

무지의 먹방 라이브

Python3

문제 설명

무지의 먹방 라이브

* 효율성 테스트에 부분 점수가 있는 문제입니다.

평소 식욕이 왕성한 무지는 자신의 재능을 뽐내고 싶어 족고 고민 끝에 카카오 TV 라이브로 방송을하기로 마음먹었다.



그냥 먹방을 하면 다른 방송과 차별성이 없기 때문에 무지는 아래와 같이 독특한 방식을 생각해냈다.

회전판에 먹어야 할 N 개의 음식이 있다.

각 음식에는 1부터 N 까지 번호가 붙어있으며, 각 음식을 섭취하는데 일정 시간이 소요된다.

무지는 다음과 같은 방법으로 음식을 섭취한다.

- 무지는 1번 음식부터 먹기 시작하며, 회전판은 번호가 증가하는 순서대로 음식을 무지 앞으로 가져다 놓는다.
- 마지막 번호의 음식을 섭취한 후에는 회전판에 의해 다시 1번 음식이 무지 앞으로 온다.
- 무지는 음식 하나를 1초 동안 섭취한 후 남은 음식은 그대로 두고, 다음 음식을 섭취한다.
 - 다음 음식이란, 아직 남은 음식 중 다음으로 섭취해야 할 가장 가까운 번호의 음식을 말한다.
- 회전판이 다음 음식을 무지 앞으로 가져오는데 걸리는 시간은 없다고 가정한다.

무지가 먹방을 시작한 지 K 초 후에 네트워크 장애로 인해 방송이 잠시 중단되었다.

무지는 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 섭취해야 하는지를 알고자 한다.

각 음식을 모두 먹는데 필요한 시간이 담겨있는 배열 `food_times`, 네트워크 장애가 발생한 시간 K 초가 매개변수로 주어질 때 몇 번 음식부터 다시 섭취하면 되는지 `return` 하도록 `solution` 함수를 완성하라.

제한사항

- `food_times` 는 각 음식을 모두 먹는데 필요한 시간이 음식의 번호 순서대로 들어있는 배열이다.
- `k` 는 방송이 중단된 시간을 나타낸다.
- 만약 더 섭취해야 할 음식이 없다면 `-1` 을 반환하면 된다.

정확성 테스트 제한 사항

- `food_times` 의 길이는 `1` 이상 `2,000` 이하이다.
- `food_times` 의 원소는 `1` 이상 `1,000` 이하의 자연수이다.
- `k` 는 `1` 이상 `2,000,000` 이하의 자연수이다.

효율성 테스트 제한 사항

- food_times 의 길이는 1 이상 200,000 이하이다.
- food_times 의 원소는 1 이상 100,000,000 이하의 자연수이다.
- k는 1 이상 2×10^{13} 이하의 자연수이다.

입출력 예

food_times	k	result
[3, 1, 2]	5	1

입출력 예 설명

입출력 예 #1

- 0~1초 동안에 1번 음식을 섭취한다. 남은 시간은 [2,1,2] 이다.
- 1~2초 동안 2번 음식을 섭취한다. 남은 시간은 [2,0,2] 이다.
- 2~3초 동안 3번 음식을 섭취한다. 남은 시간은 [2,0,1] 이다.
- 3~4초 동안 1번 음식을 섭취한다. 남은 시간은 [1,0,1] 이다.
- 4~5초 동안 (2번 음식은 다 먹었으므로) 3번 음식을 섭취한다. 남은 시간은 [1,0,0] 이다.
- 5초에서 네트워크 장애가 발생했다. 1번 음식을 섭취해야 할 때 중단되었으므로, 장애 복구 후에 1번 음식부터 다시 먹기 시작하면 된다.

solution.py

```
1 def solution(food_times, k):
2     answer = 0
3     return answer
```

실행 결과

실행 결과가 여기에 표시됩니다.

초기화

코드 실행

제출 후 채점하기