1860. Incremental Memory Leak

Description

You are given two integers `memory1` and `memory2` representing the available memory in bits on two memory sticks. There is currently a faulty program running that consumes an increasing amount of memory every second.

At the `i th` second (starting from 1), `i` bits of memory are allocated to the stick with **more available memory** (or from the first memory stick if both have the same available memory). If neither stick has at least `i` bits of available memory, the program **crashes**.

Return *an array containing* `[crashTime, memory1 crash, memory2 crash]` *, where* `crashTime` *is the time (in seconds) when the program crashed and* `memory1 crash` *and* `memory2 crash` *are the available bits of memory in the first and second sticks respectively*.

Example 1:

```
Input: memory1 = 2, memory2 = 2
Output: [3,1,0]
Explanation: The memory is allocated as follows:
- At the 1st second, 1 bit of memory is allocated to stick 1. The first stick now has 1 bit of available memory.
- At the 2nd second, 2 bits of memory are allocated to stick 2. The second stick now has 0 bits of available memory.
- At the 3rd second, the program crashes. The sticks have 1 and 0 bits available respectively.
```

Example 2:

```
Input: memory1 = 8, memory2 = 11
Output: [6,0,4]
Explanation: The memory is allocated as follows:
- At the 1st second, 1 bit of memory is allocated to stick 2. The second stick now has 10 bit of available memory.
- At the 2nd second, 2 bits of memory are allocated to stick 2. The second stick now has 8 bits of available memory.
- At the 3rd second, 3 bits of memory are allocated to stick 1. The first stick now has 5 bits of available memory.
- At the 4th second, 4 bits of memory are allocated to stick 2. The second stick now has 4 bits of available memory.
- At the 5th second, 5 bits of memory are allocated to stick 1. The first stick now has 0 bits of available memory.
- At the 6th second, the program crashes. The sticks have 0 and 4 bits available respectively.
```

Constraints:

- `0 <= memory1, memory2 <= 2 31 - 1`