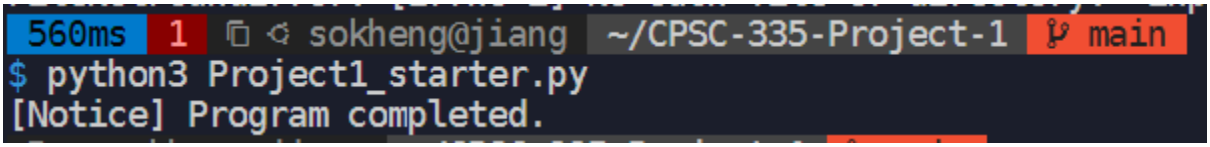
Project 1: Implementing Algorithm

Fall 2023 CPSC 335.01 - Algorithm Engineering
Instructor: Divyansh Mohan Rao
student : Sokheng Teang && Jacob Armstrong
Email: [sokhengteang@csu.fullerton.edu] & [jarmstrong31@csu.fullerton.edu]

Github repo: https://github.com/Vyadin/CPSC-335-Project-1

Screenshot of outputs showing 10 test cases 2 must be edge cases below.



```
==================================
      Example 1
==================================

Person 1's schedule: [['9:30-12:15'], ['13:30-15:45']]
Person 1's login time: [['9:00-16:30']]
Person 2's schedule: [['9:00-12:15'], ['13:30-15:45']]
Person 2's login time: [['9:00-16:00']]
Minimum meeting length: 30 minutes.

+-----------------+
| Compatible Times | 12:15-13:30
+-----------------+

==================================
      Example 2
==================================

Person 1's schedule: [['9:00-10:30'], ['12:00-13:00'], ['16:00-18:00']]
Person 1's login time: [['9:00-19:00']]
Person 2's schedule: [['9:00-10:30'], ['12:20-13:30'], ['14:00-15:00'],
['16:00-17:00']]
Person 2's login time: [['9:00-18:30']]
Minimum meeting length: 30 minutes.

+-----------------+
| Compatible Times | 10:30-12:00, 13:30-14:00, 15:00-16:00, 18:00-18:30
+-----------------+

==================================
      Example 3
==================================

Person 1's schedule: [['9:00-12:00'], ['12:30-14:00'], ['16:00-16:30']]
Person 1's login time: [['8:30-17:00']]
Person 2's schedule: [['9:00-11:30'], ['12:00-12:30'], ['14:00-16:30']]
Person 2's login time: [['9:00-16:30']]
Minimum meeting length: 30 minutes.

[Notice] No compatible times found.
```

```
================================
      Example 4
================================

Person 1's schedule: [['9:15-12:00'], ['12:45-13:30'], ['14:00-19:00']]
Person 1's login time: [['8:00-19:00']]
Person 2's schedule: [['10:00-11:30'], ['13:00-13:30'], ['14:00-15:30']]
Person 2's login time: [['10:00-15:30']]
Minimum meeting length: 45 minutes.

+-----------------+
| Compatible Times | 12:00-12:45
+-----------------+

================================
      Example 5
================================

Person 1's schedule: [['9:00-10:30'], ['12:00-13:00'], ['16:00-18:00']]
Person 1's login time: [['9:00-19:00']]
Person 2's schedule: [['9:00-10:30'], ['12:20-13:30'], ['14:00-15:00'],
['16:00-17:00']]
Person 2's login time: [['9:00-18:30']]
Minimum meeting length: 60 minutes.

+-----------------+
| Compatible Times | 10:30-12:00, 15:00-16:00
+-----------------+

================================
      Example 6
================================

Person 1's schedule: [['9:00-10:30'], ['12:00-13:00'], ['16:00-18:00']]
Person 1's login time: [['9:00-19:00']]
Person 2's schedule: [['9:00-10:30'], ['12:20-13:30'], ['14:00-15:00'],
['16:00-17:00']]
Person 2's login time: [['9:00-18:30']]
Minimum meeting length: 90 minutes.

+-----------------+
| Compatible Times | 10:30-12:00
+-----------------+
```

```
==================================
      Example 7
==================================

Person 1's schedule: [['9:00-10:30'], ['12:00-13:00'], ['16:00-18:00']]
Person 1's login time: [['9:00-19:00']]
Person 2's schedule: [['9:00-10:30'], ['12:20-13:30'], ['14:00-15:00'],
['16:00-17:00']]
Person 2's login time: [['9:00-18:30']]
Minimum meeting length: 120 minutes.

[Notice] No compatible times found.

==================================
      Example 8
==================================

Person 1's schedule: [['9:00-11:15'], ['11:30-12:00'], ['15:12-17:00']]
Person 1's login time: [['8:30-17:00']]
Person 2's schedule: [['8:00-11:00'], ['11:30-12:00'], ['15:00-16:30']]
Person 2's login time: [['8:00-16:30']]
Minimum meeting length: 180 minutes.

+-----------------+
| Compatible Times | 12:00-15:00
+-----------------+

==================================
      Example 9
==================================

Person 1's schedule: [['9:00-10:00'], ['11:45-12:00'], ['15:12-15:30'],
['15:45-16:30']]
Person 1's login time: [['8:30-17:00']]
Person 2's schedule: [['8:00-10:00'], ['11:58-12:00'], ['12:59-15:30'],
['15:45-16:30']]
Person 2's login time: [['8:00-16:30']]
Minimum meeting length: 59 minutes.

+-----------------+
| Compatible Times | 10:00-11:45, 12:00-12:59
+-----------------+
```

```
==================================
        Example 10
==================================

Person 1's schedule: [['9:00-9:15'], ['9:45-10:30'], ['12:15-12:45'],
['13:00-13:10'], ['14:00-16:45']]
Person 1's login time: [['9:00-17:00']]
Person 2's schedule: [['8:00-8:15'], ['9:30-10:00'], ['12:30-12:45'],
['13:00-13:10'], ['13:35-14:25'], ['15:00-16:45']]
Person 2's login time: [['8:00-17:00']]
Minimum meeting length: 15 minutes.

+------------------+
| Compatible Times | 09:15-09:30, 10:30-12:15, 12:45-13:00, 13:10-13:35, 16:45-17:00
+------------------+
```

brief proof argument for the time complexity of my algorithm, including step-counts

First loop for the schedule1 and nested loop for the schedule2

So the combine time complexity is **O(N*M)**

N: is the length of the schedule1

M: is the length of the schedule2

step-count:

```
# Algorithm begins here
for i in range((len(schedule1)-1)):                                  → O(N)
    time1Start = datetime.strptime(schedule1[i][1], timeFormat)      → O(1)
    time1End = datetime.strptime(schedule1[i+1][0], timeFormat)      → O(1)
    time1span = time1End - time1Start                                → O(1)
    for j in range(len(schedule2)-1):                                → O(M)
        time2Start = datetime.strptime(schedule2[j][1], timeFormat)  → O(1)
        time2End = datetime.strptime(schedule2[j+1][0], timeFormat)  → O(1)
        time2span = time2End - time2Start                            → O(1)

        if time1Start < time2End and time1End >= time2Start and time1span >= minTime and time2span >= minTime:

            startTime = max(time1Start, time2Start)
            endTime = min(time1End, time2End)

            if endTime - startTime < minTime:
                continue

            meetingTime = startTime.strftime("%H:%M") + "-" + endTime.strftime("%H:%M")
            compatibleTimes.append(meetingTime)

            timeFound = True

if not timeFound:
    outputFile.write("[Notice] No compatible times found.\n")
else:
    outputFile.write("+------------------+\n")
    outputFile.write("| Compatible Times | ")
    for i in range(len(compatibleTimes)):
        outputFile.write(compatibleTimes[i])
        if i != len(compatibleTimes)-1:
            outputFile.write(", ")
    outputFile.write("\n+------------------+\n")

outputFile.write("\n")
```

If we assume the schedule1 and schedule2 is about the same size, we can assume that the time complexity is O(N^2) where N=M

Consider the loop is the dominant operation since other operator like parsing schedules does not affect overall time complexity.