trustingsocial

CANDIDATE ASSESSMENT EXERCISES

[Big Data Engineer Position]

	Test purpose	Task name
Task 1	Algorithm and coding	Find the actual activation date of phone number
Task 2	Bigdata system design	Design a simple data pipeline for ETL and data aggregation

Task 1 [Algorithm and coding]: Find the actual activation date of a phone number

Statement:
Given a list of N = 500,000,000 records (in CSV or Parquet format). Each record describes an usage period of a specific mobile phone number.

Note that one phone number can occurs multiple times in this list, because of 2 reasons:

- This phone number can change from prepaid plan to postpaid plan, or vice versa, at anytime just by sending an SMS to the operator.
- Or, the owner of this phone number can stop using it, and after 1-2 months, it is reused by another person.

Also remember that, the reason is not recorded in the data, we just have the phone number and its activation or deactivation date for a usage period record.

- Activation date is the date that the phone number is started being used by a owner with a specific plan (prepaid or postpaid).
- Deactivation date is the date that the phone number is stopped being used by a owner with the registered plan.

Moreover, the records don't need to follow any specific order of time, and the records of the same number don't need to be consecutive.

This is an example of the list, given as a CSV file:

```
PHONE_NUMBER,ACTIVATION_DATE,DEACTIVATION_DATE
0987000001,2016-03-01,2016-05-01
0987000002,2016-02-01,2016-03-01
0987000001,2016-01-01,2016-03-01
0987000001,2016-12-01,
0987000002,2016-03-01,2016-05-01
0987000003,2016-01-01,2016-01-10
0987000001,2016-09-01,2016-12-01
0987000002,2016-05-01,
0987000001,2016-06-01,2016-09-01
```

In this list, ACTIVATION_DATE and DEACTIVATION_DATE are represented with YYYY-MM-DD format. DEACTIVATION_DATE may be empty, which means that the phone is still being used by its current owner.
From the given data, we want to find a list of unique phone numbers together with the actual activation date when its current owner started using it. Note that what we need is the first activation date of current owner, not previous owner, and not the date when current owner changes prepaid/postpaid plans.

For example: The prepaid phone number `0987000001` was used by A from `2016-01-01` to `2016-03-01`, then it was changed to postpaid. A continued using it until `2016-05-01` and stopped using this number. After 1 month, on `2016-06-01`, this phone number was reused by B

with prepaid plan. B used it until `2016-09-01` then changed to postpaid, and finally changed back to prepaid on `2016-12-01` and he's still using it until now. In this case, the actual activation date of current owner B of `0987000001` that we want to find is `2016-06-01`.

Requirement:

- Describe in detail your strategy and algorithm to solve this problem.
- Analyze the time complexity and memory complexity of your algorithm (including the processing time of any the data structure that you need to use in your implementation).
- Implement your solution in **Scala, Java** or **Python** with **Apache Spark** or **MapReduce**, with input is a CSV or Parquet file as described above, and write the output as another CSV or Parquet file with following format:

```
PHONE_NUMBER,REAL_ACTIVATION_DATE
0987000001,2016-06-01
0987000002,2016-02-01
0987000003,2016-01-01
```

- Your code should be production ready, which means that it is well organised and well tested. Please send us your code with all unit tests if available.
- You should submit your project with **README** file which show how to setup, compile code, run or submit application so that we could follow to run and check your work.

Task 2 [Big Data system design]: Design a simple data pipeline for ETL and data aggregation

Statement:

We are going to develop a simple data pipeline for ETL and data aggregation:

- We are telecom operator's partner. We build a system which load data from telco's system, process, store then doing data mining and machine learning on the data to develop a credit score model.
- Input data source for our system is CDR files which could be collected from one or many servers via FTP.
- CDR files are generated in realtime. The number of files per day is about 70,000, total records is about 300 millions and total file's size is about 200GB.
- Our system main functions:
    - Collecting all data.
    - Transforming the data (cleaning, formatting, deduplication).
    - Storing in HDFS.
    - Doing data aggregation on daily basis.
- The system should be high-availability, fault tolerant, distributed, scalability, high performance.
- You have 10 - 15 servers to setup and run the whole system. It is better if you could choose the hardware specifications (CPU, RAM, Storage, …) for each server which you want to use.

More detail about the CDR data and data aggregation:

- CDR data are provided in CSV format but it can be different in number of columns, columns' name, separated character).
- We have about 10 different types: call histories, message histories, top up histories, etc.
- An example of CDR: call_histories_20151201.csv

```
FROM_PHONE_NUMBER;TO_PHONE_NUMBER;START_TIME;CALL_DURATION;IMEI;LOCATION
0987000001;0987000002;01/12/2015 04:43:33;10;352700072104120;452049C1382CF
0987000001;0987000003;01/12/2015 04:50:04;21;351556058208220;45204A5428FD2
```

- For call histories data, we need to do data aggregation daily
    - For each number we want to have:
        - Number of call, total call duration.
        - Number of call in working hour (8am to 5pm).
        - Find the IMEI which make most call.
        - Find top 2 locations which make most call.
    - The system should run this task in the end of day then store the output in HDFS for later use.
- In this exercise, we only provide the information for call histories data but you should keep in mind that there are others file type that the system need collect, process and store. The design should be flexible enough to add new file type in the future easily.

Requirement:

- Design (draw) the high level architecture of the system.
- Describe what technologies you will use to develop the system.
- Explain in detail how the system works, how data are stored, and how it meets all the listed requirements.
- Write code for data transformation, data aggregation daily.