Then

$$4(2 - F_n) = B(3A^2 + 5B^2) \equiv 4 \pmod 8,$$

because $2 \nmid F_n$ since $(n, 6) = 1$. This congruence has no solutions with $A \equiv B \pmod 2$.

Case 3: $2 + F_n\sqrt{5} = \varepsilon_0^{-1}\alpha^3$. Noting $\varepsilon_0^{-1} = (1/2)(1 - \sqrt{5})$, we argue as in Case 2, using instead

$$4(2 + F_n) = -B(3A^2 + 5B^2) \equiv 4 \pmod 8,$$

which has no solutions with $A \equiv B \pmod 2$.

Theorem 5: The set of Lucas numbers L_n with $n > 0$ of the form 2^aX^3 are $L_1 = 1$ and $L_3 = 4$.

Proof: Let $L_n = 2^aX^3$ with $n = 3^ck$ and $(k, 3) = 1$. By Lemma 2, $L_k = X^3$ so by Theorems 3 and 4, $k = 1$. If $c \geq 2$, then Lemma 2(ii) would show $L_9 = 76$ was of the form 2^aX^3, which is false.

Remark: The set of Lucas numbers of the form $2^a3^bX^3$ leads to consideration of the equation $X^3 = Y^2 + 18$. The only solutions to this equation are $(3, \pm 3)$, but the available proofs (see [1] and [3]) are complicated. General methods for solving the equation $X^3 = Y^2 + K$ for fixed K are given in [1], [4], and [5].

REFERENCES

1. F. B. Coghlan & N. M. Stephens. "The Diophantine Equation $x^3 - y^2 = k$." In *Computers and Number Theory*, ed. by A. O. L. Atkin & B. J. Birch. London: Academic Press, 1971, pp. 199-206.
2. J. H. E. Cohn. "On Square Fibonacci Numbers." *J. London Math. Soc.* 39 (1964):537-540.
3. R. Finkelstein & H. London. "On Mordell's Equation $y^2 - k = x^3$." MR 49, #4928, Bowling Green State University, 1973.
4. O. Hemer. "On the Diophantine Equation $y^2 - k = x^3$." Almquist & Wiksalls, Uppsala 1952, MR 14, 354; *Ark. Mat.* 3 (1954):67-77, MR 15, 776.
5. W. Ljunggren. "On the Diophantine Equation $y^2 - k = x^3$." *Acta Arithmetica* 8 (1963):451-463.
6. E. K. Selmer. *Tables for the Purely Cubic Field* $K(\sqrt[3]{m})$. Oslo: Avhandlinger utgitte av det Norski Videnskapss, 1955.
7. C. L. Siegel. "Zum Bewise des Starkschen Satzes." *Inventiones Math.* 5 (1968):180-191.
8. H. Wada. "A Table of Fundamental Units of Pure Cubic Fields." *Proceedings of the Japan Academy* 46 (1970):1135-1140.

*****

THE NUMBER OF STATES IN A CLASS OF SERIAL QUEUEING SYSTEMS

MICHAEL J. MAGAZINE*

University of Waterloo, Waterloo, Ontario, Canada

ABSTRACT

It is shown that the number of states in a class of serial production or service systems with N servers is the $(2N - 1)$st Fibonacci number. This has proved useful in designing efficient systems.

This research was supported by National Research Council Grant No. A4142.

In studying queueing systems in series, it is useful to know precisely the number of different states that might occur. In particular, in [1], this number is crucial in determining approximate solutions to the allocation of a fixed resource to the individual servers or for scheduling servers with variable serving times. For a particular class of these problems, this number possesses an interesting property.

The system can be described in general as follows:

> N (single-server) service facilities (usually corresponding to N work stations of a production line) are arranged in series. Customers completing service at station i proceed to station $i+1$ and commence service there if it is free, or join a queue if the server is busy. The limitation on space restricts the number who can wait before station i to be W_i. If service is completed at station i and the waiting space before station $i+1$ is full, then the customer completing service cannot advance and station i becomes "blocked." Any station that is idle is said to be "starved." Station 1 cannot be starved, as a customer is always ready for processing (raw materials) and station N can never be blocked. Customers are not permitted to renege (see Figure 1).

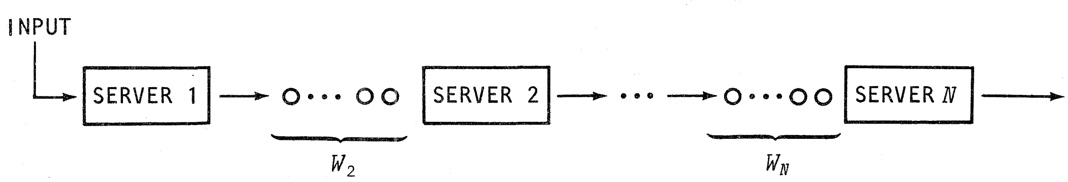

Fig. 1

The design problem is to consider how to divide the work among the N stations (or, equivalently, to determine the order of service) to maximize, among other objectives, the rate at which customers leave the system. The problem is complicated by having operation times that are not deterministic and are given only by a random variable. This optimization involves inverting a stochastic matrix whose dimension is the number of states in the system. Our problem here is to determine the number of possible states.

Without loss of generality, we can assume that $W_i = 0$, $i = 2, 3, \ldots, N$, that is, there is no waiting space before each server. This is done by assuming each waiting space is another service station with 0 service time. Hence, each station can be busy (state 1), all but station 1 can be starved (state 0), and all but station N can be blocked (state b). An N-tuple of 1's, 0's, and b's represents a state of the system. Obviously, not all combinations are allowed, for instance, a "b" must be followed by a "b" or a "1."

Theorem: Let S_N be the number of states when N servers are in series. Then $S_N = F_{2N-1}$.

Proof: When $N = 2$, the only possible states are (1, 1), (1, 0), (b, 1) and $S_2 = F_3 = 3$. Assume that $S_k = F_{2k-1}$. All possible states, when $N = k+1$, can be generated from the S_k states as follows: catenate a "1" to the right of each of the S_k states [corresponding to the ($k+1$)st server being busy]; catenate a "0" to the right of each of the S_k states; and, for each state with a "1" in the kth position, change this to a "b" and add a "1" in the ($k+1$)st position. The states with the "1" in the kth position had been similarly generated from the S_{k-1} states. This leads to the recursive relationship

$$S_{k+1} = S_k + S_k + (S_{k-1} + \cdots + S_1 + 1) = 2F_{2k-1} + \sum_{j=1}^{k-1} F_{2j-1} + F_0 = F_{2k+1}.$$

This result has been most useful in developing numerical procedures for calculating or approximating the probabilities that a server is busy, which is used in finding efficient designs for this class of production systems.

REFERENCE

1. M. J. Magazine & G. L. Silver. "Heuristics for Determining Output and Work Allocations in Series Flow Lines." *Int'l. J. Prod. Research* 16-3 (1978): 169-181.

*****

THE DETERMINATION OF ALL DECADIC KAPREKAR CONSTANTS

G. D. PRICHETT, A. L. LUDINGTON, and J. F. LAPENTA
Babson College, Wellesley, MA 02157

0. INTRODUCTION

Choose a to be any r-digit integer expressed in base 10 with not all digits equal. Let a' be the integer formed by arranging these digits in descending order, and let a'' be the integer formed by arranging these digits in ascending order. Define $T(a) = a' - a''$. When $r = 3$, repeated applications of T to any starting value a will always lead to 495, which is self-producing under T, that is, $T(495) = 495$. Any r-digit integer exhibiting the properties that 495 exhibits in the 3-digit case will be called a "Kaprekar constant." It is well known (see [2]) that 6174 is such a Kaprekar constant in the 4-digit case.

In this paper we concern ourselves only with self-producing integers. After developing some general results which hold for any base g, we then characterize all decadic self-producing integers. From this it follows that the only r-digit Kaprekar constants are those given above for $r = 3$ and 4.

1. THE DIGITS OF $T(a)$

Let $r = 2n + \delta$, where

$$\delta = \begin{cases} 1 & r \text{ odd} \\ 0 & r \text{ even.} \end{cases}$$

Let a be an r-digit g-adic integer of the form

$$a = \alpha_{r-1}g^{r-1} + \alpha_{r-2}g^{r-2} + \cdots + \alpha_1 g + \alpha_0 \tag{1.1}$$

with

$$g > \alpha_{r-1} \geq \alpha_{r-2} \geq \cdots \geq \alpha_1 \geq \alpha_0, \; \alpha_{r-1} > \alpha_0.$$

Let a' be the corresponding reflected integer

$$a' = \alpha_0 g^{r-1} + \alpha_1 g^{r-2} + \cdots + \alpha_{r-2}g + \alpha_{r-1}. \tag{1.2}$$

The operation $T(a) = a - a'$ will give rise to a new r-digit integer (permitting leading zeros) whose digits can be arranged in descending and ascending order as in (1.1) and (1.2). Define

$$d_{n-i+1} = \alpha_{r-i} - \alpha_{i-1}, \; i = 1, 2, \ldots, n. \tag{1.3}$$

Thus associated with the integer a given in (1.1) is the n-tuple of differences $D = (d_n, d_{n-1}, \ldots, d_1)$ with $g > d_n \geq d_{n-1} \geq \cdots \geq d_1$. Note that $T(a)$ depends entirely upon the values of these differences. The digits of $T(a)$ are given by the following, viz.,