Our Solution(s) Run Code

Solution 1

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

// O(c1 + c2) time | O(c1 + c2) space - where c1 and c2 are the respe
function calendarMatching(calendar1, dailyBounds1, calendar2, dailyBo
  const updatedCalendar1 = updateCalendar(calendar1, dailyBounds1);
  const updatedCalendar2 = updateCalendar(calendar2, dailyBounds2);
  const mergedCalendar = mergeCalendars(updatedCalendar1, updatedCale
  const flattenedCalendar = flattenCalendar(mergedCalendar);
  return getMatchingAvailabilities(flattenedCalendar, meetingDuration)
}

function updateCalendar(calendar, dailyBounds) {
  const updatedCalendar = [['0:00', dailyBounds[0]], ...calendar, [dai
  return updatedCalendar.map(meeting => meeting.map(timeToMinutes));
}

function mergeCalendars(calendar1, calendar2) {
  const merged = [];
  let i = 0,
    j = 0;
  while (i < calendar1.length && j < calendar2.length) {
    const meeting1 = calendar1[i],
      meeting2 = calendar2[j];
    if (meeting1[0] < meeting2[0]) {
      merged.push(meeting1);
      i++;
    } else {
      merged.push(meeting2);
      j++;
    }
  }
  while (i < calendar1.length) merged.push(calendar1[i++]);
  while (j < calendar2.length) merged.push(calendar2[j++]);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function calendarMatching(calendar1, dailyBounds1, calendar2, dailyBou
  // Write your code here.
}

// Do not edit the line below.
exports.calendarMatching = calendarMatching;

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.