De Bruijn Sequences-A Model Example of the Interaction of Discrete Mathematics and Computer Science

Author(s): Anthony Ralston

Source: *Mathematics Magazine*, Vol. 55, No. 3 (May, 1982), pp. 131-143
Published by: Taylor & Francis, Ltd. on behalf of the Mathematical Association of America
Stable URL: https://www.jstor.org/stable/2690079
Accessed: 29-04-2020 22:10 UTC

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at https://about.jstor.org/terms

Taylor & Francis, Ltd., Mathematical Association of America are collaborating with JSTOR to digitize, preserve and extend access to *Mathematics Magazine*

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

De Bruijn Sequences—A Model Example of the Interaction of Discrete Mathematics and Computer Science

Combinatorics, graph theory, and abstract algebra can all be applied to the same algorithmic problem.

ANTHONY RALSTON
SUNY at Buffalo
Amherst, NY 14226

1. An exemplary problem

The mathematical tools needed in computer science are overwhelmingly those of discrete mathematics—combinatorics, graph theory, algebra and the other branches of mathematics which focus on discrete objects rather than continuous functions. It is these branches of mathematics which are crucial to the design and analysis of so many algorithms and it is algorithms which are the lifeblood of computer science. The problem we will discuss aptly illustrates the interaction of discrete mathematics and computer science. The problem is as follows.

PROBLEM. *Given $m+1$ symbols (which, without loss of generality, we take to be 0, 1, 2,..., m) and a positive integer n, find an algorithm to generate a sequence of the symbols having minimum length, which when arranged on a circle, contains as subsequences of consecutive symbols, every sequence of length n of the symbols.*

Since each of the $m+1$ symbols may be repeated as often as desired in the sequences of length n, there are

$$(m+1)^n \tag{1}$$

possible subsequences. We shall denote the sequence to be generated by any algorithm as

$$S = s_1 s_2 \ldots s_L \qquad 0 \leqslant s_i \leqslant m. \tag{2}$$

(We shall use S in this paper as a general-purpose symbol to represent sequences of the s_i.) A subsequence of consecutive symbols of length n or, an **n-sequence**, is just a string of the form

$$s_i s_{i+1} \ldots s_{i+n-1}, \tag{3}$$

where we use the convention that the subscripts in (3) are to be taken modulo L. That is, we allow n-sequences of the form

$$s_{L-j} s_{L-j+1} \ldots s_L s_1 s_2 \ldots s_{n-j-1} \qquad j = 0, 1, \ldots, n-2 \tag{4}$$

which "wrap around" from the end of the sequence S to the beginning. Thus, any circular permutation of a sequence S which solves our problem is also a solution to the problem.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

As an example, for $m = n = 2$, the sequence

$$221201100 \tag{5}$$

contains all possible sequences of length 2 of the symbols 0, 1, 2 (i.e., 22, 21, 20, 12, 11, 10, 02, 01, 00). Note that in (5), the subsequence 02 has the form (4) with $j = 0$.

Each s_i, $i = 1, \ldots, L$ in S defines the beginning of an n-sequence because of our stipulation allowing wraparound. Therefore, the minimum possible value of L is given by (1) if S is to contain all possible n-sequences. A sequence S which solves our problem and for which L has the value (1) is called a **de Bruijn sequence.** (These have been called by various other names in the literature. Recently Fredricksen [**9**] has proposed that they be called *full cycles*.) The sequence (5) is such a sequence. As we shall see, de Bruijn sequences exist for all m and n.

In the next section we note some of the history and importance of the de Bruijn sequence problem, and then, in Section 3, discuss combinatorial, graph theoretic, and abstract algebraic approaches to the problem. In Section 4 we consider various algorithms for generating de Bruijn sequences. Finally, in Section 5 we shall discuss some implications of the growing importance of discrete mathematics for the mathematical education of computer science students and for the undergraduate mathematics curriculum.

2. History and applications

Like many problems which can be approached in a variety of different ways, the de Bruijn sequence problem has been "discovered" many times. For $m = 1$ the problem appears to have been first proposed in 1894 by A. de Rivière as just that—a problem—in the French problem journal, *l'Intermédiaire des Mathématiciens*. It was solved in that journal the same year by C. Flye Sainte-Marie [**7**] using a graphical method and three years later by W. Mantel [**18**] who, using an algebraic method, found a solution valid whenever $m + 1$ is prime. (See [**5**] and [**9**] for more of this history.) The earliest mention in the modern literature of de Bruijn sequences appears to have been in a paper by Martin [**19**] for whom the motivation was an unspecified "problem in dynamics." He approached the problem combinatorially and proved the existence of de Bruijn sequences for all m and n by exhibiting an algorithm to construct such sequences. A decade after Martin, de Bruijn [**4**] and Good [**13**] independently "discovered" and solved the problem for the case $m = 1$ in graph theoretic terms. Their results are easily extended to any m. In a commentary on Good's paper, Rees [**26**] approached the problem from the viewpoint of fields of prime characteristic and irreducible polynomials over such fields. Since the problem became generally known through de Bruijn's 1946 paper, it has since been associated with his name.

Rees' approach provides a general algorithm for the generation of de Bruijn sequences, but for $m > 1$ and, particularly, when $m + 1$ is not prime, it is a difficult algorithm to implement. For the case $m = 1$, however, this approach leads naturally to algorithms based on shift register techniques as described by Eldert et al. [**6**] (see also Golomb [**12**] and Fredricksen [**9**]). This approach is also discussed by Knuth [**15**] because of its relation to the problem of generating random sequences of binary digits.

The problem was discovered again, without any indicated motivation, by Roth [**27**] who used a combinatorial approach to develop an algorithm to generate de Bruijn sequences but his algorithm, like Martin's, requires $(m + 1)^n$ units of memory (one for each digit of the sequence). Algorithms without this restriction for the case $m = 1$ have been found by Fredricksen [**8**] and Fredricksen and Kessler [**10**] and a related algorithm for general m was derived by Fredricksen and Maiorana [**11**]. Another such algorithm was found by Ralston [**24**]. Fredricksen's paper [**9**] contains a general review of combinatorial algorithms for this problem.

While the graph theoretic approach leads to some elegant results about de Bruijn sequences, it does not lead to useful algorithms for the generation of these sequences. Good general discussions of de Bruijn sequences and their properties can be found in Hall [**14**] and van Lint [**16**].

As the preceding discussion implies, our problem is related to some applications of discrete mathematics, but the inherent mathematical interest of the problem rather than the applicability

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

of its solution has motivated most of the work on it. This certainly reflects our attitude here in presenting the problem as a paradigm of the application of discrete mathematics to algorithmics.

3. The existence of de Bruijn sequences

The combinatorial argument. The main idea of this approach is to generate a sequence one symbol at a time so that no n-sequence is ever repeated. A logical way to try to achieve this is to always add the *largest* symbol such that the resulting new n-sequence has not appeared previously. This is essentially the idea of Martin [**19**] and leads to the following algorithm:

Algorithm M

Step 1: Start with the sequence of n zeros.

Step 2: (Iterative Step) Append to the sequence S already generated the largest symbol possible such that the newly formed n-sequence (i.e., the last n symbols of S) has not already appeared.

Step 3: When Step 2 can no longer be repeated, remove the last $n-1$ symbols of the sequence generated by Step 2.

Steps 1 and 2 of Algorithm M ignore the wraparound sequences discussed in Section 1, so when Step 2 terminates, the result cannot be a de Bruijn sequence. However, we shall show that Step 3 is sufficient to then produce a de Bruijn sequence. We first prove:

LEMMA 1. *Let S, the sequence generated at any stage, be denoted by*

$$S = s_1 s_2 \ldots s_j.$$

Then, if at least one of the $n-1$ symbols $s_{j-n+2}, s_{j-n+3}, \ldots, s_{j-1}, s_j$ is not zero, Step 2 of Algorithm M may be applied to add s_{j+1} to S.

Proof. It is not possible to apply Step 2 only when the sequence of $n-1$ symbols

$$s_{j-n+2} s_{j-n+3} \cdots s_j \tag{6}$$

has appeared $m+1$ times previously in S, each time followed by a different one of our $m+1$ symbols. Thus, (6) would represent the $(m+2)$th appearance of this sequence. But one of the symbols in (6) is not zero, so this sequence cannot be the same as $s_1 s_2 \ldots s_{n-1}$ since, by Step 1, these are all zeros. Thus, the $m+2$ appearances of (6) must each have been *preceded* by a different one of our symbols. Since this is impossible, the lemma is proved.

Lemma 1 implies that Step 2 of Algorithm M can terminate when and only when the last $n-1$ symbols in S are all zero. And this happens only on the $(m+2)$th occurrence of such a sequence.

Let us denote by

$$s_1^{j_1} s_2^{j_2} \ldots s_i^{j_i}$$

the sequence of j_1 instances of s_1 followed by j_2 instances of s_2, etc. Since we have shown that the sequence S generated by Algorithm M (before Step 3) has $m+2$ occurrences of 0^{n-1} (the first of which has no predecessor), it follows that S contains all sequences of the form

$$s0^{n-1} \qquad s = 0, 1, 2, \ldots, m.$$

Since, according to Step 2, $s0^{n-2}$ is succeeded by 0 only when no larger digit can be appended, it follows that S also contains all sequences of the form

$$s0^{n-2}t \qquad s, t = 0, 1, 2, \ldots, m. \tag{7}$$

We can now prove:

THEOREM 1. *When Step 2 of Algorithm* M *terminates, each possible n-sequence appears once and only once in the sequence S generated.*

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

Proof. That no sequence can occur more than once follows immediately from Step 2 of the algorithm.

Now let

$$T = t_1 t_2 \ldots t_n \tag{8}$$

be an arbitrary sequence of n symbols such that

$$t_2 \ldots t_{n-1} \neq 0^{n-2}, \tag{9}$$

since we have shown in (7) that all sequences (8) with inequality replaced by equality in (9) do occur. To show that T appears in S, it is enough, by Step 2, to show that

$$U = t_1 t_2 \ldots t_{n-1} 0 \tag{10}$$

appears. Suppose U does not appear so that $t_2 \ldots t_{n-1}0$ occurs at most m times in S. But, therefore, again by Step 2,

$$t_2 \ldots t_{n-1} 0^2 \tag{11}$$

cannot be in S since the largest possible symbol is always chosen to follow $t_2 \ldots t_{n-1}0$.

Still under the assumption that U does not appear, it follows from (7) that, in $t_3 \ldots t_{n-1}$, there must be a nonzero symbol since otherwise (11) would be in S. Applying the same argument as before with (11) now playing the role of (10), it follows that $t_3 t_4 \ldots t_{n-1} 0^2$ appears, at most, m times in S and, therefore, $t_3 t_4 \ldots t_{n-1} 0^3$ cannot occur in S. Continuing in this way we get eventually that $t_{n-1} 0^{n-1}$ cannot appear, which contradicts the fact that all sequences of the form (7) do appear. This contradiction assures that U is in S and proves the theorem.

Finally, if we apply Step 3 of Algorithm M, the result is a de Bruijn sequence. This follows from Theorem 1 and the fact that deleting the final 0^{n-1} but allowing wraparound results in the original and new sequences having the same n-sequences. Therefore, we have proved the existence of de Bruijn sequences for all m and n.

For a given m and n, all de Bruijn sequences have the same length $L = (m+1)^n$, so that we can order this set of sequences lexicographically. (That is, if $S = s_1 s_2 \ldots s_L$ and $T = t_1 t_2 \ldots t_L$, we say $S \geqslant T$ if $s_i = t_i$, $0 \leqslant i \leqslant j < L$ and $s_{j+1} > t_{j+1}$; $S = T$ only if $s_i = t_i$, $0 \leqslant i \leqslant L$.)

We note that if the de Bruijn sequence generated by Algorithm M has its n initial zeros moved to the end, then the construction in Step 2 implies that the resulting sequence is greatest (with respect to the lexicographic ordering) among all de Bruijn sequences for a given m and n. We will later refer to this particular sequence as B_{mn}.

The graph theoretic argument. The idea behind a graphical approach to the PROBLEM (the approach used by de Bruijn [4]) is to define a digraph (directed graph) whose edges are each labeled with one of the $m+1$ symbols and which has a path such that the labels of successive

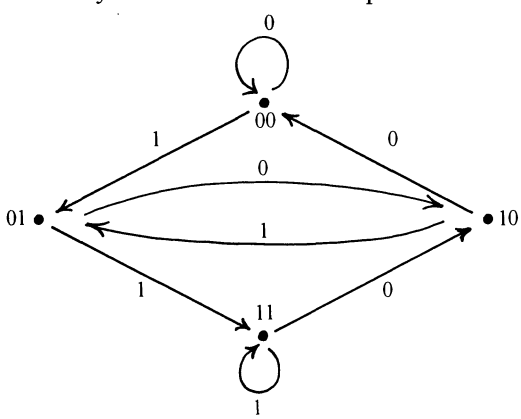

FIGURE 1. **A de Bruijn graph for $m = 1$, $n = 3$.**

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

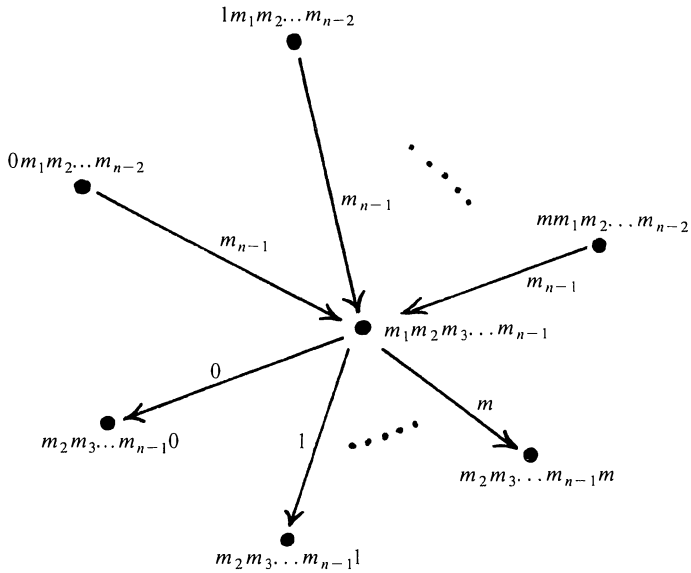

FIGURE 2. **Part of a digraph for generating de Bruijn sequences.**

edges in the path form the desired sequence. For $m=1$ and $n=3$ such a digraph is shown in FIGURE 1 with each node labeled by an $(n-1)$-sequence. Starting at the node labeled 00 and proceeding successively to the nodes labeled 01, 11, 11, 10, 01, 10, 00, 00, the corresponding edge labels form the sequence

11101000

which is a de Bruijn sequence.

Generalizing this idea, we can define a digraph G which will lead to a solution to our PROBLEM. Recall that the number of edges directed into (away from) a node is called the indegree (outdegree) of that node. G is defined by the following conditions:

(a) G has $(m+1)^{n-1}$ nodes each of which is labeled with a distinct $(n-1)$-sequence of our $m+1$ symbols.

(b) Each node $m_1m_2\ldots m_{n-1}$ has indegree and outdegree $m+1$ with the edges out to nodes $m_2m_3\ldots m_{n-1}k$, $k=0,1,\ldots,m$ labeled, respectively, $0,1,2,\ldots,m$ and with the edges in from nodes $km_1m_2\ldots m_{n-2}$, $k=0,1,\ldots,m$ all labeled m_{n-1}, the last symbol on the label of the node itself.

FIGURE 2 illustrates all of the edges which meet a single node of this digraph G.

An applicable, well-known theorem in graph theory (for instance, see [**20**]) is:

THEOREM 2. *A connected, directed graph has an Eulerian cycle* (*i.e., a path which traverses every edge once and only once and ends at the node at which it begins*) *if and only if each node has the same indegree and outdegree.*

Since our digraph G satisfies these conditions, it has an Eulerian cycle. Now suppose we have such an Eulerian cycle which starts at some node N. Suppose also that we construct a sequence S as follows:

Algorithm G

Step 1: Set S equal to the empty sequence.

Step 2: (Iterative Step) As each edge of the Eulerian path is traversed, append the label of the edge to S.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

Since the graph G has $(m+1)^{n-1}$ nodes and since each node has $m+1$ edges emanating from it, G has $(m+1)^n$ edges. Therefore, the sequence S generated by Algorithm G has $(m+1)^n$ symbols. Since, additionally, the $m+1$ edges leading from this node are each traversed once and only once in an Eulerian cycle, a unique n-sequence is formed each time a new edge is traversed. To see this note that the construction assures that, at any time, the last $n-1$ symbols added to S are the unique label of the node just arrived at (using the wraparound for the first $n-1$ nodes). Therefore, S is a de Bruijn sequence. The above is easily summarized by:

THEOREM 3. *Algorithm* G *generates a de Bruijn sequence for any m and n.*

If we choose our Eulerian cycle so that the initial node has the label 0^{n-1} and so that the first edge traversed is the loop at this node and the next edge chosen is always the one with the greatest label not yet traversed, then Algorithm G generates the same de Bruijn sequence (although a circularly permuted one) as Algorithm M. But what if we choose Eulerian cycles other than this one? For given m and n, how many distinct de Bruijn sequences are there up to a circular permutation? This combinatorial question is answered by the formula

$$(m+1)^{-n}[(m+1)!]^{(m+1)^{n-1}}. \tag{12}$$

For any n and $m=1$, there are $2^{2^{n-1}-n}$ distinct de Bruijn sequences, which, even for $n=5$, has a value of 2048. For $m=n=2$, there are 23 de Bruijn sequences in addition to (5); all 24 sequences are listed in TABLE 1.

221201100	221100201	220211001	220021101
221200110	221100120	220210011	220021011
221120100	221020011	220121100	220012110
221120010	221011200	220112100	220011210
221102001	221002011	220110021	220011021
221101200	221001120	220100211	220010211

TABLE 1. **The 24 de Bruijn sequences for $m=n=2$.**

The derivation of (12) is beyond us here. It involves the notion of *spanning trees* of a digraph, counting the number of spanning trees of graphs of the form G and then counting the number of Eulerian cycles of a graph as a function of the number of spanning trees. To do this is a major exercise in matrix algebra involving, in particular, the computation of minors and determinants of a matrix of order $(m+1)^{n-1}$. I might note that the understanding of this derivation is a good test of a student's knowledge of matrix algebra. Van Lint [**16**] gives a rather terse version of the derivation.

The finite field approach. The use of recurrence relations is a well-known way of generating sequences. If the magnitude of the terms generated by a recurrence relation is bounded, then the *period* of the sequence generated must be finite. To achieve periods of maximum length using recurrence relations is one of the aims of random number generators used on computers. From our definition, a de Bruijn sequence is the sequence of maximum length with the property that there is no repeated n-sequence. These considerations lead to a recurrence relation approach to our problem and, from it, to the theory of finite fields.

For this discussion, we *assume* $m+1=p$, *a prime*. The basic idea is, given $s_0 \neq 0$, to generate a sequence $\{s_i\}$ by using a recurrence relation

$$s_i = a_1 s_{i-1} + a_2 s_{i-2} + \cdots + a_n s_{i-n} \bmod p, \qquad i = 1,2,\ldots, \tag{13}$$

where $s_j = 0$ if $j<0$, the coefficients a_i satisfy $0 \leq a_i \leq m$ for all i and all arithmetic is performed modulo p.

How long is the period of the sequence generated by (13)? We prove first

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

LEMMA 2. *The first n-sequence* $s_j s_{j+1} s_{j+n-1}$ *to recur in the sequence* $\{s_i\}$ *generated by* (13) *is* $s_0 s_1 \ldots s_{n-1}$.

Proof. Suppose that

$$s_j s_{j+1} \ldots s_{j+n-1}, \qquad j \neq 0$$

recurs before $s_0 s_1 \ldots s_{n-1}$ so that the sequence $\{s_i\}$ has the form $s_0 s_1 \ldots s_j \ldots s_{j+n-1} \ldots s_j \ldots s_{j+n-1} \ldots$. But the digit preceding each s_j must be the same since arithmetic modulo a prime implies that (13) uniquely determines s_{j-1} from $s_j, \ldots, s_{j+n-1}$. This contradicts the assumption about $s_j s_{j+1} \ldots s_{j+n-1}$. Working back to s_0, the lemma follows.

The sequence S generated by (13), therefore, has the form

$$s_0 s_1 \ldots s_j s_0 s_1 \ldots s_j s_0 s_1 \ldots \tag{14}$$

with j the period of the "repeat." Now let

$$P(x) = 1 - a_1 x - a_2 x^2 - \cdots - a_n x^n \tag{15}$$

with the a_i's as in (13). Then, if we let $s_0 = 1$ and define

$$S(x) = 1 + \sum_{i=1}^{\infty} s_i x^i, \tag{16}$$

$S(x)$ is the formal power series expansion of $1/P(x)$ with arithmetic modulo p. This is easily shown by multiplying (15) and (16) and using (13). We may now prove

LEMMA 3. $P(x)$ *divides* $1 - x^k$ *if and only if* k *is a multiple of the period of the sequence* (14).

Proof.

$$\begin{aligned} \frac{(1 - x^k)}{P(x)} &= (1 - x^k) S(x) \\ &= s_0 + s_1 x + s_2 x^2 + \cdots + s_k x^k + s_{k+1} x^{k+1} + \cdots \\ &\quad - s_0 x^k - s_1 x^{k+1} - s_2 x^{k+2} + \cdots . \end{aligned} \tag{17}$$

But, if $P(x)$ divides $1 - x^k$, then the polynomial on the right-hand side of (22) must have degree $k - n$. Thus

$$s_i = s_{k+i} \qquad i = 0, 1, \ldots, .$$

Since the truth of the above relation implies, from (17), that $P(x)$ divides $1 - x^k$, this proves the lemma. Note also that

$$s_{k-n+1} = s_{k-n+2} \ldots = s_{k-1} = 0$$

in order for the polynomial to have the correct degree.

We now need the following theorem from the theory of finite fields (Albert [**2**], Berlekamp [**3**]):

THEOREM 4. *There exists an irreducible polynomial* $P(x)$ *of degree* n *with coefficients in a finite field of characteristic* p *which divides the polynomial* $1 - x^{p^n - 1}$ *and such that no polynomial of the form* $1 - x^j$ *having degree less than* $p^n - 1$ *is a multiple of* $P(x)$.

If we identify a polynomial $P(x)$ satisfying Theorem 4 with (15), then by Lemma 3, $p^n - 1$ must be the period of S. Of course, the period of S could not have been greater than $p^n - 1$, since there are only p^n possible n-sequences and the sequence 0^n cannot be in S (for otherwise, from (13), it would repeat immediately). Thus we have shown that the sequence generated by (13) contains all strings of length n of the digits $0, 1, \ldots, p - 1 = m$ except the string of n zeros. Thus the first period of the sequence generated by (13) may be converted to a de Bruijn sequence by inserting a zero at some place where there is a sequence of $n - 1$ zeros.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

We have shown that equation (13) is effectively a recurrence relation for generating de Bruijn sequences. In algorithmic form we have

Algorithm F (assumes $m+1=p$, a prime)
Step 1: Set $s_0=1$.
Step 2: (Iterative Step) Use (13) to generate s_i, $i=1,2,\ldots,p^n-2$, where the a_i are coefficients of a polynomial $P(x)$ satisfying Theorem 4.
Step 3: Insert a zero in the sequence next to a subsequence of $n-1$ consecutive zeros.

A point worth noting here is that, when $n=1$, the recurrence relation (13) is of the same form as the relation used for the multiplicative congruential generation of a sequence $\{r_i\}$ of random numbers. This relation is

$$r_i \equiv ar_{i-1} \bmod \alpha,$$

where α is often chosen to be 2^β and β is the (binary) word length of the computer on which the random numbers are to be generated.

To see an example of the generation of de Bruijn sequences by Algorithm F, let $m=1$ and $n=4$. In this case, the polynomial $P(x)=1-x-x^4$ is irreducible and divides $1-x^{15}$ but does not divide $1-x^k$ for any $k<15$. Now, using (13) with $s_0=1$, $a_1=1$, $a_2=0$, $a_3=0$, $a_4=1$, we obtain the sequence

$$1111010110010000 \tag{18}$$

with the final zero added to make a de Bruijn sequence. For the case $m=2$ and $n=2$, the polynomial $P(x)=1-x-x^2$ is irreducible and divides $1-x^8$ but does not divide $1-x^k$ for $k<8$. Applying Algorithm F, we obtain the de Bruijn sequence

$$112002210 \tag{19}$$

in which the second of the two consecutive 0's has been added (per Step 3).

Two questions remain:

1. How do we find the coefficients a_i which satisfy the conditions of Theorem 4?
2. How can we extend Algorithm F to the case when $m+1$ is not prime?

For an answer to the first question, see Knuth [**15**] or Alanen and Knuth [**1**]. For the second question, we need the following two lemmas due to Rees [**26**] which we present without proof.

LEMMA 4. *If p is prime and q is a power of p, we can construct a de Bruijn sequence with $m+1=p^q$ and any n as follows*:
(a) *Use* (13) *with $m+1=p$ and $n'=qn$ to generate a sequence $S=s_0s_1s_2\ldots$ and then*
(b) *Construct $T=t_0t_1t_2\ldots$ where*

$$t_i=s_{qi}+s_{qi+1}p+s_{qi+2}p^2+\cdots+s_{q(i+1)-1}p^{q-1} \tag{20}$$

so that $0\leq t_i\leq p^q-1$. Take the first $(p^q)^n-1$ symbols of T and insert a zero next to a subsequence of $n-1$ zeros.

If q is not a power of p, a modest modification of the procedure results in a de Bruijn sequence.

As an example, let $p=q=n=2$ so that $n'=4$ and $m=1$. We use for S two periods of (18) (less its final zero) to get

$$1111\ 0101\ 1001\ 0001\ 1110\ 1011\ 0010\ 00$$

and then use (20) with $t_i=s_{2i}+2s_{2i+1}$, to obtain

$$332212023113010 \tag{21}$$

which contains all sequences of length $n=2$ (except 00) of 0, 1, 2, and 3. To obtain a de Bruijn sequence we just insert a 0 at the end.

The next lemma indicates how to generate de Bruijn sequences for any n and $m+1=p_1p_2$ where p_1 and p_2 are primes or powers of primes.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

LEMMA 5. *Given positive integers n, p_1 and p_2, with p_1 and p_2 relatively prime, we can construct a de Bruijn sequence with $m+1=p_1p_2$ and n as follows.*

Let the de Bruijn sequences generated by Algorithm F (*or Lemma* 4), *corresponding to p_1 and p_2, respectively, be $S_1=s_{01}s_{11}s_{21}\ldots$ and $S_2=s_{02}s_{12}s_{22}\ldots$. Then the desired sequence is given by*

$$T=t_0t_1t_2\ldots$$

with

$$t_i=s_{i1}p_2+s_{i2}. \tag{22}$$

To generate a de Bruijn sequence for any m and n we need only find the prime factorization of $m+1$, use Lemma 4 to generate a de Bruijn sequence for each prime factor and n and then use Lemma 5 repeatedly.

As an example, let $m=11$ and $n=2$ so that $m+1=2^2\cdot 3$. For the 2^2 factor we first find the de Bruijn sequence with $m=1$, $n=4$ as given by (18) and then the sequence with $m=3$ $(=2^2-1)$ and $n=2$ as given by (21) with an additional zero appended. For the 3 factor, a de Bruijn sequence with $m=2$ $(=3-1)$ and $n=2$ is given by (19). Then, combining (19) and (21) and using A for 10 and B for 11, we apply (22) with $p_2=3$ to get the first digits of the 144 digit sequence for $m=11$ and $n=2$ as:

3322 1202 3113 0100 3322 1202 3113 0100
1120 0221 0112 0022 1011 2002 2101 1200
↓
*AA*86 3827 944*B* 0322 *A*977 5608 *B*43*A* 1500

4. Combinatorial algorithms for generating de Bruijn sequences

Although all three approaches discussed in Section 3 produce a de Bruijn sequence for a given m and n, Algorithm F has the disadvantage of requiring a calculation of the coefficients $a_1,\ldots,a_n$ of a polynomial $P(x)$ satisfying Theorem 4. (For m not prime, several of these calculations may be required.) The combinatorial approach, represented by Algorithm M, poses fewer computational difficulties, and has been the one most commonly used in practice. In this section we focus on other combinatorial algorithms which generate de Bruijn sequences.

We begin by noting that Algorithm M may be extended in a simple and elegant way to generate *all* de Bruijn sequences for any m and n. This extension involves one of the most powerful ideas in algorithmics, namely, **backtracking.**

Algorithm B

Step 1: Start with the de Bruijn sequence B_{mn} generated by Algorithm M with the initial n zeros moved to the end; $B_{mn}=s_1s_2\ldots s_L$ with $L=(m+1)^n$. Find the largest $j<L$ such that $s_j\neq 0$. Let $S=s_1s_2\ldots s_{j-1}$.

Step 2: Apply Step 2 of Algorithm M to S (except that at the first step look for the largest symbol *less than* s_j) until either

(a) a new de Bruijn sequence has been generated. In this case, return to Step 1 above with the new sequence playing the role of B_{mn}.

(b) Step 2 of Algorithm M fails with $S=s_1s_2\ldots s_k$, $k<L$ (i.e., no symbol s_{k+1} can be added such that the new n-sequence has not already appeared). In this case go to Step 3 below.

Step 3: (Backtrack) Choose the largest $j\leq k$ such that $s_j\neq 0$ and return to Step 2 above with $S=s_1s_2\ldots s_{j-1}$. The algorithm terminates when $j\leq n$.

The sequences in TABLE 1 for $m=n=2$ were all generated using this algorithm. We illustrate in TABLE 2 the generation of the second and third sequences.

Why does Algorithm B succeed in generating all de Bruijn sequences? The answer is that the backtracking idea effectively tests *all possible sequences* in decreasing lexicographic order and, therefore, must "catch" each de Bruijn sequence. Algorithm B is described in Fredricksen [**9**] who ascribes it to Alberts.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	j	
2	2	1	2	0	1	$\underline{1}$	0	0	7	[B_{22} from Algorithm M]
2	2	1	2	0	1	0	$\underline{2}$	x	8	[x represents failure]
2	2	1	2	0	$\underline{1}$	0	0	x	6	
2	2	1	2	0	0	$\underline{2}$	x		7	
2	2	1	2	0	0	1	$\underline{1}$	0	8	[Success]
2	2	1	2	0	0	$\underline{1}$	0	x	7	
2	2	1	$\underline{2}$	0	0	x			4	[Immediate failure at j]
2	2	1	1	2	0	$\underline{2}$	x		7	
2	2	1	1	2	0	$\underline{1}$	0	0	7	[Success]

TABLE 2

If we are only interested in generating a single de Bruijn sequence for given m and n, then Algorithm M poses a serious problem for all but small values of m and n. Unfortunately, it requires $(m+1)^n$ units of memory since, at every application of Step 2, we must be able to determine if the newly-formed n-sequence has appeared previously in the sequence. Algorithm M also requires an average of $(m+1)/2$ lookups in this table of $(m+1)^n$ entries every time a new symbol is added to the string.

It is most desirable to use a **memoryless algorithm,** that is, one which does not require the availability of the entire string already generated. We consider two such algorithms which are similar but have some distinctive differences. (For another such algorithm and a general discussion, see [**8**] and [**9**].) We begin with two definitions (see [**11**]).

DEFINITION. A **necklace** S of length n is an n-sequence with the property that $S \geqslant T$ for every n-sequence T which is a cyclic permutation of S. Thus, if $S = s_1 s_2 \ldots s_n$ is a necklace

$$S \geqslant s_i s_{i+1} \ldots s_n s_1 \ldots s_{i-1} \tag{23}$$

for $2 \leqslant i \leqslant n$.

An interesting question is: How many necklaces $Z(m, n)$ are there of length n containing the symbols $0, 1, 2, \ldots, m$? The answer, whose derivation is beyond us here (see Golomb [**12**]), is

$$Z(m, n) = \frac{1}{n} \sum_{d|n} \phi(d)(m+1)^{n/d} \tag{24}$$

where ϕ is Euler's totient function (i.e., $\phi(d)$ is the number of positive integers less than d which are relatively prime to d with $\phi(1) = 1$) and the summation is over all positive d which divide n. For example, when $m = 3$, $n = 6$, then d takes on the values 1, 2, 3, 6 and $\phi(1) = 1$, $\phi(2) = 1$, $\phi(3) = 2$, $\phi(6) = 2$ so that (24) becomes

$$Z(3,6) = \tfrac{1}{6}[4^6 + 4^3 + 2 \cdot 4^2 + 2 \cdot 4] = 700.$$

DEFINITION. Let $S = s_1 s_2 \ldots s_n$ and let $T = s_1 s_2 \ldots s_j$, $j < n$. Denote by T^k the subsequence of k consecutive repetitions of T. If $S = T^k$ when $k > 1$, we say that S is **periodic** with repetition (or periodicity) k and T is its **periodic reduction**. If there is no T with $k > 1$ for which $S = T^k$, we say that S is **aperiodic.**

Given positive integers m and n, the following recursive algorithm, due to Fredricksen and Maiorana [**11**], produces a de Bruijn sequence by generating successive necklaces of length n.

In the algorithm we denote by $B_{mn}^{(\mathrm{FM})}$ the de Bruijn sequence to be generated, and define $B_{0n}^{(\mathrm{FM})} = 0$.

Algorithm FM

Step 1: Start with the empty string.

Step 2: (Iterative Step) Generate the necklaces of length n whose first symbol is m, in decreasing lexicographic order. Append to the string already generated each necklace if it is aperiodic or its periodic reduction otherwise.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

Step 3: (Recursive Step) Append $B^{(FM)}_{m-1,n}$ to the string already generated.

Note that it is not necessary to state this algorithm in recursive form. Step 2 could have said just, "Generate the necklaces of length n in decreasing lexicographic order" and then Step 3 would have been unnecessary.

Clearly the crucial part of Algorithm FM is the generation of successive necklaces. Suppose $S = s_1 s_2 \ldots s_n$ is a necklace and we wish to generate its successor necklace (i.e., the next smallest necklace lexicographically). Algorithm N below generates the successor of S; in the formulation of Step 2, S_k denotes the string $s_1 s_2 \ldots s_k$.

Algorithm N

Step 1: Find j such that $s_j \neq 0$ but $s_{j+1} = s_{j+2} = \cdots = s_n = 0$.

Step 2: Generate $T = [S_{j-1}(s_j - 1)]^q S_{n-qj}$ where $0 \leqslant n - qj < j$. (That is, T consists of q repetitions of the first $j-1$ symbols of S and the jth symbol reduced by 1 followed by as many of the first $j-1$ symbols as needed to obtain n symbols.)

Step 3: Test whether T is a necklace [i.e., that it satisfies (23)]. If so, stop. If not, return to Step 1 with S replaced by T.

Fredricksen and Maiorana prove that this algorithm will produce the next necklace in at most $(1/2)(n+1)$ steps. For example with $n = 12$ and $m = 2$

$$S = 20002\ 00000\ 00$$

is a necklace. From this Algorithm N generates successively

20001	20001	20	$[j=5, \quad q=2]$
20001	20001	12	$[j=11, \quad q=1]$
20001	20001	11	$[j=12, \quad q=1]$

with the last string the necklace next smaller than S. Note that any time the last symbol is m, the result cannot be a necklace so that this may immediately be reduced to $m-1$.

Fredricksen and Maiorana also prove that Algorithm FM indeed generates a de Bruijn sequence. From a computational point of view, Algorithm FM has the modest drawback that every necklace must be generated and, thus, Algorithm N must be executed many times. (Each necklace must also be tested for periodicity but this is trivial since we must only see if j in Algorithm N divides n.) An algorithm which ameliorates this problem and which we shall call Algorithm R has been developed by Ralston [**24**]. Although in some ways similar to Fredricksen and Maiorana's algorithm, Algorithm R has the following interesting features:

1. It is truly recursive in the sense that $B^{(R)}_{mn}$, the de Bruijn sequence generated by Algorithm R for given m and n, consists of an initial segment followed by $B^{(R)}_{m-1,n}$.

2. The initial segment requires the generation only of those necklaces containing just m's and $(m-1)$'s. To each of these necklaces which is aperiodic is appended a sequence of n-sequences, each of which
   (a) contains m's only where the necklace contains m's and
   (b) is such that it has the greatest possible value less than the preceding n-sequence.

For example, when $m = 3$, $n = 5$, 33232 is a necklace and it is followed by

$$33231\ 33230\ 33132\ 33131\ 33130\ 33032\ 33031\ 33030.$$

3. For periodic necklaces of m's and $(m-1)$'s let
   t be the number of $(m-1)$'s in the periodic reduction,
   $u = m^t - 1$,
   k be the length of the periodic reduction.
   Then the periodic reduction of the necklace is followed by a sequence of k-sequences whose values are determined using $B^{(R)}_{u,n/k}$.

The details of how this is done are too complex to present here. But it is particularly worth

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

noting that, because of the use of $B^{(R)}_{u,n/k}$ in the generation of $B^{(R)}_{mn}$, Algorithm R contains that rather rare situation of a recursion within an already recursive algorithm.

The following instances of Algorithms FM and R for $n = 4$, $m = 2$ are taken, respectively, from Fredricksen and Maiorana [**11**] and Ralston [**24**].

$B^{(FM)}_{24} = 2$
2221 2220 2211 2210 2201 2200
21
2120 2111 2110 2101 2100
20
2011 2010 2001 2000
1
1110 1100
10
1000
0

$B^{(R)}_{24} = 2$
2221 2220
2211 2210 2201 2200
2121 2020
2111 2110 2101 2100 2011 2010 2001 2000
1
1110
1100
10
1000
0

For $n = 2$ and $n = 3$ the two algorithms give the same result. For $n \geq 4$, however, the results are always different. One noteworthy feature of Algorithm FM is that it generates exactly the same sequence as Algorithm M (after moving the n initial zeros to the end).

A few words are in order about the subject of the correctness of the algorithms we have discussed in this and the previous section. For those in Section 3 our discussion amounted to something close to "classical" (although very informal) mathematical proofs that they perform as claimed. For Algorithm B of this section we did no more than sketch a proof. And for Algorithms FM and R we referred only to the papers in which they were first presented and where standard mathematical proofs of their correctness are given. Of interest here is that a very active area of research in computer science concerns formal proofs of algorithms—and their implementations as computer programs—using the tools of mathematical logic. Although such proofs of even very simple algorithms tend to be quite difficult and involved, this is a most important and interesting area of research for both mathematicians and computer scientists. For a recent review of the status of research in this area see London [**17**].

5. Curriculum implications

The de Bruijn sequence problem is exceptional, yielding to successful solutions from several branches of discrete mathematics. Although few problems can be solved in such a variety of ways, the use of each of these branches—combinatorial analysis, graph theory, linear algebra, abstract algebra—is quite typical of the way these areas of mathematics impinge on computer science. The growing importance of discrete mathematics relative to more classical areas of mathematics and the symbiosis of discrete mathematics and computer science has clear and important educational implications for computer science and, although more speculative, there are also implications for the mathematics community.

It is not always fully recognized in the computer science community that mathematics should be an important component of any computer science curriculum. While there is room for

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms

argument about just how much of the calculus-linear algebra sequence should be taught to computer science majors, to this author it is clear that discrete mathematics should play *at least an equal role* to the classical subject matter in the first two years. In addition, a one- or two-year sequence balanced between discrete and continuous mathematics would be of more professional value to students in the social, management, and behavioral sciences than the calculus sequence many such students are now required to take.

In my opinion, computer science will provide the largest source of problems for mathematicians for years to come. Thus, it is reasonable to predict that mathematical research will become increasingly oriented toward discrete mathematics. This suggests that an undergraduate major in mathematics better balanced between discrete and continuous mathematics should be developed. Undoubtedly for most readers, the points above raise more questions than they answer. Some of the author's answers to these questions can be found in Ralston [**21, 22, 23, 25**].

(*Editor's note*. The cover of this *Magazine* shows five de Bruijn sequences arranged on concentric circles. Beginning with the innermost circle and moving outwards, the sequences are for the following pairs (m, n): (1,2), (1,3), (2,2), (1,4), (3,2).)

References

[1] J. Alanen and D. E. Knuth, Tables of finite fields, Sankhyā (Calcutta), 26 (1964) 305–328 (see also Math. Rev. 32, 4122).

[2] A. A. Albert, Fundamental Concepts of Higher Algebra, Univ. of Chicago Press, 1956, pp. 128–131.

[3] E. R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, 1968, chapt. 4.

[4] N. G. de Bruijn, A combinatorial problem, Nederl. Akad. Wetensch. Proc., 49 (1946) 758–764.

[5] _____, Acknowledgment of priority to C. Flye Sainte-Marie on the counting of circular arrangements of 2^n zeros and ones that show each n-letter word exactly once, Report 75-WSK-06, Technical University Eindhoven, 1975.

[6] C. Eldert, H. J. Gray, Jr., H. M. Gurk, and M. Rubinoff, Shifting counters, AIIE Trans., 77 (1958) 70–74.

[7] C. Flye Sainte-Marie, Solution to problem number 58, l'Intermédiaire des Mathématiciens, 1 (1894) 107–110.

[8] H. Fredricksen, A class of nonlinear de Bruijn cycles, J. Combin. Theory, 19 (1975) 191–199.

[9] _____, A survey of full cycle algorithms, SIAM Review, (to appear).

[10] H. Fredricksen and I. J. Kessler, Lexicographic compositions and de Bruijn sequences, J. Combin. Theory, 22 (1977) 17–30.

[11] H. Fredricksen and J. Maiorana, Necklaces of beads in k colors and k-ary de Bruijn sequences, Discrete Math., 23 (1978) 207 – 210.

[12] S. W. Golomb, Shift Register Sequences, Holden-Day, San Francisco, 1967, pp. 118–122.

[13] I. J. Good, Normal recurring decimals, J. London Math. Soc., 21 (1946) 167–169.

[14] M. Hall, Jr., Combinatorial Theory, Blaisdell, Waltham, MA, 1967.

[15] D. E. Knuth, The Art of Computer Programming, Addison-Wesley, Reading, MA, 1969, vol. 2, p. 27.

[16] J. H. van Lint, Combinatorial Theory Seminar, vol. 382, Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1974.

[17] R. L. London, Program Verification in Research Directions in Software Technology, MIT Press, Cambridge, MA, 1979, pp. 302–315.

[18] W. Mantel, Resten van wederkeerige reeksen, Nieuw Arch. Wisk., Ser. 2, (1897) 172–184.

[19] M. H. Martin, A problem in arrangements, Bull. Amer. Math. Soc., 40 (1934) 859–864.

[20] F. P. Preparata and R. T. Yeh, Introduction to Discrete Structures, Addison-Wesley, Reading, MA, 1973, p. 74.

[21] A. Ralston, The twilight of the calculus, Southeast Asian Bull. Math., 3 (1979) 49–56.

[22] A. Ralston and M. Shaw, Curriculum 78—is computer science really that unmathematical?, Comm. ACM, 23 (1980) 67–70.

[23] A. Ralston, Computer science, mathematics and the undergraduate curricula in both, Report No. 161, Dept. of Computer Science, SUNY at Buffalo, 1980.

[24] _____, A new memoryless algorithm for de Bruijn sequences, J. Algorithms, 2 (1981) 50–62.

[25] _____, Computer science, mathematics and the undergraduate curricula in both, Amer. Math. Monthly, 88 (1981) 472–484 (a shortened version of [**23**]).

[26] D. Rees, Note on a paper by I. J. Good, J. London Math. Soc., 21 (1946) 169–172.

[27] E. Roth, Permutations arranged around a circle, Amer. Math. Monthly, 78 (1971) 990–992.

This content downloaded from 142.132.1.147 on Wed, 29 Apr 2020 22:10:38 UTC
All use subject to https://about.jstor.org/terms