



파이썬 알고리즘 - 백준 6615번: 콜라츠 추측

콜라츠 추측

성공 다국어

★ 한국어 ▾

3 실버 III

| 시간 제한 | 메모리 제한 | 제출 | 정답 | 맞힌 사람 | 정답 비율 |
|-------|--------|-----|-----|-------|---------|
| 1 초 | 128 MB | 427 | 112 | 89 | 31.338% |

문제

콜라츠 추측은 흥미로운 현상이다. 이 법칙은 간단해보이지만, 수학적으로 아직까지 증명되어있지 않은 문제이다. 우리는 이 추측이 옳다고 받아들이겠다.

콜라츠 추측을 설명하면 다음과 같다. 우선 다음과 같은 양의 정수 수열 x_i 를 생각하자.

만약 x_i 가 짝수이면, $x_{i+1} = x_i / 2$

만약 x_i 가 홀수이면, $x_{i+1} = 3 * x_i + 1$ 이다.

콜라츠 추측은 이렇게 만든 수열은 결국 1이 된다는 것이다.

과학자들은, 컴퓨터를 이용하여 첫 번째 수열이 2^{58} 보다 작으면, 이 추측은 참이라고 증명했다.

이제 문제를 보자.

두개의 양의 정수를 준다. 각각의 수에 대해서 콜라츠 추측으로 만든 수열을 생각하자.

각각의 수열을 비교하였을때 처음으로 같은 숫자가 나왔을때, 각각 몇번째 수열에서 만나는지 구해본다. 문제의 편의를 위해, 이 수열은 1이 나오면 더이상 진행하지 않는다고 하자. (1 다음에 나올 수열을 생각하면, 1, 4, 2, 1, 4, 2, 1로 반복되기 때문이다.)

입력

입력은 몇개의 테스트 케이스로 구성된다. 각 테스트 케이스는 두개의 정수 A와 B가 주어진다. ($1 \leq A, B \leq 1,000,000$) 마지막 줄은 두개의 0으로 구성된다.

출력

각각의 테스트 케이스마다 다음과 같은 문장을 한줄에 출력한다.

"A needs S_A steps, B needs S_B steps, they meet at C"

S_A 와 S_B 는 A와 B로 수열을 만들고, 처음으로 같은 숫자 C가 나왔을때 각각의 수열에서 몇번째 인지 알려주는 숫자이다.

<https://www.acmicpc.net/problem/6615>

소재를 보고 풀게 된 문제.

콜라츠 추측은 정말 흥미로운 난제입니다.



제발 풀려고 하지 마세요! (아무도 못푼 쉬운 문제. 콜라츠 추측)

제발 풀려고 하지 마세요! (아무도 못푼 쉬운 문제. 콜라츠 추측)



```

def collatz_conjecture(a, b):
    s = []
    x = a
    s.append(x)
    while x != 1:
        if x % 2 == 0:
            x = x // 2
        else:
            x = 3*x + 1
        s.append(x)
    x = b
    steps = 0
    while x != 1:
        if x in s:
            break
        if x % 2 == 0:
            x = x // 2
        else:
            x = 3*x + 1
        steps += 1
    result = f"{a} needs {s.index(x)} steps, {b} needs {steps} steps, they meet at {x}"
    print(result)

while True:
    a, b = map(int, input().split())
    if a == b == 0:
        break
    collatz_conjecture(a, b)

```

함수 안 while문 덩어리가 마음에 들지는 않지만, 따지려 들면 끝도 없습니다.

콜라츠 추측으로 만든 각각의 수열을 비교하여 처음으로 같은 숫자가 나왔을 때, 각각 몇 번째 수열에서 만나는지 구해야 합니다.

그러니 수열 하나 먼저 구해놓고, 다음 수열 구할 때 이전 수열에 포함된 숫자가 나올 때까지 수를 구해나가면 됩니다.