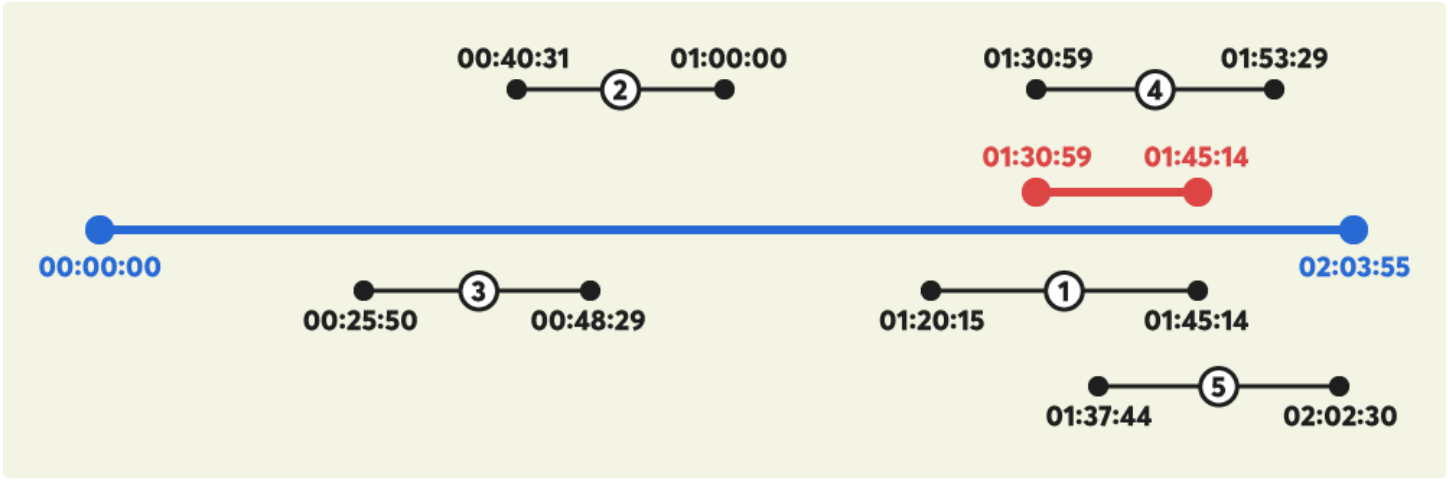
광고 삽입 Python3

문제 설명

카카오TV에서 유명한 크리에이터로 활동 중인 죠르디는 환경 단체로부터 자신의 가장 인기있는 동영상에 지구온난화의 심각성을 알리기 위한 공익광고를 넣어 달라는 요청을 받았습니다. 평소에 환경 문제에 관심을 가지고 있던 "죠르디"는 요청을 받아들였고 광고효과를 높이기 위해 시청자들이 가장 많이 보는 구간에 공익광고를 넣으려고 합니다. "죠르디"는 시청자들이 해당 동영상의 어떤 구간을 재생했는 지 알 수 있는 재생구간 기록을 구했고, 해당 기록을 바탕으로 공익광고가 삽입될 최적의 위치를 고를 수 있었습니다.
참고로 광고는 재생 중인 동영상의 오른쪽 아래에서 원래 영상과 동시에 재생되는 PIP(Picture in Picture) 형태로 제공됩니다.

다음은 "죠르디"가 공익광고가 삽입될 최적의 위치를 고르는 과정을 그림으로 설명한 것입니다.



- 그림의 파란색 선은 광고를 검토 중인 "죠르디" 동영상의 전체 재생 구간을 나타냅니다.
  - 위 그림에서, "죠르디" 동영상의 총 재생시간은 `02시간 03분 55초` 입니다.
- 그림의 검은색 선들은 각 시청자들이 "죠르디"의 동영상을 재생한 구간의 위치를 표시하고 있습니다.
  - 검은색 선의 가운데 숫자는 각 재생 기록을 구분하는 ID를 나타냅니다.

    - 검은색 선에 표기된 왼쪽 끝 숫자와 오른쪽 끝 숫자는 시청자들이 재생한 동영상 구간의 시작 시각과 종료 시각을 나타냅니다.
    - 위 그림에서, 3번 재생 기록은 `00시 25분 50초` 부터 `00시 48분 29초` 까지 총 `00시간 22분 39초` 동안 죠르디의 동영상을 재생했습니다. [1]
    - 위 그림에서, 1번 재생 기록은 `01시 20분 15초` 부터 `01시 45분 14초` 까지 총 `00시간 24분 59초` 동안 죠르디의 동영상을 재생했습니다.
- 그림의 빨간색 선은 "죠르디"가 선택한 최적의 공익광고 위치를 나타냅니다.
    - 만약 공익광고의 재생시간이 `00시간 14분 15초`라면, 위의 그림처럼 `01시 30분 59초` 부터 `01시 45분 14초` 까지 공익광고를 삽입하는 것이 가장 좋습니다. 이 구간을 시청한 시청자들의 누적 재생시간이 가장 크기 때문입니다.
    - `01시 30분 59초` 부터 `01시 45분 14초` 까지의 누적 재생시간은 다음과 같이 계산됩니다.
        - `01시 30분 59초` 부터 `01시 37분 44초` 까지 : 4번, 1번 재생 기록이 두차례 있으므로 재생시간의 합은 `00시간 06분 45초` X 2 = `00시간 13분 30초`
        - `01시 37분 44초` 부터 `01시 45분 14초` 까지 : 4번, 1번, 5번 재생 기록이 세차례 있으므로 재생시간의 합은 `00시간 07분 30초` X 3 = `00시간 22분 30초`
        - 따라서, 이 구간 시청자들의 누적 재생시간은 `00시간 13분 30초` + `00시간 22분 30초` = `00시간 36분 00초`입니다.

[문제]

"죠르디"의 동영상 재생시간 길이 play_time, 공익광고의 재생시간 길이 adv_time, 시청자들이 해당 동영상을 재생했던 구간 정보 logs가 매개변수로 주어질 때, 시청자들의 누적 재생시간이 가장 많이 나오는 곳에 공익광고를 삽입하려고 합니다. 이때, 공익광고가 들어갈 `시작 시각`을 구해서 return 하도록 solution 함수를 완성해주세요. 만약, 시청자들의 누적 재생시간이 가장 많은 곳이 여러 곳이라면, 그 중에서 `가장 빠른 시작 시각`을 return 하도록 합니다.

[제한사항]

- play_time, adv_time은 길이 8로 고정된 문자열입니다.
    - play_time, adv_time은 `HH:MM:SS` 형식이며, `00:00:01` 이상 `99:59:59` 이하입니다.
    - 즉, 동영상 재생시간과 공익광고 재생시간은 `00시간 00분 01초` 이상 `99시간 59분 59초` 이하입니다.
    - 공익광고 재생시간은 동영상 재생시간보다 짧거나 같게 주어집니다.
- logs는 크기가 1 이상 300,000 이하인 문자열 배열입니다.
    - logs 배열의 각 원소는 시청자의 재생 구간을 나타냅니다.
    - logs 배열의 각 원소는 길이가 17로 고정된 문자열입니다.
    - logs 배열의 각 원소는 `H1:M1:S1-H2:M2:S2` 형식입니다.
        - `H1:M1:S1`은 동영상이 시작된 시각, `H2:M2:S2`는 동영상이 종료된 시각을 나타냅니다.
        - `H1:M1:S1`는 `H2:M2:S2`보다 1초 이상 이전 시각으로 주어집니다.
        - `H1:M1:S1`와 `H2:M2:S2`는 play_time 이내의 시각입니다.
- 시간을 나타내는 `HH, H1, H2`의 범위는 00~99, 분을 나타내는 `MM, M1, M2`의 범위는 00~59, 초를 나타내는 `SS, S1, S2`의 범위는 00~59까지 사용됩니다. 잘못된 시각은 입력으로 주어지지 않습니다. (예: `04:60:24`, `11:12:78`, `123:12:45` 등)
- return 값의 형식
    - 공익광고를 삽입할 시각을 `HH:MM:SS` 형식의 8자리 문자열로 반환합니다.

[입출력 예]

play_time	adv_time	logs	result
`"02:03:55"`	`"00:14:15"`	`["01:20:15-01:45:14", "00:40:31-01:00:00", "00:25:50-00:48:29", "01:30:59-01:53:29", "01:37:44-02:02:30"]`	`"01:30:59"`

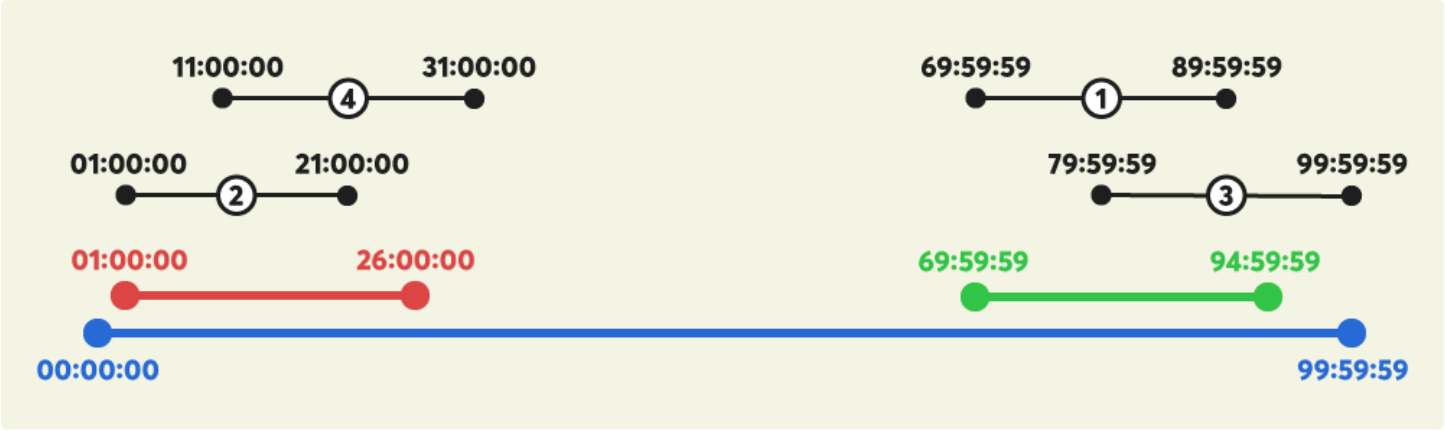
"99:59:59"	"25:00:00"	["69:59:59-89:59:59", "01:00:00-21:00:00", "79:59:59-99:59:59", "11:00:00-31:00:00"]	"01:00:00"
"50:00:00"	"50:00:00"	["15:36:51-38:21:49", "10:14:18-15:36:51", "38:21:49-42:51:45"]	"00:00:00"

입출력 예에 대한 설명

입출력 예 #1

문제 예시와 같습니다.

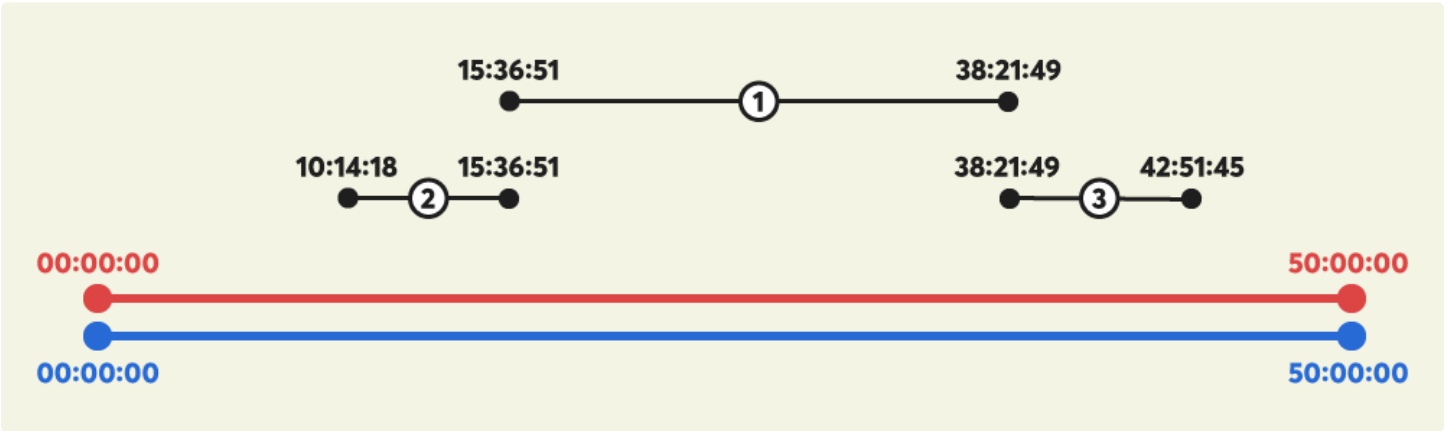
입출력 예 #2



`01:00:00`에 공익광고를 삽입하면 `26:00:00`까지 재생되며, 이곳이 가장 좋은 위치입니다. 이 구간의 시청자 누적 재생시간은 다음과 같습니다.

- `01:00:00-11:00:00` : 해당 구간이 1회(2번 기록) 재생되었으므로 누적 재생시간은 `10시간 00분 00초` 입니다.
- `11:00:00-21:00:00` : 해당 구간이 2회(2번, 4번 기록) 재생되었으므로 누적 재생시간은 `20시간 00분 00초` 입니다.
- `21:00:00-26:00:00` : 해당 구간이 1회(4번 기록) 재생되었으므로 누적 재생시간은 `05시간 00분 00초` 입니다.
- 따라서, 이 구간의 시청자 누적 재생시간은 `10시간 00분 00초` + `20시간 00분 00초` + `05시간 00분 00초` = `35시간 00분 00초` 입니다.
- 초록색으로 표시된 구간(`69:59:59-94:59:59`)에 광고를 삽입해도 동일한 결과를 얻을 수 있으나, `01:00:00`이 `69:59:59` 보다 빠른 시각이므로, `"01:00:00"`을 return 합니다.

입출력 예 #3



동영상 재생시간과 공익광고 재생시간이 같으므로, 삽입할 수 있는 위치는 맨 처음(`00:00:00`)이 유일합니다.

---

1. `동영상 재생시간 = 재생이 종료된 시각 - 재생이 시작된 시각`(예를 들어, `00시 00분 01초`부터 `00시 00분 10초`까지 동영상이 재생되었다면, 동영상 재생시간은 `9초` 입니다.) ↩

solution.py

```python
def solution(play_time, adv_time, logs):
    answer = ''
    return answer
```

실행 결과

실행 결과가 여기에 표시됩니다.

초기화 코드 실행 제출 후 채점하기