



Mancala'h

locked

by IEEEExtreme

Problem

Submissions

Leaderboard

Discussions

You and your friend Alessandro are taking part to an archaeological mission that aims to explore a newly discovered tomb of an ancient pharaoh in Egypt. After an adventurous trip through tunnels, doors and rooms, you and your fellow archaeologists arrive in front of a huge closed door and find a mysterious artefact that appears to be a sort of puzzle for opening the door. The artefact is composed of some inscriptions and a massive wooden disc with many bins carved at the perimeter. Some of the bins contain seeds.

On a side of the artefact, an inscription says: "Harvest all the seeds in the first bin. Sow them in the following bins, one-by-one. Rotate the disc by one step. Repeat."

On the other side of the artefact, a second inscription says: "How far you can go before starting to endless repeat the same harvesting pattern? How far can you go backwards?"

After a fellow archaeologist translated the inscriptions, your friend Alessandro exclaims: "I know this game: it's *Mancala'h*! I know that from every possible configuration, the game evolves to a periodic status, meaning that at some point you start repeating the same pattern. For example, the configuration $[1, 1, 1]$ evolves to $[0, 2, 1]$ because you take the seed in the first bin and put it in the second bin. Then, after a clockwise rotation of the disc, the second bin with two seeds becomes the first and we can remove the leading zero from notation. So, the configuration is now $[2, 1]$, which evolves to $[2, 1]$ itself, because you take the two seeds in the first bin and put one in the second bin and one in the third bin, which was empty. In this case we can say that the depth of the original $[1, 1, 1]$ configuration is equal to 1, meaning that in a single step we reach a periodic status."

"The situation can be more complex, for example the configuration $[4]$ evolves to $[1, 1, 1, 1]$, which in turn evolves to $[2, 1, 1]$, which evolves to $[2, 2]$, which evolves to $[3, 1]$, which evolves again to $[2, 1, 1]$, and so on. In this case the depth of the configuration $[4]$ is 2 and the period is 3 steps long."

"The first thing we need to find is exactly the depth of the configuration we have found here."

"Going backwards in the game evolution is not as easy as going forward: for example, we have seen that configurations $[1, 1, 1, 1]$ and $[3, 1]$ evolve both to $[2, 1, 1]$ in one step. On the contrary, the configuration $[1, 2]$ cannot be the evolution of any valid configuration, because we *exclude* all the so-called *not connected* configurations, which are those that contain a zero, such as $[1, 0, 2]$."

"The second thing we need to find, in fact, is the height of the given configuration, which is equal to the maximum number of backwards steps or, in other words, the distance (in terms of number of steps) of the farthest previous configuration. We cannot iterate backwards over the period, so the height is the length of the longest sequence of *unique* configurations that leads to the current configuration."

Help the archaeologists to solve the mystery by finding the depth D and the height H of given Mancala'h configurations.

Input Format

The input contains a single Mancala'h configuration. A Mancala'h configuration is defined by the sequence $[N_1, N_2, N_3, \dots, N_L]$ of L ($1 \leq L \leq 100$) integers separated by a blank-space. Each integer N_i ($1 \leq N_i \leq 500$) represents the number of seeds in the i^{th} bin.

Constraints

The size of the Mancala'h board is large enough that you will never have a board in which all of the bins are filled. In other words, there are always more bins than seeds.

The number of unique configurations reachable from any input (either forwards or backwards) is at most $5 \cdot 10^6$.

Output Format

The output is a single line containing two integers D and H , separated by a blank-space. The first integer D is the depth of the Mancala'h configuration specified in the input. The second integer H is the height of the Mancala'h configuration specified in the input.

Sample Input

```
2 1 1
```

Sample Output

0 3

Explanation

The depth of the Mancala'h configuration [2, 1, 1] is 0 because the period includes the given configuration:

[2, 1, 1] → [2, 2] → [3, 1] → [2, 1, 1]

The height is 3 because there are three backwards evolutions, two of which contain 3 steps:

[2, 1, 1] ← [1, 1, 1, 1] ← [4] ← [1, 3]
 [2, 1, 1] ← [3, 1] ← [2, 2] ← [1, 1, 2]
 [2, 1, 1] ← [3, 1] ← [1, 2, 1]

Note that for the following is not a valid backwards evolution because it repeats the configuration [2, 1, 1]:

[2, 1, 1] ← [3, 1] ← [2, 2] ← [2, 1, 1]

As an additional example, suppose that the input was:

3 1 2 3 2

For this Mancala'h, the depth is 6:

[3, 1, 2, 3, 2] → [2, 3, 4, 2] → [4, 5, 2] →
 [6, 3, 1, 1] → [4, 2, 2, 1, 1, 1] → [3, 3, 2, 2, 1] →
 [4, 3, 3, 1] → [4, 4, 2, 1] → [5, 3, 2, 1] →
 [4, 3, 2, 1, 1] → [4, 3, 2, 2] → [4, 3, 3, 1]

The height is 1 because the only previous configuration possible is:

[3, 1, 2, 3, 2] ← [1, 2, 1, 2, 3, 2]

Max Score: 82pts dynamic

Submissions: 327

Max Score: 82

Difficulty: Hard

[More](#)

Current Buffer (saved locally, editable)  

Java 8   

```

1  import java.io.*;
2  import java.util.*;
3
4  public class Solution {
5
6      public static void main(String[] args) {
7          /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
8      }
9  }

```

Line: 1 Col: 1

 [Upload Code as File](#)
☐ [Test against custom input](#)

Run Code

Submit Code

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Interview Prep](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [Terms Of Service](#) | [Privacy Policy](#)