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

Because the data is not in any specific order but the algorithm is working related on date data types so I need to sort the data by ACTIVATION_DATE field. When data was sorted, it uncovers some useful information.

- Firstly, because each time range (ACTIVATION_DATE; DEACTIVATION_DATE) record describes a usage period of a specific mobile phone number so it will not be overlapping time, that means each record of specific phone number will look like time series when sorted.

- Secondly. The records have no DEACTIVATION_DATE, it means the phone number still being used by the current owner, so when sorted the data, this record will be in the last line

- Thirdly, I noticed if a user switching from prepaid plan to post-paid plan and vice-versa, it will on the same day, so I can detect current owner by checking the time range between 2 data fields DEACTIVATION_DATE and ACTIVATION_DATE, if the time range has a gap between, that means the phone number have switched to new ower

Summary, my algorithm to solve this problem will look like the

following:

Step 1: Read the CSV file into memory, I using the pandas library because it’s data structure was optimized for working with large datasets.
Step 2: Sort the data by the ACTIVATION_DATE filed, I also sorted by the phone number so the records of same phone number lie side by side. I use the heapsort sorting algorithm of pandas which have time complexity O(n log(n)) and memory complexity is O(1) and sorted it in-place to saving memory.

Step 3: Initialize a dictionary, I coding in Python (I know Golang but familiar with Python more) to store real activation date of phone numbers. Because dictionary in python is hash table data structure so it is O(1) when access, insert, update.

Step 4: Pandas will read entirely the dataset into a data structure called DataFrame.
Looping through each row in the dataset (skipping the header) and check if the phone number in current row exist in dictionary or not

- If the phone number did not exist in the dictionary, I save the current row to dictionary with the phone number as key and activation date and deactivation date as value.
- If the phone number exists, that means this phone number has more than one record:
    - If the deactivation_date of the current phone number in dictionary equal the activation_date of current row< that

means this phone numbers still being used by the current owner, it just switching the plan, so I update deactivation_date to current row to dictionary, it just use for comparing the record next time.

- If the activation date of current row greater than deactivation_date in the dictionary, that means to have a gap in the time range, so the phone numbers have taken by another owner, I update real activation date and deactivation date to the dictionary

Step 5: Convert the dictionary back to Dataframe, do some cleaning and save to CSV file

The overall time complexity of the algorithm is:
O(n)<Reading to memory> + Θ(n log(n))<sorting> + O(n)* O(1)<loop, processing>
and the memory time complexity is 0(n)