THE DIVISION ALGORITHM

MICHELLE HEWSON

Abstract. The Division Algorithm formalizes Euclidean division by providing a concrete method to calculate the quotient and remainder of two integers. The algorithm states that for any two integers a and b, there must exist unique quotient and remainder integer values. The applications derived from the algorithm itself and its proof are important in both theoretical and applied mathematics.

1. Introduction

Division is one of the basic mathematical operations that we use in everyday life. Countless tasks that are incorporated into our daily lives, including driving in a car or portioning out dinner, utilize division to some degree. Another familiar form of division is the division of integers. Examples like $15 \div 3$ and $72 \div 9$ can be intuitively calculated, but examples like $83 \div 20$ pose a slight problem. Since 83 can not be evenly divided by 20, in order to get an integer quotient, there must be an integer remainder. The Division Algorithm formalizes the cases where two integers, when divided by each other, have an integer remainder. Although the algorithm itself is a seemingly elementary concept, it is a key result in number theory and serves many purposes in several realms of mathematics. The goals of this paper include introducing the Division Algorithm and its functionalities, explaining its proof, and discussing the applications of the algorithm.

2. Background

In this section, the definitions and theorems that are relevant to the Division Algorithm, its proof, and its results will be introduced.

The following definitions and terms are for the conceptualization of the Division Algorithm and how it works mathematically. The *quotient* is the result of dividing one integer by another, given by the equation

$$\text{dividend} \div \text{divisor} = \text{quotient}.$$

The *remainder* is the 'left-over' integer after dividing one integer by another in order to produce an integer quotient. The remainder and quotient are expressed together by the equation

$$\text{dividend} \div \text{divisor} = \text{quotient} + \text{remainder}.$$

The *greatest common divisor*, or *GCD*, of two integers is the greatest positive integer that is a divisor of both integers. The GCD is denoted by $d = \gcd(a, b)$, where d is the GCD of a and b, which are two positive, non-zero integers. The greatest common divisor is a useful tool in the application of the Division Algorithm. Regarding the algorithm's proof, the utilization of the Well-Ordering Principle is essential in proving a key part.

Theorem 1 (Well-Ordering Principle)**.** *Every nonempty set of non-negative integers has a smallest element.*

Proof. Consider a set S that is a subset of the positive integers and has no smallest element. We want to show that S is equal to the empty set by using induction with some integer k. It follows that $1 \notin S$ because if so, $1 = \min(\mathbb{Z}^+) = \min(S)$, and that contradicts our assumption that S has no smallest element. Now consider $\{1, 2, \cdots, k\} \cap S = \emptyset$. We want to show that $k+1 \notin S$. So, if $k+1 \in S$, then $k+1 \neq \min(S)$ based on our initial assumption that S does not contain a smallest element. Thus, it must be the case that there exists an integer m such that $m \in S$ and $m < k+1$. However, this is a contradiction because there cannot be a smallest element in S. Thus, $k+1 \notin S$, therefore $S = \emptyset$. □

The Well-Ordering Principle is strictly applicable to sets that are nonempty and have positive elements. This is a crucial aspect in the upcoming proof of the Division Algorithm[3].

3. Theorem and Proof

Theorem 2 (The Division Algorithm)**.** *If a and b are integers, with $b > 0$, there exist unique integers q and r such that $a = qb + r \quad$ with $\quad 0 \leq r < b$.*

The proof of the Division Algorithm is a prime example of the existence-and-uniqueness proof style. This style is customary in number theory and abstract algebra. First, the integers r and q must be proven to exist. This is done utilizing the Well-Ordering Principle. Then, it must be proven that they are unique by introducing new integer values and proving they are equal. [1]

Proof. To prove the existence of q and r, consider the set $S = \{a - bx | x \in \mathbb{Z}, a - bx \geq 0\}$. This is the set of generalized remainders when dividing a by b. To prove that q (which can be viewed as x in the set's definition) exists and r exists within S, we must show that S is nonempty. First, assume S is not empty. Then consider case 1: $a \geq 0$. Then let $x = 0$. Then, plug x into $a - bx \Rightarrow a - b(0) \Rightarrow a \in S$. Thus, $a \in S$ so S is not empty. Now consider case 2: $a < 0$. Then, let $x = a$ and in similar fashion to case 1, plug x into $a - bx \Rightarrow a - b(a) \Rightarrow a(1-b) \in S$. It is clear that $a(1-b) \in S$ because we assumed $a < 0$ and $b > 0$, therefore $(1-b) \leq 0$. Thus, it must be $a(1-b) \geq 0$. In both cases, we have shown that S is nonempty.

By the Well-Ordering Principle, we know that S has a least element because it is both nonempty and only contains positive elements. So, let r be the least element in S. With algebraic manipulation, $r = \min(S)$ and $a = bq + r \Rightarrow r = a - bq$. We want to prove the bounds of r in the theorem are correct.

For the sake of contradiction, assume $r \geq b$. Then $r = a - bq \geq b$, and after subtracting b throughout, we get

$$\begin{aligned} r - b &= a - bq - b \\ r - b &= a - b(q-1) \end{aligned}$$

where $r - b = a - b(q-1) \geq b - b = 0$. So, $r - b \in S$ because $r - b \geq 0$. This contradicts our assumption that r is the least element in S because $r - b < r$ since $b > 0$. Thus, $r < b$. So, r and q both exist.

Now that we have proven that q and r exist, we must prove that q and r are unique integer values. Suppose there exist integers r, r', q, q' such that $a = bq + r$, $a = bq' + r'$, $0 \leq r < b$, and $0 \leq r' < b$. Then, since both equations are equal to a, we can set them equal to each other and get $bq + r = bq' + r'$. Without loss of generality, assume $r' \geq r$. A similar series of steps follows if $r \geq r'$. Then

$$\begin{aligned} bq + r &= bq' + r' \\ bq - bq' &= r' - r \\ b(q - q') &= r' - r. \end{aligned}$$

Notice that the left hand side of the equation is a multiple of b. Regarding the right hand side, remember that $0 \leq r' - r < b$. Since 0 is the only multiple of $b \in [0, b)$, it must be true that $b(q - q') = r' - r = 0$, so $r = r'$ and since $b > 0$, $q = q'$. Thus, q and r are unique integer values. □

4. Applications

Consider the example introduced earlier, $83 \div 20$. It is clear that $83 = 20(4) + 3$, where $q = 4$ and $r = 3$. This result is intuitive because 83 and 20 are relatively small values and this can be done without the help of mathematical software. The Division Algorithm is especially helpful when the two integers, a and b, are relatively large values. Consider $8523 \div 735$. In this example, $a = 8523$ and $b = 735$, and we will utilize the Division Algorithm to find q and r. One of the methods that can be used to visualize the Divison Algorithm is repeated subtraction, where b is repeatedly subtracted from a until b is greater than the difference. This method is shown below:

$$8523 - 735 = 7788 \tag{1}$$
$$7788 - 735 = 7053 \tag{2}$$
$$7053 - 735 = 6318 \tag{3}$$
$$6318 - 735 = 5583 \tag{4}$$
$$5583 - 735 = 4848 \tag{5}$$
$$4848 - 735 = 4113 \tag{6}$$
$$4113 - 735 = 3378 \tag{7}$$
$$3378 - 735 = 2643 \tag{8}$$
$$2643 - 735 = 1908 \tag{9}$$
$$1908 - 735 = 1173 \tag{10}$$
$$1173 - 735 = 438 \tag{11}$$

Notice that the difference at step 11, 438 is less than 735. This means that 8523 is divided by $735 * 11 + 438$, where 11 is the quotient and 438 is the remainder.

The Division Algorithm's applications diversely extend to both theoretical and applied mathematics. While it is an elementary number theory result, it is the basis of further results and concepts in the field. For example, the Euclidean Algorithm is an algorithm that allows one to find the greatest common divisor of two integers by repeatedly performing the Division Algorithm until the desired integer is reached[5]. The Euclidean Algorithm spells out the following steps to compute $d = \gcd(a, b)$, where a, b, and d are integers:

$$\begin{aligned}
a &= bq_1 + r_1 \\
b &= r_1q_2 + r_2 \\
r_1 &= r_2q_3 + r_3 \\
&\vdots \\
r_{n-2} &= r_{n-1}q_n + r_n \\
r_{n-1} &= r_nq_{n+1}
\end{aligned}$$

Once $r_n = d$, the greatest common divisor has been found. To find the greatest common divisor of the integers from our previous example, $a = 8523$ and $b = 735$, we will continuously perform the Division

Algorithm:

$$8523 = 735(11) + 438 \tag{12}$$

$$735 = 438(1) + 297 \tag{13}$$

$$438 = 297(1) + 141 \tag{14}$$

$$297 = 141(2) + 15 \tag{15}$$

$$141 = 15(9) + 6 \tag{16}$$

$$15 = 6(2) + 3 \tag{17}$$

$$6 = 3(2) + 0 \tag{18}$$

Step 18 is the last step because there is no remainder associated with $6 = 3(2)$. Notice in step 13 that $r_n = 3$, which is the remainder. The Euclidean Algorithm tells us that after repeating the Division Algorithm until $r_n = d = 3$, the greatest common divisor of 8523 and 735 is 3.

Another interesting application of the Division Algorithm is in the Extended Euclidean Algorithm. The Extended Euclidean Algorithm is an expansion of the previously discussed Euclidean Algorithm, and it allows for the representation of the greatest common divisor, d, as a linear combination of a and b. This means $d = ar + bs$ where a and b are the two integers derived from $\gcd(a, b) = d$ and r and s are two newly introduced integers. The Extended Euclidean Algorithm is truly an extension of the Euclidean Algorithm because it begins with the final result in the Euclidean Algorithm and continues from there. To illustrate, to find r and s, the Extended Euclidean Algorithm is as follows:

$$\begin{aligned}
d &= r_n \\
&= r_{n-2} - r_{n-1}q_n \\
&= r_{n-2} - q_n(r_{n-3} - -q_{n-1}r_{n-2} \\
&= -q_n r_{n-3} + (1 + q_n q_{n-1})r_{n-2} \\
&\vdots \\
&= ra + sb.
\end{aligned}$$

If we were to represent $d = 3$ from our previous example as a linear combination of $a = 8523$ and $b = 735$, we would have:

$$\begin{aligned}
d &= 3 \\
&= 15 - 6(2) \\
&= 15 - 2(141 - 9(15)) \\
&= -2(141) + (1 + 2(15))(141) \\
&\vdots \\
&= -99(8523) + 1148(735)
\end{aligned}$$

Thus, $r = -99$ and $s = 735$.

While both the Euclidean Algorithm and the Extended Euclidean Algorithm are examples of applications of the Division Algorithm in theoretical mathematics, together they have a pertinent role in applied mathematics and computer science. Together, these algorithms aid in creating and deciphering keys in RSA cryptography. RSA cryptography is the most popular public-key cryptography in today's society and assists in keeping online transactions, emails, and other personal information safe[2]. For RSA cryptography, the encoding key is public knowledge, but the decrypting key needs to be solved for. In order to obtain the private key required to decipher a message, one must utilize the Extended Euclidean Algorithm to find a number D such that $DE \equiv 1 \mod (m)$, where E and m are relatively prime integers related to the keys. Large numbers are utilized to make the possibility of finding D more difficult for attackers, thus the Extended Euclidean Algorithm is used. So, this additional application shows how substantial the role of the Division Algorithm is in both traditional mathematics and real-world mathematics.

5. Comments

Overall, the Division Algorithm, although simple, serves several purposes in the realms of both theoretical and applied mathematics. It is the basis of several number theory concepts like the Euclidean Algorithm, which is dated as one of the oldest known algorithms having been created around 300 BC. The Division Algorithm is also the basis of other key theorems, such as the Extended Euclidean Algorithm, whose uses extend to modern-day problems and topics such as cryptography and data encryption. So, even though division can be found in everyday tasks, the Division Algorithm helps prove why it is extremely important in all branches of mathematics.

References

1. T. Judson, "Abstract Algebra: Theory and Applications," 1^{st} edition, Virginia Commonwealth University Mathematics **7**, 2009.
2. E. Lehman, *Mathematics for computer science,* Technical report, Massachusetts Institute of Technology **1** (2006).
3. L. Ohman, *Are Induction and Well-Ordering Equivalent?,* The Mathematical intelligencer **41(3)** (2019).
4. F. and SF. Obermann, *An Analysis of Division Algorithms and Implementations,* echnical Report CSL-TR-95-675, Stanford University. **41** (1995).
5. J. Shallitt, "Origins of the analysis of the Euclidean algorithm," 4^{th} edition, Historia Mathematica, 1994.