ITERATING THE DIVISION ALGORITHM

MICHAEL E. MAYS
West Virginia University, Morgantown, WV 26506

(Submitted June 1985)

INTRODUCTION

The division algorithm guarantees that when an arbitrary integer b is divided by a positive integer a there is a unique quotient q and remainder r satisfying

$$0 \leqslant r < a$$

so that

$$b = qa + r.$$

We will assume that $0 < a \leqslant b$ in this paper.

Euclid's algorithm iterates this division as

$$\begin{aligned} b &= q_1 a + r_1, \quad 0 < r_1 < a \\ a &= q_2 r_1 + r_2, \quad 0 < r_2 < r_1 \\ r_1 &= q_3 r_2 + r_3, \quad 0 < r_3 < r_2 \\ &\vdots \\ r_{n-3} &= q_{n-1} r_{n-2} + r_{n-1}, \quad 0 < r_{n-1} < r_{n-2} \\ r_{n-2} &= q_n r_{n-1} + 0. \end{aligned}$$

Euclid's algorithm terminates when $r_n = 0$. What makes the algorithm useful is that r_{n-1} is then the greatest common divisor of a and b. The worst case, in the sense that the algorithm takes the longest possible number of iterations to terminate, is when the sequence

$$a > r_1 > r_2 > \cdots > r_n = 0$$

decreases to 0 as slowly as possible. The smallest pairs (b,a) for which this happens are found by choosing each quotient q_i to be 1 except the last one, where $r_{n-2} = 2$ and $r_{n-1} = 1$ forces $q_n = 2$. This makes $r_{n-3} = r_{n-2} + r_{n-1}$, $r_{n-4} = r_{n-3} + r_{n-2}$, and so on, back until we have that a and b are consecutive Fibonacci numbers. Lamé first noticed the connection between Fibonacci numbers and Euclid's algorithm in 1844 (see [3]).

General results based on this insight include:

1. If $a < F_n$, then Euclid's algorithm terminates in at most $n - 2$ steps, and the smallest pair (b,a) taking exactly $n - 2$ steps is (F_n,F_{n-1}).

2. If (b_n,a_n) denotes the pair (b,a) with smallest b for which Euclid's algorithm first takes n steps to terminate, then

$$\lim_{n\to\infty} b_n/a_n = \lim_{n\to\infty} F_{n+2}/F_{n+1} = (1 + 5^{1/2})/2.$$

The intermediate steps in Euclid's algorithm can be unwound to find integers x and y satisfying

$$d = ax + by,$$

where d is the greatest common divisor of a and b. A short BASIC program for iterating the division algorithm is given in Figure 1.

```
60 PRINT "WHAT TWO NUMBERS TO START WITH";:INPUT B,A
70 Q=INT(B/A):R=B-Q*A
80 PRINT B,"=";Q,"*";A,"+";R
90 B=A:A=R
100 IF A=0 THEN GOTO 120
110 GOTO 70
120 PRINT "ALGORITHM TERMINATES."
```

Figure 1. A BASIC Program for Euclid's Algorithm

The algorithm for radix conversion can also be written as a succession of divisions. Starting with b positive and $a \geqslant 2$, we can write

$$\begin{aligned} b &= q_1 a + r_1, \quad 0 \leqslant r_1 < a \\ q_1 &= q_2 a + r_2, \quad 0 \leqslant r_2 < a \\ &\vdots \\ q_{n-2} &= q_{n-1} a + r_{n-1}, \quad 0 \leqslant r_{n-1} < a \\ q_{n-1} &= q_n a + r_n, \quad 0 \leqslant r_n < a. \end{aligned}$$

In the i^{th} step, $q_i = [b/a^i]$, so, using the natural stopping place $q_n = 0$, the algorithm takes n steps to complete, where $a^{n-1} \leqslant b < a^n$. The value of this algorithm is that successive substitution gives

$$\begin{aligned} b &= r_1 + aq_1 = r_1 + a(r_2 + aq_2) = \cdots \\ &= r_1 + a(r_2 + a(r_3 + a(\ldots(r_{n-1} + ar_n)\ldots))) \\ &= r_1 + ar_2 + a^2 r_3 + \cdots + a^{n-1} r_n, \end{aligned}$$

which says that the remainders can be interpreted as successive digits (from right to left) in the expansion of b using the base a.

The BASIC program used for Euclid's algorithm works here as well with only minor modifications. Line 90 becomes

```
90 B=Q
```

and the test for completion in line 100 uses B instead of A.

Whatever number is used for b, it is clear there is no value for a that can make the algorithm take longer to terminate than $a = 2$. With this choice for a, the first b that makes the algorithm terminate in exactly n steps is 2^{n-1}.

In this paper we investigate ways in which the four numbers b, a, q, and r of the division algorithm can be rearranged to give a terminating sequence of quotients q_i and remainders r_i when the division algorithm is iterated. The combinatorial and number theoretic properties of some of the sequences so generated are of interest.

ALTERNATE ALGORITHMS

Line 90 of the BASIC program in Figure 1 provides the pattern for iterating the divisions in Euclid's algorithm. The substitution made is that the old A becomes the new B, and the old R becomes the new A. In the radix conversion algorithm, the old A never changes, and the new B is the old Q. We classify

possible algorithms by analyzing possible replacement lines for line 90 in the BASIC program. Naively, there are sixteen possibilities, summarized in Figure 2, but ten of these are uninteresting in that their behavior is independent of the particular numbers a and b we start with. There is a single equation which repeats, a pair of equations which replace one another, or a sequence of equations that terminates to avoid a zero division. Of the six interesting cases, two are the radix conversion algorithm and Euclid's algorithm. The others are merely labelled in the table, and their analysis occupies the remainder of the paper.

A \ B	B	A	Q	R
B	b = q a + r b = 1 b + 0 repeats	b = q a + r a = 0 b + a b = q a + r cycles	b = q a + r q = 0 b + q 0 = 0 q + 0 terminates	b = q a + r r = 0 b + r r = 1 r + 0 0 = 0 r + 0 terminates
A	b = q a + r repeats	b = q a + r a = 1 a + 0 repeats	Radix Conversion	b = q a + r r = 0 a + r repeats
Q	b = q a + r b = a q + r cycles once r < min(a,q)	Algorithm 5	b = q a + r q = 1 q + 0 1 = 1 1 + 0 repeats	Algorithm 4
R	Algorithm 6	Euclid's Algorithm	Algorithm 3	b = q a + r r = 1 r + 0 repeats

Figure 2. Possibilities for Line 90

ALGORITHM 3

Iterate the division algorithm as

$$\begin{aligned} b &= q_1 a + r_1, \quad 0 < r_1 < a \\ q_1 &= q_2 r_1 + r_2, \quad 0 < r_2 < r_1 \\ q_2 &= q_3 r_2 + r_3, \quad 0 < r_3 < r_2 \\ &\vdots \\ q_{n-2} &= q_{n-1} r_{n-2} + r_{n-1}, \quad 0 < r_{n-1} < r_{n-2} \\ q_{n-1} &= q_n r_{n-1} + 0, \quad r_n = 0. \end{aligned}$$

Stretching the algorithm out as long as possible is accomplished by taking $r_{n-1} = 1$, $r_{n-2} = 2$, ..., $r_1 = n - 1$. Then the smallest possible choices for the q_i would be given by

$$
\begin{aligned}
q_{n-1} &= 0\ 1 + 0 = 0\\
q_{n-2} &= 0\ 2 + 1 = 1\\
q_{n-3} &= 1\ 3 + 2 = 5\\
q_{n-4} &= 5\ 4 + 3 = 23\\
&\vdots\\
q_{n-i} &= q_{n-i+1}i + i - 1\\
&\vdots
\end{aligned}
$$

This implies that $q_{n-i} = i! - 1$, and hence $a = n$ and $b = n! - 1$. Thus, we obtain

Theorem 1: If $b < n! - 1$, then Algorithm 3 terminates in $<n$ steps. Algorithm 3 terminates in exactly n steps when $b = n! - 1$ and $a = n$.

Back substituting in Algorithm 3 gives an interesting pattern for the r's in terms of the q's. We have

$$
\begin{aligned}
r_{n-1} &= q_{n-1}/q_n,\\
r_{n-2} &= (q_{n-2} - (q_{n-1}/q_n))/q_{n-1},\\
r_{n-3} &= (q_{n-3} - (q_{n-2} - (q_{n-1}/q_n)/q_{n-1})/q_{n-2},\\
&\vdots
\end{aligned}
$$

and so on back in an inverted continued fraction expansion, to

$$a = (b - (q_1 - (q_2 - (\cdots - (q_{n-2} - (q_{n-1}/q_n)/q_{n-1})/\cdots/q_2)/q_1.$$

As a one-line summary of Algorithm 3 more in the spirit of radix conversion, we have

$$
\begin{aligned}
b &= r_1 + aq_1 = r_1 + a(r_2 + r_1q_2) = \cdots\\
&= r_1 + a(r_2 + r_1(r_3 + r_2(\ldots(r_{n-2} + r_{n-3}(r_{n-2} + r_{n-1}q_n))\ldots))).
\end{aligned}
$$

In the worst case $b = n! - 1$, $a = n$ of Theorem 1, we generate here a representation in the factorial number system (see [2]).

ALGORITHM 4

Here the division algorithm is iterated as

$$
\begin{aligned}
b &= q_1a + r_1,\ 0 \leqslant r_1 < a\\
r_1 &= q_2q_1 + r_2,\ 0 \leqslant r_2 < q_1\\
r_2 &= q_3q_2 + r_3,\ 0 \leqslant r_3 < q_2\\
&\vdots\\
r_{n-2} &= q_{n-1}q_{n-2} + r_{n-1},\ 0 \leqslant r_{n-1} < q_{n-2}\\
r_{n-1} &= 0q_{n-1} + r_n,\ 0 \leqslant r_n < q_{n-1}.
\end{aligned}
$$

This time the algorithm terminates just before the first zero division, i.e., when $q_n = 0$. It could be considered the dual of Algorithm 3 in that the roles of the A and B assignments in line 90 of the BASIC program are reversed.

We build backwards to see what the smallest possible values are for b and a to give a certain number of steps before the algorithm terminates. It is clear that the sequence of r's is strictly decreasing until the next to last

term. If q_n is the first quotient that is 0, the smallest possible choice for q_{n-1} is 1. Since $r_n < q_{n-1}$, that forces $r_n = 0$. Then

$$r_{n-1} = q_n q_{n-1} + r_n = 0\,1 + 0 = 0,$$

and since $q_{n-2} > r_{n-1}$, $q_{n-2} = 1$ is the smallest possible choice. Then

$$r_{n-2} = q_{n-1} q_{n-2} + r_{n-1} = 1\,1 + 0 = 1,$$

and $q_{n-3} > r_{n-2}$ gives $q_{n-3} = 2$ as the smallest possible choice. We continue building the sequences of q's and r's backward from their n^{th} values by

$$r_{n-i} = q_{n-i+1} q_{n-i} + r_{n-i+1}$$

$$q_{n-i-1} = r_{n-i} + 1.$$

Writing $f(m) = r_{n-m}$, the sequence of r's is described by the recurrence

$$f(0) = f(1) = 0,$$

$$f(m) = (f(m-2) + 1)(f(m-1) + 1) + f(m-1) \text{ for } m > 1.$$

Writing $q_{n-m} = g(m) = f(m-1) + 1$, we obtain the neater recurrence

$$g(n+1) = g(n)(g(n-1) + 1).$$

This is summarized in

Theorem 2: Define $g(n)$ for $n \geqslant 0$ by

$$g(0) = 0,\ g(1) = 1,$$

$$g(n+1) = g(n)(g(n-1) + 1) \text{ for } n \geqslant 1.$$

Then the pair (b_n, a_n) for which Algorithm 4 first takes n steps to terminate is given by

$$b_n = g(n+2) - 1,\ a_n = g(n+1).$$

The sequence $b_1, b_2, b_3, \ldots$ begins

1, 3, 11, 59, 779, 47579, 37159979, ...

and the sequence $a_1, a_2, a_3, \ldots$ starts out

1, 2, 4, 12, 60, 780, 47580, ...

Neither of these sequences, nor any of their more obvious variants, seems to occur in Sloane's *Handbook* [5].

$\lim_{n\to\infty} b_n/a_n = \infty$ for Algorithm 4, but

$$\lim_{n\to\infty} \ln b_n/\ln a_n = (1 + 5^{1/2})/2.$$

This can be seen by noting that

$$\lim_{n\to\infty} \ln b_n/\ln a_n = \lim_{n\to\infty} \ln(b_n + 1)/\ln a_n = \lim_{n\to\infty} \ln g(n+2)/\ln g(n+1)$$

$$= \lim_{n\to\infty} (\ln g(n+1) + \ln(g(n) + 1))/\ln g(n+1)$$

$$= 1 + 1/\lim_{n\to\infty} (\ln g(n+1)/\ln g(n)),$$

and this process can be iterated to produce as many convergents to the continued fraction for $(1 + 5^{1/2})/2$ as desired. The limit has to be well behaved by the inequality

$$2^{F_{n-1}} \leqslant g(n) \leqslant 2^{F_n - 1},$$

which is easy to establish for $n \geq 1$ by induction.

ALGORITHM 5

Iterate the division algorithm as

$$\begin{aligned} b &= q_1 a + r_1, \quad 0 \leq r_1 < a \\ a &= q_2 q_1 + r_2, \quad 0 \leq r_2 < q_1 \\ q_1 &= q_3 q_2 + r_3, \quad 0 \leq r_3 < q_2 \\ &\vdots \\ q_{n-3} &= q_{n-1} q_{n-2} + r_{n-1}, \quad 0 \leq r_{n-1} < q_{n-2} \\ q_{n-2} &= 0 q_{n-1} + r_n, \quad 0 \leq r_n < q_{n-1}. \end{aligned}$$

The iteration should end just before a zero division, i.e., when $q_n = 0$. $q_1, q_2, \ldots$ form a strictly decreasing sequence out to q_{n-2}, so the algorithm is guaranteed to terminate. Choosing r's and q's so as to build the longest possible algorithm for the smallest possible b and a, we find $q_n = 0$ and $q_{n-1} = 1$ forces $r_n = 0$, since $r_n < q_{n-1}$, and then $q_{n-2} = q_n q_{n-1} + r_n = 0$, which cannot happen. $q_n = 0$, $q_{n-1} = 2$, and $r_n = 1$ gives $q_{n-2} = 0\,2 + 1 = 1$. Now, $r_{n-1} = 0$ gives no trouble, and $q_{n-3} = q_{n-1} q_{n-2} + r_{n-3} = 2\,1 + 0 = 2$, and all the other r's $= 0$ give the q's satisfying the recurrence

$$q_{n-k} = q_{n-k+1} q_{n-k+2},$$

with $q_n = 0$, $q_{n-1} = 2$. Thus, we obtain, in general, that

$$q_{n-k} = 2^{F_{k-2}},$$

with the $(k-2)^{\text{th}}$ Fibonacci number in the exponent. This is summarized in

Theorem 3: Writing (b_n, a_n) as the pair for which Algorithm 5 first takes n iterations to finish, we have, for $n \geq 2$,

$$b_n = 2^{F_{n-1}} \quad \text{and} \quad a_n = 2^{F_{n-2}}.$$

Thus,

$$\lim_{n \to \infty} \ln b_n / \ln a_n = (1 + 5^{1/2})/2.$$

Successive substitution provides a one-line summary of Algorithm 5:

$$\begin{aligned} b &= q_1 a + r_1 = r_1 + q_1 (r_2 + q_2 q_1) = \cdots \\ &= r_1 + q_1 (r_2 + q_2 (r_3 + q_3 (\ldots (r_{n-1} + q_{n-1} r_n) \ldots))). \end{aligned}$$

Multiply this out to obtain the "mixed radix expansion" of b relative to the sequence of quotients $q_1, q_2, q_3, \ldots$

$$b = r_1 + r_2 (q_1) + r_3 (q_1 q_2) + \cdots + r_n (q_1 q_2 \cdots q_{n-1}).$$

The relationship between systems of numeration and the division algorithm is explored by Fraenkel (see [2]).

ALGORITHM 6

The last variation we consider is

$$\begin{aligned} b &= q_1 a + r_1, \quad 0 < r_1 < a \\ b &= q_2 r_1 + r_2, \quad 0 < r_2 < r_1 \end{aligned}$$

$$b = q_3 r_2 + r_3, \quad 0 < r_3 < r_2$$
$$\vdots$$
$$b = q_{n-1} r_{n-2} + r_{n-1}, \quad 0 < r_{n-1} < r_{n-2}$$
$$b = q_n r_{n-1} + 0, \quad r_n = 0.$$

If the sequence of r's is chosen to decrease as slowly as possible so that $r_n = 0$, $r_{n-1} = 1$, $r_{n-2} = 2$, ..., then b would satisfy the system of congruences

$$b = 1 \pmod 2$$
$$b = 2 \pmod 3$$
$$\vdots$$
$$b = n - 1 \pmod n.$$

The smallest such b is clearly l.c.m. $(2, 3, \ldots, n) - 1$, with $a = n$. For $n \geq 4$, however, there are smaller values of b that provide an algorithm terminating after n steps. Table 1 summarizes "worst case" behavior up to $n = 16$.

Table 1. b_n, a_n that First Make Algorithm 6 Run for n Steps

n	b_n	a_n	n	b_n	a_n
1	1	1	9	53	32
2	3	2	10	95	61
3	5	3	11	103	65
4	11	4	12	179	115
5	11	7	13	251	161
6	19	12	14	299	189
7	35	22	15	503	316
8	47	30	16	743	470

We bound the number of steps that Algorithm 6 can take in the next result.

Theorem 4: Given b, no value for a makes Algorithm 6 take more than $2b^{1/2} + 2$ iterations to terminate.

Proof: Given b, form the sequence $R_1, R_2, \ldots, R_b$ of remainders associated with dividing b by each of the numbers 1, 2, ..., b. Applying Algorithm 6 to a pair (b, a) is equivalent to picking out the increasing subsequence

$$0 = R_{n_1} < R_{n_2} < \cdots < R_{n_m} = R_a$$

satisfying

$$R_{n_{i+1}} = n_i.$$

The sequence $R_1, R_2, \ldots, R_b$ has its last $b - [b/2]$ elements decreasing by 1 (corresponding to quotients 1 in the divisions), preceded by $[b/2] - [b/3]$ elements decreasing by 2, preceded by $[b/3] - [b/4]$ elements decreasing by 3, and so on back. Most of the larger values for j have no elements between $[b/j]$ and $[b/j + 1]$. Choose $k = [b^{1/2}]$, and consider as a worst case that there could be an increasing subsequence with

$$R_{n_1} = 0,\ R_{n_2} = 1,\ \ldots,\ R_{n_{[b^{1/2}]+1}} = [b^{1/2}]$$

and working backward from the other end,

R_{n_m} one of the last $b - [b/2]$ elements

$R_{n_{m-1}}$ one of the next to last $[b/2] - [b/3]$ elements

$\vdots$

$R_{n_{m-[b^{1/2}]+1}}$ between $[b/[b^{1/2}]]$ and $[b/([b^{1/2}] + 1)]$.

This would yield an increasing subsequence of maximum length

$$[b^{1/2}] + 1 + [b/[b^{1/2}]] \leqslant 2b^{1/2} + 2.$$

One would expect that the longest sequences would be obtained from pairs (b, a) such that the sequence of quotients $q_1, q_2, q_3, \ldots$ grows as slowly as possible and the sequence of remainders $r_n, r_{n-1}, r_{n-2}, \ldots$ also stays as small as possible. Keeping the remainders small is achieved by choosing b to satisfy a number of low-order congruences. The quotients' size is controlled by the relative sizes of b and a.

Theorem 5: Let $\{(b_n, a_n)\}$ be any sequence of ordered pairs of integers with the property that for any positive integer m there exists an N such that, when Algorithm 6 is applied to (b_n, a_n) for $n > N$, $q_i = i$ for $i = 1, 2, \ldots, m$. Then

$$\lim_{n\to\infty} b_n/a_n = e/(e - 1).$$

Proof: A pair (b, a) with $q_1 = 1$ satisfies $b = 1a + r_1$, with $r_1 < a$. Hence,

$b < 2a$, so $b/a < 2$;

$q_2 = 2$ implies $b = 2r_1 + r_2 < 3r_1 = 3(b - a)$, so $b/a > 3/2$;

$q_3 = 3$ implies $b = 3r_2 + r_3 < 4r_2 = 4(2a - b)$, so $b/a < 8/5$;

$q_4 = 4$ implies $b < 5(4b - 6a)$, so $b/a > 30/19$;

$q_5 = 5$ implies $b < 6(24a - 15b)$, so $b/a < 144/91$.

Continue this procedure to build a sequence of fractions

$$\{f(n)/g(n)\} = 2/1,\ 3/2,\ 8/5,\ 30/19,\ 144/91,\ 840/531,\ 5760/3641,\ \ldots$$

satisfying

$$f(2)/g(2) < f(4)/g(4) < \cdots < b/a < \cdots < f(3)/g(3) < f(1)/g(1).$$

It is easy to establish that $f(n) = (n + 1)(n - 1)!$.

$g(n)$ arises as the sum of coefficients of b in the inequalities generated from the assumptions

$$q_{n+1} = n + 1 \text{ and } q_n = n.$$

This sequence of coefficients,

$$\{c_n\} = 1,\ 1,\ 4,\ 15,\ 76,\ 455,\ \ldots,$$

has arisen in the literature before in an analysis of the game of Mousetrap [6], and satisfies the recurrence

$$c_n = nc_{n-1} + (-1)^{n+1}.$$

The analogy with subfactorials is compelling. See the note by Rumney and Primrose [4] for an analysis of the sequence $\{u_n\}$, which satisfies

$$u_{n-1} = f(n) - g(n).$$

A combinatorial interpretation of this sequence in terms of consecutive ascending pairs of numbers in permutation is given in [1]. Properties of $\{u_n\}$ can be used to establish the recurrences

$$g(n) = ng(n - 1) + \sum_{i=2}^{n-1} (-1)^{i+1} g(n - i)$$

$$= (n - 1)g(n - 1) + (n - 2)g(n - 2)$$

and the formula

$$g(n) = (n + 1)(n - 1)!(1 - 1/2! + 1/3! - \cdots + (-1)^{n+1}/(n + 1)!).$$

Since the sum is a truncated series expansion for $1 - 1/e$, the theorem is established.

Examples of pairs (b, a) for which Algorithm 6 takes a relatively large number of iterations to terminate can be constructed by starting with two consecutive convergents a/b and c/d in the continued fraction expansion of

$$e/(e - 1) = [1, 1, 1, 2, 1, 1, 4, 1, 1, 6, \ldots]$$

and then choosing positive integers x and y so that the numerator of the intermediate fraction

$$(ax + cy)/(bx + dy)$$

satisfies a number of low-order congruences.

Algorithm 6 provides a weaker statement about divisibility than Euclid's Algorithm does. It is easy to show that, if Algorithm 6 ends at the n^{th} step with $b = q_n r_{n-1} + 0$, then $\gcd(b, a)$ divides r_{n-1}, which in turn divides b.

The k^{th} quotient q_k is given in terms of b, a, and earlier quotients by

$$q_k = [b/(b - q_{k-1}(b - q_{k-2}(b - \cdots q_2(b - q_1 a) \cdots)))].$$

$r_{n-1} = 1$ is a sufficient condition for $\gcd(b, a) = 1$. It is not necessary, because, for example, $b = 9999$ and $a = 343$ ends with $r_{n-1} = 9$.

The iterations in Algorithm 6 say that $b = r_{k+1} \pmod{r_k}$. Thus, we are led to the following number theory problem: Given n, for each decreasing sequence of positive integers

$$x_1, x_2, x_3, \ldots, x_n$$

find the smallest positive number b satisfying

$$b = x_2 \pmod{x_1}$$
$$b = x_3 \pmod{x_2}$$
$$\vdots$$
$$b = x_n \pmod{x_{n-1}},$$

if a solution exists. A solution is guaranteed to exist if, for example, the numbers $x_1, x_2, \ldots, x_{n-1}$ are pairwise relatively prime. If a solution does exist, it is unique $(\text{mod } \text{lcm}(x_1, x_2, \ldots, x_{n-1}))$. What is the smallest solution b among all possible decreasing sequences of n terms? It is the same b as first makes Algorithm 6 take exactly n steps to terminate.

REFERENCES

1. F. M. David, M. G. Kendall, & D. E. Barton. *Symmetric Functions and Allied Tables*. Cambridge: Cambridge University Press, 1966.
2. A. S. Fraenkel. "Systems of Numeration." *Amer. Math. Monthly* **92** (1985): 105-14.
3. G. Lamé. "Note sur la limite du nombre des divisions dans la recherche du plus grand commun diviseur entre deux nombres entiers." *Compte Rendu* **19** (Paris, 1844):867-69.
4. M. Rumney & E. J. F. Primrose. "A Sequence Connected with the Sub-Factorial Sequence." *Math. Gazette* **52** (1968):381-82.
5. N. J. A. Sloane. *A Handbook of Integer Sequences*. New York and London: Academic Press, 1973.
6. A. Steen. "Some Formulae Respecting the Game of Mousetrap." *Q. J. Pure and App. Math.* **15** (1878):230-41.

◆◇◆◇◆