THE UNIVERSITY OF MANITOBA

A COMPUTATIONAL ALGORITHM FOR DAVENPORT-SCHINZEL SEQUENCES

BY

CRAIG R. PETERKIN

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

WINNIPEG, MANITOBA

OCTOBER, 1973

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank Dr. R. G. Stanton for his advice and encouragement throughout the preparation of this thesis. I would also like to thank Miss J. Labovitch and Mrs. M. Williams for their assistance in typing the thesis.

ABSTRACT

This thesis presents an algorithm for finding Davenport-Schinzel sequences. Details of its implementation on a high-speed computer are given. An overview of the current literature on Davenport-Schinzel sequences is included. The following new results are obtained:

1) the value of $N(5,6)$,

2) a lower bound for $N(5,n)$, $n > 4$,

3) a correction of the previously determined value of $N(6,5)$,

4) a lower bound for $N(6,n)$, $n > 5$.

CHAPTER I

DAVENPORT-SCHINZEL SEQUENCES

1.1 Problem Definition.

In 1965, Davenport and Schinzel [1] proposed the following combinatorial problem.

Suppose J(n) denotes the set of integers 1,2,...,n. We wish to form sequences of integers using the elements of J(n) subject to the following two constraints:

1) no two adjacent elements are equal;

2) no sequence contains a subsequence of the form abab... of length greater than d, for a and b elements of J(n) and a not equal to b.

N(d,n) denotes the maximal length of any such sequence for given d and n. We are interested in finding the value of N(d,n) for various d and n. Previous literature has placed d as a subscript on N; the current notation is adopted to facilitate use of the text-editing program which produced this copy of the thesis.

It is worthwhile at this point to clarify what is meant by an abab... subsequence. Despite the form in which this is written, the elements need not appear as adjacent entries in the sequence. Thus the sequence 1 2 1 3 2 3 1 2 1 3 contains a 1212... subsequence of length 7, a 1313... subsequence of length 4, and a 2323... subsequence of length 6.

Throughout the discussion, we will call a sequence admissible if it satisfies the definition just given. We call a sequence a Davenport-Schinzel sequence (DS sequence) if it is of length $N(d,n)$.

It is obvious from the definition that $N(d,n)$ is finite. A crude upper bound may be given as follows.

Consider all possible pairs of the integers $1,2,\ldots,n$. Because of the second constraint in the definition, an admissible sequence can contain any given pair of integers at most $d/2 + 1$ times. The number of pairs of unequal integers taken from $1,2,\ldots,n$, is $n(n-1)$. Thus,

$$N(d,n) \leq (d/2 + 1)\ n(n-1)$$
$$= n(n-1)(d+3)/2, \quad n > 1.$$

Before proceeding further, we define a normal sequence. A sequence is said to be normal if, before any occurrence of the element k, there occur all elements less than k. In other words, the elements occur in increasing order. In the search for $N(d,n)$ and DS sequences, we need consider only normal sequences, since a suitable permutation of the elements of $J(n)$ will give a normal sequence. That permutation may be made as follows. Take the first two entries to be 1 and 2, the next entry not equal to either of these to be 3, and so on, naming the entries in order of their first appearance in the sequence. Clearly, the constraints imposed in the definition are unaffected by this permutation; so there is no loss in generality incurred.

1.2 Elementary Results.

Several fundamental known results are presented in this section.

$N(1,n)=1$

There can be no subsequence ab. Thus the sequence can contain only 1 element.

$N(d,1) = 1$

There is only one symbol to be entered.

$N(2,n) = n$

For d = 2, there can be no subsequence of the form aba. Thus each of the integers 1,2,...,n, appears only once. The only normal DS sequence is 1 2 3 ... n.

$N(d,2) = d$

The only normal DS sequence is 1212... to length d, since there are no other symbols to be entered.

1.3 Discussion of N(3,n).

$N(3,n) = 2n - 1$

Consider the following sequences:

1 2 1 3 1 4 ... 1 n-1 1 n 1

1 2 3 4 ... n n-1 n-2 ... 3 2 1.

Both sequences have length 2n - 1 and both are clearly admissible. Thus, $N(3,n) \geq 2n - 1$. We will show that these are DS sequences by proving that $2n - 1$ is also an upper bound.

Two distinct derivations have been given [1] for $N(3,n)$, one based on induction, the other on the fact that there is an integer from $J(n)$ which occurs only once in a DS sequence of length $N(3,n)$. The latter proof is given below.

Suppose S is a DS sequence. Let a be any integer which occurs twice in S, so that

$$i(x) = a \quad \text{and} \quad i(y) = a \quad \text{for } x < y.$$

There must be some integer b which occurs between these occurrences of a, that is,

$$i(z) = b, \quad x < z < y.$$

b can not occur before the first occurrence or after the second occurrence of a, since that would produce subsequences baba or abab, respectively, each of which has length $d + 1$. If b occurs only once, we have established the result. If it occurs more than once, we can repeat the argument replacing a by b and taking some new element as b. Eventually the set of n integers will be exhausted, and we will have the result.

Suppose then that the element n occurs only once in the DS sequence S. If n is deleted from the sequence, we then have a sequence made up of entries from $J(n-1)$ which still satisfies the subsequence criterion. If the elements on either side of n in S were equal, the new sequence now has adjacent elements which are equal. We delete one of them, and the sequence again becomes admissible. Thus, by deleting at most two entries from S, we have an admissible sequence constructed of entries from $1, 2, \ldots, n-1$. This gives

$$N(3,n) - 2 \le N(3,n-1),$$

$$N(3,n) \le N(3,n-1) + 2.$$

Noting that $N(1) = 1$, and recalling that the lower bound is $N(3,n) \ge 2n - 1$, we have $N(3,n) = 2n - 1$.

1.4 Discussion of $N(d,3)$.

$N(d,3) = 3d - 4$, $\quad$ d even, $d > 3$

$N(d,3) = 3d - 5$, $\quad$ d odd, $d > 3$

Case 1. d even.

Let the frequencies of the entries 1, 2, and 3, in a DS sequence be $f(1)$, $f(2)$, and $f(3)$, respectively. Obviously, $N(d,3) = f(1) + f(2) + f(3)$. We may assume without loss of generality that $f(1) \ge f(2) \ge f(3)$.

Lemma

$f(1) = d - 1$

Consider the sequences

1 2 1 2 1 2 1 3 1 3 1 3 1 3 2 3 2 3 2 3

$1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 2\ 3\ 2\ 3\ 2\ 3\ 2\ 3.$

These sequences show that $N(8,3) \geq 20$, $N(10,3) \geq 26$, and, in general,

$$N(d,3) \geq 2(d/2 - 1) + 2(d/2) + 2(d/2 - 1) = 3d - 4.$$

Thus $f(1) \geq (3d - 4)/3 = d-4/3$, that is, $f(1) \geq d - 1$.

Consider the spaces between the 1's in a DS sequence. If $f(1) \geq d - 1$, then there are at least $d - 2$ spaces to be filled. Each space must contain either a single 2 or a single 3. The following argument shows that the entries 2 and 3 can appear in at most $(d-2)/2$ spaces.

If a 2 appeared in $(d-2)/2 + 1 = d/2$ spaces, then there would be a subsequence of the form 1212...1 which would have length $2(d/2) + 1$, and this is impossible. A similar argument applies to 3. Thus there are at most $d - 2$ spaces between the 1's, with an equal number of 2's and 3's filling them. This proves that $f(1) = d - 1$.

Taking the $d - 1$ occurrences of the element 1, the $(d-2)/2$ interior 2's, and the $(d-2)/2$ interior 3's, we have subsequences 12121... and 13131... of length $d - 1$. Thus 2's may be added before the first 1 or following the last 1. The same holds for 3's.

If the 2's and 3's are added at opposite ends of the sequence, only 1 of each can be added. This does not produce a DS sequence, since it is of length

$$(d-1) + (d-2) + 2 = 2d-1 < 3d-4.$$

Adding the 2's and 3's at the same end allows us to add a subsequence of the form 2323... with length $d - 1$. A previous or later 2 or 3 will extend this subsequence to length d. With this construction,

$$N(d,3) \le d-1 + (d-2)/2 + (d-2)/2 + (d-2)/2 + N(d-1,2)$$
$$= 3d - 4.$$

Thus $N(d,3) = 3d - 4$, d even, $d > 3$.

Case 2. d odd.

Lemma

$f(1) = d - 1$

Consider the sequences

1 2 1 2 1 2 3 2 3 2 3 1 3 1 3 1

1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 1 3 1 3 1 3 1.

These show that $N(7,3) \ge 16$, $N(9,3) \ge 22$, and, in general,

$$N(d,3) \ge d - 2 + 2(d - 1)/2 + d - 2 = 3d - 5.$$

Making the same assumptions on symbol frequency as in the case with d even, we have

$$f(1) \ge d - 5/3,$$
$$f(1) \ge d - 1.$$

Suppose that $f(1) > d - 1$. Then we have at least $d - 1$ interior positions to be filled with single 2's or 3's. 2 and 3 can occur in at most $(d - 1)/2$ of these positions, since if a 2 or 3 occurred in $(d - 1)/2 + 1 = (d + 1)/2$ positions, then there would be a subsequence having length

$2(d + 1)/2 + 1 = d + 2$.

If $f(1) = d + 1$, we would require d interior positions to be filled by d - 1 2's and 3's; consequently, we see that $f(1) \leq d$.

Suppose that $f(1) = d$. The subsequences 12121... and 13131... have length d; consequently, neither the element 2 nor the element 3 can occur before the first 1 or following the last 1. Thus the sequence has length $d+d-1 = 2d - 1$. This sequence can not be maximal, since $N(d,3) \geq 3d - 5$. Thus $f(1) < d$, and $f(1) = d - 1$.

We know then that there are d - 2 interior positions, and each of 2 or 3 can appear in at most $(d - 1)/2$ positions. Consider the following two cases.

Case A.

Each of 2 and 3 occurs in $(d - 1)/2$ positions. This results in a total of d - 1 occurrences, that is, there is some interior position in which both 2 and 3 occur. In this overlapping interior position, we can add at most a sequence of length $N(d-1,2)$ to give a sequence of length at most

$$(d - 1) + (d - 3) + (d - 1) = 3d - 5.$$

The examples given are of this length; hence, $N(d,3) = 3d-5$, provided Case B does not yield a longer sequence.

Case B.

Suppose that 2 occurs in $(d-1)/2$ positions and 3 in

$(d-3)/2$ positions; this fills all interior positions with single symbols. Then one further 3 can be added at the beginning and the end of the sequence to give a sequence of length

$$1 + (d - 1) + (d - 2) + 1 = 2d - 1.$$

Since this is less than $3d - 5$ for all d being considered here (that is, $d = 5,7,9,\ldots$), the result in Case A holds. Thus we have that, for d odd, $d > 3$,

$$N(d,3) = 3d - 5.$$

Stanton and Roselle claim [2] that the construction just given given for $N(d,3)$ is unique for both even and odd d. It is instructive to clarify their use of the term "unique". In the case d even, the construction produces exactly one DS sequence, since the subsequence of length $N(d-1,2)$ must be added at the end of the sequence. However, for odd d, the construction actually produces $d - 2$ normal DS sequences, none of which may be obtained from another by a simple permutation of the symbols.

Consider $d = 5$. The three normal DS sequences are

1 2 1 . 2 3 2 3 . 1 3 1

1 2 1 3 1 . 3 2 3 2 . 1

1 . 2 3 2 3 . 1 3 1 2 1

We see that the sequences of length $N(d-1,2)$ may be inserted in any of the $d - 2$ interior positions, with each insertion giving a different DS sequence. The construction is "unique" insofar as it is the only way to construct a DS sequence for $n = 3$ and d odd, but it does produce $d - 2$ normal DS sequences.

1.5 Discussion of $N(4,n)$, $n > 4$.

Consider the following sequences:

$n=5$, $d=4$ 1 2 1 3 1 4 1 . 4 3 2 . 5 2 5 3 5 4 5

$n=6$, $d=4$ 1 2 1 3 1 4 1 5 1 . 5 4 3 2 . 6 2 6 3 6 4 6 5 6

The construction of these sequences shows that $N(4,5) \geq 17$, $N(4,6) \geq 22$, and, in general, that

$$N(4,n) \geq 2(n-1) + (n-3) + 2(n-2) + 1 = 5n - 8.$$

(These examples were found to be unique DS sequences in an exhaustive search using the program which will be described in Chapter II.)

Stanton and Roselle have shown [5] that

$$5n - 8 \leq N(4,n) \leq n\ N(4,n-1)/(n-1) + 2.$$

This gives

$N(4,7) = 27$,

$N(4,8) = 32$.

J. H. Conway has shown that

$N(4,9) = 37$,

$N(4,10) = 42$.

W. H. Mills has shown that

$N(4,11) = 47$,

$N(4,12) = 53$,

$N(4,13) = 58$.

$n = 11$ is the last value for which $N(4,n) = 5n - 8$. Davenport has proved [4] that

$$N(4,km+1) \geq 6km - m - 5k + 2.$$

This implies that $N(4,n) > 5n - 8$ for odd $n \geq 13$ and for even $n \geq 18$. He notes also [4] that Z. Kolba has proved that $N(4,2m) \geq 11m - 13$. Combining these results, we have that $N(4,n) > 5n - 8$ for all $n \geq 12$.

For values of n such that $14 \leq n \leq 20$, the following bounds are listed according to the material outlined above.

$64 \leq N(4,14) \leq 64$	Kolba
$69 \leq N(4,15) \leq 70$	Davenport
$75 \leq N(4,16) \leq 77$	Kolba
$80 \leq N(4,17) \leq 84$	Davenport
$86 \leq N(4,18) \leq 91$	Davenport
$91 \leq N(4,19) \leq 98$	Davenport
$98 \leq N(4,20) \leq 105$	Davenport

1.6 Discussion of $N(d,4)$.

$N(d,4) = 6d - 13$, d even, $d > 4$

$N(d,4) = 6d - 14$, d odd, $d > 4$

We begin by establishing a lower bound.

Case 1. d even, $d > 4$.

Let f be the maximum number of occurrences of any element in the DS sequence. Without loss of generality, we choose the element 1. This implies that there are $f - 1$ interior blocks of elements, each of which contains elements chosen from $2, 3, \ldots, n$, and is bounded on both sides by 1's. None of the remaining $n - 1$ elements may appear in as many as $d/2$ interior blocks, since this would give a subsequence of the form 1a1a... of length $d + 1$. Thus, the maximum number of blocks in which an element other than 1 appears is $d/2 - 1 = (d - 2)/2$. Also, a pair of 1's can not appear as adjacent elements; thus, the number of interior blocks must be equal to or less than the total number of elements appearing in the blocks. This gives

$$f - 1 \leq (n - 1)(d - 2)/2,$$

$$f \leq (n - 1)(d - 2)/2 + 1.$$

The following construction shows that this bound can be attained.

List $(n - 1)(d - 2)/2 + 1$ 1's. Insert n between the first $(d - 2)/2$ 1's, n-1 between the next $(d - 2)/2$ 1's, etc., down to the element 2. Finally, to the right of the last 1, enter a normal DS sequence of length $N(d-1,n-1)$ made up of the elements $2, \ldots, n$. Certainly there are no adjacent elements equal. We can see that there is no abab... subsequence of length greater than d by noting that the DS sequence of length $N(d-1,n-1)$ has no abab... subsequence (a>b) of length greater than d - 2. Thus, the sequence we have just constructed is admissible. This proves that, for d even, $d > n$,

$$N(d,n) \geq (n-1)(d-2) + 1 + N(d-1,n-1).$$

In particular, for $n = 4$,

$$N(d,4) \geq 3d - 6 + 1 + 3(d-1) - 5 = 6d - 13.$$

Case 2. d odd, $d > 4$.

Again let f denote the maximum frequency of any element, say 1. In this case, none of the remaining elements may appear in more than $(d-1)/2$ of the $f-1$ interior blocks, since more appearances would result in an abab... subsequence of length greater than d. Thus,

$$f - 1 \leq (n-1)/2,$$

$$f \leq (n-1)(d-1)/2 + 1.$$

This bound is attainable as follows.

List $(n-1)(d-1)/2 - n + 3$ 1's. Insert n between the first $(d-3)/2$ of the 1's, $n-1$ between the next $(d-3)/2$ of the 1's, etc.; between the last two 1's, insert a normal DS sequence of length $N(d-1,n-1)$. This sequence is admissible, and its existence shows that, for d odd, $d > n$,

$$N(d,n) \geq (n-1)(d-3) + 1 + N(d-1,n-1) + 1.$$

In particular, for $n = 4$,

$$N(d,4) \geq 3(d-3) + 2 + 3(d-1) - 4 = 6d - 14.$$

We will now show that, for $n = 4$, the bounds

$$N(d,4) \geq 6d - 13, \quad (d \text{ even}),$$

$$N(d,4) \geq 6d - 14, \quad (d \text{ odd}),$$

are also upper bounds.

<u>$n = 4$; d even, $d > 4$.</u>

We know that

$$f \leq 3(d - 2)/2 + 1 = (3d - 6)/2 + 1 = 3d/2 - 2,$$

and $N(d,4) \geq 6d - 13$. Since there are 4 symbols, and f denotes the maximum frequency of the most common one,

$$4f \geq N(d,4) \geq 6d - 13.$$

This gives

$$f \geq (6d - 13)/4 = 3d/2 - 3,$$

$$3d/2 - 2 \geq f \geq 3d/2 - 3.$$

Suppose that $f = 3d/2 - 3$. Then there are $3d/2 - 4$ interior positions. If there is no overlapping of elements, that is, each position is filled by a single symbol, then the number of blocks filled by the remaining symbols 2,3, and 4, must be $(d-2)/2$, $(d-2)/2$, and $(d-4)/2$. If overlapping does occur, we see, by the following argument, that it can happen in just one interior block.

If we have one entry in each interior block, there are then $3d/2 - 4$ appearances of elements. Adding an element in one of these interior blocks will increase this number to $3d/2 - 3$. But 2,3, and 4, can occur in at most $(d - 2)/2$ blocks; hence, the maximum number of block occupancies is $3(d - 2)/2$, and we see that there can be no more overlapping.

Case A. No Overlapping.

Let us take the number of blocks containing 2, 3, and 4, to be $(d-2)/2$, $(d-2)/2$, and $(d-4)/2$, respectively. Since the sequence already contains subsequences 12121... and 13131... of length $2(d-2)/2 + 1 = d - 1$, we can not add either 2 or 3 at both ends of the sequence. However, 4 could occur at both ends. If 2 and 4 appear at one end, with 3 and 4 at the other end, then we have a sequence whose length is at most

$$f + f - 1 + 2N(d-1,2)$$

$$= 3d/2 - 3 + 3d/2 - 4 + 2d - 2 = 5d - 9.$$

If 2, 3, and 4, appear at the same end of the sequence, with a single 4 at the other end, we have maximum length

$$1 + f + f - 1 + N(d-1,3)$$

$$= 1 + 3d - 7 + 3(d-1) - 5 = 6d - 14.$$

But we know that $N(d,4) \geq 6d - 13$ for $d > 4$; so neither of these is a DS sequence.

Case B. Single Overlap.

Let us assume 3 and 4 occur in the same interior block. The sequence already contains subsequences 1a1a... of length $d - 1$ for a = 2, 3, and 4; consequently, each of these entries can occur at only one end of the sequence.

If 4 is at one end and 2 and 3 at the other end, then the sequence can not be longer than

$$1 + f + f - 2 + 2\,N(d-1,2) = 5d - 9 < 6d - 13.$$

If 2 is at one end and 3 and 4 at the other end, then the sequence can not be longer than

$$1 + f + f - 2 + 2 + N(d-1,2) = 4d - 8 < 6d - 13.$$

If 2, 3, and 4, appear at one end, the length is at most

$$f + f - 2 + N(k+2,2) + N(d-k,2), \quad k \geq 1.$$

This gives $6d - 2k - 10$ for k even, and $6d - 2k - 11$ for k odd. The lower bound $N(d,4) \geq 6d - 13$ implies that $k = 1$.

It has been shown [3] that the DS sequence of length $N(d,3)$ for odd d contains a subsequence abab... of length at least $d - 2$ for all a and b. The sequence for $k = 1$ above must then contain a subsequence which has length at least $1+3+d-1-2 = d+1$, and hence is inadmissible. We have exhausted all possibilities, and can conclude that f can not be equal to $3d/2 - 3$.

The construction used in establishing the lower bound $N(d,4) \geq 6d - 13$ gives a sequence in which $f = 3d/2 - 2$. It remains only to examine the possible constructions of sequences with this frequency of 1's.

There can be no overlapping, since the number of entries to be made is at most $f - 1 = 3d/2 - 3$, and the maximum number of interior blocks in which 2, 3, or 4, can occur is $(d - 2)/2$. The sequence already contains 1a1a... subsequences of length d-1; thus, additional entries of 2, 3, and 4, must appear at only one end of the sequence.

If 2 occurs at one end and 3 and 4 at the other, then the sequence has maximum length

$$1 + f + f - 1 + N(d-1,2)$$

$$= \quad 3d - 4 + d - 1 = 4d - 5.$$

This is not a DS sequence, since $6d - 13 > 4d - 5$ for $d > 4$.

If 2, 3, and 4, occur at the same end, we have the sequence constructed for the lower bound, which is of length

$$f + f - 1 + N(d-1,3)$$

$$= 3d/2 - 2 + 3d/2 - 3 + 3(d - 1) - 5 = 6d - 13.$$

Thus we have shown that, for d even, $d > 4$,

$$N(d,4) = 6d - 13.$$

<u>$n = 4$; d odd, $d > 4$</u>

From the development of the lower bound, we have

$(6d - 14)/4 = 3(d - 1)/2 - 2 \le f \le 3(d - 1)/2 + 1.$

We gave the construction of a sequence of length $6d - 14$ for which $f = 3(d - 1)/2 - 1$. If $f = 3(d - 1)/2 - 2$, then $N(d,4) = 6d - 14$; so we need only consider

$$3(d - 1)/2 - 1 \le f \le 3(d - 1)/2 + 1.$$

If $f = 3(d - 1)/2 + 1$, then each interior position is occupied by a single element, since each of 2,3, and 4, can occur in at most $(d - 1)/2$ blocks. This sequence contains all subsequences 1a1a... of length $2(d - 1)/2 + 1 = d$; so no further entries can be made at either end. The length of this sequence is

$$f + f - 1 = 2(3(d-1)/2 + 1) - 1 = 3d - 2 < 6d - 14.$$

If $f = 3(d - 1)/2$, there can be an overlap in at most one of the interior blocks.

If there is no overlap, the sequence has length

$$f + f - 1 + 1 = 2f = 3d - 3 \quad < \quad 6d - 14.$$

If there is an overlap in one interior position, then the length is

$$f + f - 2 + N(d-1,2) = 4d - 6 \quad < \quad 6d - 14.$$

The last possibility for f is $3(d-1)/2 - 1$. This allows overlapping to occur in at most two interior positions. If there is no overlapping, the sequence can not be longer than

$$f + f - 1 + N(d-1,2) + 2 = 4d - 3 \quad < \quad 6d - 14.$$

If there is overlapping in two of the interior positions, we may have 2 and 3 together in both of them, or 2 and 3 in one and 2 and 4 in the other. In the first case, the sequence has maximum length

$$f + f - 3 + N(d-k,2) + N(k+1,2).$$

As before, we see that $k = 1$, and the length is $4d - 7$. The second case gives a longer sequence, with length

$$f + f - 3 + 2N(d-1,2) = 5d - 10 \quad < \quad 6d - 14.$$

The final possibility, that of overlap in one interior block, gives the longest sequence, as follows.

If only two distinct elements occur in the overlap, the sequence has length at most

$$f + f - 2 + N(d-1,2) + 2 = 4d - 6 \quad < \quad 6d - 14.$$

If 2,3, and 4, all occur in the overlapping position, the sequence can not be longer than

$$f + f - 2 + N(d-1,3) = 6d - 14.$$

All possibilities are exhausted, and we have proved that, for d odd, $d > 4$,

$$N(d,4) = 6d - 14.$$

1.7 Results Derived for N(5,6).

The prime purpose of the computational algorithm which will be described in Chapter II is to find the value of N(5,6). We shall show that this number is:

$$N(5,6) = 29.$$

There are thirty-five DS sequences for $n = 6$, $d = 5$; these are listed below in the order in which they appeared in the search.

1 2 1 2 3 2 4 2 5 2 5 4 3 6 3 6 4 6 5 6 1 6 1 5 1 4 1 3 1

1 2 1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 6 5 6 1 6 3 6 4 6 4 3 1

1 2 1 2 3 2 4 2 5 2 5 4 5 3 5 6 5 6 3 6 4 6 1 6 1 4 1 3 1

1 2 1 2 3 2 4 2 5 2 6 2 6 5 6 4 6 3 6 3 4 5 1 5 1 4 1 3 1

1 2 1 2 3 4 5 4 5 3 5 2 5 6 5 6 2 6 3 6 4 6 1 6 1 4 1 3 1

1 2 1 3 1 3 2 3 4 3 5 3 5 4 2 6 2 6 4 6 5 6 1 6 1 5 1 4 1

1 2 1 3 1 3 2 3 4 3 5 3 5 4 5 2 5 6 5 6 2 6 4 6 1 6 1 4 1

1 2 1 3 1 3 2 3 4 3 5 3 6 3 6 5 6 4 6 2 6 2 4 5 1 5 1 4 1

1 2 1 3 1 3 2 4 5 4 5 2 5 3 5 6 5 6 3 6 2 6 4 6 1 6 1 4 1

1 2 1 3 1 4 1 4 3 2 5 2 5 3 5 4 5 6 5 6 4 6 3 6 2 6 1 6 1

1 2 1 3 1 4 1 4 3 4 2 4 5 4 5 2 3 6 3 6 2 6 5 6 1 6 1 5 1

1 2 1 3 1 4 1 4 3 4 2 4 5 4 5 2 5 3 5 6 5 6 3 6 2 6 1 6 1

1 2 1 3 1 4 1 4 3 4 2 4 5 4 6 4 6 5 6 2 6 3 6 3 2 5 1 5 1

1 2 1 3 1 4 1 5 1 5 4 3 6 3 6 4 6 5 6 2 6 2 5 2 4 2 3 2 1

1 2 1 3 1 4 1 5 1 5 4 5 3 5 2 5 2 3 4 6 4 6 3 6 2 6 1 6 1

1 2 1 3 1 4 1 5 1 5 4 5 3 5 2 5 6 5 6 2 6 3 6 4 6 4 3 2 1

1 2 1 3 1 4 1 5 1 5 4 5 3 5 6 5 6 3 6 4 6 2 6 2 4 2 3 2 1

1 2 1 3 1 4 1 5 1 6 1 6 5 6 4 6 3 6 3 4 5 2 5 2 4 2 3 2 1

1 2 1 3 4 5 4 5 3 5 1 5 6 5 6 1 6 3 6 4 6 2 6 2 4 2 3 2 1

1 2 3 2 3 1 3 4 3 5 3 5 4 1 6 1 6 4 6 5 6 2 6 2 5 2 4 2 1

1 2 3 2 3 1 3 4 3 5 3 5 4 5 1 5 2 5 6 5 6 2 6 1 6 4 6 4 1

1 2 3 2 3 1 3 4 3 5 3 5 4 5 1 5 6 5 6 1 6 4 6 2 6 2 4 2 1

1 2 3 2 3 1 3 4 3 5 3 6 3 6 5 6 4 6 1 6 1 4 5 2 5 2 4 2 1

1 2 3 2 3 1 4 5 4 5 1 5 3 5 6 5 6 3 6 1 6 4 6 2 6 2 4 2 1

1 2 3 2 4 2 4 3 1 5 1 5 3 5 4 5 6 5 6 4 6 3 6 1 6 2 6 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 3 6 3 6 1 6 5 6 2 6 2 5 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 5 3 5 6 5 6 3 6 1 6 2 6 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 6 4 6 5 6 1 6 3 6 3 1 5 2 5 2 1

1 2 3 2 4 2 5 2 5 4 3 6 3 6 4 6 5 6 1 6 1 5 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 1 3 4 6 4 6 3 6 1 6 2 6 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 6 5 6 1 6 3 6 4 6 4 3 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 6 5 6 3 6 4 6 1 6 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 6 2 6 5 6 4 6 3 6 3 4 5 1 5 1 4 1 3 1 2 1

1 2 3 4 3 4 2 4 1 4 5 4 5 1 5 2 5 3 5 6 5 6 3 6 2 6 1 6 1

1 2 3 4 5 4 5 3 5 2 5 6 5 6 2 6 3 6 4 6 1 6 1 4 1 3 1 2 1

1.8 A Lower Bound for $N(5,n)$, $n > 4$.

Observation of the sequences in the previous section allows us to develop a lower bound for $N(5,n)$, $n > 4$. Consider the following sequence for $d = 5$, $n = 6$.

$$1\ 2\ 1\ 2\ 3\ 2\ 4\ 2\ 5\ 2\ .\ 5\ 4\ 3\ .\ 6\ 3\ 6\ 4\ 6\ 5\ 6\ .$$

$$1\ 6\ 1\ 5\ 1\ 4\ 1\ 3\ 1$$

Consider also the following sequence for $d = 5$, $n = 7$.

$$1\ 2\ 1\ 2\ 3\ 2\ 4\ 2\ 5\ 2\ 6\ 2\ .\ 6\ 5\ 4\ 3\ .\ 7\ 3\ 7\ 4\ 7\ 5\ 7\ 6\ 7\ .$$

$$1\ 7\ 1\ 6\ 1\ 5\ 1\ 4\ 1\ 3\ 1$$

These sequences are certainly admissible, and show that $N(5,6) \geq 29$, $N(5,7) \geq 36$. Construction of admissible sequences of the same form for larger n is clearly possible, and thus we have that, for $n > 5$,

$$N(5,n) \geq 2(n-1) + (n-3) + 2(n-3) + 1 + 2(n-2) + 1 = 7n - 13.$$

1.9 Discussion of $N(d,5)$.

Case 1. d even, $d > 5$.

We recall that in obtaining $N(d,4)$, we proved that

$$f \leq (n-1)(d-2)/2 + 1,$$

where f is the frequency of 1's in a DS sequence. For $n=5$, we have $f \leq 2d-3$. We recall also that

$$N(d,n) \geq (n-1)(d-2) + 1 + N(d-1,n-1), \quad d \text{ even, } d > n.$$

This gives $N(d,5) \geq 10d-27$, and we see immediately that

$$2d-3 \geq f \geq 2d-5.$$

Stanton and Roselle claim [5] that this bound of 10d-27 is also an upper bound, and occurs when f = 2d-3. Their method is similar to that used in obtaining N(d,4), that is, they consider occurrences of overlap within interior blocks, and then the possible entries exterior to the blocks. They claim that, for f = 2d-4 and f = 2d-5, it is not possible to construct a sequence of length 10d-27. Their results, however, do not apply to the case d = 6; there does exist a unique maximal sequence of length 34 with f = 2d-5, as we shall show below.

The following lemma is useful.

Lemma

In a DS sequence of length N(d,3), all subsequences abab... occur at least d-1 times if d is even and at least d-2 times if d is odd.

The result can readily be seen by considering sequences of the following type.

N(8,3)	1 2 1 2 1 2 1 3 1 3 1 3 1 . 3 2 3 2 3 2 3
N(7,3)	1 2 1 2 1 . 2 3 2 3 2 3 . 1 3 1 3 1

Let us now consider N(6,5). Suppose that f = 2d-5 = 7. If there is no overlap in the 6 interior positions, the interior block occurrences of 2,3,4, and 5, may be (d-2)/2, (d-2)/2, (d-2)/2, and (d-6)/2, respectively, or (d-2)/2, (d-2)/2, (d-4)/2, and (d-4)/2, respectively. It is in the first case that a sequence of length 34 may be found.

Suppose that the interior block frequencies are $(d-2)/2$ for 2,3, and 4, and $(d-6)/2$ for 5. This gives the following initial entries in the sequence:

$$1\ 2\ 1\ 2\ 1\ 3\ 1\ 3\ 1\ 4\ 1\ 4\ 1.$$

It is particularly important to notice that, for $d = 6$, the element 5 does not occur in an interior block; it is this zero frequency, for $d = 6$, which was ignored by Stanton and Roselle. Let us now add to these elements a sequence of length $N(d-2,3) = 8$ made up of the elements 2,3, and 4. By the Lemma, this part of the sequence will contain all subsequences aba to length at least 3. Now consider the lengths of abab... subsequences thus far in the sequence.

1212, 1313, 1414 - already of length d (=6)

2323, 2424, 3434 - now of length $d - 1$ (2+3=5)

We may then add the following elements:

$$5\ 2\ 5\ 2\ 5\ 3\ 5\ 3\ 5\ 4\ 5\ 4\ 5.$$

These will increase the length of the 2323, 2424, and 3434 subsequences to length d, and will introduce a5a5 subsequences of length d for $a = 2,3$, and 4. The last subsequence 1515... has length only 2. All subsequences except this one then are of length d, and no further 1's may be added because of the fact that $f = 2d-5$. Thus no further elements can be added, and the sequence is maximal. Since the DS sequence for $N(4,3)$ is unique, the above construction is unique also, and gives the following corresponding DS sequence

1 2 1 2 1 3 1 3 1 4 1 4 1 . 4 3 4 2 4 2 3 2 .

5 2 5 2 5 3 5 3 5 4 5 4 5

Combining this new result with the proofs in [5], we have:

$$N(d,5) = 10d - 27, \quad d \text{ even}, \; d > 6,$$

and

$$N(6,5) = 34.$$

<u>Case 2. d odd, d > 5.</u>

We know that

$$N(d,n) \geq (n-1)(d-3) + 2 + N(d-1,n-1), \quad d \text{ odd}.$$

This gives

$$N(d,5) \geq 10d-29, \quad d \text{ odd}.$$

Also,
$$f \leq (n-1)(d-1)/2 + 1, \quad d \text{ odd},$$

and we see that

$$2d-5 \leq f \leq 2d-1.$$

Consideration of each of these possible frequencies for f follows. The principles are similar to those in previous proofs; hence, many of the details will be omitted here.

Suppose that $f = 2d-1$.

We know that the elements other than 1 can not occur in interior blocks with frequency greater than $(d-1)/2$. But $4(d-1)/2 = 2d-2 = f-1$. There is no overlap, and the sequence has length

$$f + f - 1 = 4d - 3 < 10d - 29.$$

Suppose that $f = 2d-2$.

In this case, we may have no overlap or one overlap. If there is no overlap, we have interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-3)/2$. This yields a sequence not longer than

$$f + f - 1 + 2 = 4d - 3 < 10d - 29.$$

If there is an overlap, the sequence has maximum length

$$f + f - 2 + N(d-1,2) = 5d - 7 < 10d - 29.$$

Suppose that $f = 2d-3$.

In this case, there is a maximum of two overlaps. If there is no overlap, the interior frequencies may be $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-5)/2$, which produces a sequence not longer than

$$f + f - 1 + 2 = 4d - 5 < 10d - 29.$$

Alternatively, the frequencies may be $(d-1)/2$, $(d-1)/2$, $(d-3)/2$, and $(d-3)/2$, which produces a sequence not longer than

$$f + f - 1 + 1 + N(d-1,2) = 5d - 7 < 10d - 29.$$

If there is one overlap, the interior frequencies may be $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-3)/2$. This will produce a sequence of length not exceeding

$$f + f - 2 + 2 + N(d-1,2) = 5d - 7 < 10d - 29.$$

Finally, if two overlaps occur, the interior frequencies are all $(d-1)/2$. The longest sequence will be produced when these overlaps occur in one interior block. This results in a sequence of length

$$f + f - 2 + N(d-1,3) = 7d - 15 < 10d - 29.$$

Suppose that $f = 2d-5$.

If there is no overlap, we have several possibilities for interior frequencies. They may all be $(d-3)/2$. This will yield a sequence of length not greater than

$$f + f - 1 + 1 + N(d-1,4) = 10d - 29.$$

Consideration of the subsequences 5a5a... in this case shows that the sequence is not a DS sequence.

Next, we may have interior frequencies $(d-1)/2$, $(d-3)/2$, $(d-3)/2$, and $(d-5)/2$. This gives a sequence not longer than

$$f + f - 1 + 1 + N(d-1,3) = 7d - 17 < 10d - 29.$$

For $d > 7$, we may have interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-3)/2$, and $(d-7)/2$. This will produce a sequence not longer than

$$f + f - 1 + 1 + N(d-1,2) = 5d - 11 < 10d - 29.$$

Or, if the interior frequencies are $(d-1)/2$, $(d-1)/2$, $(d-5)/2$, and $(d-5)/2$, we obtain a sequence of length

$$f + f - 1 + 1 + N(d-1,2) = 5d - 11 < 10d - 29.$$

Finally, for $d > 9$, the interior frequencies may be $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-9)/2$. The sequence has length

$$f + f - 1 + 2 = 4d - 9 < 10d - 29.$$

We now consider possible distributions in cases of a single overlap.

If we have interior frequencies $(d-1)/2$, $(d-3)/2$, $(d-3)/2$, and $(d-3)/2$, the sequence has length

$$f + f - 2 + 1 + N(d-1,3) + N(d-1,2) = 8d - 19 < 10d - 29.$$

Or, we may have interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-3)/2$, and $(d-5)/2$. The sequence then has length

$$f + f - 2 + 1 + 2N(d-1,2) = 6d - 13 < 10d - 29.$$

Finally, for $d > 7$, we consider interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-7)/2$. The sequence is then of length

$$f + f - 2 + 1 + N(d-1,2) + 1 = 5d - 11 < 10d - 29.$$

Now we look at the possibility of two overlaps. The interior frequencies may be $(d-1)/2$, $(d-1)/2$, $(d-3)/2$, and $(d-3)/2$; if the overlaps occur in different blocks, the sequence has length

$$f + f - 3 + 2N(d-1,2) + 1 + N(d-1,2) = 7d - 15 < 10d - 29.$$

In the case where the overlaps occur in the same block, the sequence has length

$$f + f - 2 + N(d-1,3) + 1 + N(d-1,2) = 8d - 19 < 10d - 29.$$

Finally, we may have interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-5)/2$. The sequence length will be either

$$f + f - 2 + 2 + N(d-1,3) = 7d - 17 < 10d - 29,$$

or, alternatively,

$$f + f - 3 + 2 + 2N(d-1,2) = 6d - 13 < 10d - 29.$$

If there are three overlaps, the interior frequencies may be $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-3)/2$. The three possible lengths are:

$$f + f - 4 + 2 + 3N(d-1,2) = 7d - 15 < 10d - 29,$$

$$f + f - 3 + 2 + N(d-1,2) + N(d-1,3) = 8d - 19 < 10d - 29,$$

$$f + f - 2 + 2 + N(d-1,4) = 10d - 29.$$

As before, it is possible to eliminate the last case by considering the length of subsequences in the construction.

It is possible to have four overlaps, with interior frequencies all $(d-1)/2$. This gives rise to the following lengths:

$$f + f - 5 + 4N(d-1,2) = 8d - 19 < 10d - 29,$$

$$f + f - 4 + 2N(d-1,2) + N(d-1,3) = 9d - 23 < 10d - 29,$$

$$f + f - 3 + N(k,3) + N(m,3), \text{ where } k-2+m-2 \le d,$$

$$f + f - 3 + N(d-1,2) + N(d-1,3) = 11d - 33.$$

The second-last length beccmes

$$4d-13-3k+3m-8 \le 4d-21+3(d+7) = 7d-9 < 10d - 29.$$

The last result may be improved by considering subsequences. Since 45 occurs in more than one block, we may reduce the length to

$$f + f - 3 + N(d-k,4) + N(k+2,2)$$

$$= 4d - 13 + 6d - 6k - 13 + k + 2 = 10d - 5k - 24$$

$$(k \text{ odd}), \text{ or},$$

$$= 4d - 13 + 6d - 6k - 14 + k + 3 = 10d - 5k - 24$$

$$(k \text{ even}).$$

Because of the lower bound, k must be equal to 1, and we find that the length is less than or equal to $10d - 29$. This may then be ruled out by considering 3434... subsequences.

Thus, all possibilities for $f = 2d-5$ have been exhausted, and none has produced a sequence of length $10d - 29$.

The final possibility for f is $2d-4$.

If there are no overlaps, the interior frequencies may be $(d-1)/2$, $(d-3)/2$, $(d-3)/2$, and $(d-3)/2$. This gives a sequence with length at most

$$f + f - 1 + 1 + N(d-1,3) = 7d - 15 < 10d - 29.$$

If the interior frequencies are $(d-1)/2$, $(d-1)/2$, $(d-3)/2$, and $(d-5)/2$, the sequence has length

$$f + f - 1 + 1 + N(d-1,2) = 5d - 9 < 10d - 29.$$

For $d > 7$, and interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-7)/2$, the sequence can not be longer than

$$f + f - 1 + 2 = 4d - 7 < 10d - 29.$$

If there is one overlap, we may have $(d-1)/2$, $(d-1)/2$, $(d-3)/2$, and $(d-3)/2$, with length

$$f + f - 2 + 1 + 2N(d-1,2) = 6d - 11 < 10d - 29;$$

or, we may have interior frequencies $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-5)/2$. The sequence then has length

$$f + f - 2 + N(d-1,2) + 2 = 5d - 9 < 10d - 29.$$

Two overlaps might result from frequencies $(d-1)/2$, $(d-1)/2$, $(d-1)/2$, and $(d-3)/2$. There are two possibilities for the length of this sequence, namely,

$$f + f - 2 + 2 + N(d-1,3) = 7d - 15 < 10d - 29,$$

$$f + f - 3 + 2 + 2N(d-1,2) = 6d - 11 < 10d - 29.$$

If three overlaps occur, with all frequencies $(d-1)/2$, the possibilities for length are:

$$f + f - 4 + 3N(d-1,2) = 7d - 15 < 10d - 29,$$

$$f + f - 3 + N(d-1,2) + N(d-1,3) = 8d - 19 < 10d - 29,$$

$$f + f - 2 + N(d-1,4) = 10d - 29.$$

The sequence corresponding to the last possibility is the one constructed for the lower bound.

All possibilities have been exhausted, and we have shown that, for $d > 5$, d odd,

$$N(d,5) = 10d - 29.$$

1.10 A Lower Bound for $N(6,n)$, $n > 5$.

We are able to extend the construction given in the determination of $N(6,5)$ to produce a lower bound for larger n. We recall that the sequence of length 34 for $N(6,5)$ was constructed as follows:

$$1\ 2\ 1\ 2\ 1\ 3\ 1\ 3\ 1\ 4\ 1\ 4\ 1,$$

plus

a sequence of length $N(4,3)$ on the elements 2,3, and 4,

plus

$$5\ 2\ 5\ 2\ 5\ 3\ 5\ 3\ 5\ 4\ 5\ 4\ 5.$$

Because of the properties of DS sequences for $n = 3$, we saw that the sequence was maximal for this particular frequency of 1's.

Let us consider a generalization of this construction. Take a sequence

$$1\ 2\ 1\ 2\ 1\ 3\ 1\ 3\ 1\ \ldots\ n-1\ 1\ n-1\ 1,$$

plus

a sequence of length $N(4,n-2)$ on the elements $2,3,\ldots,n-1$,

plus

$$n\ 2\ n\ 2\ n\ 3\ n\ 3\ \ldots\ n\ n-1\ n\ n-1\ n.$$

We must now determine whether, for $n > 3$ and $d = 4$, we can always find a DS sequence which, when inserted between the initial and terminal elements, will leave the sequence admissible. We show that this is possible by considering the sequence construction used earlier in obtaining the lower bound for $N(4,n)$. Two examples were given ($n=5$ and $n=6$). With a suitable renaming of the elements, we can always insert this sequence into the structure above and maintain an admissible sequence. By using the lengths of these sequences we find that, for $n > 4$,

$$\begin{aligned} N(6,n) &\geq 4(n-2) + 1 + N(4,n-2) + 4(n-2) + 1 \\ &\geq 8n - 14 + 5(n-2) - 8 \\ &= 13n - 32. \end{aligned}$$

1.11 The Diagonal Entries: $N(d,n)$ for $d = n$.

For $d = n$, the Davenport-Schinzel problem seems not to share the characteristics of either the case $d > n$ or $d < n$. Most of the literature involves one or other of these cases, and analysis of the diagonal entries using non-computational methods has not proved to be very successful.

The first unknown diagonal entry is, of course, $N(6,6)$. Some spot searches were done for this case using the program in Chapter II, but no sequences of length greater than 46 (the length given by the bound in the section above) were found. However, since this is considerably longer than any previously known sequence for $n = d = 6$, we note it here as

1 2 1 2 1 3 1 3 1 4 1 4 1 5 1 5 1 .

5 4 5 3 5 3 4 2 4 2 3 2 .

6 2 6 2 6 3 6 3 6 4 6 4 6 5 6 5 6

For $n = d = 3$, there are 2 DS sequences (length 5).

For $n = d = 4$, there are 5 DS sequences (length 12).

For $n = d = 5$, there are 63 DS sequences (length 22).

The sequences for these diagonal entries are listed for reference in Appendix A.

We conclude this chapter by giving the following table, which contains the known values of $N(d,n)$ for $n,d \le 10$.

		d									
		1	2	3	4	5	6	7	8	9	10
	1	1	1	1	1	1	1	1	1	1	1
	2	1	2	3	4	5	6	7	8	9	10
	3	1	3	5	8	10	14	16	20	22	26
	4	1	4	7	12	16	23	28	35	40	47
n	5	1	5	9	17	22	34	41	53	61	73
	6	1	6	11	22	29					
	7	1	7	13	27						
	8	1	8	15	32						
	9	1	9	17	37						
	10	1	10	19	42						

CHAPTER II

THE COMPUTATIONAL ALGORITHM

2.1 Introduction.

The basic algorithm used in constructing DS sequences is outlined in succeeding sections.

We begin with the entries 1 2. Since the next entry may be either another 1 or a new symbol 3, these are the only entries that may be made, in general. For certain runs of the program, additional entries were made, in order to decrease computation time. Having placed these original entries in the sequence, we must then make subsequent entries according to the definition of a valid DS sequence in Chapter I.

The discussion in the following sections does not consider the trivial cases $n = 2$ or $d = 2$. These are discussed in Chapter I.

2.2 Criteria for Valid Entries.

There are three basic considerations involved in making subsequent entries.

The first is that no adjacent entries may be identical. This condition is easily handled. Before an entry is placed in the sequence, a check is done to determine if it is equal to the last entry. If so, the new entry is rejected as invalid.

The second consideration is that of normality (See Chapter I). Before an entry is made in the sequence, a check must be done to determine if that entry would invalidate the normality of the sequence. This check is accomplished by retaining the largest entry made in the sequence thus far. If the largest entry is m, then a subsequent entry k must be such that $k < m + 2$, if the sequence is to remain normal. Any new entry which violates this condition is rejected.

The third and final consideration is that the sequence contain no subsequence of the form abab... with length greater than d. Clearly, we require some method of examining a sequence and determining if a particular entry may be made without creating a subsequence of the above type. We shall employ a concept defined in the next section.

2.3 Pair-Receptiveness: Definition and Examples.

The basic concept involved in subsequence checking is that of "pair-receptiveness". An entry a in the sequence is said to be "pair-receptive" to b if placing b in the sequence would increase the length of the abab... subsequence. Two examples follow to illustrate this concept.

Suppose $n = 6$. Consider the following entries in a sequence: 1 2 1 3 2 3 1.

1 is pair-receptive to all entries 2,...,6.

2 is pair-receptive to 4,5,6 but not 3. Inserting a 3 in the sequence as the next element would not increase the length of the 2323... subsequence.

In contrast, consider the entries 1 2 3 1 2 4 3 2. In this case, 2 is pair-receptive to all entries 3,...,6. Inserting a 3 would increase the 2323... subsequence to length 6.

The number of ab pairs that must be considered is n(n-1)/2. For n = 6, the pairs are:

12	23	34	45	56
13	24	35	46	
14	25	36		
15	26			
16				

Since we are considering only normal sequences, there is no necessity to count pairs ab where a > b, in virtue of the following argument.

If S is a normal admissible sequence, that is, one with no subsequence abab... of length d + 1, then S contains no subsequence

S' = b a b a b... of length d, where b > a.

Suppose S does contain a subsequence of the form S'. Since S is a normal sequence and a < b, the element a must appear at least once before the first occurrence of b. But this gives a subsequence

$S'' = a\ b\ a\ b\ a\ b\ldots$ of length $d + 1$.

This is a contradiction.

2.4 Subsequence Checking.

The method for subsequence checking differs for odd and even d.

Case 1. d odd.

In this case, if we wish to make an entry b in the sequence, we must determine if there already exists in the sequence a subsequence of the form abab... of length d for any $a < b$. If there is such a subsequence, the entry b must be rejected, since its inclusion would produce a subsequence abab... of length $d + 1$.

The method is as follows. The entry b is placed in the sequence. If a is pair-receptive to b at this point, then b will make a pair with a. The count for the pair ab is incremented by 1. If the number of ab pairs is still less than $(d+1)/2$, b was a valid entry in the sequence. If the number of ab pairs is now equal to $(d+1)/2$, then there exists a subsequence abab... of length $d + 1$, and the entry b must be rejected. Note that, if b is rejected, the count of ab pairs must be decreased by one, since the entry b is not remaining in the sequence.

Case 2. d even.

For even d, the subsequence check is as follows. When an entry is made in the sequence, a count is retained of how many ab pairs exist in subsequences of the form abab... At the point at which an entry b brings the count of ab pairs to $d/2$, it is then impossible to add the entry a to the sequence, since that entry would produce a subsequence ababa... of length $2(d/2) + 1 = d + 1$. A flag is set to indicate that a may not be entered. All subsequent entries are checked and rejected if equal to a.

Thus, the algorithm for even d requires a flag for each of the numbers $1,2,\ldots,n$, which indicates whether the entry will violate the subsequence criterion.

2.5 Obtaining Entries for the Sequence.

The algorithm produces numbers for possible entry in the sequence, as illustrated in Figure 1.

1	2	3	4	5	...	k-1	k
1	1	→ 1	→ 1	→ 1	...	1	→ 1
→ 2	2	2	2	2	...	2	2
3	→ 3	3	3	3	...	3	3
4	4	4	4	4	...	4	4
5	5	5	5	5	...	5	5
6	6	6	6	6	...	6	6

Figure 1

The following discussion applies to the search for N(5,6). It may be generalized, however, merely by ensuring that the dimensions of the table are such that:

1) number of rows = n,

2) number of columns ≥ N(d,n) + 1.

The arrows in the table represent pointers indicating the next possible number which may be entered in the sequence at that position. Initially, all pointers are set to 1. Let us assume that the entries 1 2 have already been made in the sequence. We then begin by looking at column 3. The next possible entry is indicated by the column 3 pointer, that is, 1. This possible entry is checked according to the criteria outlined above. Since we are considering N(5,6), this entry is certainly valid; so it is entered in the sequence. The pointer for column 3 is incremented by 1 to indicate that the next possible entry for that position in the sequence is 2. We then go on to column 4.

Suppose n = 4, d = 3, and that we are about to make an entry from column 8. The table of pointers follows.

1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	→ 1
→ 2	2	→ 2	2	→ 2	2	→ 2	2
3	→ 3	3	3	3	3	3	3
4	4	4	→ 4	4	4	4	4
					→		

Figure 2

This would indicate that we have made the following entries in the sequence: 1 2 1 3 1 4 1. We are attempting to add an eighth entry to the sequence. Working in column 8, we reject 1, since it appears as the last entry. We reject 2 because it would create a subsequence 1 2 1 2 (length 4). We reject 3 and 4 for the same reason as 2. Thus we have rejected all entries for this column; this rejection indicates that sequence just constructed terminates at length 7. This length is recorded for comparison against the length of the longest sequence found previously.

It now becomes necessary to "back up" in the search, change a previous entry, and proceed forward again.

2.6 Backing Up in the Search.

Backing up in the search presents the greatest single difficulty in the algorithm. When we are progressing forward and inserting valid entries in the sequence, it is easy to retain current pair counts and pair-receptiveness data. However, backing up requires an updating of this information to reflect the new pair counts and pair-receptiveness data resulting from changing a previous entry in the sequence.

This would be quite easy if the abab... subsequences were always composed of adjacent entries. Then, backing up in the sequence and changing the last b to c would only require looking at the previous element a in order to change the pair count for ab and pair-receptiveness for a. But, since b may have made a pair with several previous entries at various points in the sequence and have affected their pair-receptiveness, the problem is much more complicated.

Three methods were considered to handle "backing up" in the search. After a sequence has terminated, the pointer in the column in which no entry could be made is reset to 1. The entry pointer in the previous column is incremented by 1 and placed in the sequence. Pair counts and pair-receptiveness must now be updated before proceeding further in the search.

Method 1

We return to the first element in the existing sequence and go through the entire sequence counting pairs, and recording pair-receptiveness. When we reach the last column in which an entry has been made, the pair counts and pair-receptiveness data will reflect the current status of the sequence. This method, though simple, is very time-consuming, particularly when one is searching for long sequences.

Method 2

In order to obviate the need to rescan the entire sequence each time a "back-up" is necessary, the pair counts and pair-receptiveness data are stored as each entry is made. When the pair counts and pair-receptiveness data have been updated for a particular entry, and we are ready to move on to the next column, the status of the sequence (that is, the pair counts and pair-receptiveness data) at that point is recorded. Then, if we are required to back up in the search and place a new entry in column n, we can recall the information stored for column n-1, and proceed with testing entries for column n.

Certainly, this method requires overhead, since every advancement in the sequence requires a storing of data, and every back up requires a retrieval for the previous column (for odd d, this amounts to 96 bytes of memory per column; for even d, 104 bytes). For long sequences, however, there is a considerable gain in the speed of execution of the program. In the search for N(5,6), the increase in speed was roughly by a factor of three.

Method 3

The third method is an extension of Method 2. Since the addition of an element will increment the pair counts only for certain pairs and change the pair-receptiveness only for certain elements, only such changes are recorded. When the search backs up to a particular column and changes an entry, one retrieves the changes which that entry made in the status of the sequence, and "undoes" the changes, thus restoring the status to its former state. This avoids the necessity of storing all the status data for each column. The implementation of this method is somewhat more difficult than that of method 2. In addition, it was tested on part of the search for N(5,6), and did not produce a significant improvement in performance. Thus, the second method is the one used for the majority of production runs.

2.7 Restarting Facility

Despite the fact that the coding used in the search for

N(5,6) is reasonably efficient, the magnitude of the search is so great that it is not possible to complete it in one computer run. This necessitates incorporating some mechanism for restarting the search at the point at which it left off at the end of a previous run. It was decided at the outset that the restarting should be done by hand, rather than by the computer system. This facilitates checking on the progress of the search. It also allows one to make spot searches for maximal sequences, when an exhaustive search is unreasonable, simply by "restarting" with certain promising entries at the beginning of the sequence.

In order to restart, one needs essentially three pieces of information: 1) the last valid sequence found, 2) the pair counts and pair-receptiveness data at each entry in that sequence, and 3) the status of the pointer array immediately following generation of the last sequence.

With the current program, only the first of these three must be supplied. The program inputs this sequence, and calls an initializing routine which prepares for continuing the search.

The initializing routine makes one pass at the sequence, storing pair counts and pair-receptiveness data for each entry. At the end of the pass, it returns the current status of the sequence so that the search may continue. The third requirement, that of reproducing the pointer array, is easily handled by setting the pointer for each column at a value one greater than the corresponding entry in the sequence. The initializing routine then passes control back to the main program, and the search continues, beginning at the sequence which was read in.

2.8 Notes on the Programs.

The programs used to implement the algorithm just described are listed in Appendix C. There are separate programs for even and odd d, since the subsequence checking is different for each. Both programs are written in the FORTRAN H programming language, and were run at the Computer Centre of the University of Manitoba. Once the programs had been debugged, a load module was created and stored on disk for use in production runs. The load module requires approximately 32K of memory in which to execute.

Some comments on the code itself are in order.

Subroutine 'INIT' contains much code which is duplicated in the main program. It would have been possible, from a programming point of view, to eliminate the subroutine entirely and to set switches in the main program to indicate whether or not initialization was being performed. However, this would require checking the switches millions of times during execution, resulting in a reduction in program efficiency.

In order to reduce time spent doing I/O, the test value for sequence length (PRTLEN) was kept as large as possible. For values of n and d which are small enough to allow completion of the search in just one run, this length was set at N(d,n), resulting in the printing only of DS sequences. If it were not for the necessity of seeing the last sequence in order to restart, and if we were interested only in the value of N(d,n) and not the corresponding DS sequences, this I/O statement could be eliminated entirely, thus reducing output to the final result for N(d,n).

The resetting of the current maximum (MAXLEN) was omitted from the production runs on N(5,6). One can easily find the longest sequences and their lengths from the output listing, and thus save executing several instructions each time there is a backing up in the search.

Pair counting, as can be seen from the listing, is effected by means of an array of counters. Each possible (normal) pair is assigned a unique position in that array, using the two integers making up the pair. In order to run the program for a different value of n, this section of code (and the corresponding section in the subroutine 'INIT') must be revised. The program could have been made more general by including pair-counting code for several possible values of n. This generalization, however, would require a test on n followed by a branch to the appropriate section of pair-counting code for every potential entry in the sequence.

Storing and restoring the pair counts and pair-receptiveness data was made more efficient by equivalencing the arrays in which this data was stored to a single one-dimensional array (R). For example, this allows restoring the sequence status for cclumn 'COLN' with the following two statements:

```
        DO   10   J = 1,LENGR
   10   R(J)  = STORE(COLN,J)
```

This is much more efficient than restoring into the two-dimensional array 'RECEPT' and the one-dimensional array 'PAIR' separately. For even d, the vector (VIO) indicating which elements can not be entered in the sequence is also equivalenced to part of the large array R.

Finally, the program listing indicates that there is actually no two-dimensional array of possible entries for sequences, as is suggested by Figures 1 and 2. Since the sequence entries are simply the integers from 1 to n, the values of the "pointers" themselves may be placed in the sequence, thus avoiding a look-up in a two-dimensional table. Two-subscript table lcok-ups are very costly in computer time, despite their relatively simple form in a high-level programming language such as FORTRAN H. The two-dimensicnal form in Figures 1 and 2 was used merely to aid in comprehension of the way in which elements are selected for entry in the sequence.

APPENDIX A

For reference, the following is a listing of Davenport-Schinzel sequences for certain values of d and n.

n=3, d=3, length=5

1 2 1 3 1

1 2 3 2 1

n=3, d=4, length=8

1 2 1 3 1 3 2 3

n=3, d=5, length=10

1 2 1 2 3 2 3 1 3 1

1 2 1 3 1 3 2 3 2 1

1 2 3 2 3 1 3 1 2 1

n=3, d=6, length=14

1 2 1 2 1 3 1 3 1 3 2 3 2 3

n=3, d=7, length=16

1 2 1 2 1 2 3 2 3 2 3 1 3 1 3 1

1 2 1 2 1 3 1 3 1 3 2 3 2 3 2 1

1 2 1 2 3 2 3 2 3 1 3 1 3 1 2 1

1 2 1 3 1 3 1 3 2 3 2 3 2 1 2 1

1 2 3 2 3 2 3 1 3 1 3 1 2 1 2 1

n=3, d=8, length=20

1 2 1 2 1 2 1 3 1 3 1 3 1 3 2 3 2 3 2 3

n=3, d=9, length=22

1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 1 3 1 3 1 3 1

1 2 1 2 1 2 1 3 1 3 1 3 1 3 2 3 2 3 2 3 2 1

1 2 1 2 1 2 3 2 3 2 3 2 3 1 3 1 3 1 3 1 2 1

1 2 1 2 1 3 1 3 1 3 1 3 2 3 2 3 2 3 2 1 2 1

1 2 1 2 3 2 3 2 3 2 3 1 3 1 3 1 3 1 2 1 2 1

1 2 1 3 1 3 1 3 1 3 2 3 2 3 2 3 2 1 2 1 2 1

1 2 3 2 3 2 3 2 3 1 3 1 3 1 3 1 2 1 2 1 2 1

n=3, d=10, length=26

1 2 1 2 1 2 1 2 1 3 1 3 1 3 1 3 1 3 2 3 2 3 2 3 2 3

n=4, d=3, length=7

1 2 1 3 1 4 1

1 2 1 3 4 3 1

1 2 3 2 1 4 1

1 2 3 2 4 2 1

1 2 3 4 3 2 1

n=4, d=4, length=12

1 2 1 3 1 3 2 4 2 4 3 4

1 2 1 3 1 4 1 4 2 4 3 4

1 2 1 3 1 4 1 4 3 2 3 4

1 2 1 3 1 4 1 4 3 4 2 4

1 2 3 2 1 4 1 4 2 4 3 4

n=4, d=5, length=16

1 2 1 2 3 2 4 2 4 3 4 1 4 1 3 1

1 2 1 3 1 3 2 3 4 3 4 2 4 1 4 1

1 2 1 3 1 4 1 4 3 4 2 4 2 3 2 1

1 2 3 2 3 1 3 4 3 4 1 4 2 4 2 1

1 2 3 2 4 2 4 3 4 1 4 1 3 1 2 1

n=4, d=6, length=23

1 2 1 2 1 3 1 3 1 4 1 4 1 4 2 4 2 3 2 3 4 3 4

1 2 1 2 1 3 1 3 1 4 1 4 1 4 3 2 3 2 4 2 4 3 4

1 2 1 2 1 3 1 3 1 4 1 4 1 4 3 4 2 4 2 3 2 3 4

1 2 1 2 1 3 1 3 1 4 1 4 1 4 3 4 3 2 3 2 4 2 4

1 2 1 2 3 2 3 1 3 1 4 1 4 1 4 2 4 2 4 3 4 3 4

1 2 1 2 3 2 3 1 3 1 4 1 4 1 4 3 4 2 4 2 4 3 4

1 2 1 2 3 2 3 1 3 1 4 1 4 1 4 3 4 3 4 2 4 2 4

1 2 1 3 1 3 1 2 1 4 1 4 1 4 2 4 2 3 2 3 4 3 4

1 2 1 3 1 3 2 3 2 1 4 1 4 1 4 2 4 2 4 3 4 3 4

1 2 3 2 3 1 3 1 2 1 4 1 4 1 4 2 4 2 4 3 4 3 4

n=4, d=7, length=28 (incomplete search)

1 2 1 2 1 2 3 2 3 2 4 2 4 2 4 3 4 3 4 1 4 1 4 1 3 1 3 1

1 2 1 2 1 3 1 3 1 3 2 3 2 3 4 3 4 3 4 2 4 2 4 1 4 1 4 1

1 2 1 2 1 3 1 3 1 4 1 4 1 4 3 4 3 4 2 4 2 4 2 3 2 3 2 1

1 2 1 2 3 2 3 2 3 1 3 1 3 4 3 4 3 4 1 4 1 4 2 4 2 4 2 1

1 2 1 2 3 2 3 2 4 2 4 2 4 3 4 3 4 1 4 1 4 1 3 1 3 1 2 1

1 2 1 3 1 3 1 3 2 3 2 3 4 3 4 3 4 2 4 2 4 1 4 1 4 1 2 1

1 2 1 3 1 3 1 4 1 4 1 4 3 4 3 4 2 4 2 4 2 3 2 3 2 1 2 1

n=4, d=8, length=35 (incomplete search)

1 2 1 2 1 2 1 3 1 3 1 3 1 4 1 4 1 4 1 4 +

2 4 2 4 2 3 2 3 2 3 4 3 4 3 4

3 2 3 2 3 2 4 2 4 2 4 3 4 3 4

3 4 2 4 2 4 2 3 2 3 2 3 4 3 4

3 4 3 2 3 2 3 2 4 2 4 2 4 3 4

3 4 3 4 2 4 2 4 2 3 2 3 2 3 4

3 4 3 4 3 2 3 2 3 2 4 2 4 2 4

1 2 1 2 1 2 3 2 3 2 3 1 3 1 3 1 4 1 4 1 4 1 4 +

2 4 2 4 2 4 3 4 3 4 3 4

3 4 2 4 2 4 2 4 3 4 3 4

3 4 3 4 2 4 2 4 2 4 3 4

3 4 3 4 3 4 2 4 2 4 2 4

n=5, d=3, length=9

1 2 1 3 1 4 1 5 1

1 2 1 3 1 4 5 4 1

1 2 1 3 4 3 1 5 1

1 2 1 3 4 3 5 3 1

1 2 1 3 4 5 4 3 1

1 2 3 2 1 4 1 5 1

1 2 3 2 1 4 5 4 1

1 2 3 2 4 2 1 5 1

1 2 3 2 4 3 5 2 1

1 2 3 2 4 5 4 2 1

1 2 3 4 3 2 1 5 1

1 2 3 4 3 2 5 2 1

1 2 3 4 3 5 3 2 1

1 2 3 4 5 4 3 2 1

n=5, d=4, length=17

1 2 1 3 1 4 1 4 3 2 5 2 5 3 5 4 5

n=5, d=5, length=22

1	2	1	2	3	2	4	2	4	3	4	1	4	5	4	5	1	5	3	5	3	1
1	2	1	2	3	2	4	2	4	3	5	3	5	4	5	1	5	1	4	1	3	1
1	2	1	2	3	2	4	2	5	2	5	3	5	4	5	1	5	1	4	1	3	1
1	2	1	2	3	2	4	2	5	2	5	4	3	4	5	1	5	1	4	1	3	1
1	2	1	2	3	2	4	2	5	2	5	4	5	3	5	1	5	1	3	1	4	1
1	2	1	2	3	2	4	2	5	2	5	4	5	3	5	1	5	1	3	4	3	1
1	2	1	2	3	2	4	2	5	2	5	4	5	3	5	1	5	1	4	1	3	1
1	2	1	2	3	2	4	2	5	2	5	4	5	3	5	3	4	1	4	1	3	1
1	2	1	2	3	4	3	2	5	2	5	3	5	4	5	1	5	1	4	1	3	1
1	2	1	2	3	4	3	4	2	4	5	4	5	2	5	3	5	1	5	1	3	1
1	2	1	3	1	3	2	3	4	3	4	2	5	2	5	4	5	1	5	1	4	1
1	2	1	3	1	3	2	3	4	3	5	3	5	2	5	4	5	1	5	1	4	1
1	2	1	3	1	3	2	3	4	3	5	3	5	4	2	4	5	1	5	1	4	1
1	2	1	3	1	3	2	3	4	3	5	3	5	4	5	2	5	1	5	1	4	1
1	2	1	3	1	3	2	3	4	3	5	3	5	4	5	2	5	2	4	1	4	1
1	2	1	3	1	3	2	4	2	3	5	3	5	2	5	4	5	1	5	1	4	1
1	2	1	3	1	3	2	4	2	4	3	4	5	4	5	3	5	2	5	1	5	1
1	2	1	3	1	3	4	3	2	3	5	3	5	2	5	4	5	1	5	1	4	1
1	2	1	3	1	4	1	4	2	4	3	4	5	4	5	3	5	2	5	1	5	1
1	2	1	3	1	4	1	4	3	2	3	4	5	4	5	3	5	2	5	1	5	1
1	2	1	3	1	4	1	4	3	4	2	4	2	3	5	3	5	2	5	1	5	1
1	2	1	3	1	4	1	4	3	4	2	4	5	4	5	2	3	2	5	1	5	1
1	2	1	3	1	4	1	4	3	4	2	4	5	4	5	2	5	3	5	1	5	1
1	2	1	3	1	4	1	4	3	4	2	4	5	4	5	2	5	3	5	3	2	1

1	2	1	3	1	4	1	4	3	4	2	4	5	4	5	3	5	2	5	1	5	1
1	2	1	3	1	4	1	4	3	5	3	5	4	5	2	5	2	4	2	3	2	1
1	2	1	3	1	4	1	5	1	5	3	5	4	5	2	5	2	4	2	3	2	1
1	2	1	3	1	4	1	5	1	5	4	3	4	5	2	5	2	4	2	3	2	1
1	2	1	3	1	4	1	5	1	5	4	5	3	5	2	5	2	3	2	4	2	1
1	2	1	3	1	4	1	5	1	5	4	5	3	5	2	5	2	3	4	3	2	1
1	2	1	3	1	4	1	5	1	5	4	5	3	5	2	5	2	4	2	3	2	1
1	2	1	3	1	4	1	5	1	5	4	5	3	5	3	4	2	4	2	3	2	1
1	2	1	3	4	3	1	5	1	5	3	5	4	5	2	5	2	4	2	3	2	1
1	2	1	3	4	3	4	1	4	2	4	5	4	5	2	5	1	5	3	5	3	1
1	2	1	3	4	3	4	1	4	5	4	5	1	5	3	5	2	5	2	3	2	1
1	2	3	2	1	4	1	4	2	4	3	4	5	4	5	3	5	2	5	1	5	1
1	2	3	2	3	1	3	4	3	4	1	4	2	4	5	4	5	2	5	1	5	1
1	2	3	2	3	1	3	4	3	4	1	5	1	5	4	5	2	5	2	4	2	1
1	2	3	2	3	1	3	4	3	5	3	5	1	5	4	5	2	5	2	4	2	1
1	2	3	2	3	1	3	4	3	5	3	5	4	1	4	5	2	5	2	4	2	1
1	2	3	2	3	1	3	4	3	5	3	5	4	5	1	5	1	4	2	4	2	1
1	2	3	2	3	1	3	4	3	5	3	5	4	5	1	5	2	5	2	1	4	1
1	2	3	2	3	1	3	4	3	5	3	5	4	5	1	5	2	5	2	4	2	1
1	2	3	2	3	1	4	1	3	5	3	5	1	5	4	5	2	5	2	4	2	1
1	2	3	2	3	1	4	1	4	3	4	5	4	5	3	5	1	5	2	5	2	1
1	2	3	2	3	4	3	1	3	5	3	5	1	5	4	5	2	5	2	4	2	1
1	2	3	2	4	2	4	1	4	3	4	5	4	5	3	5	1	5	2	5	2	1
1	2	3	2	4	2	4	3	1	3	4	5	4	5	3	5	1	5	2	5	2	1

1 2 3 2 4 2 4 3 4 1 4 1 3 5 3 5 1 5 2 5 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 3 1 5 2 5 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 5 3 5 2 5 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 5 3 5 3 1 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 3 5 1 5 2 5 2 1

1 2 3 2 4 2 4 3 5 3 5 4 5 1 5 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 5 3 5 4 5 1 5 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 5 4 3 4 5 1 5 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 1 3 1 4 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 1 3 4 3 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 3 4 1 4 1 3 1 2 1

1 2 3 4 3 2 5 2 5 3 5 4 5 1 5 1 4 1 3 1 2 1

1 2 3 4 3 4 2 4 1 4 5 4 5 1 5 2 5 3 5 3 2 1

1 2 3 4 3 4 2 4 5 4 5 2 5 3 5 1 5 1 3 1 2 1

n=5, d=6, length=34

1 2 1 2 1 3 1 3 1 4 1 4 1 +

4 3 4 2 4 2 3 2 5 2 5 2 5 3 5 3 5 4 5 4 5

n=6, d=3, length=11

1	2	1	3	1	4	1	5	1	6	1
1	2	1	3	1	4	1	5	6	5	1
1	2	1	3	1	4	5	4	1	6	1
1	2	1	3	1	4	5	4	6	4	1
1	2	1	3	1	4	5	6	5	4	1
1	2	1	3	4	3	1	5	1	6	1
1	2	1	3	4	3	1	5	6	5	1
1	2	1	3	4	3	5	3	1	6	1
1	2	1	3	4	3	5	3	6	3	1
1	2	1	3	4	3	5	6	5	3	1
1	2	1	3	4	5	4	3	1	6	1
1	2	1	3	4	5	4	3	6	3	1
1	2	1	3	4	5	4	6	4	3	1
1	2	1	3	4	5	6	5	4	3	1
1	2	3	2	1	4	1	5	1	6	1
1	2	3	2	1	4	1	5	6	5	1
1	2	3	2	1	4	5	4	1	6	1
1	2	3	2	1	4	5	4	6	4	1
1	2	3	2	1	4	5	6	5	4	1
1	2	3	2	4	2	1	5	1	6	1
1	2	3	2	4	2	1	5	6	5	1
1	2	3	2	4	2	5	2	1	6	1
1	2	3	2	4	2	5	2	6	2	1
1	2	3	2	4	2	5	6	5	2	1

1 2 3 2 4 5 4 2 1 6 1

1 2 3 2 4 5 4 2 6 2 1

1 2 3 2 4 5 4 6 4 2 1

1 2 3 2 4 5 6 5 4 2 1

1 2 3 4 3 2 1 5 1 6 1

1 2 3 4 3 2 1 5 6 5 1

1 2 3 4 3 2 5 2 1 6 1

1 2 3 4 3 2 5 2 6 2 1

1 2 3 4 3 2 4 5 4 2 1

1 2 3 4 3 5 3 2 1 6 1

1 2 3 4 3 5 3 2 6 2 1

1 2 3 4 3 5 3 6 3 2 1

1 2 3 4 3 5 6 5 3 2 1

1 2 3 4 5 4 3 2 1 6 1

1 2 3 4 5 4 3 6 3 2 1

1 2 3 4 5 4 6 4 3 2 1

1 2 3 4 5 6 5 4 3 2 1

n=6, d=4, length=22

1 2 1 3 1 4 1 5 1 5 4 3 2 6 2 6 3 6 4 6 5 6

n=6, d=5, length=29

1 2 1 2 3 2 4 2 5 2 5 4 3 6 3 6 4 6 5 6 1 6 1 5 1 4 1 3 1

1 2 1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 6 5 6 1 6 3 6 4 6 4 3 1

1 2 1 2 3 2 4 2 5 2 5 4 5 3 5 6 5 6 3 6 4 6 1 6 1 4 1 3 1

1 2 1 2 3 2 4 2 5 2 6 2 6 5 6 4 6 3 6 3 4 5 1 5 1 4 1 3 1

1 2 1 2 3 4 5 4 5 3 5 2 5 6 5 6 2 6 3 6 4 6 1 6 1 4 1 3 1

1 2 1 3 1 3 2 3 4 3 5 3 5 4 2 6 2 6 4 6 5 6 1 6 1 5 1 4 1

1 2 1 3 1 3 2 3 4 3 5 3 5 4 5 2 5 6 5 6 2 6 4 6 1 6 1 4 1

1 2 1 3 1 3 2 3 4 3 5 3 6 3 6 5 6 4 6 2 6 2 4 5 1 5 1 4 1

1 2 1 3 1 3 2 4 5 4 5 2 5 3 5 6 5 6 3 6 2 6 4 6 1 6 1 4 1

1 2 1 3 1 4 1 4 3 2 5 2 5 3 5 4 5 6 5 6 4 6 3 6 2 6 1 6 1

1 2 1 3 1 4 1 4 3 4 2 4 5 4 5 2 3 6 3 6 2 6 5 6 1 6 1 5 1

1 2 1 3 1 4 1 4 3 4 2 4 5 4 5 2 5 3 5 6 5 6 3 6 2 6 1 6 1

1 2 1 3 1 4 1 4 3 4 2 4 5 4 6 4 6 5 6 2 6 3 6 3 2 5 1 5 1

1 2 1 3 1 4 1 5 1 5 4 3 6 3 6 4 6 5 6 2 6 2 5 2 4 2 3 2 1

1 2 1 3 1 4 1 5 1 5 4 5 3 5 2 5 2 3 4 6 4 6 3 6 2 6 1 6 1

1 2 1 3 1 4 1 5 1 5 4 5 3 5 2 5 6 5 6 2 6 3 6 4 6 4 3 2 1

1 2 1 3 1 4 1 5 1 5 4 5 3 5 6 5 6 3 6 4 6 2 6 2 4 2 3 2 1

1 2 1 3 1 4 1 5 1 6 1 6 5 6 4 6 3 6 3 4 5 2 5 2 4 2 3 2 1

1 2 1 3 4 5 4 5 3 5 1 5 6 5 6 1 6 3 6 4 6 2 6 2 4 2 3 2 1

1 2 3 2 3 1 3 4 3 5 3 5 4 1 6 1 6 4 6 5 6 2 6 2 5 2 4 2 1

1 2 3 2 3 1 3 4 3 5 3 5 4 5 1 5 2 5 6 5 6 2 6 1 6 4 6 4 1

1 2 3 2 3 1 3 4 3 5 3 5 4 5 1 5 6 5 6 1 6 4 6 2 6 2 4 2 1

1 2 3 2 3 1 3 4 3 5 3 6 3 6 5 6 4 6 1 6 1 4 5 2 5 2 4 2 1

1 2 3 2 3 1 4 5 4 5 1 5 3 5 6 5 6 3 6 1 6 4 6 2 6 2 4 2 1

1 2 3 2 4 2 4 3 1 5 1 5 3 5 4 5 6 5 6 4 6 3 6 1 6 2 6 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 3 6 3 6 1 6 5 6 2 6 2 5 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 5 1 5 3 5 6 5 6 3 6 1 6 2 6 2 1

1 2 3 2 4 2 4 3 4 1 4 5 4 6 4 6 5 6 1 6 3 6 3 1 5 2 5 2 1

1 2 3 2 4 2 5 2 5 4 3 6 3 6 4 6 5 6 1 6 1 5 1 4 1 3 1 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 1 3 4 6 4 6 3 6 1 6 2 6 2 1

1 2 3 2 4 2 5 2 5 4 5 3 5 1 5 6 5 6 1 6 3 6 4 6 4 3 1 2 1

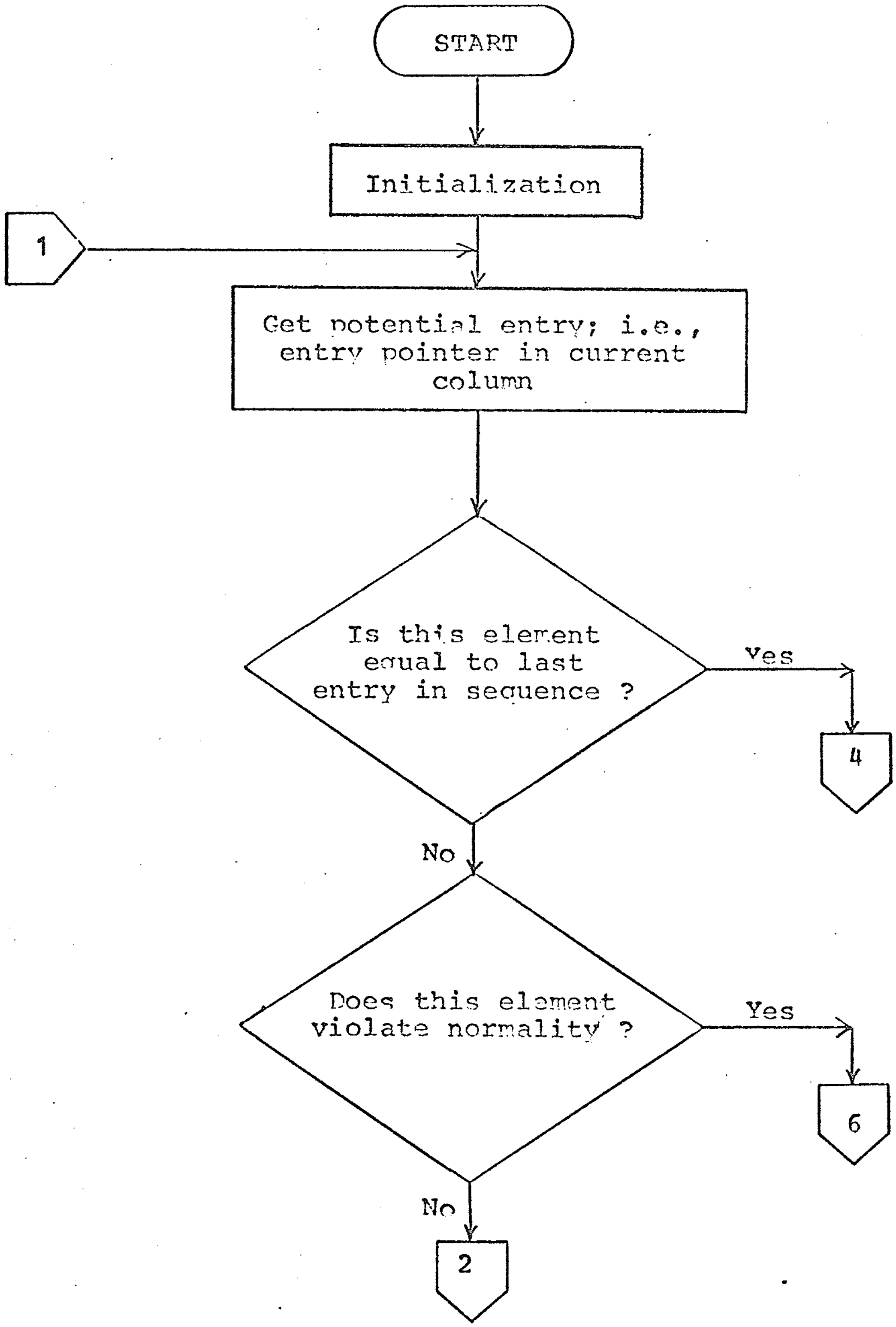
1 2 3 2 4 2 5 2 5 4 5 3 5 6 5 6 3 6 4 6 1 6 1 4 1 3 1 2 1

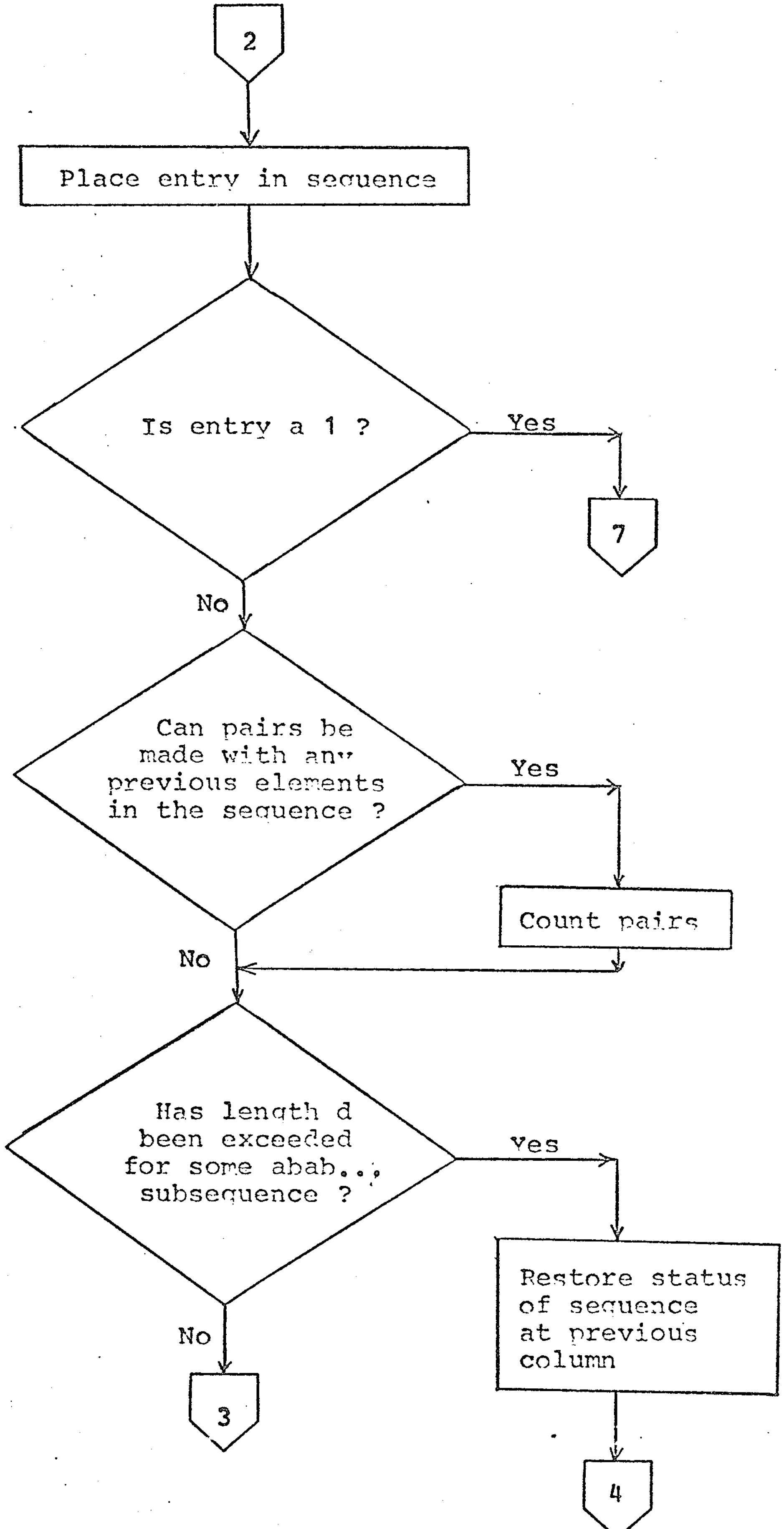
1 2 3 2 4 2 5 2 6 2 6 5 6 4 6 3 6 3 4 5 1 5 1 4 1 3 1 2 1

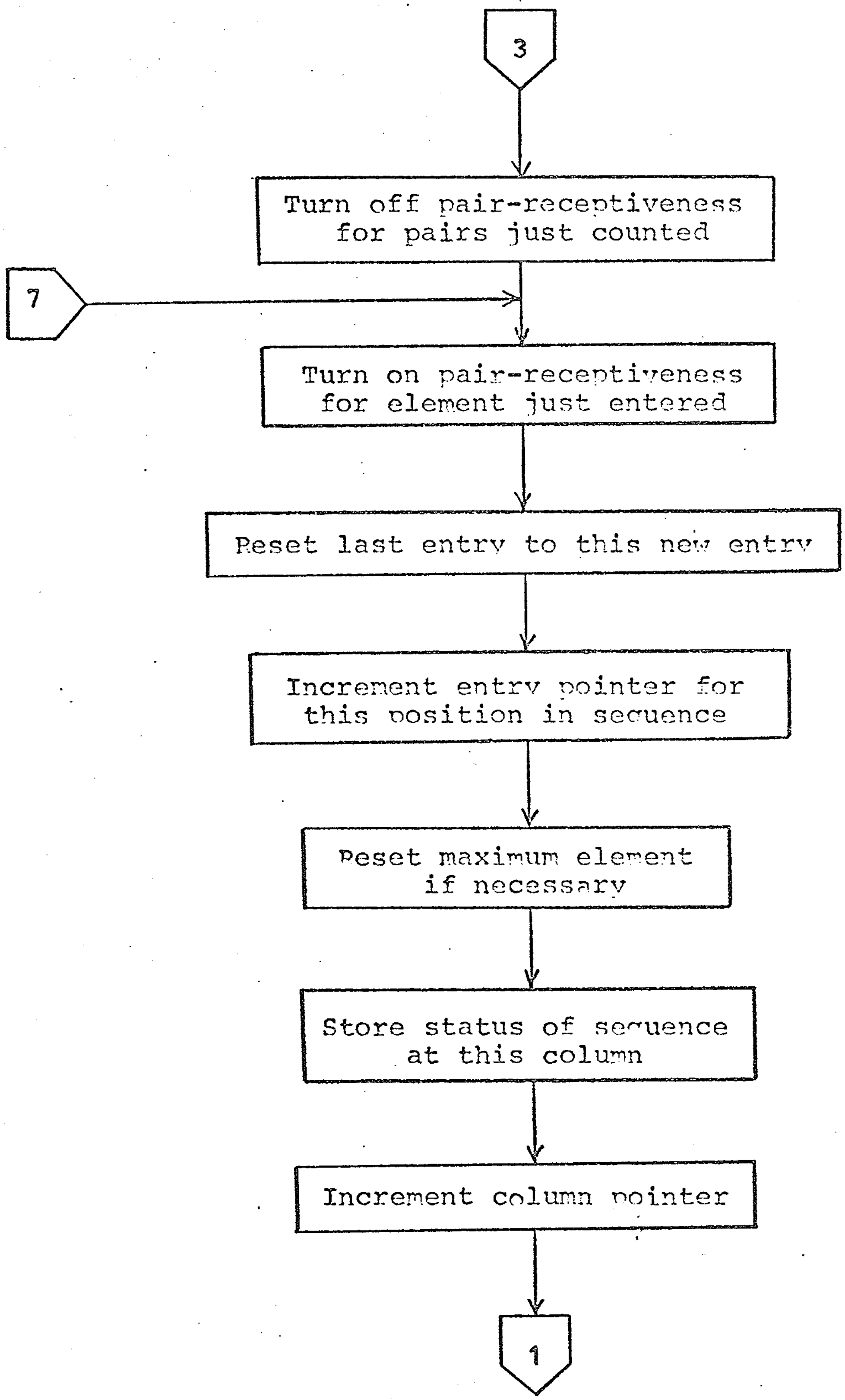
1 2 3 4 3 4 2 4 1 4 5 4 5 1 5 2 5 3 5 6 5 6 3 6 2 6 1 6 1

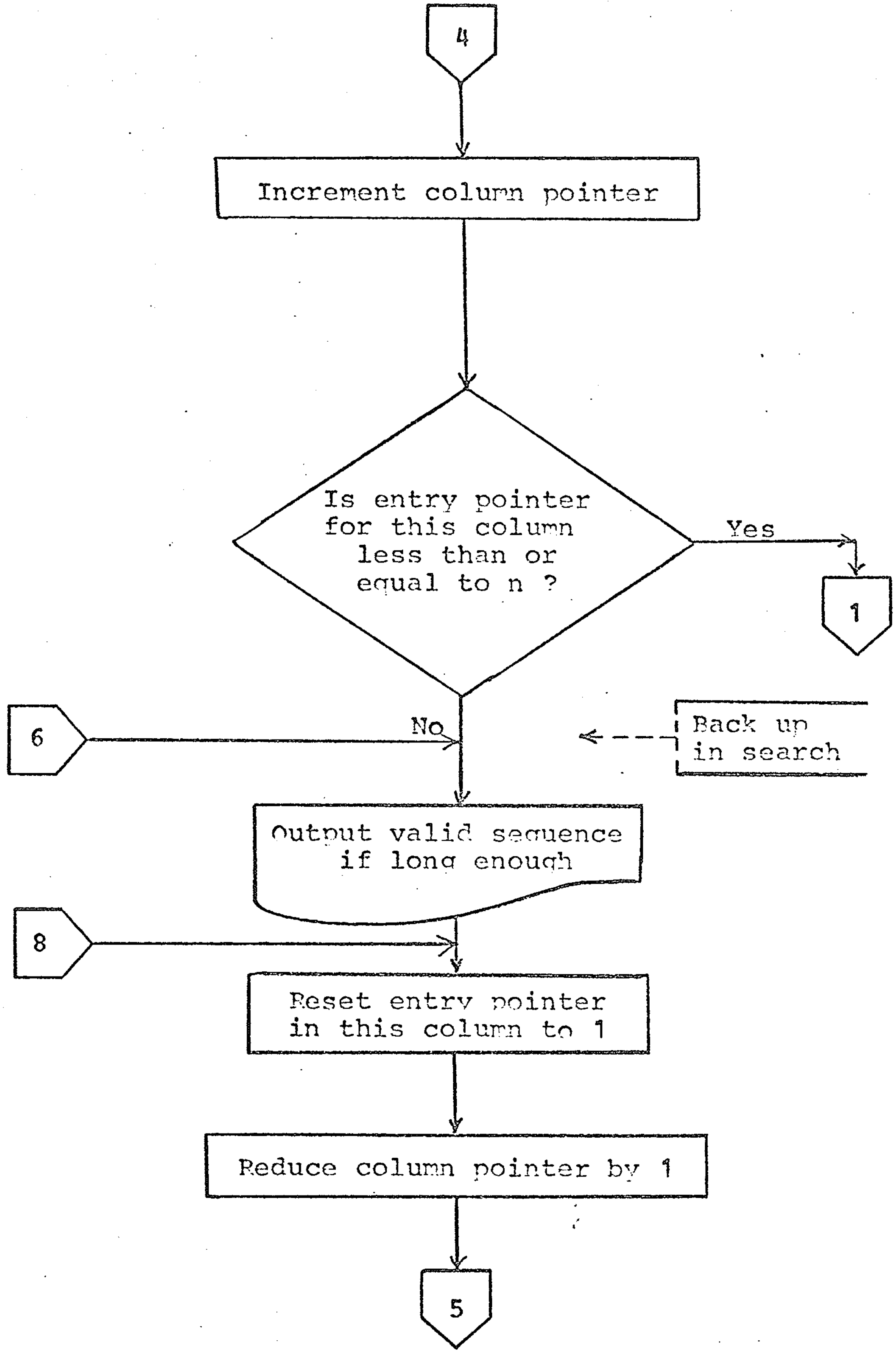
1 2 3 4 5 4 5 3 5 2 5 6 5 6 2 6 3 6 4 6 1 6 1 4 1 3 1 2 1

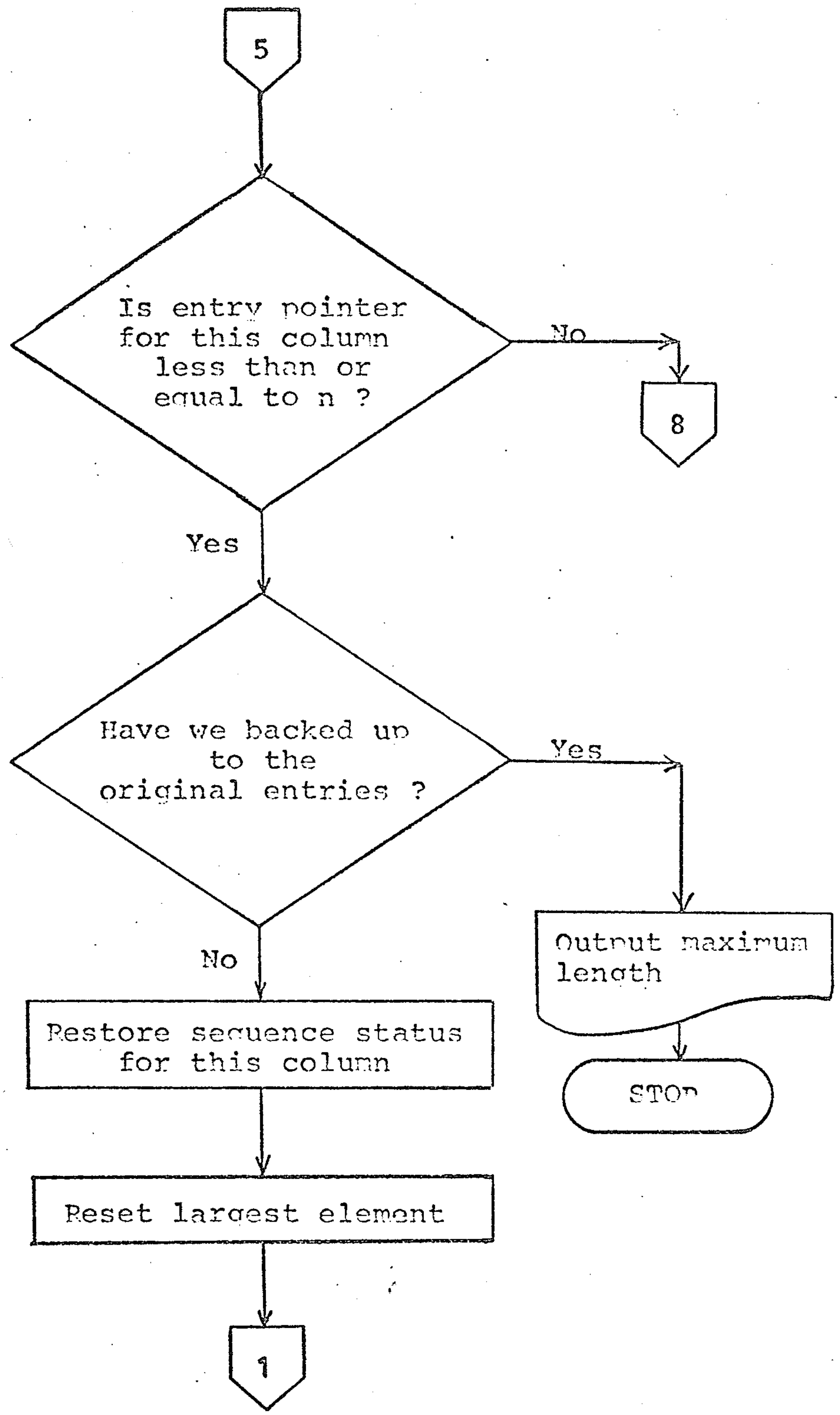
General Flowchart of Algorithm




2
Place entry in sequence
Is entry a 1 ?
Yes
7
No
Can pairs be made with any previous elements in the sequence ?
Yes
Count pairs
No
Has length d been exceeded for some abab... subsequence ?
Yes
Restore status of sequence at previous column
No
3
4


3
Turn off pair-receptiveness for pairs just counted
7
Turn on pair-receptiveness for element just entered
Reset last entry to this new entry
Increment entry pointer for this position in sequence
Reset maximum element if necessary
Store status of sequence at this column
Increment column pointer
1


4
Increment column pointer
Is entry pointer for this column less than or equal to n ?
Yes
1
No
6
Back up in search
Output valid sequence if long enough
8
Reset entry pointer in this column to 1
Reduce column pointer by 1
5


5
Is entry pointer for this column less than or equal to n ?
No
8
Yes
Have we backed up to the original entries ?
Yes
Output maximum length
STOP
No
Restore sequence status for this column
Reset largest element
1

```
C****************************************************************************
C****************************************************************************
C  N = 6 ,    D EVEN.
C****************************************************************************
      IMPLICIT INTEGER (A-Z)
      INTEGER SEQ(50),PTR(50),PAIR(15),R(26),STORE(50,26),CURRNT(26)
      LOGICAL*1 RECEPT(6,6),REC,VIO(5)
      EQUIVALENCE (RECEPT(1,1),R(1)),(PAIR(1),R(10)),(VIO(1),R(25))
      DATA PTR,MAXLEN,MAXELT,LINIT/50*1,3*2/
C
C**************************************************
C  READ IN THE SEQUENCE AT WHICH THE SEARCH IS TO BEGIN
C**************************************************
C
      READ1020,COL,(SEQ(J),J=1,COL)
C
C*********************************************************************
C  READ IN NUMBER OF SYMBOLS, MAXIMUM LENGTH OF ABAB... SUBSEQUENCE,
C  AND MINIMUM LENGTH OF SEQUENCE TO BE PRINTED.
C  OUTPUT THESE AND ANY INITIAL ENTRIES IN THE SEQUENCE
C*********************************************************************
C
      READ1000,NOSYM,D,PRTLEN
      PRINT1005,NOSYM,D,PRTLEN
C
C*****************
C  INITIALIZATION
C*****************
C
      LENGR=26
      DHALF=D/2
      LINIT1=LINIT+1
      NOSYM1=NOSYM-1
      COL1=COL-1
C
C*************************
C  ININTIALIZE FOR RESTART
C*************************
C
      CALL INIT(SEQ,COL,STORE,NOSYM,DHALF,PTR,LINIT1,CURENT,LENGR)
C
C***********************************
C  RESTORE CURRENT STATUS OF SEQUENCE
C***********************************
C
      DO 110 J=1,LENGR
110   R(J)=CURRNT(J)
C
C*****************************
C  RECORD LAST ENTRY IN SEQUENCE
C*****************************
C
20    LASTEL=SEQ(COL1)
C
C
C
```

```
C
C**********************************
C  GET ENTRY FROM ARRAY OF SYMBOLS
C**********************************
C
25    ELT=PTR(COL)
C
C*********************************************
C  CHECK IF ENTRY IS EQUAL TO LAST ENTRY MADE
C*********************************************
C
      IF(ELT.EQ.LASTEL)GO TO 30
C
C**********************************************
C  CHECK THAT ENTRY DOES NOT VIOLATE NORMALITY
C**********************************************
C
      IF(ELT.GT.MAXELT+1)GO TO 200
C
C*****************************
C  PLACE ENTRY IN THE SEQUENCE
C*****************************
C
      SEQ(COL)=ELT
C
C****************************************************************************
C  CHECK THAT ENTRY DOES NOT EXTEND AN ABAB... SUBSEQUENCE TO LENGTH D+1
C****************************************************************************
C
145   IF(VIO(ELT))GO TO 30
C
C********************************************************
C  IF THE ENTRY IS A 1, THERE IS NO NEED TO COUNT PAIRS
C********************************************************
C
      IF(ELT.EQ.1)GO TO 150
C
C******************************************************************
C  COUNT PAIRS FOR ANY ELEMENTS WHICH ARE PAIR-RECEPTIVE TO 'ELT'
C******************************************************************
C
      ELT1=ELT-1
      DO 170 K=1,ELT1
      REC=RECEPT(K,ELT)
      IF(.NOT.REC)GO TO 170
C
C**************************************
C  K IS PAIR RECEPTIVE.  PAIR IS (K,ELT)
C  PAIR ALLOCATION FOR N=6.  15 PAIRS
C  12=2  23=7  34=11  45=14  56=1
C  13=3  24=8  35=12  46=15
C  14=4  25=9  36=13
C  15=5  26=10
C  16=6
C**************************************
C
```

```
      GO TO (50,60,70,80),K
      PAIRNO=1
      GO TO 100
50    PAIRNO=ELT
      GO TO 100
60    PAIRNO=ELT+4
      GO TO 100
70    PAIRNO=ELT+7
      GO TO 100
80    PAIRNO=ELT+9
C
C***************************************
C  INCREMENT PAIR COUNT FOR VALID PAIR
C***************************************
C
100   PAIR(PAIRNO)=PAIR(PAIRNO)+1
C
C**************************************************************
C  SET FLAG IF  K  CAN NO LONGER BE ENTERED IN THE SEQUENCE
C**************************************************************
C
      IF(PAIR(PAIRNO).EQ.CHALF)VIO(K)=.TRUE.
C
C***************************************************************************
C  TURN OFF PAIR-RECEPTIVENESS FOR PAIR JUST COUNTED AND GET NEXT ELEMENT
C***************************************************************************
C
      RECEPT(K,ELT)=.FALSE.
170   CONTINUE
C
C***********************************************************
C  THE CURRENT ENTRY IS VALID;
C  MAKE IT PAIR-RECEPTIVE TO ALL ELEMENTS GREATER THAN IT
C***********************************************************
C
150   ELTP1=ELT+1
      IF(ELT.EQ.NOSYM)ELTP1=NOSYM
      DO 160 L=ELTP1,NOSYM
160   RECEPT(ELT,L)=.TRUE.
C
C********************************************************
C  RESET LAST ENTRY;  INCREMENT POINTER IN THIS COLUMN
C********************************************************
C
165   LASTEL=ELT
      PTR(COL)=ELT+1
C
C**********************************************************
C  RESET MAXIMUM IF ENTRY IS GREATER THAN CURRENT MAXIMUM
C**********************************************************
C
      IF(ELT.GT.MAXELT)MAXELT=ELT
C
C******************************************
C  STORE CURRENT STATUS OF THE SEQUENCE
C******************************************
```

```
C
      DO 280 J=1,LENGR
280   STORE(COL,J)=R(J)
C
C*****************************************************
C  INCREMENT COLUMN COUNTER AND GO ON TO NEXT COLUMN
C*****************************************************
C
      COL=COL+1
      GO TO 25
C
C**************************************************************************
C  ENTRY WAS INVALID;  INCREMENT POINTER IN THIS COLUMN AND TRY NEXT ELEMENT,
C  UNLESS THIS COLUMN IS EXHAUSTED
C**************************************************************************
C
30    PTR(COL)=ELT+1
      IF(PTR(COL).LE.NCSYM)GO TO 25
C
C*******************************************************************
C  BACK UP ONE COLUMN;
C  PRINT SEQUENCE IF LENGTH IS EQUAL TO OR GREATER THAN THE MIMIMUM;
C  RESET LENGTH OF LONGEST SEQUENCE IF NECESSARY
C*******************************************************************
C
200   LEN=COL-1
      IF(LEN.GE.PRTLEN)PRINT2000,LEN,(SEQ(J),J=1,LEN)
      IF(LEN.GT.MAXLEN)MAXLEN=LEN
C
C*********************************************
C  RESET ENTRY POINTER FOR BACK UP COLUMN;
C  IF PREVIOUS COLUMN EXHAUSTED, BACK UP AGAIN
C*********************************************
C
15    PTR(COL)=1
      COL=COL-1
      IF(PTR(COL).GT.NCSYM)GO TO 15
C
C*****************************************************************
C  IF WE HAVE BACKED UP TO THE ORIGINAL ENTRIES, WE ARE FINISHED
C*****************************************************************
C
      IF(COL.EQ.LIMIT)GO TO 220
C
C************************************************
C  RESTORE STATUS OF SEQUENCE FOR PREVIOUS COLUMN
C************************************************
C
      COL1=COL-1
      DO 285 J=1,LENGR
285   R(J)=STORE(COL1,J)
C
C**********************************
C  RESET LARGEST ELEMENT FOR NORMALITY
C**********************************
C
```

```
      MAXELT=2.
      DO 205 I=1,COL1
205   IF(SEQ(I).GT.MAXELT)MAXELT=SEQ(I)
      GO TO 20
C
C*************************************************************************
C  SINCE WE HAVE BACKED UP TO THE ORIGINAL ENTRIES, WE ARE DONE;
C  OUTPUT  N, D, AND N(D,N)
C*************************************************************************
C
220   PRINT2005,NOSYM,D,MAXLEN
      STOP
C
C*************************************
1000  FORMAT(3I3)
1005  FORMAT('1'////' NUMBER OF SYMBOLS',I5,5X,'MAXIMUM LENGTH OF ABAB..
     X. SUBSEQUENCE IS',I5/' SEQUENCES OF LENGTH',I5,'  OR LONGER ARE PR
     XINTED'////'            LENGTH',25X,'SEQUENCE')
1020  FORMAT(40I2)
2000  FORMAT(' ',10X,I5,10X,50I2)
2005  FORMAT(/////' MAXIMUM LENGTH FOR N=',I5,3X,'D=',I5,5X,'IS',I6)
      END
C
C*****************************************************************************************
C
      SUBROUTINE INIT(SEQ,COL,STORE,NOSYM,DHALF,PTR,LINIT1,CURRNT,LENGR)
      IMPLICIT INTEGER(A-Z)
      INTEGER SEQ(50),PTR(50),STORE(50,26),PAIR(15),R(26),CURRNT(26)
      LOGICAL*1 RECEPT(6,6),REC,VID(6)
      EQUIVALENCE (RECEPT(1,1),R(1)),(PAIR(1),R(10)),(VID(1),R(25))
C
C*****************
C  INITIALIZATION
C*****************
C
      DO 10 I=1,LENGR
10    R(I)=0
C
C*****************************************************************************
C  SET POINTERS.  EACH POINTER IS ONE GREATER THAN ENTRY IN SEQUENCE
C  IN CORRESPONDING POSITION
C*****************************************************************************
C
      COL1=COL-1
      DO 20 J=2,COL1
20    PTR(J)=SEQ(J)+1
C
C*************************************************************
C  MAKE 1 PAIR-RECEPTIVE TO ALL SUBSEQUENT ENTRIES
C*************************************************************
C
      DO 30 L=2,NOSYM
30    RECEPT(1,L)=.TRUE.
C
C
C
```

```
C*********************************************************************
C  SCAN SEQUENCE TO RECORD PAIR COUNTS AND PAIR-RECEPTIVENESS DATA
C*********************************************************************
C
      DO 40 COLN=2,COL1
      ELT=SEQ(COLN)
      IF(ELT.EQ.1)GO TO 150
      ELT1=ELT-1
      DO 170 K=1,ELT1
      REC=RECEPT(K,ELT)
      IF(.NOT.REC)GO TO 170
      GO TO (50,60,70,80),K
      PAIRNO=1
      GO TO 100
50    PAIRNO=ELT
      GO TO 100
60    PAIRNO=ELT+4
      GO TO 100
70    PAIRNO=ELT+7
      GO TO 100
80    PAIRNO=ELT+9
100   PAIR(PAIRNO)=PAIR(PAIRNO)+1
      IF(PAIR(PAIRNO).EQ.CHALF)VIC(K)=.TRUE.
      RECEPT(K,ELT)=.FALSE.
170   CONTINUE
150   ELTP1=ELT+ 1
      IF(ELT.EQ.NOSYM)ELTP1=NOSYM
      DO 160 L=ELTP1,NOSYM
160   RECEPT(ELT,L)=.TRUE.
C
C*********************************************************
C  STORE STATUS AT THIS COLUMN AND GO ON TO NEXT COLUMN
C*********************************************************
C
      DO 165 J=1,LENGR
165   STORE(COLN,J)=R(J)
40    CONTINUE
C
C*******************************************
C  PASS BACK CURRENT STATUS OF THE SEQUENCE
C*******************************************
C
      DO 180 J=1,LENGR
180   CURRNT(J)=R(J)
      RETURN
      END
C
C*************************************************************************
C*************************************************************************
```

```
C*************************************************************************************
C*************************************************************************************
C  N = 6 ,    D ODD.
C*************************************************************************************
C
      IMPLICIT INTEGER (A-Z)
      INTEGER SEQ(50),PTR(50),PAIR(15),R(24),STORE(50,24),CURRNT(24),
      LOGICAL*1 RECEPT(6,6),REC
      EQUIVALENCE (RECEPT(1,1),R(1)),(PAIR(1),R(10))
      DATA PTR,MAXLEN,MAXELT,LINIT/50*1,3*2/
C
C**********************************************************
C  READ IN THE SEQUENCE AT WHICH THE SEARCH IS TO BEGIN
C**********************************************************
C
      READ1010,COL,(SEQ(J),J=1,COL)
C
C*********************************************************************
C  READ IN NUMBER OF SYMBOLS, MAXIMUM LENGTH OF ABAB... SUBSEQUENCE,
C  AND MINIMUM LENGTH OF SEQUENCE TO BE PRINTED.
C  OUTPUT THESE AND ANY INITIAL ENTRIES IN THE SEQUENCE
C*********************************************************************
C
      READ1000,NOSYM,D,PRTLEN
      PRINT1005,NOSYM,D,PRTLEN,(SEQ(J),J=1,LINIT)
C
C*****************
C  INITIALIZATION
C*****************
C
      LENGR=24
      COL1=COL-1
      DHALF=D/2
      DHALF1=DHALF+1
      LINIT1=LINIT+1
      NOSYM1=NOSYM-1
C
C**************************
C  INITIALIZE FOR RESTART
C**************************
C
      CALL INIT(SEQ,COL,NOSYM,STORE,PTR,LINIT1,CURRNT,LENGR)
C
C*************************************
C  RESTORE CURRENT STATUS OF SEQUENCE
C*************************************
C
      DO 130 J=1,LENGR
130   R(J)=CURRNT(J)
C
C*******************************
C  RECORD LAST ENTRY IN SEQUENCE
C*******************************
C
20    LASTEL=SEQ(COL1)
C
```

```
C*********************************
C  GET ENTRY FROM ARRAY OF SYMBOLS
C*********************************
C
25     ELT=PTR(COL)
C
C**************************************************
C  CHECK IF ENTRY IS EQUAL TO THE LAST ENTRY MADE
C**************************************************
C
       IF(ELT.EQ.LASTEL)GO TO 30
C
C************************************************
C  CHECK THAT ELEMENT DOES NOT VIOLATE NORMALITY
C************************************************
C
       IF(ELT.GT.MAXELT+1)GO TO 200
C
C*****************************
C  PLACE ENTRY IN THE SEQUENCE
C*****************************
C
       SEQ(COL)=ELT
C
C**********************************************************
C  IF THE ENTRY IS A 1, THERE IS NO NEED TO COUNT PAIRS
C**********************************************************
C
145    IF(ELT.EQ.1)GO TO 150
C
C******************************************************************
C  COUNT PAIRS FOR ANY ELEMENTS WHICH ARE PAIR-RECEPTIVE TO 'ELT'
C******************************************************************
C
       ELT1=ELT-1
       DO 170 K=1,ELT1
       REC=RECEPT(K,ELT)
       IF(.NOT.REC)GO TO 170
C
C*************************************
C  K IS PAIR RECEPTIVE.  PAIR IS (K,ELT)
C  PAIR ALLOCATION FOR N=6.   15 PAIRS
C  12=2  23=7  34=11  45=14  56=1
C  13=3  24=8  35=12  46=15
C  14=4  25=9  36=13
C  15=5  26=10
C  16=6
C*************************************
C
      GO TO (50,60,70,80),K
      PAIRNO=1
      GO TO 100
50    PAIRNO=ELT
      GO TO 100
60    PAIRNO=ELT+4
      GO TO 100
```

```
 70    PAIRNO=ELT+7
       GO TO 100
 80    PAIRNO=ELT+9
 C
 C*************************************
 C  INCREMENT PAIR COUNT FOR VALID PAIR
 C*************************************
 C
 100   PAIR(PAIRNO)=PAIR(PAIRNO)+1
       PRNO=PAIR(PAIRNO)
 C
 C*********************************************
 C  CHECK IF SUBSEQUENCE LENGTH IS GREATER THAN D
 C*********************************************
 C
       IF(PRNO.EQ.DHALF1)GO TO 45
 C
 C*************************************************************************
 C  TURN OFF PAIR-RECEPTIVENESS FOR PAIR JUST COUNTED AND GET NEXT ELEMENT
 C*************************************************************************
 C
       RECEPT(K,ELT)=.FALSE.
 170   CONTINUE
 C
 C*********************************************************
 C  THE CURRENT ENTRY IS VALID;
 C  MAKE IT PAIR-RECEPTIVE TO ALL ELEMENTS GREATER THAN IT
 C*********************************************************
 C
 150   ELTP1=ELT+1
       IF(ELT.EQ.NOSYM)ELTP1=NOSYM
       DO 160 L=ELTP1,NOSYM
 160   RECEPT(ELT,L)=.TRUE.
 C
 C****************************************************
 C  RESET LAST ENTRY, INCREMENT POINTER IN THIS COLUMN
 C****************************************************
 C
 165   LASTEL=ELT
       PTR(CCL)=ELT+1
 C
 C*********************************************************
 C  RESET MAXIMUM IF ENTRY IS GREATER THAN CURRENT MAXIMUM
 C*********************************************************
 C
       IF(ELT.GT.MAXELT)MAXELT=ELT
 C
 C*************************************
 C  STORE CURRENT STATUS OF THE SEQUENCE
 C*************************************
 C
       DO 175 J=1,LENGR
 175   STORE(CCL,J)=R(J)
 C
 C
 C
```

```
C*************************************************************
C  INCREMENT COLUMN COUNTER AND GO ON TO NEXT COLUMN
C*************************************************************
C
      COL=COL+1
      GO TO 25
C
C*****************************************************************************
C  ENTRY WAS INVALID BECAUSE OF PAIR COUNTS;
C  DECREMENT COUNTER FOR PAIR JUST COUNTED;  RESTORE SEQUENCE STATUS;
C  INCREMENT POINTER IN THIS COLUMN AND TRY NEXT ELEMENT,
C  UNLESS THIS COLUMN IS EXHAUSTED
C*****************************************************************************
C
45    PAIR(PAIRNO)=PRNC-1
      COL1=COL-1
      DO 35 J=1,LENGR
35    R(J)=STORE(COL1,J)
30    PTR(COL)=ELT+1
      IF(PTR(COL).LE.NOSYM)GO TO 25
C
C*****************************************************************************
C  BACK UP ONE COLUMN;
C  PRINT SEQUENCE IF LENGTH IS EQUAL TO OR GREATER THAN THE MINIMUM;
C  RESET LENGTH OF LONGEST SEQUENCE IF NECESSARY
C*****************************************************************************
C
200   LEN=COL-1
      IF(LEN.GE.PRTLEN)PRINT1015,LEN,(SEQ(J),J=1,LEN)
      IF(LEN.GT.MAXLEN)MAXLEN=LEN
C
C*********************************************************
C  RESET ENTRY POINTER FOR BACK UP COLUMN;
C  IF PREVIOUS COLUMN EXHAUSTED, BACK UP AGAIN
C*********************************************************
C
15    PTR(COL)=1
      COL=COL-1
      IF(PTR(COL).GT.NOSYM)GO TO 15
C
C*************************************************************************
C  IF WE HAVE BACKED UP TO THE ORIGINAL ENTRIES, WE ARE FINISHED
C*************************************************************************
C
      IF(COL.EQ.LINIT)GO TO 220
C
C*************************************************************
C  RESTORE STATUS OF SEQUENCE FOR PREVIOUS COLUMN
C*************************************************************
C
      COL1=COL-1
      DO 280 J=1,LENGR
280   R(J)=STORE(COL1,J)
C
C
C
```

```
C****************************************
C  RESET LARGEST ELEMENT FOR NORMALITY
C****************************************
C
      MAXELT=2
      DO 205 I=1,COL1
205   IF(SEQ(I).GT.MAXELT)MAXELT=SEQ(I)
      GO TO 20
C
C*************************************************************
C  SINCE WE HAVE BACKED UP TO THE ORIGINAL ENTRIES, WE ARE DONE;
C  OUTPUT  N, D, AND N(D,N)
C*************************************************************
C
220   PRINT1020,NOSYM,D,MAXLEN
      STOP
C
C****************************************
1010  FORMAT(40I2)
1000  FORMAT(3I3)
1005  FORMAT('1'////' NUMBER OF SYMBOLS',I5,5X,'MAXIMUM LENGTH OF ABAB..
     X. SUBSEQUENCE IS',I5/' SEQUENCES OF LENGTH',I5,'  OR LONGER ARE PR
     XINTED'//' BEGINNING ENTRIES IN ALL SEQUENCES',5X,2I3///
     X '              LENGTH',25X,'SEQUENCES')
1015  FORMAT(' ',2X,I5,10X,50I2)
1020  FORMAT(/////' MAXIMUM LENGTH FOR N=',I5,3X,'D=',I5,5X,'IS',I6)
      END
C
C*************************************************************************
C
      SUBROUTINE INIT(SEQ,COL,NOSYM,STORE,PTR,LINIT1,CURRNT,LENGR)
      IMPLICIT INTEGER (A-Z)
      DIMENSION SEQ(50),STORE(50,24),PTR(50),PAIR(15),R(24),CURRNT(24)
      LOGICAL*1 RECEPT(6,6),REC
      EQUIVALENCE (RECEPT(1,1),R(1)),(PAIR(1),R(10))
C
C******************
C  INITIALIZATION
C******************
C
      DO 10 I=1,LENGR
10    R(I)=0
C
C*************************************************************************
C  SET POINTERS.  EACH POINTER IS ONE GREATER THAN ENTRY IN SEQUENCE
C  IN CORRESPONDING POSITION
C*************************************************************************
C
      COL1=COL-1
      DO 20 J=LINIT1,COL1
20    PTR(J)=SEQ(J)+1
C
C*****************************************************
C  MAKE 1 PAIR-RECEPTIVE TO ALL SUBSEQUENT ENTRIES
C*****************************************************
C
```

```
      DO 30 L=2,NOSYM
30    RECEPT(1,L)=.TRUE.
C
C*******************************************************************
C  SCAN SEQUENCE TO RECORD PAIR COUNTS AND PAIR-RECEPTIVENESS DATA
C*******************************************************************
C
      DO 40 CCLN=2,CCL1
      ELT=SEQ(CCLN)
      IF(ELT.EQ.1)GO TO 150
      ELT1=ELT-1
      DO 170 K=1,ELT1
      REC=RECEPT(K,ELT)
      IF(.NOT.REC)GO TO 170
      GO TO (50,60,70,80),K
      PAIRNO=1
      GO TO 100
50    PAIRNO=ELT
      GO TO 100
60    PAIRNO=ELT+4
      GO TO 100
70    PAIRNO=ELT+7
      GO TO 100
80    PAIRNO=ELT+9
100   PAIR(PAIRNO)=PAIR(PAIRNO)+1
      RECEPT(K,ELT)=.FALSE.
170   CONTINUE
150   ELTP1=ELT+1
      IF(ELT.EQ.NOSYM)ELTP1=NOSYM
      DO 160 L=ELTP1,NOSYM
160   RECEPT(ELT,L)=.TRUE.
C
C*******************************************************
C  STORE STATUS AT THIS COLUMN AND GO ON TO NEXT COLUMN
C*******************************************************
C
165   DO 175 J=1,LENGR
175   STORE(CCLN,J)=R(J)
40    CONTINUE
C
C*****************************************
C  PASS BACK CURRENT STATUS OF THE SEQUENCE
C*****************************************
C
      DO 180 J=1,LENGR
180   CURRNT(J)=R(J)
      RETURN
      END
C
C**************************************************************************
C**************************************************************************
```

REFERENCES

1. H. Davenport and A. Schinzel, "A Combinatorial Problem Connected With Differential Equations", American Journal of Mathematics, Vol. 87(1965), pp. 684-694.

2. R.G. Stanton and D.P. Roselle, "A Result On Davenport-Schinzel Sequences", Colloquia Mathematica Societatis Janos Bolyai 4, Combinatorial Theory and Its Applications, Balatonfured (Hungary), 1969, pp. 1023-1027.

3. R.G. Stanton and D.P. Roselle, "Some Properties of Davenport-Schinzel Sequences", Acta Arithmetica XVII (1971), pp. 355-362.

4. H. Davenport, "A Combinatcrial Problem Connected With Differential Equations II", Acta Arithmetica XVII (1971), pp. 363-372.

5. R.G. Stanton and D.P. Roselle, "Results On Davenport-Schinzel Sequences", Proceedings of the Louisiana Conference On Combinatorics, Graph Theory, and Computing, Louisiana State University, Baton Rouge, March 1970, pp. 249-267.

6. R.G. Stanton and R.C. Mullin, "A Map-Theoretic Approach to Davenport-Schinzel Sequences", Pacific Journal of Mathematics, Vcl. 40, No.1 (1972), pp. 167-172.