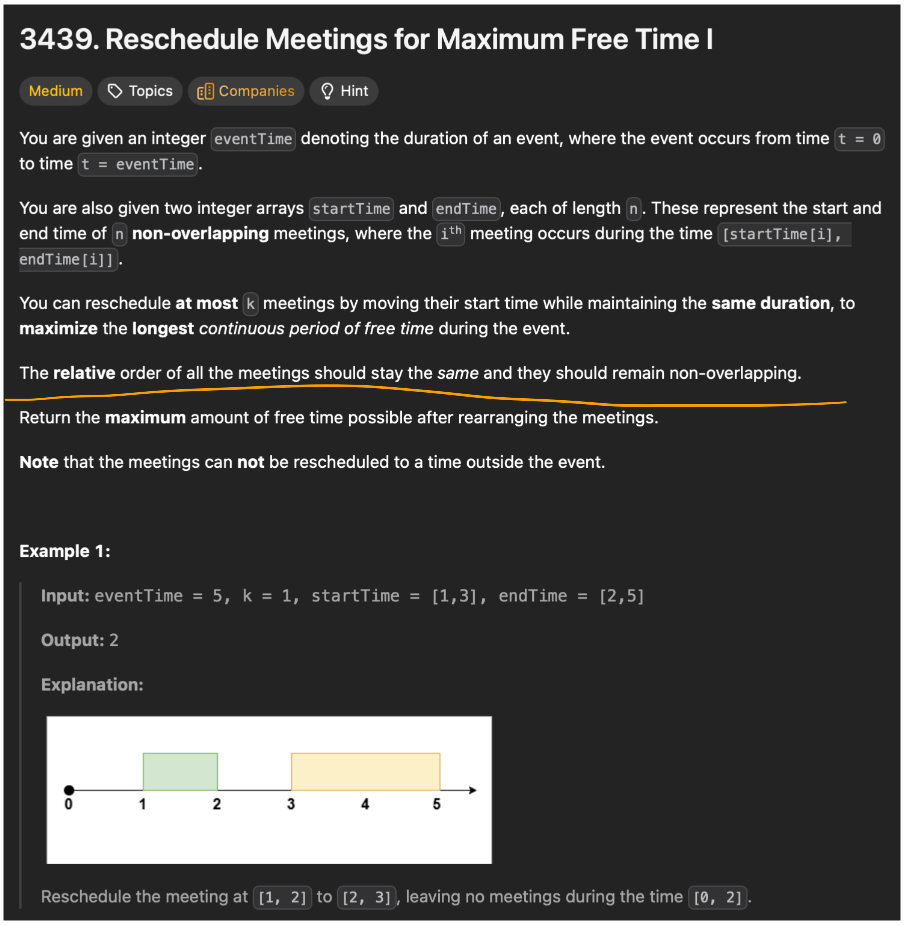


3439. Reschedule Meetings for Maximum Free Time I

Medium · Topics · Companies · Hint

You are given an integer `eventTime` denoting the duration of an event, where the event occurs from time `t = 0` to time `t = eventTime`.

You are also given two integer arrays `startTime` and `endTime`, each of length `n`. These represent the start and end time of `n` **non-overlapping** meetings, where the `i^th` meeting occurs during the time `[startTime[i], endTime[i]]`.

You can reschedule **at most** `k` meetings by moving their start time while maintaining the **same duration**, to **maximize** the **longest** *continuous period of free time* during the event.

The **relative** order of all the meetings should stay the *same* and they should remain non-overlapping.

Return the **maximum** amount of free time possible after rearranging the meetings.

Note that the meetings can **not** be rescheduled to a time outside the event.

Example 1:

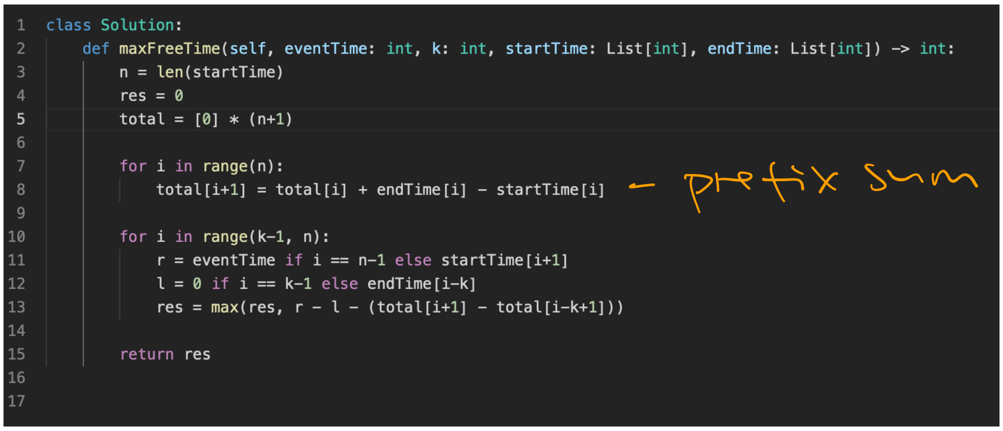
> **Input:** eventTime = 5, k = 1, startTime = [1,3], endTime = [2,5]
>
> **Output:** 2
>
> **Explanation:**
>
>
>
> Reschedule the meeting at `[1, 2]` to `[2, 3]`, leaving no meetings during the time `[0, 2]`.

- Greedy

move k times →

combine at most k+1

gap together



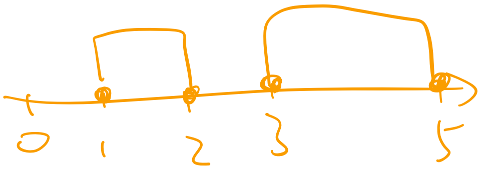
```python
class Solution:
    def maxFreeTime(self, eventTime: int, k: int, startTime: List[int], endTime: List[int]) -> int:
        n = len(startTime)
        res = 0
        total = [0] * (n+1)

        for i in range(n):
            total[i+1] = total[i] + endTime[i] - startTime[i]

        for i in range(k-1, n):
            r = eventTime if i == n-1 else startTime[i+1]
            l = 0 if i == k-1 else endTime[i-k]
            res = max(res, r - l - (total[i+1] - total[i-k+1]))

        return res
```

← prefix sum



total = [0|1|3]

0 2

for i in range(k-1, n):

i=0, r=3, l=0, res = max(0, 3-0-(1-0)) = 2

i=1, r=5, l=2, res = max(2, 5-2-(3-1)) = 2