Codility

Candidate Report: training64BV8E-ACA

Check out Codility training tasks

Test Name:

Summary | Timeline

Tasks summary

Task	Time spent	Score
ChocolatesByNumbers Python	6 min	100%

Total score

Tasks Details

Easy

1. ChocolatesByNumbers

There are N chocolates in a circle. Count the number of chocolates you will eat.

Task Score	Correctness	Performance
100%	100%	100%

Task description

Two positive integers N and M are given. Integer N represents the number of chocolates arranged in a circle, numbered from 0 to N − 1.

You start to eat the chocolates. After eating a chocolate you leave only a wrapper.

You begin with eating chocolate number 0. Then you omit the next M − 1 chocolates or wrappers on the circle, and eat the following one.

More precisely, if you ate chocolate number X, then you will next eat the chocolate with number (X + M) modulo N (remainder of division).

You stop eating when you encounter an empty wrapper.

For example, given integers N = 10 and M = 4. You will eat the following chocolates: 0, 4, 8, 2, 6.

The goal is to count the number of chocolates that you will eat, following the above rules.

Write a function:

```
def solution(N, M)
```

that, given two positive integers N and M, returns the number of chocolates that you will eat.

For example, given integers N = 10 and M = 4. the function should return 5, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N and M are integers within the range [1..1,000,000,000].

Solution

Programming language used:	Python
Total time used:	6 minutes
Effective time used:	6 minutes
Notes:	*not defined yet*

Task timeline

22:48:21 22:53:43

Code: 22:53:43 UTC, py, final, score: 100

show code in pop-up

```
1  # you can write to stdout for debugging purposes, e.g.
2  # print("this is a debug message")
3
4  def solution(N, M):
5      # write your code in Python 3.6
6      if N==M:
7          return 1
8      if N==1:
9          return 1
10     if M==1:
11         return N
12
13     i=1
14     while True:
15         j=(N*i)//M-1
16         while N*i > M*j:
17             j+=1
18             if N*i==M*j:
19                 return j
20         i+=1
21
22     return 0
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(log(N + M))**

expand all	Example tests	
▶	example example test	✔ OK
expand all	**Correctness tests**	
▶	extreme_small very small N and M	✔ OK
▶	simple simple test, N = 24, M = 18	✔ OK
▶	small1 small tests	✔ OK
▶	small2 small tests	✔ OK
expand all	**Performance tests**	
▶	medium medium tests	✔ OK
▶	large large tests	✔ OK
▶	large2 N = (3**9)*(2**14), M=(2**14)*(2**14)	✔ OK
▶	extreme_large maximal and minimal values	✔ OK