***Maximum Solution***

C371\_Coding\_November2022

**Topic**: Searching Algorithms

**Difficulty Level:** Medium

**Question / Problem Statement**:

Jason and Caleb decided that problem **X** needs at least **N** distinct solutions to be written. It does not matter how many solutions each of them will write, they need to write at least **N** solutions in total.

Jason needs **T**1 units of time to write a solution, and Caleb needs **T**2 units of time. They start to work simultaneously at time 0. For example, Jason finishes writing his first solution at time **T**1, his second solution at 2∗**T**1, and so on.

Jason and Caleb are working using the same algorithm. Each time Jason or Caleb finishes writing a solution, he checks on how many solutions have already been written up to the current time moment **T**. If the number of such solutions is strictly less than **N**, then he starts writing the next solution. If a member begins working on a problem, he does not stop working under any circumstances and he will surely finish it.

Jason and Caleb realise that if they act on this algorithm, they will not necessarily write exactly **N** solutions in total.

Write a program to find how many solutions they wrote in total and the moment when the latest solution was finished.

**Note**

Considering that Jason and Caleb work non-stop.

**Function Description**

In the provided code snippet, implement the provided **maximumSolution(...)** method using the variables to find how many solutions they wrote in total and the moment when the latest solution was finished .You can write your code in the space below the phrase **“WRITE YOUR LOGIC HERE”**.   
  
There will be multiple test cases running so the Input and Output should match exactly as provided.  
The base Output variable **result** is set to a default value of **-404** which can be modified. Additionally, you can add or remove these output variables.

**Input Format**

The first line of input contains three space-separated integers N, T1 and T2.

**Sample Input**

2 1 2 - denotes N, T1 and T2.

**Constraints**

1 <= **N, T1, T2** <= 10^9.

**Output Format**

Output should find how many solutions they wrote in total and the moment when the latest solution was finished.

**Sample Output**

3 2

**Explanation**

Jason will finish writing his first solution at T = 1. At this moment, only 1 solution is completed. Therefore, he will start writing his second solution and finish at T = 2. At this moment, Caleb will also finish writing his first solution. Thus, a total of 3 solutions will be written and the latest time is T = 2.

**Solution Steps**

1. Since both Jason and Caleb will only write solutions when the current sum of solutions written is less than N, therefore it is easy to deduce that the maximum number of solutions that can be written are only N+1.

2. Thus, we only need to find who is going to write the last solution. This can be done by using binary search on the number of solutions written by both Jason and Caleb. Let's say Jason and Caleb will write b and a solutions respectively.

T = finish time = max(T1\*b, T2\*a).

S = number of solutions written = min(N+1, T/T1 + T/T2).

**Running Solution in C++** :

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n, t1, t2;

cin >> n >> t1 >> t2;

long long lo = 0, hi = 1.1e18;

while (hi - lo > 1) {

long long mid = (lo + hi) / 2;

(mid / t1 + mid / t2 < n ? lo : hi) = mid;

}

long long ans = hi;

cout << (ans + t1 - 1) / t1 + (ans + t2 - 1) / t2 << ' ';

if (ans % t1)

ans = (ans + t1 - 1) / t1 \* t1;

else

ans = (ans + t2 - 1) / t2 \* t2;

cout << ans << '\n';

}

Input:

2 2 4

Output:

3 4

**Test Cases [ Qty: 12 ]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No** | **Input** | **Output** | **Score** |
| 1 | 2 1 2 | 3 2 | 0 |
| 2 | 2 2 4 | 3 4 | 0 |
| 3 | 4 5 5 | 4 10 | 1 |
| 4 | 352 6973 4250 | 353 934382 | 1 |
| 5 | 7650 2079 1669 | 7651 7083236 | 1 |
| 6 | 7556 5635 9262 | 7557 26480058 | 1 |
| 7 | 4145 6915 6812 | 4146 14230268 | 1 |
| 8 | 3788 9296 7366 | 3789 15571724 | 1 |
| 9 | 805 59019665 41414377 | 806 19630414698 | 1 |
| 10 | 916 59696054 71240121 | 917 29788330946 | 1 |
| 11 | 704 75295791 97523048 | 705 29967724818 | 1 |
| 12 | 760 75228849 799161 | 761 601830792 | 1 |

Plagiarism found – No

Clarity of the problem statement - Yes

Clarity of the example in the problem statement - Yes

Clarity of sample test cases - Yes

Clarity of test cases (Dual output) – Yes

Clarity of explanations - Yes

Provided Solution running – Yes

EEOC complaint (using abusive words/Indian Names/) - No

Similar Question in System - No

Difficulty Level – Medium

Question w.r.t Searching algorithms concepts- Yes

Final Comment: **Accepted**