An Instructive Algorithm Involving a Number Theoretic Problem

by *William J. Bruce*
University of Alberta
Edmonton, Alberta

The following article is reprinted with permission from the Ontario Mathematics Gazette, *Volume 17, Number 2, December 1978.*

This activity-oriented problem is rich in mathematical experiences that are adaptable to students from elementary through high school. The emphasis is on simple arithmetical operations, flow charts, and graphical representation. Another such algorithm has been considered previously by the author.[1]

Consider any natural number N > 1. If N is divisible by 3, divide it by 3 and repeat the division by 3 as often as possible. When N is not divisible by 3 and is an odd number, subtract 1 from N and try division by 3 again. Repeat this division if possible. Should N be even and not divisible by 3, multiply N by 2 and add 1. Again try division by 3 and repeat if possible. Division by 3 is attempted at every step. Whenever N is even and division by 3 fails, multiply N by 2 and add 1, but if N is odd and division by 3 fails, subtract 1. The procedure continues until N is reduced to 1.

For N = 14 we tabulate as follows:

Step Number	Procedure	n
0		14
1	M2 + 1	29
2	S1	28
3	M2 + 1	57
4	D3	19
5	S1	18
6	D3	6
7	D3	2
8	M2 + 1	5
9	S1	4
10	M2 + 1	9
11	D3	3
12	D3	1

[1]William J. Bruce, "Crazy Roller Coasters." *The Mathematics Teacher,* Vol. 71, No. 1, January 1978, pp.45-49.

In this table, M2+1 means multiply by 2 and add 1, whereas, S1 means subtract 1, while D3 means divide by 3. The natural number n is the number obtained in each step when one of these operations has been performed.

For n = 57 the tabulation becomes:

Step Number	Procedure	n
0		57
1	D3	19
2	S1	18
3	D3	6
4	D3	2
5	M2 + 1	5
6	S1	4
7	M2 + 1	9
8	D3	3
9	D3	1

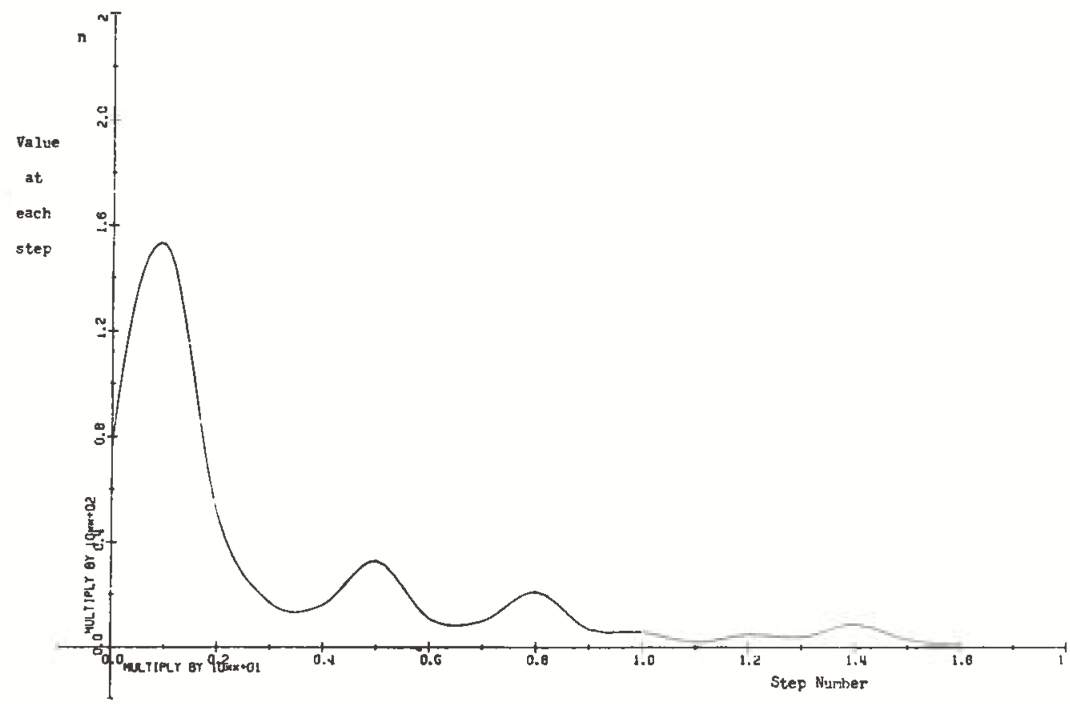
It will be shown later in this paper that for all natural numbers $N > 1$ the final number in the chain is always 1. The following table gives the number of steps required to complete the chains for natural numbers $2 \leq N \leq 101$. Clearly the number of steps do not tend to increase very rapidly. In this table, the largest number of steps is 24 and this occurs for N = 80 and N = 86. Even for numbers up to 500 the greatest number of steps is 31 and this occurs for N = 494. Incidentally, N = 194 requires 30 steps.

N	Number of Steps	N	Number of Steps	N	Number of Steps	N	Number of Steps
2	5	27	3	52	13	77	17
3	1	28	10	53	14	78	16
4	3	29	11	54	8	79	17
5	4	30	10	55	9	80	24
6	6	31	11	56	21	81	4
7	7	32	18	57	9	82	11
8	14	33	11	58	8	83	12
9	2	34	10	59	9	84	11
10	9	35	11	60	8	85	12
11	10	36	5	61	9	86	24
12	4	37	6	62	16	87	12
13	5	38	18	63	9	88	11
14	12	39	6	64	16	89	12
15	5	40	5	65	17	90	11
16	12	41	6	66	8	91	12
17	13	42	13	67	9	92	11
18	7	43	14	68	16	93	12
19	8	44	13	69	9	94	11
20	7	45	6	70	16	95	12
21	8	46	13	71	17	96	19
22	7	47	14	72	16	97	20
23	8	48	13	73	17	98	19
24	15	49	14	74	16	99	12
25	16	50	13	75	17	100	11
26	15	51	14	76	16	101	12

New chains can be generated by reference to previously determined chains without doing every step because, eventually, a number less than N is reached. An interesting phenomenon frequently occurs for certain groups of adjacent numbers. For example, for 87 ≤ N ≤ 95 the numbers of steps are, respectively, 12, 11, 12, 11, 12, 11, 12, 11, 12. For 334 ≤ N ≤ 350 the numbers of steps are 20, 21, 20, ..., 21, 20. Also, by arguing backward, it is easily shown that, whenever three or more steps are needed, the last three values of n are always 9, 3, 1.

Graphical experience can be provided along with each chain completed by having the values at each step plotted against the step numbers to obtain a set of points. If desired, these points can be connected by line segments or by a smooth curve. The result, in the latter case, tends to remind one of a roller coaster run. Some of these are quite interesting and can, again, be associated with the term crazy roller coasters.[2] Since every graph will be different, pupils can produce their personalized roller coaster run. Figure 1 illustrates the plot for the chain for which N = 76. A cubic curve has been fitted to the set of points to produce a smooth path. This chain requires only 16 steps so is very quickly completed. Some attention to scale units will be necessary in that smaller scale units on the vertical scales than on the horizontal scales often will be advisable. In this example, the largest value of n is 153 and this occurs in step 1.

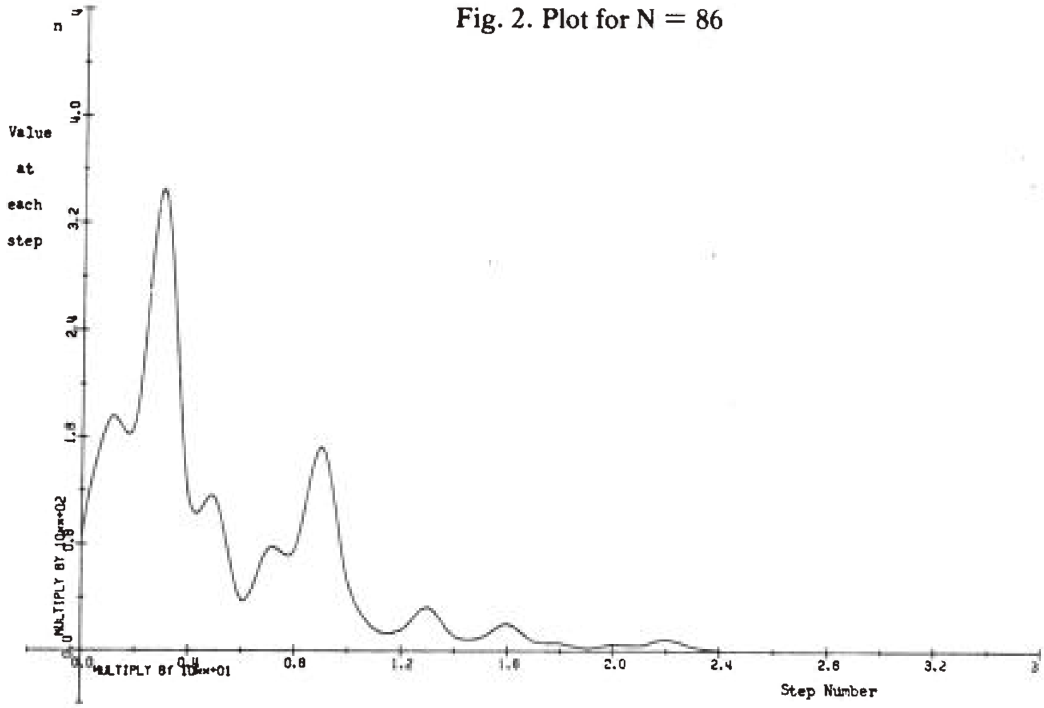
Fig. 1. Plot for N = 76



Consider now the chain associated with N = 86. Only 24 steps are needed to find the values of n. The plot, in this case, with a smoothly fitted cubic curve is shown in Figure 2. The graph reaches its highest point quickly in step 3 and attains a value n = 345 at this time, after which it drops quickly for awhile, but in step 9 it peaks again at n = 153 before dropping quickly again, and then slowly the rest of the way.

[2]Ibid.

Fig. 2. Plot for N = 86



The chain for N = 494, which has only 31 steps, builds up very quickly until it peaks in step 3 at a value of n = 1977. Its next greatest value is n = 1317 at step 6. Certainly this run should provide quite a thrilling ride. Power assist will be necessary only to step 3 and from there on free descent will be rapid and exciting. Note, in Figure 3, that the vertical scale probably should be given so as to require 100 as a multiplier. Such an interesting example could be used as a pin-up model.

Fig. 3. Plot for N = 494

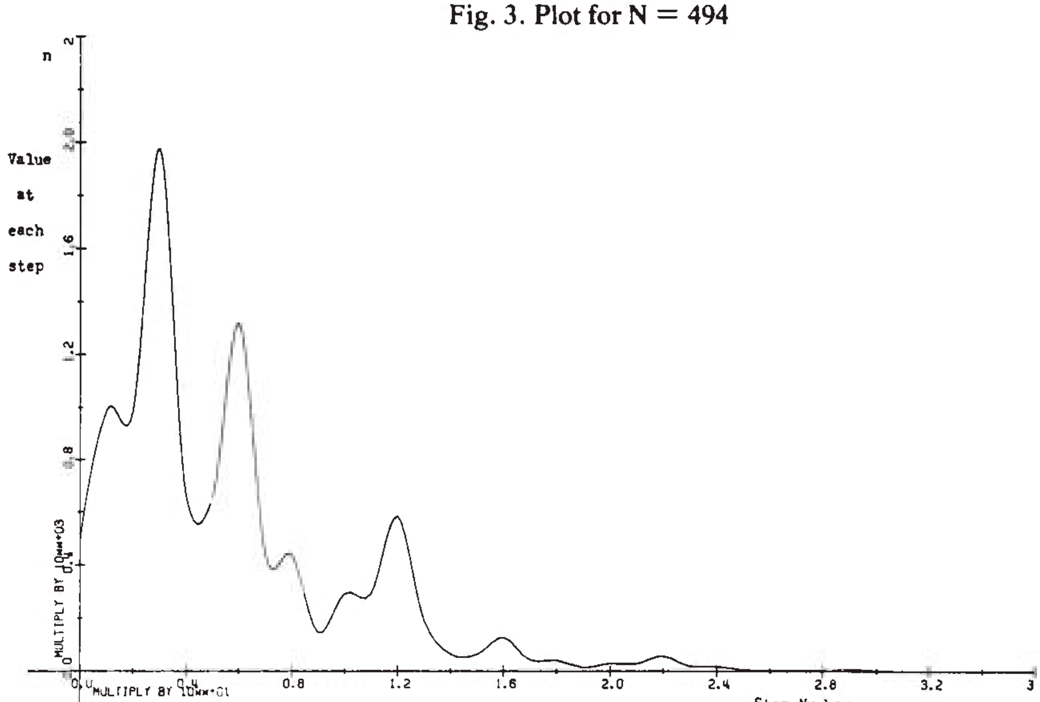


Fig. 3. Plot for N = 494

Since the algorithm used to generate the chains is very simple, the concept of a flow chart can be introduced again, as recommended previously.[3] In this case, the flow chart is only slightly more difficult to construct than before. Certainly, one based entirely on the computational procedure, as outlined in the first paragraph, should be attempted at first. Then one based on the language of the Fortran program could be tried. Figure 4 is one suggested format for the latter.

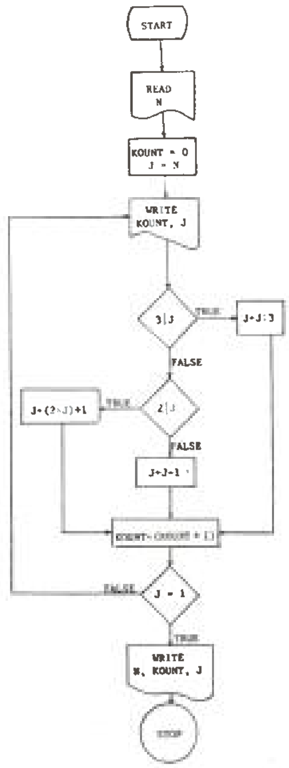


Fig. 4. Flow chart for a chain with initial N

```
#
#
#list c
      100 FORMAT(I4)
          READ(5,100)N
     C
          J=N
     C    THE NUMBER TO BE TESTED IS N.
     C    IT WILL TAKE ON VALUES NAMED J
     C    AS IT IS REDUCED TO 1.
     C
          KOUNT=0
     C    WE WILL COUNT THE NUMBER OF STEPS
     C    IT TAKES TO REDUCE J TO 1.
     C
       66 WRITE(6,200)KOUNT,J
      200 FORMAT(1X,2I5)
     C
          IF(J.EQ.3*(J/3))GO TO 3
     C    WE TEST TO SEE IF J IS
     C    EVENLY DIVISIBLE BY 3.
     C
          IF(J.EQ.2*(J/2))GO TO 4
     C    WE TEST TO SEE IF J IS EVEN.
     C
          J=J-1
          GO TO 5
     C
        3 J=J/3
          GO TO 5
     C
        4 J=1+2*J
     C
        5 KOUNT=KOUNT+1
     C    WE INCREMENT THE COUNTER
     C    EVERY TIME AN OPERATION
     C    IS PERFORMED ON J.
     C
          IF(J.EQ.1) GO TO 16
          GO TO 66
     C
     16   WRITE(6,200) KOUNT,J
          WRITE(6,300) N,KOUNT
      300 FORMAT(1X,/,'THE NUMBER',I4,
         *' WAS REDUCED TO 1 IN',
         *I4,' STEPS.',//)
     C
          STOP
          END
#END OF FILE
#
```

Fig. 5. Annotated Fortran program.

For those who want to generate chains for selected large values of N, Figure 5 provides an annotated Fortran program to facilitate application of the algorithm.

That the algorithm is always true for any natural number $N > 1$ can be shown easily by mathematical induction.[4] It is clearly true for $N = 2$. We assume the algorithm true for $N \leq m - 1$ and show that it is then true for $N = m$. It suffices to consider m even, since if m were odd, and whether m is divisible by 3 or not, the algorithm will reduce it to a number less than m in one step.

Since m is either divisible by 3 or off by 1 or 2, three cases need by considered, namely, (i) $m = 3p$, (ii) $m = 3p + 1$, and (iii) $m = 3p + 2$. In case (iii) there are three sub-cases since one must consider whether p itself is divisible by 3 or is off by 1 or 2. If we let $p = 3k$, $p = 3k + 1$, and $p = 3k + 2$, we get the three sub-cases (a) $m = 9k + 2$, (b) $m = 9k + 5$, and (c) $m = 9k + 8$. Apply the algorithm to all of the cases as follows:

(i)

Procedure	n
	3p
D3	p < m.

(ii)

Procedure	n
	3p + 1
M2 + 1	6p + 3
D3	2p + 1 < m.

(iii)

---

[3]Ibid.

[4]In consultation with A. Meir, Professor of Mathematics, University of Alberta.

(a)		(b)		(c)	
Procedure	n	Procedure	n	Procedure	n
	9k + 2		9k + 5		9k + 8
M2 + 1	18k + 5	M2 + 1	18k + 11	M2 + 1	18k + 17
S1	18k + 4	S1	18k + 10	S1	18k + 16
M2 + 1	36k + 9	M2 + 1	36k + 21	M2 + 1	36k + 33
D3	12k + 3	D3	12k + 7	D3	12k + 11
D3	4k + 1 < m.	S1	12k + 6	S1	12k + 10
		D3	4k + 2 < m.	M2 + 1	24k + 21
				D3	8k + 7 < m.

The requirements of induction have been satisfied. Hence the algorithm is true for any natural number $N > 1$.

Another proof that requires little more than a knowledge of elementary arithmetic is possible. This proof considers all of the natural numbers as being written in base 3 or in the equivalent expanded forms. For example,

91 (base 10) $= 3^4 + 3^2 + 1 =$ 10101 (base 3)

19 (base 10) $= 2(3^2) + 1 =$ 201 (base 3)

16 (base 10) $= 3^2 + 2(3) + 1 =$ 121 (base 3).

In base 3 the last two digits of any natural number $N > 1$ can be none other than one of the following:

00	10	20
01	11	21
02	12	22

We call the starting number N and the number at each step is called n. For numbers that end in 00, 10, and 20 division by 3 is immediate and we get $n < N$. The other endings need to be examined further. Since every odd number can be converted to an even number by subtraction of 1, as provided for in the algorithm, we need consider only even numbers. Thus the remaining six endings are taken as even at the outset and these give rise to six cases. The algorithm is applied in the following table to these six cases. Note that cases 5 and 6 in step 4 lead to three different routes in each case. Eventually, in all of these cases, the starting number N is reduced so that $n < N$. Thus we have shown that reduction occurs in all possible cases and since all possible endings have been included, the algorithm eventually will reduce all natural numbers $N > 1$ to unity.

(1)			(2)		
Procedure	Endings for n base 3		Procedure	Endings for n base 3	
	01	Even		11	Even
M2 + 1	10	Odd	M2 + 1	00	Odd
D3	01, 11, 21	n < N	D9	00	n < N

(3)

Procedure	Endings for n base 3	
	21	Even
M2 + 1	20	Odd
D3	02, 12, 22	n < N

(4)

Procedure	Endings for n base 3	
	02	Even
M2 + 1	12	Odd
S1	11	Even
M2 + 1	00	Odd
D9	00	n < N

(5)

	12	Even
M2 + 1	02	Odd
S1	01	Even
M2 + 1	10	Odd
D3	(a) 01, (b) 11, (c) 21	

(a)

	01	Odd
S1	00	Even
D9	00	n < N

(b)

	11	Odd
S1	10	Even
D3	01, 11, 21	n < N

(c)

	21	Odd
S1	20	Even
D3	02, 12, 22	n < N

(6)

	22	Even
M2 + 1	22	Odd
S1	21	Even
M2 + 1	20	Odd
D3	(a) 02, (b) 12, (c) 22	

(a)

	02	Odd
S1	01	Even
M2 + 1	10	Odd
D3	01, 11, 21	n < N

(b)

	12	Odd
S1	11	Even
M2 + 1	00	Odd
D9	00	n < N

(c)

	22	Odd
S1	21	Even
M2 + 1	20	Odd
D3	02, 12, 22	n < N