

[1차] 추석 트래픽

Python3

문제 설명

추석 트래픽

이번 추석에도 시스템 장애가 없는 명절을 보내고 싶은 어피치는 서버를 증설해야 할지 고민이다. 장애 대비용 서버 증설 여부를 결정하기 위해 작년 추석 기간인 9월 15일 로그 데이터를 분석한 후 초당 최대 처리량을 계산해보기로 했다. **초당 최대 처리량**은 요청의 응답 완료 여부에 관계없이 임의 시간부터 1초(=1,000밀리초)간 처리하는 요청의 최대 개수를 의미한다.

입력 형식

- `solution` 함수에 전달되는 `lines` 배열은 $N(1 \leq N \leq 2,000)$ 개의 로그 문자열로 되어 있으며, 각 로그 문자열마다 요청에 대한 응답완료시간 **S**와 처리시간 **T**가 공백으로 구분되어 있다.
- 응답완료시간 **S**는 작년 추석인 2016년 9월 15일만 포함하여 고정 길이 `2016-09-15 hh:mm:ss.sss` 형식으로 되어 있다.
- 처리시간 **T**는 `0.1s`, `0.312s`, `2s` 와 같이 최대 소수점 셋째 자리까지 기록하며 뒤에는 초 단위를 의미하는 `s`로 끝난다.
- 예를 들어, 로그 문자열 `2016-09-15 03:10:33.020 0.011s`은 "2016년 9월 15일 오전 3시 10분 **33.010초**"부터 "2016년 9월 15일 오전 3시 10분 **33.020초**"까지 "**0.011초**" 동안 처리된 요청을 의미한다. (**처리시간은 시작시간과 끝시간을 포함**)
- 서버에는 타임아웃이 3초로 적용되어 있기 때문에 처리시간은 $0.001 \leq T \leq 3.000$ 이다.
- `lines` 배열은 응답완료시간 **S**를 기준으로 오름차순 정렬되어 있다.

출력 형식

- `solution` 함수에서는 로그 데이터 `lines` 배열에 대해 **초당 최대 처리량**을 리턴한다.

입출력 예제

예제1

- 입력: [
"2016-09-15 01:00:04.001 2.0s",
"2016-09-15 01:00:07.000 2s"
]
- 출력: 1

예제2

- 입력: [
"2016-09-15 01:00:04.002 2.0s",
"2016-09-15 01:00:07.000 2s"
]
- 출력: 2
- 설명: 처리시간은 시작시간과 끝시간을 **포함**하므로
첫 번째 로그는 `01:00:02.003 ~ 01:00:04.002`에서 2초 동안 처리되었으며,
두 번째 로그는 `01:00:05.001 ~ 01:00:07.000`에서 2초 동안 처리된다.
따라서, 첫 번째 로그가 끝나는 시점과 두 번째 로그가 시작하는 시점의 구간인 `01:00:04.002 ~ 01:00:05.001` 1초 동안 최대 2개가 된다.

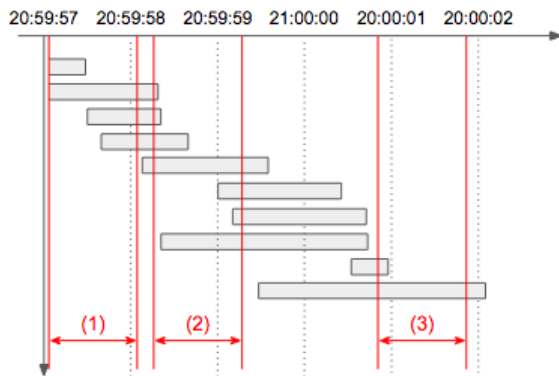
예제3

• 입력: [

"2016-09-15 20:59:57.421 0.351s",
 "2016-09-15 20:59:58.233 1.181s",
 "2016-09-15 20:59:58.299 0.8s",
 "2016-09-15 20:59:58.688 1.041s",
 "2016-09-15 20:59:59.591 1.412s",
 "2016-09-15 21:00:00.464 1.466s",
 "2016-09-15 21:00:00.741 1.581s",
 "2016-09-15 21:00:00.748 2.31s",
 "2016-09-15 21:00:00.966 0.381s",
 "2016-09-15 21:00:02.066 2.62s"
]

• 출력: 7

- 설명: 아래 타임라인 그림에서 빨간색으로 표시된 1초 각 구간의 처리량을 구해보면 (1)은 4개, (2)는 7개, (3)는 2개임을 알 수 있다. 따라서 **초당 최대 처리량**은 7이 되며, 동일한 최대 처리량을 갖는 1초 구간은 여러 개 존재할 수 있으므로 이 문제에서는 구간이 아닌 개수만 출력한다.

[해설 보러가기](#)

solution.py

```
1 def solution(lines):
2     answer = 0
3     return answer
```

실행 결과

실행 결과가 여기에 표시됩니다.

초기화

코드 실행

제출 후 채점하기