Our Solution(s) Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(c1 + c2) time | O(c1 + c2) space - where c1 and c2 are the respec
def calendarMatching(calendar1, dailyBounds1, calendar2, dailyBounds2
    updatedCalendar1 = updateCalendar(calendar1, dailyBounds1)
    updatedCalendar2 = updateCalendar(calendar2, dailyBounds2)
    mergedCalendar = mergeCalendars(updatedCalendar1, updatedCalendar
    flattenedCalendar = flattenCalendar(mergedCalendar)
    return getMatchingAvailabilities(flattenedCalendar, meetingDuratio


def updateCalendar(calendar, dailyBounds):
    updatedCalendar = calendar[:]
    updatedCalendar.insert(0, ["0:00", dailyBounds[0]])
    updatedCalendar.append([dailyBounds[1], "23:59"])
    return list(map(lambda m: [timeToMinutes(m[0]), timeToMinutes(m[1]


def mergeCalendars(calendar1, calendar2):
    merged = []
    i, j = 0, 0
    while i < len(calendar1) and j < len(calendar2):
        meeting1, meeting2 = calendar1[i], calendar2[j]
        if meeting1[0] < meeting2[0]:
            merged.append(meeting1)
            i += 1
        else:
            merged.append(meeting2)
            j += 1
    while i < len(calendar1):
        merged.append(calendar1[i])
        i += 1
    while j < len(calendar2):
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def calendarMatching(calendar1, dailyBounds1, calendar2, dailyBounds2,
    # Write your code here.
    pass

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.