Overview

- We are looking for your ability to write and work with Python code
- Please do not put this on public GitHub - there is no confidential data, but we do not want to leak the prompt, the data, or the solutions. It takes us time to develop a prompt, and we want to reuse it
- You will have a week to complete this exercise from the time you will get this prompt and the data files. We will not be able to offer time extensions to keep it fair for all candidates, so plan appropriately
- We are expecting you to spend no more than 2-3 hours on this exercise; there is always more one can do, but please focus on the essential functionality first
- It is OK to submit incomplete solution, we will still evaluate it and decide whether to proceed
- There will be a “live” (Zoom or in-person) review of your solution submission for this prompt, potentially writing more code - please make sure you can run the code and demonstrate that it works then

Problem setup

A hypothetical company has several locations. Employees access these locations using badges. Each badge has its own ID and is permanently assigned to an employee.

In order to improve our security, we would like to start monitoring for unusual access patterns based on the badge swipe data that is collected at every location for all employees.

We have historical data that we would like you to analyze to see if these patterns have occurred in the past. We would like to create a setup that will allow us to perform exploratory analysis in the future.

Suspicious pattern detection

We would like you to implement a detection for a specific kind of suspicious activity: whether the same badge has been swiped in two locations less than 6 hours apart. It is suspicious, since it is physically impossible to travel between these locations so fast.

We know this has happened in the past; we need your help to find these cases in the data files provided.

Details

Each location has its own system that records the badge swipes. As a result, the data available to us will be in slightly different format, although at the end of the day it amounts to:

```
{badge_id, local_timestamp, location_id}
```

Please note that the `local_timestamp` field will contain the local time for that location, for example 10AM in California is actually 1PM in New York. This will be important to keep in mind when working across locations.

Sample files we provided contain swipe data for holidays, but not weekends. There is also swipe data outside of work hours. It is not considered suspicious for the purposes of this exercise.

Sample data

Please study the annotated example below to guide your implementation (do not forget that we provide you with actual data files in the handout to use in your implementation):

badge_id	location	local_timestamp	notes
123	NY	Nov 21 2023 8:55 AM	Valid morning workday entry
123	CA	Nov 21 2023 10:55 AM	Suspicious entry, since it is only 5 hrs after NY
123	CA	Nov 21 2023 11:55 AM	Another suspicious entry, since it is still too close to NY
123	NY	Nov 21 2023 6:55 PM	Another suspicious entry, since it is 5 hrs after CA
321	CA	Nov 22 2023 11:01 AM	Valid morning workday entry
321	NY	Nov 22 2023 5:02 PM	Suspicious entry, since it is only 3 hrs after CA
321	CA	Nov 23 2023 9:00 AM	Valid entry, since it is more than 8 hours after NY

Acceptance Criteria

You write the code in Python to read and process the data files provided by us and output the suspicious badge swipes, including the timestamp, the location, and the badge ID.

Guidelines

1. In order to avoid being stuck, please go ahead and make assumptions, but document them, and move on. There is no need to verify every decision with us
2. Document any other significant decisions you make (choice of tools, data structures, performance, etc)
3. Use any Python libraries you want, or any other tool you want—just document your choice
4. It is always possible to do more, but focus on answering the main question first
5. Use the files with swipe data that we have provided in the handout
6. If you ran out of time - submit what you've got; we will still evaluate it and decide whether to proceed

Submission

- Use `git-bundle` to package up your work, including any scripts, test data, etc
- Email it to `interview@coastpay.com`
- Prepare to run your code and possibly add extra functionality during the in-person part of the interview