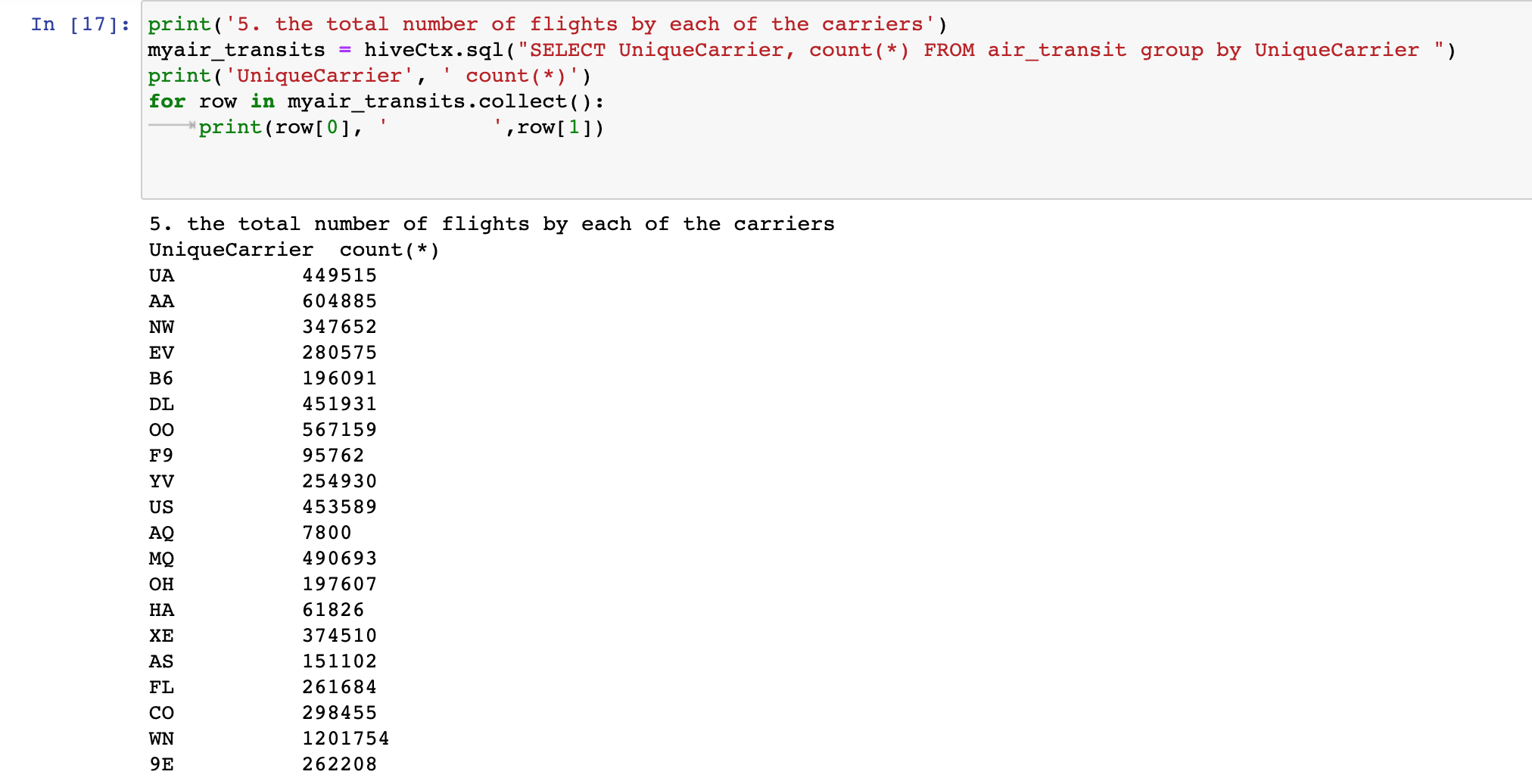


Figure 18: Results of the task (d)

**(e)**

****

print('5. the total number of flights by each of the carriers')

myair\_transits = hiveCtx.sql("SELECT UniqueCarrier, count(\*) FROM air\_transit group by UniqueCarrier ")

print('UniqueCarrier', ' count(\*)')

for row in myair\_transits.collect():

print(row[0], row[1])

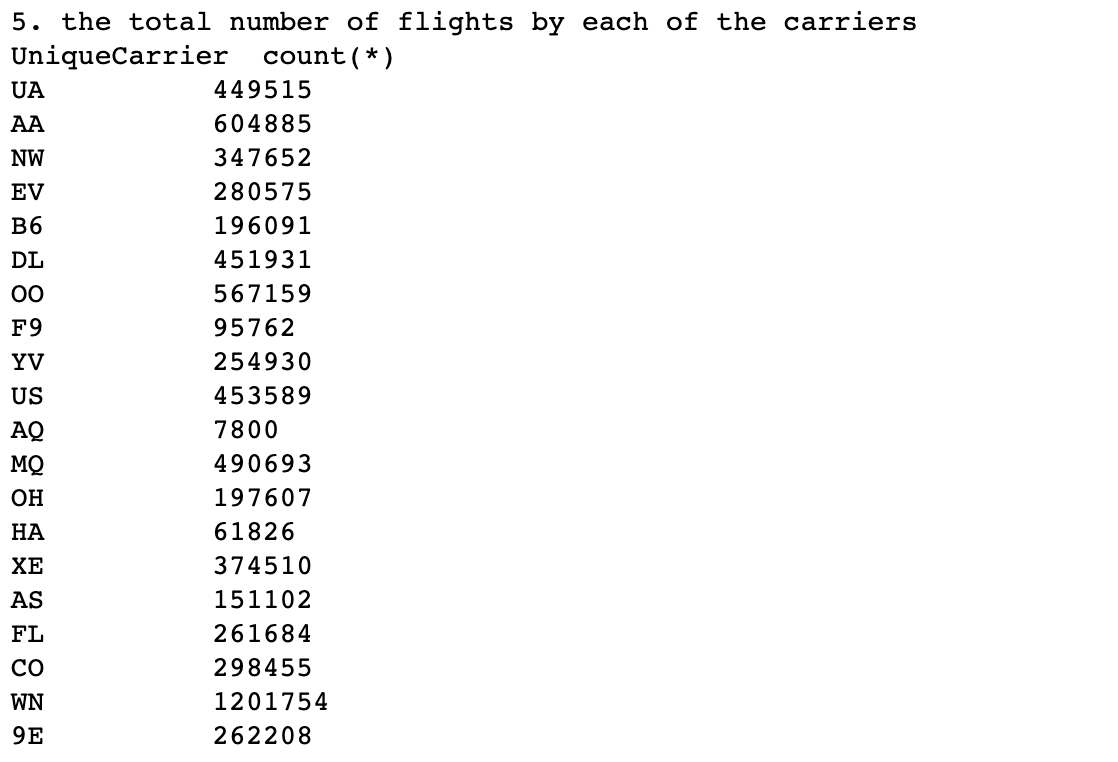
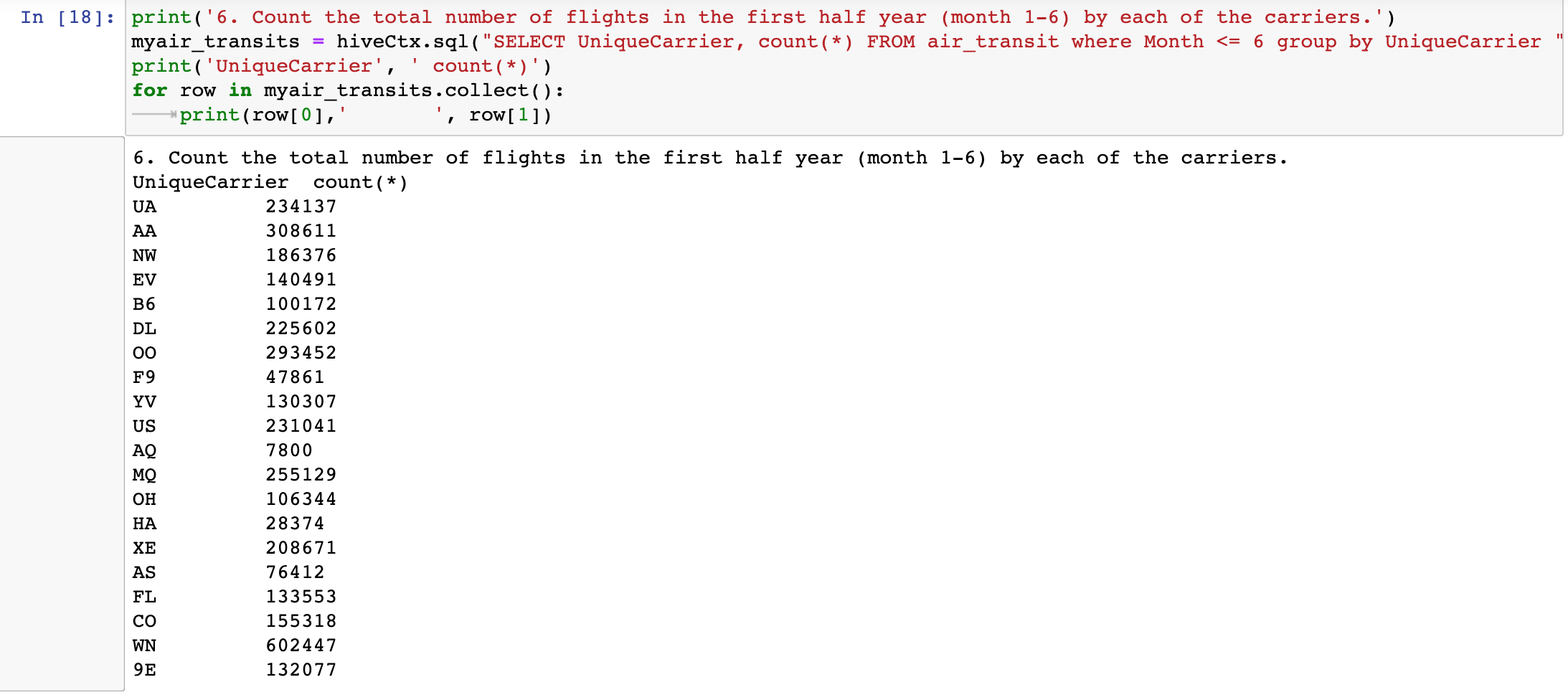


Figure 19: Results of the task (e)

**(f)**

****

print('6. Count the total number of flights in the first half year (month 1-6) by each of the carriers.')

myair\_transits = hiveCtx.sql("SELECT UniqueCarrier, count(\*) FROM air\_transit where Month <= 6 group by UniqueCarrier ")

print('UniqueCarrier', ' count(\*)')

for row in myair\_transits.collect():

print(row[0], row[1])

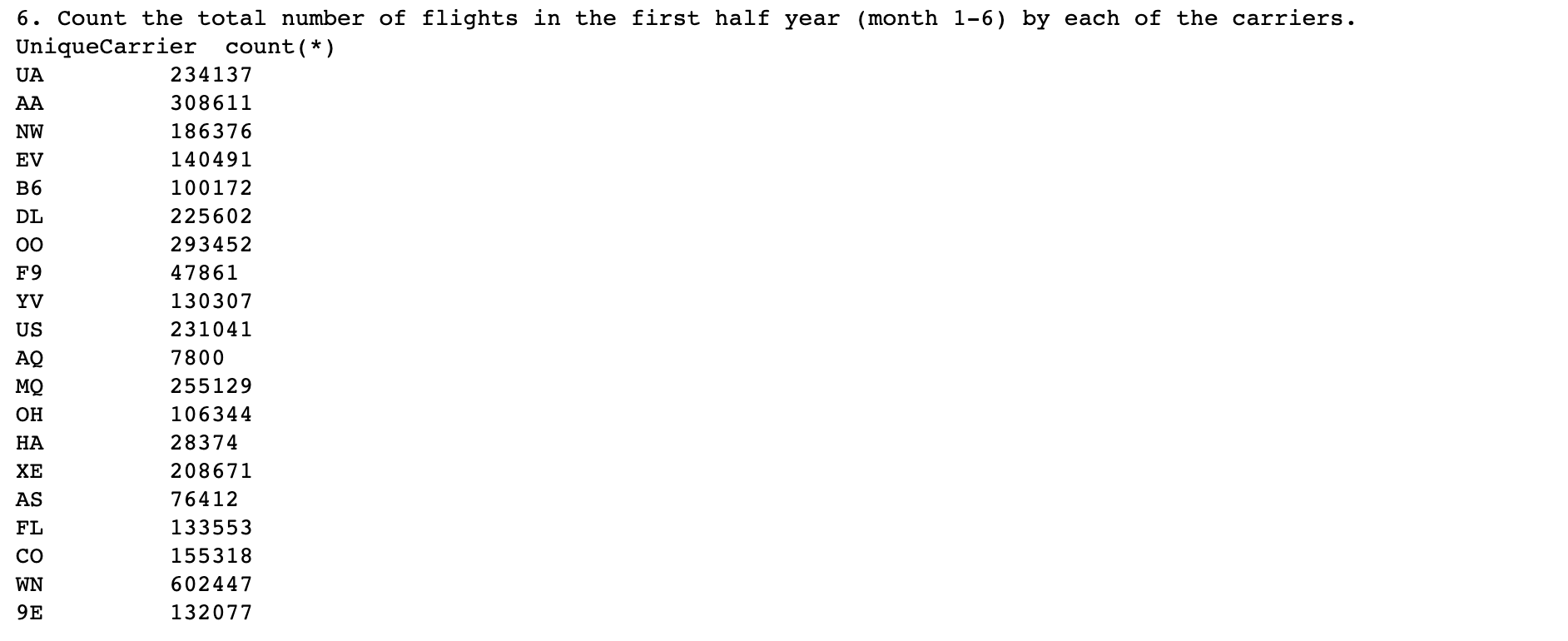


Figure 20: Results of the task (f)