

[문제 설명]

제로는 아름다운 배열 개념의 창시자입니다. 아름다운 배열이란 다음의 식과 같습니다.

$$a_l + a_{l+1} + \cdots + a_r = \frac{1}{2}(a_l + a_r) * (r - l + 1)$$

이 식에 따라, 주어진 배열을 아름다운 배열로 만들고 싶습니다. 저희가 할 수 있는 일은 기존 배열에서 숫자를 하나씩 바꿔가며 아름다운 배열이 되도록 만들 수 있는데, 계속 반복하여 아름다운 배열이 될 때까지 몇번의 작업이 최소한으로 필요한지 반환해주시면 됩니다.

예를 들어 [1, 1, 2, 2] 배열이 주어졌을 때, 이 배열을 아름다운 배열로 만들 수 있는 최소한의 횟수는 2 회로 [1, 1, 1, 1]으로 바꾸면 아름다운 배열이 됩니다.

또한, 소수점으로도 이 배열을 바꿀 수 있습니다. 만약 [3, -2, 4, -1, -4, 0]으로 배열이 주어졌을 때, 이 배열을 [-2.5, -2, -1.5, -1, -0.5, 0]으로 변경하면 아름다운 배열이 됩니다. 이 때 소요된 횟수는 3 회입니다. 3 회보다 더 작은 횟수로 아름다운 배열을 만드는 방법은 없습니다.

[제한 사항]

- 배열의 길이는 1 이상 70 이하입니다.
- 배열의 각 원소는 -100 부터 100 까지 존재합니다.

[입력 형식]

- beautiful 의 이름을 가진 배열이 주어집니다.

[출력 형식]

- 배열을 아름답게 만드는 최소 횟수를 구해주세요. 이미 아름다운 배열이라면 0 을 반환해주시면 됩니다.

제한 시간: 25 분

문제 유형: 완전탐색

난이도: 중

매개변수

| beautiful |
|--------------|
| Array / list |

리턴타입

| 리턴타입 |
|------|
| int |

초기코드 Python

```
def solution(beautiful):  
    ...  
    :param beautiful: list  
    :return: int  
    ...  
    answer = 0  
  
    return answer
```

초기코드 JavaScript

```
/**  
 * @param {array} beautiful  
 * @return {int}  
 */  
function solution(beautiful) {  
    return 0  
}
```

테스트 케이스

예제용

| 입력값 beautiful |
|---------------|
| [1, 2, 3, 4] |

| 출력값 |
|-----|
| 0 |

채점용

| 입력값 beautiful | 출력값 |
|--|-----|
| [9, 0, 0, 36, 45, 54, 0, 72, 0, 0, 0, 100, 100, 100, 0, 100, 100, 0, 0, 0, 100, 100, 0, 0, 100, 100, 0, 0, 100, 100, 0, 100, 100, 100, 100, 0, 0, 0, 0, 0, 100, 100, 0, 0, 100, 0, 100, 0, 0, 0, 100, 100, 0, 0, 0, 0, 0, 100, 0, 100, 100, 0, 0, 0, 0, 100, 0, 100, 100, 0, 0, 0, 0, 100, 100, 100, 0, 0, 0, 100, 0, 0] | 31 |

| 입력값 beautiful | 출력값 |
|--|-----|
| [-4, 0, 0, -22, -28, -34, -40, -46, 0, -58, 0, -70, -76, 0, 0, 0, 0, 0, -100, 0, -100, -100, 0, 0, 0, 0, 0, 0, -100, -100, -100, 0, -100, -100, -100, -100, 0, 0, -100, -100, -100, -100, -100, 0, -100, -100, 0, 0, -100, -100, 0, -100, 0, 0, -100, 0] | 35 |

| | |
|--|--|
| 0, -100, 0, -100, 0, 0, -100, -100, 0, -100, -100, 0, 0] | |
|--|--|

| 입력값 beautiful | 출력값 |
|---|-----|
| [1, -1, 0, -1, -1, 1, -1, 0, 0, -1, -1, 0, 1, 0, -1, -1, -1, -1, 0, 1, 0, -1, -1, 1, -1, 0, -1, 1, -1, -1, 0, 0, 0, 0, 0, 1, 1, -1, 1, 0, 0, 0, 1, -1, 0, 0, 0, -1, 1, 1, 1, 1, -1, 0, -1, 0, -1, 1, 0, -1, 0, -1, -1, -1, 0, -1, 0, 1, 0, 1] | 43 |

| 입력값 beautiful | 출력값 |
|---|-----|
| [-2, -2, -2, 2, -2, 0, -1, 0, 0, 2, 1, -2, 0, 1, -2, 1, -1, 1, -1, -1, 1, 1, -2, 1, -1, 0, 0, -1, -2, 2, 0, -1, -2, 2, 2, 0, 1, -2, 0, -2, -1, 0, 1, 0, 2, 1, -2, -1, 0, 0, -1, -1, -1, -2, 2, -2, 0, 0, 1, -1, -1, 1, 2, 1, 2, -2, 1, -2, -2, 2] | 53 |

| 입력값 beautiful | 출력값 |
|---|-----|
| [22, 67, -52, 67, -28, -94, 5, -86, -67, 51, -28, -80, 35, 33, 11, 35, -94, 76, -80, 86, 45, 95, -43, 34, -31, -55, 93, 76, -55, -32, -78, -30, -2, -75, -8, -58, 82, -34, 57, 45, -10, -68, -20, 23, -63, -13, -42, 72, 90, -76, -45, 35, 62, -60, 50, 79, -25, 71, -41, 17, | 63 |

| | |
|---|--|
| 69, -79, 49, -13, 27, -64, 81] | |
|---|--|