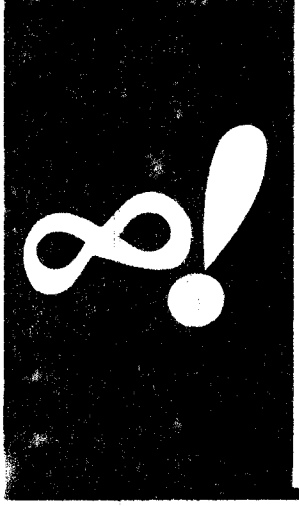

M·A·T·H·E·M·A·T·I·C·A·L R·E·C·R·E·A·T·I·O·N·S

Mathematical programming

Euclid's Algorithm

GCDs, LCMs, and repeating decimals

BY ROBERT T. KUROSAKA

DIVISIBILITY TESTS.

In my last installment in November, I examined fractions that form repeating decimals. This month, I am going the other way and will look at how to represent repeating decimals as fractions. I concluded the November column with a presentation of the algorithm for converting repeating decimals to fractions. This month's column will include a program to implement that algorithm, which is shown in figure 1.

The most irritating part of the algorithm is step 5, reducing the fraction to lowest terms. Just how do we do that? First, we must find a common divisor, an integer that divides into both the numerator and the denominator. (For this column, "divides into" also implies "without remainder.") Although 24/30 can be reduced to 12/15 by dividing the numerator and denominator by 2, we must reduce again to 4/5 by dividing through by 3. However, I'm sure we all saw that the greatest common divisor (GCD) was 6 and reduced the fraction in one step. As the numerator and denominator get larger, it becomes more difficult to determine what divides into them.

There are a surprisingly large number of folkways for finding divisors of large numbers. Everyone knows that a number is divisible by 2 if its last (rightmost) digit is divisible by 2. A number is divisible by 4 if the last two digits are divisible by 4. For example, 7536 is divisible by 4 since 36 is divisible by 4. This basic pattern can be extended to higher powers of 2: A number is divisible by 8 if its last three digits form a number divisible by 8, and so on.

Similarly, we all know that a number is divisible by 5 if it ends in 0 or 5. A number is divisible by 25 if its last two digits are divisible by 25, by 125 if the final three-digit number is divisible by 125, etc.

As I mentioned in the last column, 2 and 5 are special cases in base 10, so it's not surprising that we can't generalize this to numbers other than powers of 2 and 5. However, there are some methods for other numbers. I'll briefly run down the list of techniques for other numbers up to 9.

A number is divisible by 3 if the sum of its digits is divisible by 3. For example, 312,798 is divisible by 3 since 3+1+2+7+9+8=30. Further, it is divisible by 6 because any even number that is divisible by 3 is divisible by 6. A number is divisible by 9 if the sum of its digits is divisible by 9. Thus, 312,798 is not divisible by 9, but 312,795 is (3+1+2+7+9+5=27).

The test for divisibility by 7 is rather amusing: "Detach" the last digit and double it, then subtract the result from the rest of the number. If the answer is divisible by 7, the original number is divisible by 7. For example (not that this method needs any clarification), to test 378, we detach the 8 and double it, then subtract 16 from 37. Since the answer, 21, is divisible by 7, so is 378. Try a larger number, 33,929. Detach the 9, double it, and subtract from 3392, giving 3374. Now, is 3374 divisible by 7? You don't know? Apply the test to 3374. (Why is no one laughing?)

Before giving up on these strange tests, let's look at 11. The divisibility test for 11 is a bit complicated but rather impressive. Add every other (alternate) digit in the number; add the remaining digits; if the difference of the two sums is divisible by 11, the number is divisible by 11. In 9,370,845, the first sum is 9+7+8+5=29, and the second sum is 3+0+4=7. Since their difference, 22, is divisible by 11, the entire number is divisible by 11. Note also that the sum of all the digits is 36, which is divisible by 9, and that the number ends in 5. Hence, with simple eyeballing, you can be the hit of the party by announcing that 9,370,845 is divisible by 495 (11×9×5).

Similar to our last example, you can determine that a number is divisible by 100 by applying our rules for divisibility by 4 and 25. (What? You have an easier way?) Well, perhaps you're beginning to feel that we need a more general method for finding common divisors of fractions. Our first im-

(continued)

Robert T. Kurosaka teaches mathematics in the Massachusetts State College system. He invites your correspondence c/o BYTE, POB 372, Hancock, NH 03449.

1. Let x equal the decimal:	$x = 0.7363636\ldots$
2. Multiply the equation by 10^ (# of digits in the cycle)	$100\,x = 73.6363636\ldots$
3. Subtract 1. from 2.	$99\,x = 72.9$
4. Solve for x	$x = 72.9/99 = 729/990$
5. Reduce the fraction	$x = 81/110$

Figure 1: *The steps in converting repeating decimals to fractions.*

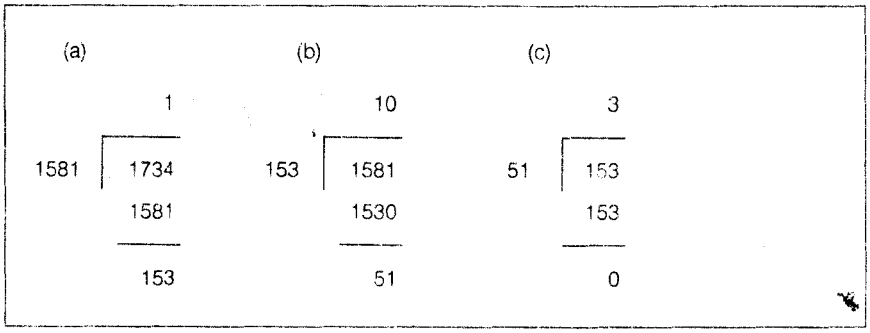

Figure 2: *Step-by-step application of Euclid's algorithm to the problem of finding the greatest common divisor of* 1734 *and* 1581.

pulse is to use prime factorization. This is certainly a correct approach, but most prime-factorization subroutines are cumbersome or time-consuming. I will show you a delightfully direct method of finding the GCD of two numbers that is easily programmed and requires no guessing, no trial and error, and no prime factorization. This remarkable method is called Euclid's algorithm.

Let us reduce the fraction 1581/1734. The steps are

1. Divide the larger number by the smaller. In figure 2a, 1734/1581 = 1 with a remainder of 153.

2. If the remainder is not 0, divide the divisor by the remainder. In figure 2b, 1581/153 = 10 with a remainder of 51.

3. Repeat step 2 until a 0 remainder occurs. In figure 2c, 153/51 = 3 with a remainder of 0.

The last divisor is the GCD of the two numbers (most texts call it the last nonzero remainder). In figure 2c, the last divisor is 51. Therefore, the GCD of 1581 and 1734 is 51. And, sure enough, the fraction reduces to 31/34. An obvious but necessary remark: If the GCD is 1, the two numbers are relatively prime; the fraction is already expressed in lowest terms.

Listing 1 offers a program that reduces a fraction to lowest terms by Euclid's algorithm. [*Editor's note: The Microsoft* BASIC *listings in this column are available for downloading from* BYTE*net Listings at* (617) 861-9764.]

An annoyance related to reducing fractions arises when adding or subtracting fractions with unequal denominators. We need to find the least common denominator (LCD). Before finishing our cyclic decimal-to-reduced-fraction routine, let's take a side trip to shine some light on LCDs.

In 1/9+5/12, we see that the LCD is 36. But precisely what are we seeing?

(continued)

Listing 1: A BASIC *program using Euclid's algorithm to reduce a fraction to its lowest terms. The routine begins at line* 310 *so that, when listing* 3 *is merged with it, the unnecessary lines* (310 *to* 410) *are overwritten.*

```
310 '*************************************************************************
320 '*              EUCLID'S ALGORITHM FOR GREATEST COMMON DIVISORS           *
330 '*                          BY ROBERT T. KUROSAKA                        *
340 '*************************************************************************
350 CLS
360 PRINT "This program calculates the greatest common divisor"
370 PRINT "of a positive fraction"
380 PRINT "and reduces the fraction to lowest terms."
390 PRINT :PRINT
400 INPUT "ENTER THE FRACTION'S NUMERATOR";NUM:NUM = ABS(NUM)
410 INPUT "ENTER THE FRACTION'S DENOMINATOR";DEN:DEN = ABS(DEN)
420 DIVISOR = NUM:DIVIDEND = DEN 'SAVE ORIGINAL VALUES FOR LATER DISPLAY
430 REM IF EITHER TERM IS NOT A WHOLE NUMBER, CLEAR THE DECIMAL.
440 IF DIVISOR< >INT(DIVISOR) OR DIVIDEND< >INT(DIVIDEND) THEN DIVISOR = DIVISOR*10:
    DIVIDEND = DIVIDEND*10:NUM = DIVISOR:DEN = DIVIDEND:GOTO 440
450 IF DIVISOR>DIVIDEND THEN SWAP DIVISOR, DIVIDEND
460 WHILE DIVISOR>0
470       QUOTIENT = INT(DIVIDEND/DIVISOR)
480       REMAINDER = DIVIDEND - DIVISOR*QUOTIENT
490       DIVIDEND = DIVISOR:DIVISOR = REMAINDER
500 WEND
510 PRINT :PRINT
520 PRINT "THE FRACTION ";NUM;"/";DEN;" HAS A GCD OF ";DIVIDEND
530 IF DIVIDEND = 1 THEN PRINT "THE FRACTION IS ALREADY IN LOWEST TERMS.":GOTO
    560
540 PRINT "THE REDUCED FRACTION IS: ";NUM/DIVIDEND;" /";DEN/DIVIDEND;
550 IF DEN/DIVIDEND = 1 THEN PRINT " = ";NUM/DIVIDEND
560 END
```

The LCD is the smallest integer that is evenly divisible by both denominators. To put it another way, the LCD is the least common multiple (LCM) of the denominators. While any common multiple of 9 and 12 will suffice for adding the fractions (e.g., their product, 108), we prefer the least value because it will simplify reducing the fraction later.

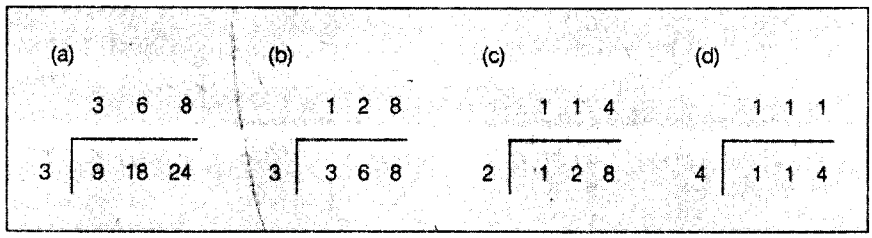

Figure 3: *Finding the least common multiple of three numbers using triple division.*

With larger denominators, the task of finding the LCD becomes increasingly difficult. In 5/12 and 3/14, it is not easily seen that the LCD is 84. Many methods have been devised for finding the LCD, most of which require prime factorization. One rather mystical method works for any number of denominators.

Suppose we want the LCM of 9, 18, and 24. Find a common divisor for all three numbers, if possible. If not, find a common divisor for any two of them, if possible. (If not, the numbers are relatively prime; the LCM is merely the product of the three numbers.)

Listing 2: A BASIC *program to find the least common multiple of a set of numbers using Euclid's algorithm.*

```
10 '*************************************************************************
20 '*                      LEAST COMMON MULTIPLE ALGORITHM                  *
30 '*                          BY ROBERT T. KUROSAKA                        *
40 '*************************************************************************
50 CLS
60 PRINT "This program calculates the least common multiple"
70 PRINT "of a set of positive integers."
80 PRINT
90 INPUT "HOW MANY INTEGERS ARE IN THE SET";TERMS:TERMS = INT(ABS(TERMS))
100 IF TERMS<2 THEN 400
110 REM NUMBER ARRAY HOLDS THE SET OF INTEGERS FOR WHICH THE LCM IS SOUGHT.
120 DIM NUMBER(TERMS)
130 PRINT :PRINT "ENTER THE INTEGERS ONE AT A TIME."
140 FOR I = 1 TO TERMS
150     INPUT NUMBER(I)
160     NUMBER(I) = INT(ABS(NUMBER(I)))
170     IF NUMBER(I) = 0 THEN PRINT "ILLEGAL ENTRY.":GOTO 150
180 NEXT I
190 REM BEGIN LCM PROCEDURE.
200 LCM = NUMBER(1)       'THE LCM OF A SINGLE NUMBER IS ITSELF.
210 FOR I = 2 TO TERMS
220     REM FIND GCD OF ACTIVE ENTRY AND WHAT PRECEDED IT (GCD WILL BE STORED
            IN 'DIVIDEND' BECAUSE LINE 290 ASSIGNS LAST VALID DIVISOR TO DIVIDEND).
230     DIVISOR = NUMBER(I):DIVIDEND = LCM
240     REM LINES 250-300 ARE THE SAME AS 450-500 OF THE GCD ROUTINE.
250     IF DIVISOR > DIVIDEND THEN SWAP DIVISOR,DIVIDEND
260     WHILE DIVISOR > 0
270             QUOTIENT = INT(DIVIDEND/DIVISOR)
280             REMAINDER = DIVIDEND - DIVISOR*QUOTIENT
290             DIVIDEND = DIVISOR:DIVISOR = REMAINDER
300     WEND
310     LCM = NUMBER(I)*LCM/DIVIDEND
320     REM THE LAST LCM WILL BE LCM OF ALL THE ENTRIES.
330 NEXT I
340 PRINT :PRINT
350 PRINT "THE LEAST COMMON MULTIPLE OF";
360 FOR I = 1 TO TERMS
370     PRINT NUMBER(I);
380 NEXT I
390 PRINT "IS";LCM
400 END
```

We will perform a triple division on these three numbers. In figure 3a, we divide the common divisor 3 into all three numbers. Note that we did not divide the first two numbers by 9 because our precedence of rules requires us to first look for divisors of all the numbers.

Next, we repeat the procedure for the three quotients: 3, 6, and 8. The first two are divisible by 3 again (figure 3b). Important: If the divisor does not divide into a particular number, merely *copy* the number. In figure 3b, the 8 is brought up. This procedure is repeated until all the quotients are 1s. In figure 3c, the 2 and 8 are divided by 2, with the 1 being brought up. Finally, in figure 3d, we divide by 4 and obtain all 1s in the quotients. The LCM is the product of all the divisors used. That is, the LCM of 9, 18, and 24 is $3\times3\times2\times4=72$.

You might enjoy trying to write a program to implement this method. There is, however, yet another method for finding the LCM, and, not surprisingly, it employs Euclid's algorithm.

The LCM of two numbers a and b is the product of the numbers divided by their GCD. That is, $LCM(a,b)=a\times b/GCD(a,b)$. This becomes apparent if we look at a simple example: 10 and 14. Their product, 140, is a multiple, but it isn't the smallest one. Since $10=2\times5$ and $14=2\times7$, their LCM needs only $2\times5\times7$, while their product is $2\times5\times2\times7$. Dividing by their GCD of 2 eliminates the overlap. In the language of elementary set theory, the LCM is the union of the two sets of factors. The formula above instructs us to "add" the two sets together and then "subtract" their intersection.

Since we already have the GCD program, we are only one step away from finding the LCM of two numbers. That's the good news. The bad news is that this method works only with two numbers at a time. To find the LCM of three or more numbers (say, 8, 10, and 14), we first find the LCM of 8 and 10 (40) and then find the LCM of 40 and 14 (280). This is no problem for a computer, but you may feel that it is less efficient than our triple-division approach. Anyway, listing 2 presents my version of the LCM routine using Euclid's algorithm.

I use strings, of course, because you can't directly enter a repeating decimal into the computer.

Finally, we are ready to return to our initial problem of converting a repeating decimal to a fraction. Listing 3 shows my routine for doing steps 1 through 4 of figure 1. The program is mostly a lot of string-handling. I use strings, of course, because you can't directly enter a repeating decimal into the computer. So I use a "_" to signify where the cyclic part begins and enter the number through one iteration of the cycle. This is analogous to the way of writing repeating decimals like 0.333 . . . as $0.\overline{3}$. Not only are you unable to enter repeating decimals into the computer, the computer is unable to hold any infinite series. However, the point of step 3 is to get rid of the cyclic part. All of the action in the method happens in the nonrepeating part and the first iteration of the cycle. So that is all that we use in our program.

The only really interesting part of this routine is in line 420. After I find the value of the unreduced fraction's numerator and denominator, I convert the values into strings and then back to numbers. Why?

When I first tried writing the routine in listing 3, I used the number $0.7\overline{36}$ as one of my test cases. The routine displayed the numerator value as 72.9 and the denominator value as 99. However, when I merged the routine with listing 1, I got strange results. It seems that Microsoft BASIC's guard digits were nonzero, so when the GCD routine tried to clear the decimal from the 72.9, the fraction became 7.29×

(continued)

Listing 3: A BASIC *program to convert a cyclic decimal number into an unreduced fraction. You can merge this program with listing 1 to make a complete implementation of the algorithm in figure 1.*

```
10 '*****************************************************************************
20 '*              REPEATING DECIMAL TO FRACTION CONVERTING ROUTINE              *
30 '*                          BY ROBERT T. KUROSAKA                            *
40 '*****************************************************************************
50 CLS
60 PRINT "This routine can be used with the greatest common denominator"
70 PRINT "program. Load the GCD program, then MERGE this routine into it."
80 PRINT "The MERGEd program is designed to determine the reduced fractional"
90 PRINT "representation of a repeating decimal.":PRINT
100 PRINT "To ENTER a repeating decimal:";
    PRINT " Type the nonrepeating part, then a '__' before the cycle."
110 PRINT "For example, 1.2__345 is the proper entry for 1.2345345345..."
120 PRINT "The decimal should always precede the '__', i.e., .333.... is entered";
    PRINT "as '.__3'. Reversing the '.' and '__' will cause an error.":PRINT
130 INPUT "ENTER REPEATING DECIMAL";NUMBER$
140 REM NONREPEATING PART OF NUMBER IS THAT PART UP TO "__". VAL OPERATOR
        IGNORES ALL NUMBERS AFTER A NONNUMERICAL CHARACTER. THUS, IN
        1.2__345,VAL("1.2__345") WILL BE 1.2, ETC.
150 NONREPEATING.PART = ABS(VAL(NUMBER$))
160 REM DEFINE A MORE READABLE FUNCTION TO USE FOR THROWING THE LEFTMOST
        CHARACTER OF A STRING AWAY.
170 DEF FNDROP.LEFT$(A$) = RIGHT$(A$,LEN(A$) - 1)
180 REM FIND DECIMAL POINT
190 WHILE LEFT$(NUMBER$,1) <> "."
200       NUMBER$ = FNDROP.LEFT$(NUMBER$)
210 WEND
220 NUMBER$ = FNDROP.LEFT$(NUMBER$)
230 REM FIND OUT HOW MANY DECIMAL PLACES THE REPEATING CYCLE IS OFFSET FROM
        THE DECIMAL POINT.
240 WHILE LEFT$(NUMBER$,1) <> "__"
250       OFFSET = OFFSET + 1
260       NUMBER$ = FNDROP.LEFT$(NUMBER$)
270 WEND
280 REM THROW AWAY REPEATING PORTION MARKER, "__"
290 NUMBER$ = FNDROP.LEFT$(NUMBER$)
300 REM HOW MANY DECIMAL PLACES ARE IN THE CYCLE? SINCE THE REPEATING CYCLE
        IS EVALUATED AFTER THROWING AWAY THE DECIMAL POINT, MULTIPLY BY
        10^ - (TOTAL NUMBER OF PLACES TO THE RIGHT IT SHOULD BE SHIFTED).
310 CYCLE.LENGTH = LEN(NUMBER$)
320 REPEATING.CYCLE = VAL(NUMBER$)*10^ - (OFFSET + CYCLE.LENGTH)
330 REM NUMBER = NONREPEATING PART + REPEATING CYCLE. SINCE THE FIRST
        ITERATION OF THE CYCLE IS THE ONLY ONE THAT DOES NOT CANCEL ON
        SUBTRACTION, ONLY USE IT.
340 NUMBER = NONREPEATING.PART + REPEATING.CYCLE
350 REM "CLEARED.NUMBER IS THE VALUE OF THE SUBTRACTION THAT DOES AWAY WITH
        THE INFINITE CYCLE (STEP 3 IN THE BYTE ARTICLE ALGORITHM).
360 CLEARED.NUMBER = NUMBER*10^CYCLE.LENGTH - NONREPEATING.PART
370 REM NOW, ASSIGN THE VALUES OF THE NUMERATOR AND DENOMINATOR TO THE
        VARIABLE NAMES USED IN THE GCD ROUTINE.
380 NUM = CLEARED.NUMBER:DEN = 10^CYCLE.LENGTH - 1
390 REM I CONVERT NUM AND DEN TO STRINGS AND THEN BACK TO CLEAR THE GUARD
        DIGITS IN THE NUM AND DEN VARIABLES. SEE BYTE ARTICLE FOR DETAILS.
400 NUM$ = STR$(NUM):DEN$ = STR$(DEN):NUM = VAL(NUM$):DEN = VAL(DEN$)
410 PRINT "THE EQUIVALENT UNREDUCED FRACTION IS:";NUM;"/";DEN
```

10^7 before the decimal-clearing routine gave up in disgust. By converting the values calculated in listing 3 to string values, I clear the guard-digit garbage out of the numerical representation. When I then reconvert the values into numbers, the GCD routine functions properly.

As always, I welcome your comments, criticisms, and suggestions. I've been getting some interesting mail on previous columns and will devote some space in an upcoming column to some of the more interesting insights. ■