Programming Assignment 2

Darin Goldstein

1 Deadline

10/1/2020 at 5 PM

2 Recursion Counting

There are multiple recursive methods for computing the greatest common divisor of two numbers, a and b. The one that is taught most often in class is the Euclidean Algorithm. However, there are other recursive methods that take advantage of the fact that, for example, computers can divide by 2 very quickly (via bit shift). Stein's algorithm (named for its discoverer Josef Stein, 1967, though this algorithm may have been known to the 1st century Chinese) works as follows.

Given two nonnegative integers, a and b, repeatedly, in order, apply the following identities.

1. $gcd(0, b) = b$, $gcd(a, 0) = a$, and $gcd(0, 0) = 0$
2. If a and b are both even, then $gcd(a, b) = 2gcd(a/2, b/2)$.
3. If a is even and b is odd, then $gcd(a, b) = gcd(a/2, b)$ (and vice versa if a is odd and b is even).
4. If a and b are both odd, then $gcd(a, b) = gcd(\max\{a, b\} - \min\{a, b\}, \min\{a, b\})$. (Stein's algorithm actually divides the first parameter $\max\{a, b\} - \min\{a, b\}$ by 2 in this step, but for the sake of making this programming assignment easier to understand, we will not.)

It is clear that because division by 2 is implemented by a simple bit shift in computers, whenever a division by 2 is needed, it is extremely fast.

On another planet, the computers are able to be designed so that it is easy to divide by multiple numbers, not necessarily just 2. In that case, steps 2 and 3 of Stein's algorithm above can be repeated in increasing order of the "easily-dividable" numbers before proceeding to the final step. For example, if the "easily-dividable" numbers are 3, 4, and 7, then the Stein-equivalent gcd algorithm would look like the following:

Repeatedly applying the following identities.

1. $gcd(0, b) = b$, $gcd(a, 0) = a$, and $gcd(0, 0) = 0$
2. If a and b are both divisble by 3, then $gcd(a, b) = 3gcd(a/3, b/3)$.
3. If a is divisible by 3 and b is not, then $gcd(a, b) = gcd(a/3, b)$ (and vice versa if a is not and b is).
4. If a and b are both divisble by 4, then $gcd(a, b) = 4gcd(a/4, b/4)$.
5. If a is divisible by 4 and b is not, then $gcd(a, b) = gcd(a/4, b)$ (and vice versa if a is not and b is).
6. If a and b are both divisble by 7, then $gcd(a, b) = 7gcd(a/7, b/7)$.
7. If a is divisible by 7 and b is not, then $gcd(a, b) = gcd(a/7, b)$ (and vice versa if a is not and b is).
8. If we reach this point, then $gcd(a, b) = gcd(\max\{a, b\} - \min\{a, b\}, \min\{a, b\})$.

If you are designing a computer system to perform well on a particular set of problems, then it is clearly in your favor to do more of the middle steps than having to drop to the final "If we reach this point..." step, and in particular, it is strongly in your favor to do more of the "If a and b are *both* divisible by..." steps than any other.

You are in charge of designing a computer system that has the option of choosing multiple numbers to be "easily dividable" in Stein's algorithm. You will be given n options to choose from and your goal will be to compute the score for each, given a set S of representative gcd problems. A number x that is a potential "easily dividable" number will be awarded *points* as follows.

- If, when performing $gcd(a, b)$, the rule "If a and b are both divisble by x..." is applied any time during the computation, then x is awarded 2 points each time the rule is applied.
- If, when performing $gcd(a, b)$, the rule "If a is divisible by x and b is not..." (or vice versa) is applied, then x is awarded 1 point each time the rule is applied *before the last step is reached.* If the last step is reached, then any application of this rule afterwards is not awarded a point.

2.1 Inputs and outputs

The first line of the input will be the number of "easily-dividable" numbers n.

The following n numbers, one per line, will be the "easily-dividable" numbers. You can expect each of these to fit inside of a standard Java integer.

All following lines will be gcd problems in the set S. Each line will consist of a number a, a space, and then another number b. (Note that you will distinguish these lines from the single numbers by the fact that each of these lines contains a space.) These numbers could become arbitrarily large.

Each line of the output should consist of an "easily-dividable" number followed by a space and then its score.