The work of the first author was supported by NSF Grant MCS77-02113, and that of the second author by NSF Grant MCS78-01794.

References

1. E. A. Bender, The number of fanout-free functions with various gates, J. Assoc. Comput. Mach., 27 (1980) 181–190.
2. E. A. Bender and J. T. Butler, Enumeration of functions realized by fanout-free networks of general multi-valued gates, Proc. Intl. Symp. Multi-Valued Logic (1979), 94–103.
3. J. T. Butler, On the number of functions realized by cascades and disjunctive networks, IEEE Trans. Comput., C-24 (1975) 681–690.; MR54 4831.
4. H. A. Curtis, A New Approach to the Design of Switching Circuits, Van Nostrand, 1962, Chapter 4.

CLASSROOM NOTES

EDITED BY DEBORAH TEPPER HAIMO AND FRANKLIN TEPPER HAIMO

Material for this department should be sent to Professor Deborah Tepper Haimo, Department of Mathematical Sciences, University of Missouri, St. Louis MO 63121.

AN ALTERNATIVE TO EUCLID'S ALGORITHM

DANIEL A. MARCUS
Department of Mathematics, California State Polytechnic University, Pomona, CA 91768

Euclid's algorithm for constructing the greatest common divisor of two integers plays at least two important roles in the standard exposition of elementary number theory. Besides serving a practical purpose for solving numerical problems, it also provides the theoretical basis for the Unique Factorization Theorem. Yet the algorithm has some undesirable features. In its general form, it involves a fair amount of obscure subscripted notation. As a practical procedure, it often requires cumbersome algebraic manipulations that are highly prone to errors and are not well suited to efficient implementation on a calculator. This note will present an alternative approach that accomplishes the same results as Euclid's algorithm but which is free from the drawbacks described above.

Euclid's Algorithm. Let m and n be two positive integers. Euclid's algorithm constructs the greatest common divisor (GCD) d of m and n along with integers a and b such that $am + bn = d$. The well-known procedure involves repeated division, resulting in a sequence of equations

$$\begin{aligned} m &= q_1 n + r_1 \quad (0 \leqslant r_1 < n) \\ n &= q_2 r_1 + r_2 \quad (0 \leqslant r_2 < r_1) \\ r_1 &= q_3 r_2 + r_3 \quad (0 \leqslant r_3 < r_2) \\ &\vdots \end{aligned}$$

in which the r_i and q_i are integers. The process terminates when some remainder r_k is equal to 0, which must happen eventually. Then the GCD is r_{k-1}, and the coefficients a and b are found as follows: Solve the first equation for r_1 and substitute into the second and third equations to eliminate r_1; then solve the new second equation for r_2 and substitute into the third and fourth equations to eliminate r_2; etc. Eventually an equation of the form $am + bn = r_{k-1}$ is obtained, where a and b are integers.

The Alternative. The existence of a and b when m and n are relatively prime integers plays a crucial role in the proof of the Unique Factorization Theorem; see, for example, [2, p. 26]. It is

known, however, that this existence can be established by a simple nonconstructive argument that does not depend on Euclid's algorithm. (See [2, p. 26, Exercise 12], [1, p. 20], or [3, p. 7].) The argument runs as follows: Given relatively prime integers m and n, let S denote the set of all integers of the form $am + bn$, for all integers a and b. It is clear that S is closed under both addition and subtraction. From these properties it follows that, if s is the least positive member of S, then S contains all multiples of s and no other integers. Since m is clearly a member of S, s must be a divisor of m. Similarly, s is a divisor of n. Since m and n are assumed to be relatively prime, we conclude that $s = 1$. Thus $am + bn = 1$ for some integers a and b.

As for the practical function of Euclid's algorithm, we note first that when all that is required is the GCD of two positive integers m and n then there is no need to compute the quotients q_i, and consequently the language and techniques of modular arithmetic can be utilized: Reduce the larger of m and n modulo the smaller and let the reduced value replace the larger number. Repeat this procedure until 0 is obtained; the last nonzero value is the GCD.

What is less obvious is that the coefficients a and b in the equation $am + bn = d$ can also be obtained easily without computing the q_i. The first step is to perform the sequence of reductions described above. With m denoting the larger of the two numbers (which we will assume from now on), the results are arranged in the form

$$\begin{array}{cccccc} m & m_1 & m_1 & m_2 & m_2 & \cdots \\ n & n & n_1 & n_1 & n_2 & \cdots \end{array}$$

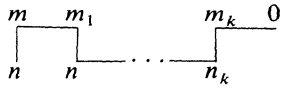
where m reduces to m_1 modulo n, n reduces to n_1 modulo m_1, etc. The process terminates in one of two ways:

$$\begin{array}{ccc} \cdots & m_k & 0 \\ \cdots & n_k & n_k, \end{array} \qquad \text{(case 1)}$$

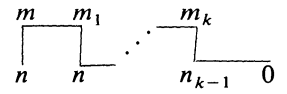
in which case $d = n_k$, or

$$\begin{array}{ccc} \cdots & m_k & m_k \\ \cdots & n_{k-1} & 0, \end{array} \qquad \text{(case 2)}$$

in which case $d = m_k$. In either case, construct a path from n to 0 consisting of alternating vertical and horizontal segments as shown:



(case 1)

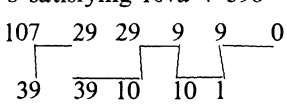


(case 2).

Now follow the path backwards from 0 to n, taking 0 as the initial value and performing the operations indicated by

$$\underset{\times}{\leftarrow} \qquad \downarrow +d) \div \qquad \uparrow -d) \div$$

on the numbers encountered along the way. That is, when moving to the left along the path, multiply by the next number on the path; when moving downward, add d and then divide by the next number; when moving upward, subtract d and divide by the next number. The result at the end of the path is the coefficient b, from which a can then be obtained easily. As an illustration of this procedure, we obtain a and b satisfying $107a + 39b = 1$.



The sequence of operations is

$$0 \times 9) + 1) \div 1) \times 10) - 1) \div 9) \times 29) + 1) \div 10) \times 39) - 1) \div 29) \times 107) + 1) \div 39 = 11,$$

where the first member of each pair of parentheses is understood to occur at the beginning of the calculation. Thus $b = 11$, and consequently $a = -4$.

The sequence of operations above is carried out quickly and easily on a calculator, without any need to store data along the way. There is also no need to write down the operations as we did above, since they can be read directly off the path.

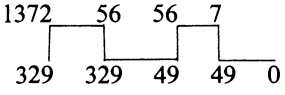
No rounding errors occur because all numbers computed turn out to be integers. This fact, which also provides a check on the calculations, will be established along with the validity of the algorithm.

The algorithm can be expressed in a particularly compact form in reverse Polish notation, utilizing the symbol "\", where "$a \setminus b$" means "b/a":

$$\text{Case 1:} \quad nm\ldots 0 * d + \setminus * d - \setminus \ldots * d + \setminus$$

$$\text{Case 2:} \quad nm\ldots 0 * d - \setminus * d + \setminus \ldots * d + \setminus$$

As another illustration, we compute a and b satisfying $1372a + 329b = 7$.



$$0 \times 49) - 7) \div 7) \times 56) + 7) \div 49) \times 329) - 7) \div 56) \times 1372) + 7) \div 329 = -25.$$

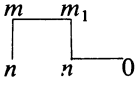
Thus $b = -25$, and consequently $a = 6$.

The proof that this algorithm gives the correct value of b and that all numbers computed are integers can be accomplished by induction on the number of steps, which we define to mean the number of horizontal segments in the path from 0 to n.

Suppose the algorithm requires only one step.

Then the GCD is n and the calculation correctly gives $0 \times m) + n) \div n = 1$ as the value of b.

Next, suppose the algorithm ends in two steps.



Then the GCD is m_1 and the calculation gives

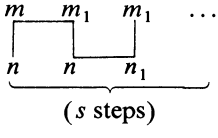
$$0 \times n) - m_1) \div m_1) \times m) + m_1) \div n = (-m + m_1) \div n$$

as the value of b. This is an integer, since $m \equiv m_1 (\bmod n)$, and the equation

$$m + \left(\frac{-m + m_1}{n}\right)n = m_1$$

shows that the value of b is correct.

Finally, we establish the inductive step. Consider a case in which the algorithm requires $s \geqslant 3$ steps and suppose the algorithm is known to give the correct result and only integer values throughout the calculation whenever it ends in fewer than s steps. The induction will proceed from $s - 2$ steps to s steps.



Since $s \geqslant 3$, m_1 and n_1 are nonzero. The first $s - 2$ steps of the calculation result in an integer b_1 such that

$$a_1 m_1 + b_1 n_1 = d$$

for some integer a_1. (Note that the GCD of m_1 and n_1 is the same as that of m and n.) Also, all values calculated up to that point are integers. The entire calculation gives

$$b_0 = b_1 n - d) \div m_1) \times m) + d) \div n.$$

The congruence

$$b_1 n - d \equiv b_1 n_1 - d = -a_1 m_1 \equiv 0 \pmod{m_1},$$

shows that

$$r = (b_1 n - d) \div m_1$$

is an integer. Moreover

$$b_0 = (rm + d) \div n$$

is an integer since

$$rm + d \equiv rm_1 + d = b_1 n \equiv 0 \pmod{n}.$$

Finally, the equation

$$-rm + b_0 n = d$$

shows that $b_0 = b$. The induction is now complete.

It is useful to note that $a = -r$ in the notation above and that r is the number calculated by the first $s - 1$ steps of the algorithm.

Another helpful shortcut is the following: Whenever the algorithm requires at least two steps, the calculation can begin directly to the left of the GCD on the path. In Case 1, it begins at n_{k-1} and the initial value is taken to be n_{k-1}. Thus, in the first example given above, the calculation could begin with

$$10 - 1) \div 9) \times 29) \ldots .$$

In Case 2, the calculation starts at m_{k-1}, but the initial value must be taken to be $-m_{k-1}$. In the second example we would calculate

$$-56 + 7) \div 49) \times 329) \ldots .$$

It is clear that the algorithm described above can be adapted to any Euclidean domain, such as the ring of polynomials over a field.

The algorithm was introduced by the author in a third-year-level course in number theory at California State Polytechnic University, Pomona. As expected, the students quickly became adept at the calculator implementation of the algorithm.

References

1. G. H. Hardy and E. M. Wright, An Introduction to the Theory of Numbers, 4th ed., Clarendon Press, Oxford, 1960.

2. W. J. LeVeque, Elementary Theory of Numbers, Addison-Wesley, Reading, Mass., 1962.

3. I. Niven and H. Zuckerman, An Introduction to the Theory of Numbers, 4th ed. Wiley, New York, 1980.

CONSTRUCTING LOOPS FROM GROUPS

ARTHUR H. COPELAND, JR., AND ALBERT O. SHAR
Department of Mathematics & Computer Science, University of New Hampshire, Durham, NH 03824

In any course in group theory it is convenient to have examples of algebraic systems in which most, but not all, of the group postulates are satisfied. For certain kinds of groups $(G, \cdot)$ one can define a new operation "$*$" in terms of "$\cdot$" so that the resulting binary system $(G, *)$ retains all of the defining properties of a group except associativity. Although there are several ways of doing this, the authors have encountered an especially interesting one while studying H-spaces.

If $(G, \cdot)$ is a connected topological group with multiplication $\cdot$ then a new multiplication $*$ may