NUC-TN-381

Copy 83

# NAVAL UNDERSEA RESEARCH and DEVELOPMENT CENTER

## DERIVATION OF CUMULATIVE MULTI-PING DETECTION PROBABILITY MODEL

By

L. K. Arndt
Performance Modeling & Operations Analysis Division

January 1970

San Diego, California

U0133742
*Title:* Derivation of Cumulative Multiple-Ping Detection Probability Model.
*AD Number:* ADA072100
*Corporate Author:* NAVAL UNDERSEA WARFARE CENTER SAN DIEGO CA
*Personal Author:* Arndt,L K
*Report Date:* April 01, 1968
*Media:* 9 Page(s)
*Distribution Code:* 01 - APPROVED FOR PUBLIC RELEASE
*Report Classification:* Unclassified
*Source Code:* 403023
From the collection:
Technical Reports

attached to U-133,742

DEPARTMENT OF THE NAVY
NAVAL UNDERSEA RESEARCH AND DEVELOPMENT CENTER
SAN DIEGO, CALIFORNIA 92132

IN REPLY REFER TO:
506/efa
5605
Ser 506-5
AUG 05 1970

From: Commander, Naval Undersea Research and Development Center
To: Distribution List

Subj: NUC Technical Note; forwarding of

Encl: NUC Tech Note No. 381, "Derivation of Cumulative Multi-Ping Detection Probability Model"

1. Enclosure (1) is forwarded for information and retention.

R.H. ALLEN
By direction

Distribution:
NWSC (T.G. Bell) (Code 2040) (Cy 1)
(F.B. Rakoff) (Code 2131) (Cy 2)
(L.T. Einstein) (Code 2071) (Cy 3)
(S.A. Peterson) (Code 2010) (Cy 4)
(S. Slavtcheff) (Code 2132) (Cy 5)
(R.W. Hasse) (Code 2210) (Cy 6)
(H. Wilms) (Code 2011) (Cy ~~6)~~ 7)
(H. Fridge) (Code 2041) (Cy 7)
NOL (R.L. Shaffer) (Code ASW 1425) (Cy 8)
(A. Fanjoy) (Cy 9)
IDA (C.J. Loda) (Cy 10)
NAVAIRDEVCEN (C.L. Bartberger) (AWRD) (Cy 11)
NWL (Code KRT-3) (Cy 12)
SHIPRES&DEV (Panama City) (Code 716) (Cy 13)
NAVOCEANO (Library) (Cy 14)
(Code 1640) (Cy 15)
(Code 3300) (Cy 16)
(Code 0314D) (Cy 17)
(Code 3400) (Cy 18)
(Code 70) (Cy 19)
COMKWESTEVDET (LCDR W.A. Estell) (Cy 20)
COMASWFORPAC (CDR John D. Hague) (Cy 21)
COMASWFORLANT (Cy 22)
CHNAVMAT (NMAT 0331) (Cy 23)
(NMAT 03L) (Cy 24)

506/efa
5605
Ser 506-5

Subj: NUC Technical Note; forwarding of

Distribution: (Cont'd)
CINCPACFLT (Code 93) (Cy 25)
CINCLANTFLT (Cy 26)
CNO (OP-03EG) (Cy 27)
(OP-322C) (Cy 28)
(OP-07T) (Cy 29)
(OP-713) (Cy 30)
(OP-095B) (Cy 31)
COMNAVSHIPSYSCOM (NSHIP OOV) (Cy 32)
(NSHIP OOV1) (Cy 33)
(NSHIP OOV1B) (Cy 34 and 35)
(NSHIP OOV2) (Cy 36)
(NSHIP OOV3) (Cy 37)
(NSHIP PMS 385) (Cy 38)
(NSHIP PMS 386) (Cy 39)
(NSHIP PMS 387) (Cy 40)
(NSHIP PMS 388) (Cy 41)
COMNAVAIRSYSCOM (NAIR 604) (Cy 42)
COMNAVORDSYSCOM (NORD 0322) (Cy 43)
(NORD 05A) (Cy 44)
(NORD 9132) (Cy 45)
NAVSEC (Code 6170) (Cy 46)
COMSUBDEVGRUONE (Cy 47)
COMSUBDEVGRUTWO (Cy 48)
NELC (Library) (Cy 49)
UNIV OF CA - SD (Marine Physical Lab) (Cy 50)
(Scripps Inst of Oceano) (Cy 51)
(Security Officer, Attn: Library) (Cy 52)
NAVAL UNDERWATER WEAPONS RESEARCH & ENGINEERING STATION (Stanley Erickson) (Cy 53)
NAVPGSCOL (Dept of Oceano) (Cy 54)
(Library) (Cy ~~55~~) 83
(Dr. G.H. Jung) (~~Cy 56~~) (84)
FLENUMWEACEN (Cy 57)
NAVOCEANO, SD (Cy 58)
COMNAVWEASERV (Cy 59)
FLEWEACEN (Norfolk, Va.) (Cy 60)
ONR (Code 102-OS) (Dr. J. Brackett Hersey) (Cy 61)
(Code 102-OS) (CDR Peter Tatro) (Cy 62)
DEPCOMOPTEVFORPAC (Cy 63)
COMCRUDESPAC (Code 425) (Cy 64)
COMCRUDESLANT (Cy 65)
COMSUBPAC (Cy 66)
COMSUBLANT (Cy 67)
COMFIRSTFLT (Cy 68)
COMSECONDFLT (Cy 69)
COMSIXTHFLT (Cy 70)

556/efa
5605
Ser 506-5

Subj: NUC Technical Note; forwarding of

Distribution: (Cont'd)
COMTRAPAC (Cy 71)
COMTRALANT (Cy 72)
OCEANAV (Cy 73)
COMDESGRUPAC (Cy 74)
COMFAIRWINGSLANT/COMFAIRWING FIVE (Cy 75)
COMFAIRWINGSPAC/COMFAIRMOFFETT (Cy 76)

This technical note describes the development of a computer program for the calculation of multi-ping probability of detection as a part of a total computer model for sonar prediction. This memorandum has been prepared because it is believed that the information may be useful in this form to others at Naval Undersea Research and Development Center. It should not be construed as a report as its only function is to present a portion of the work as information for others. This technical note supersedes NUWC TN 90.

This work has been supported by NAVSHIPS Code OOV1 under subproject SF 11-121-105 Task 14318.

INTRODUCTION

One type of detection criterion of interest for use in sonar systems is a "counting criterion" where multiple looks at a target area are available. The receiver output for each look or ping is compared with a threshold and generates a binary one output if that threshold is exceeded or a binary zero if it fails to exceed the threshold. The multiple looks will now appear as a binary sequence and the "counting criterion" examines a group of length m, counting the number of ones in that group. If the number of ones, or threshold crossings, equals or exceeds the integer n the detection criterion is satisfied and a target is declared present. Otherwise another ping in the sequence is examined. This counting criterion is often called an n out of m criterion. Since successive groups of the m pings have m-1 pings in common the probability of satisfying the n of m criterion at any given point of the sequence is dependent upon the m-1 past outcomes. Such a sequence is often called a Markov chain of m-1 order. It is of interest to examine the joint probability of such overlapping groups of independent pings and to determine the probability that n or more of the group (in any order) exceed a fixed threshold, thus qualifying that group as satisfying an n out of m detection criterion. In addition, the cumulative probability that any group in the sequence satisfies the n of m criterion is desired. Since adjacent groups have many pings in common the groups themselves are not independent, even though the pings are assumed to be. It is assumed that the single ping probability of exceeding the threshold, $P_k$, is available and all solutions will be in terms of the single ping detection probability.

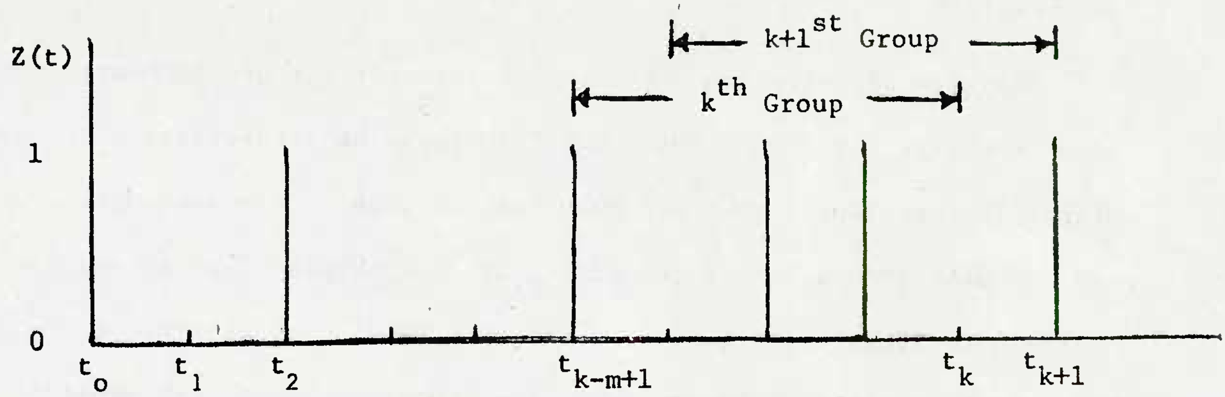


Figure 1: Binary Output of Threshold Receiver

JOINT PROBABILITY FOR $K^{th}$ GROUP

$Z(t)$ is a binary function which represents the sequence of single ping outputs. If the $k^{th}$ ping which is received at time $t_k$ exceeds a fixed receiver threshold, $Z(t_k)$ is equal to one; otherwise, $Z(t_k)$ is zero. $P_k$ is the single ping probability that the $k^{th}$ pulse exceeds the threshold and thus the probability that $Z(t_k)$ will equal one. We now examine a group of $m$ adjacent pulses consisting of the $k-m+1^{st}$ to the $k^{th}$ and write down the probability that $Z(t_k)$ for this group will contain exactly $n$ ones and $m-n$ zeros under the assumption of independence between pings.

If the $P_k$'s were equal, the expression for the probability of exactly $n$ out of $m$ is the familiar binomial form:

$$P\{n/m,\ k\} = \binom{m}{n} P^n (1-P)^{m-n}, \quad P = P_k \ \text{for all } k$$

and the probability of $n$ or more out of $m$ is

$$P\{\geq n/m,\ k\} = \sum_{j=n}^{m} \binom{m}{j} P^j (1-P)^{m-j}$$

For the more general case of unequal $P_k$'s, $P\{n/m,k\}$ is the sum of all possible products of n different $P_k$'s and m-n different $(1-P_k)$'s. Thus, if n = 3, m = 4.

$$P\{3/4,k\} = P_k P_{k-1} P_{k-2} (1-P_{k-3}) + P_k P_{k-1} (1-P_{k-2}) P_{k-3}$$

$$+ P_k (1-P_{k-1}) P_{k-2} P_{k-3} + (1-P_k) P_{k-1} P_{k-2} P_{k-3}.$$

The number of terms in the sum is

$$\binom{m}{n} \triangleq \frac{m!}{n! \quad (m-n)!}$$

In the above example $P\{\geq 3/4,k\}$ is given by

$$P\{\geq 3/4,k\} = P\{3/4,k\} + P\{4/4,k\}$$

$$= P_k P_{k-1} P_{k-2} (1-P_{k-3}) + P_k P_{k-1} (1-P_{k-2}) P_{k-3}$$

$$+ P_k (1-P_{k-1}) P_{k-2} P_{k-3} + (1-P_k) P_{k-1} P_{k-2} P_{k-3}$$

$$+ P_k P_{k-1} P_{k-2} P_{k-3}$$

If $P_k = P$ the above expression reduces to

$$P\{\geq 3/4,k\} = P^3 (1-P) + P^3 (1-P) + P^3 (1-P) + P^3 (1-P) + P^4$$

$$= 4 P^3 (1-P) + P^4$$

$$= \binom{4}{3} P^3 (1-P) + \binom{4}{4} P^4 = \sum_{k=3}^{4} \binom{4}{k} P^k (1-P)^{4-k}$$

the same as previously noted.

Cumulative Probability:

It is of interest to calculate the probability that a string of pings of length $k$ contains a sub-string of length $m$ which satisfies the $n$ of $m$ criteria. This sub-string can be located anywhere in the string. This probability will be denoted by $P_{cum}$ and is a monotonically increasing function of the string length $k$. To obtain a general algorithm for the calculation of $P_{cum}$, one approach is to generate all possible binary functions, $Z(t)$ of length $k$, selecting those that meet the criterion and calculate their probability and sum. Since the number of possible sequences is $2^k$ this procedure is not practical for large $k$. A second approach in which $P_{cum}(k)$ is a function of $P_{cum}(k-1)$ and a smaller set of binary functions will be described.

First a few definitions. Let $E(k)$ be the event that a given binary function of length $k$ has $n$ or more ones (successes) occurring within the last $m$ positions and $\overline{E(k)}$ the complement of that event. Thus, if. $Z(t)$ is written as a binary number with the right most digit representing $Z(t_k)$ then for $k = 7$, $n = 2$, $m = 3$, the string 0010010 fails while 0010101 is in event $E(7)$. With this notation $P_{cum}(k)$ is the probability that

E(k) occurs or E(k-1) or E(k-2) ... or E(m). Using DeMorgan's law

$$P_{cum}(k) = P(E(k) \cup E(k-1) \cup E(k-2) \cup \cdots \cup E(m)) \quad *$$

$$= P(E(k-1) \cup E(k-2) \cup \cdots \cup E(m))$$

$$+ P(E(k) \cap \overline{E(k-1)} \cap \overline{E(k-2)} \cap \cdots \overline{E(m)})$$

$$P_{cum}(k) = P_{cum}(k-1) + P(E(k), \overline{E(k-1)}, \overline{E(k-2)}, \cdots, \overline{E(m)}).$$

Using the notation P(A/B) for the probability of A given that B occurred, $P_{cum}(k)$ is given by

$$P_{cum}(k) = P_{cum}(k-1) + P(E(k) | \overline{E(k-1)}, \overline{E(k-2)}, \ldots \overline{E(m)})$$

$$\cdot P(\overline{E(k-1)}, \overline{E(k-2)}, \ldots \overline{E(m)})$$

but

$$P(\overline{E(k-1)}, \overline{E(k-2)}, \ldots \overline{E(m)}) = P(\overline{E(k-1) \cup E(k-2) \cup \ldots \cup E(m)})$$

$$= 1 - P_{cum}(k-1) \quad \text{giving}$$

$$P_{cum}(k) = P_{cum}(k-1) + P(E(k) | \overline{E(k-1)}, \overline{E(k-2)}, \ldots, \overline{E(m)}) \cdot (1 - P_{cum}(k-1)).$$

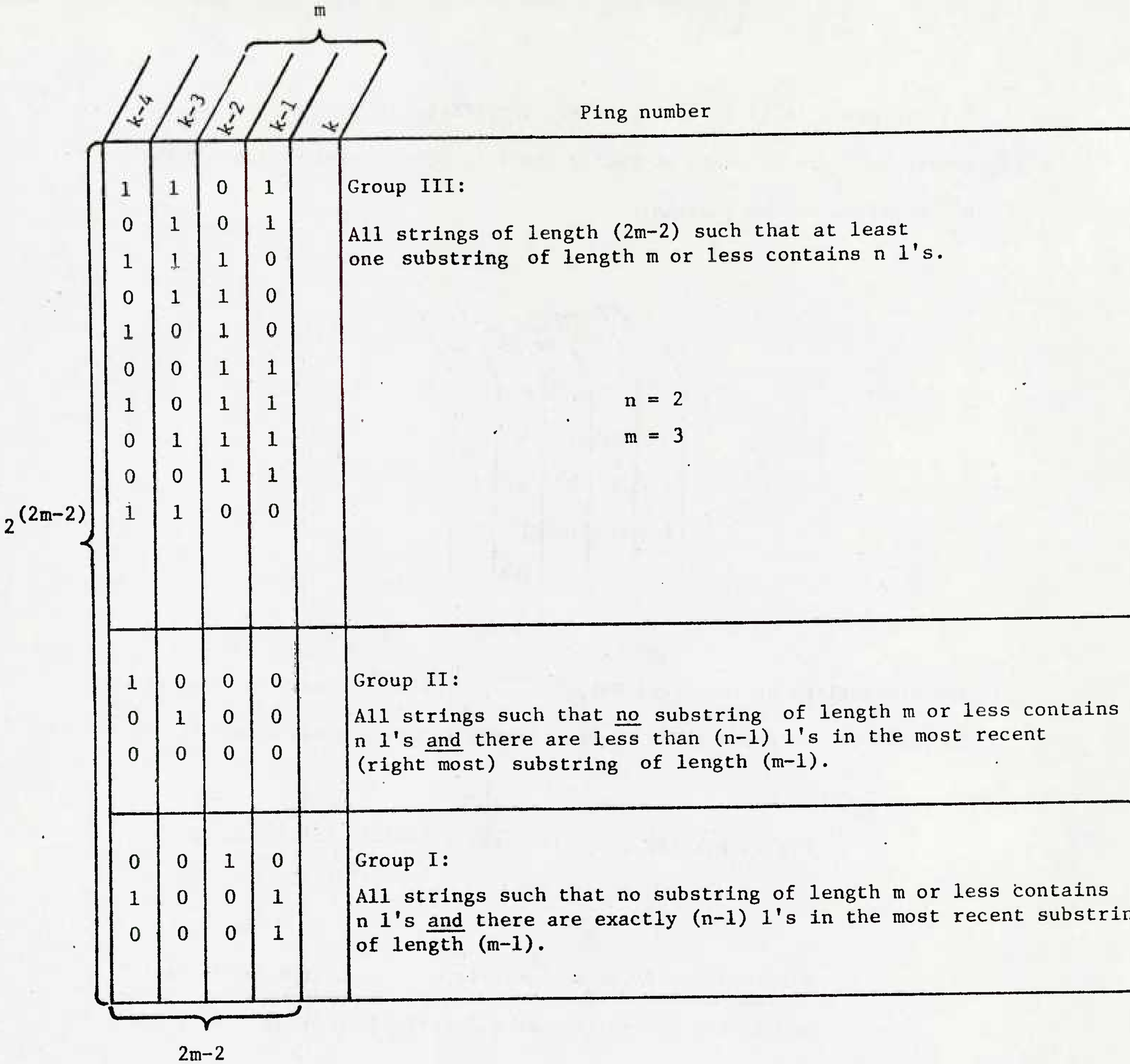
---

*U is union, ∩ is intersection hereafter written with a comma

The E(k) and E(k-m), have no common pings which determine the events success or failure; therefore,

$$P(E(k)|\overline{E(k-1)}, \overline{E(k-2)}, \ldots, \overline{E(m)}) = P(E(k)|\overline{E(k-1)}, \ldots, \overline{E(k-m+1)}).$$

Since the event E(k-m+1) depends on pings number (k-m+1) back to (k-m+1) - m+1 or (k-2m+2) and the event E(k) depends only on pings number k back to k-m+1 the $P(E(k)|\overline{E(k-1)}, \ldots, \overline{E(k-1-m+1)})$ may be determined by examining all possible binary sequences of length k - (k-2m+2) + 1 or 2m-1 ending with the $k^{th}$. There are $2^{(2m-1)}$ possible sequences which can be partitioned into the following groups using n=2, m=3 as an example.




Only those strings (of length 2m-2) in Group I can result in the event $E(k)|E(\overline{k-1}) \ldots E(\overline{k-m+1})$ no matter what the outcome of the $k^{th}$ ping. If the $k^{th}$ ping is a success (1) the event is satisfied by the strings in Group I and none of the strings in Group II or Group III. If the outcome of the $k^{th}$ ping is a failure

(0) the event fails for all groups. Therefore, <u>all</u> successful strings of length 2m-1 are obtained by taking those in Group I and adding a one in the $k^{th}$ position in the example:

| k-4 | k-3 | k-2 | k-1 | k |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |

The probability of the event $E(k)|\overline{E(k-1)}, .. \overline{E(k-m+1)}$ may now be written in terms of the probability of the strings in Group I and Group II.

$$P(E(k)|\overline{E(k-1)}, .. , \overline{E(k-m+1)}) = \frac{P(E(k), \overline{E(k-1)}, .., \overline{E(k-m+1)})}{P(\overline{E(k-1)}, ..., \overline{E(k-m+1)})}$$

$$= \frac{P(k) \cdot (\text{Sum of the probability of all strings in Group I})^*}{(\text{Sum of the probability of all strings in Group I or Group II})}$$

$$= \frac{P(k) \cdot P(Gp\ I)}{P(Gp\ I) + P(Gp\ II)}$$

*Note: In all successful strings the $k^{th}$ ping must be a 1 (success) and the k-m ping a 0 (failure).

In the example of n=2, m=3, k=5, and P(i) the probability that the $i^{th}$ ping exceeds a fixed threshold.

$$P_k(GP\ I) = (1-P(1)) \cdot (1-P(2)) \cdot P(3) \cdot (1-P(4))$$
$$+ P(1) \cdot (1-P(2)) \cdot (1-P(3)) \cdot P(4)$$
$$+ (1-P(1)) \cdot (1-P(2)) \cdot (1-P(3)) \cdot P(4)$$

$$P_k(GP\ II) = P(1) \cdot (1-P(2)) \cdot (1-P(3)) \cdot (1-P(4))$$
$$+ (1-P(1)) \cdot P(2) \cdot (1-P(3)) \cdot (1-P(4))$$
$$+ (1-P(1)) \cdot (1-P(2)) \cdot (1-P(3)) \cdot (1-P(4))$$

and

$$P_{cum}(k) = P_{cum}(k-1) + \frac{P(k) \cdot P_k\ (GP\ I)}{P_k(GP\ I) + P_k(GP\ II)} \{1 - P_{cum}(k-1)\}$$

After the strings of length 2m-2 in Group I and Group II have been generated (for a fixed n and m) they are retained for calculating $P(E(k))|E(\overline{k-1}), .., E(\overline{k-m+1}))$ for all k > 2 m-1 by changing only the individual ping probabilities, P(k), which enter into the calculation.

A general flow diagram for the calculation of $P_{cum}(k)$ for a given n and m is given in the appendix. The writing and check-out of this program was completed by Mr. K. A. Faucher of Code 556.

Appendix: Flow Diagram

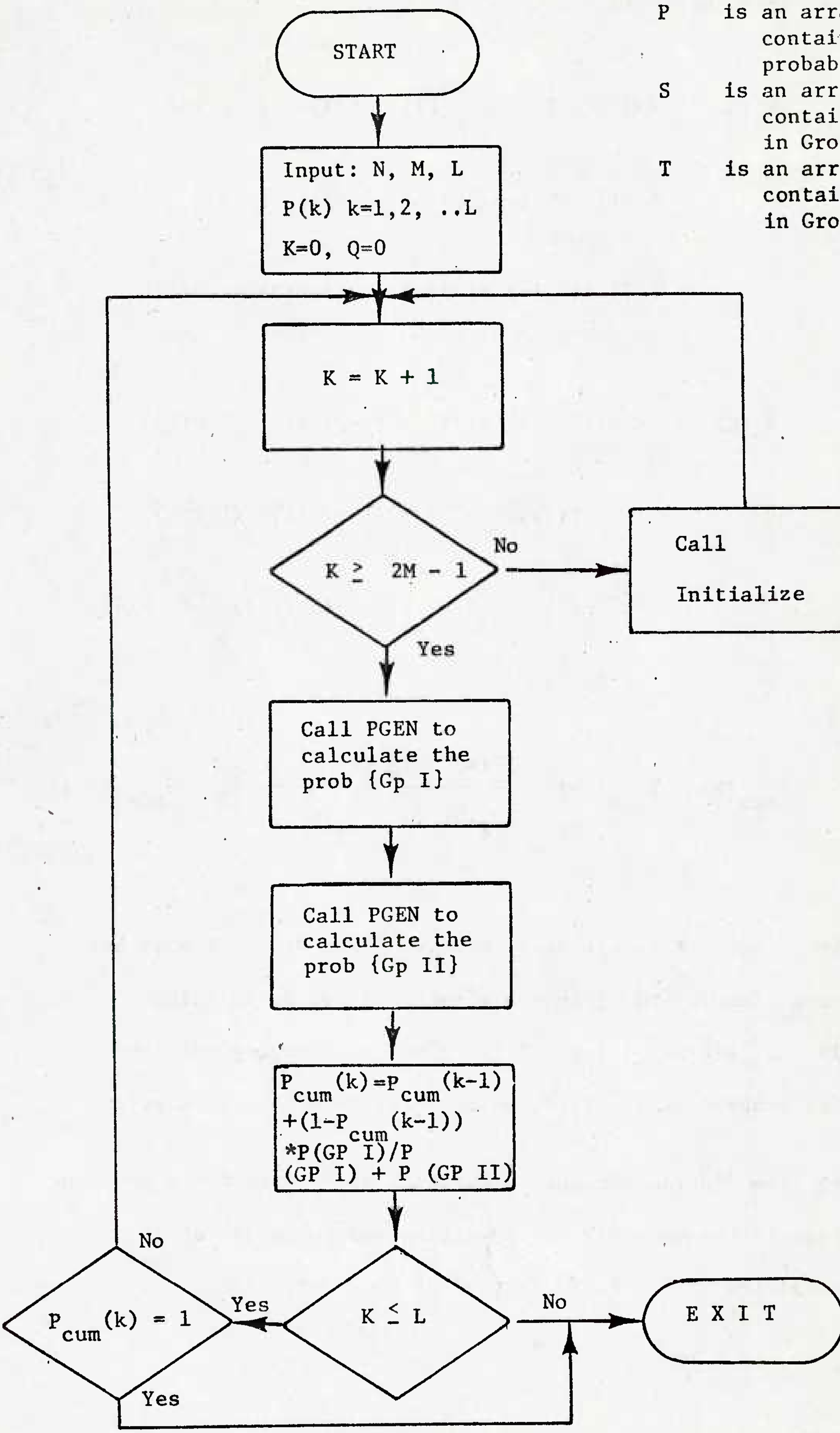
MAIN PROGRAM

$P_{cum}$ is an array of length L containing the cumulative probability.

P is an array of length L containing the ping probabilities.

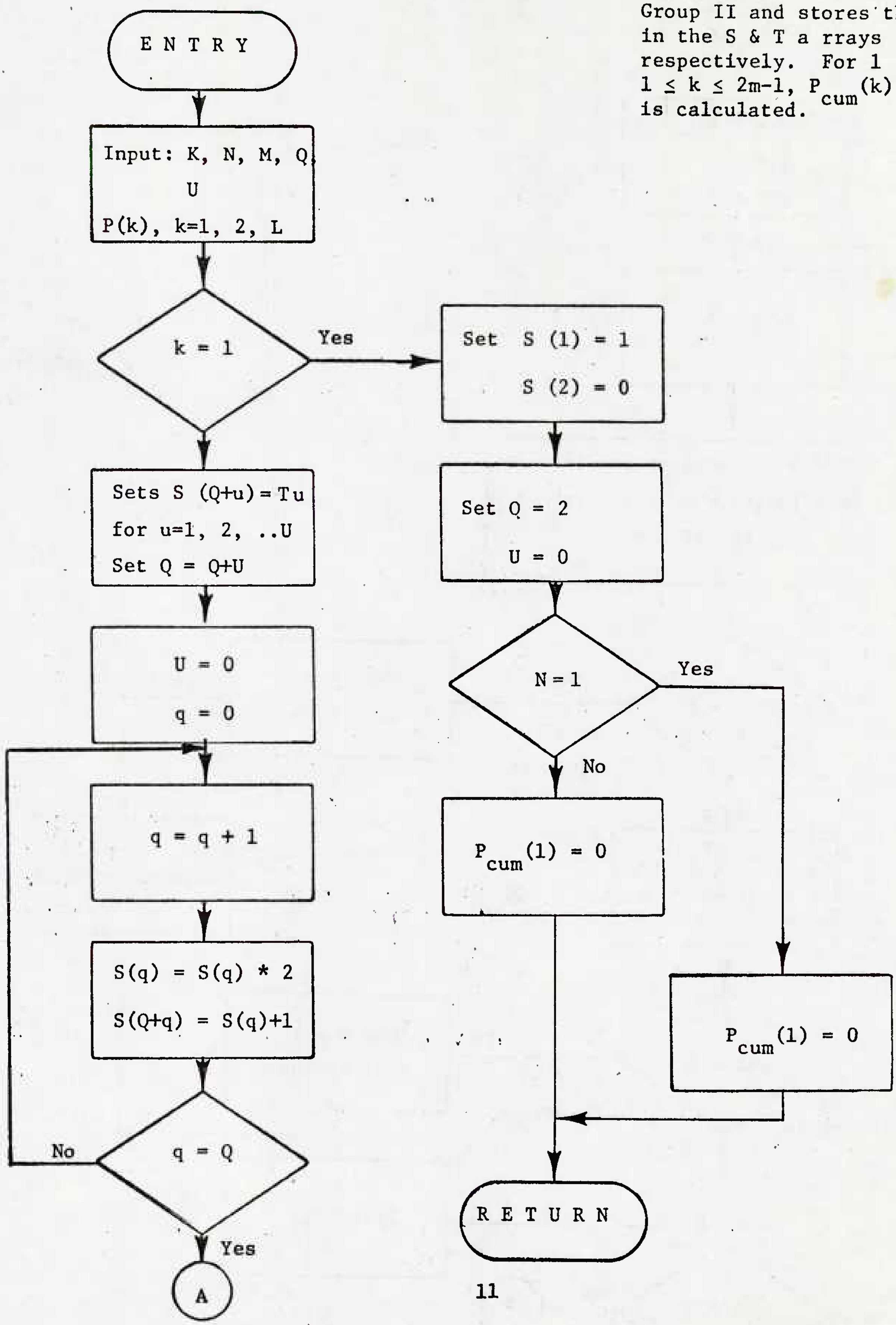
S is an array of length Q containing the sequences in Group I.

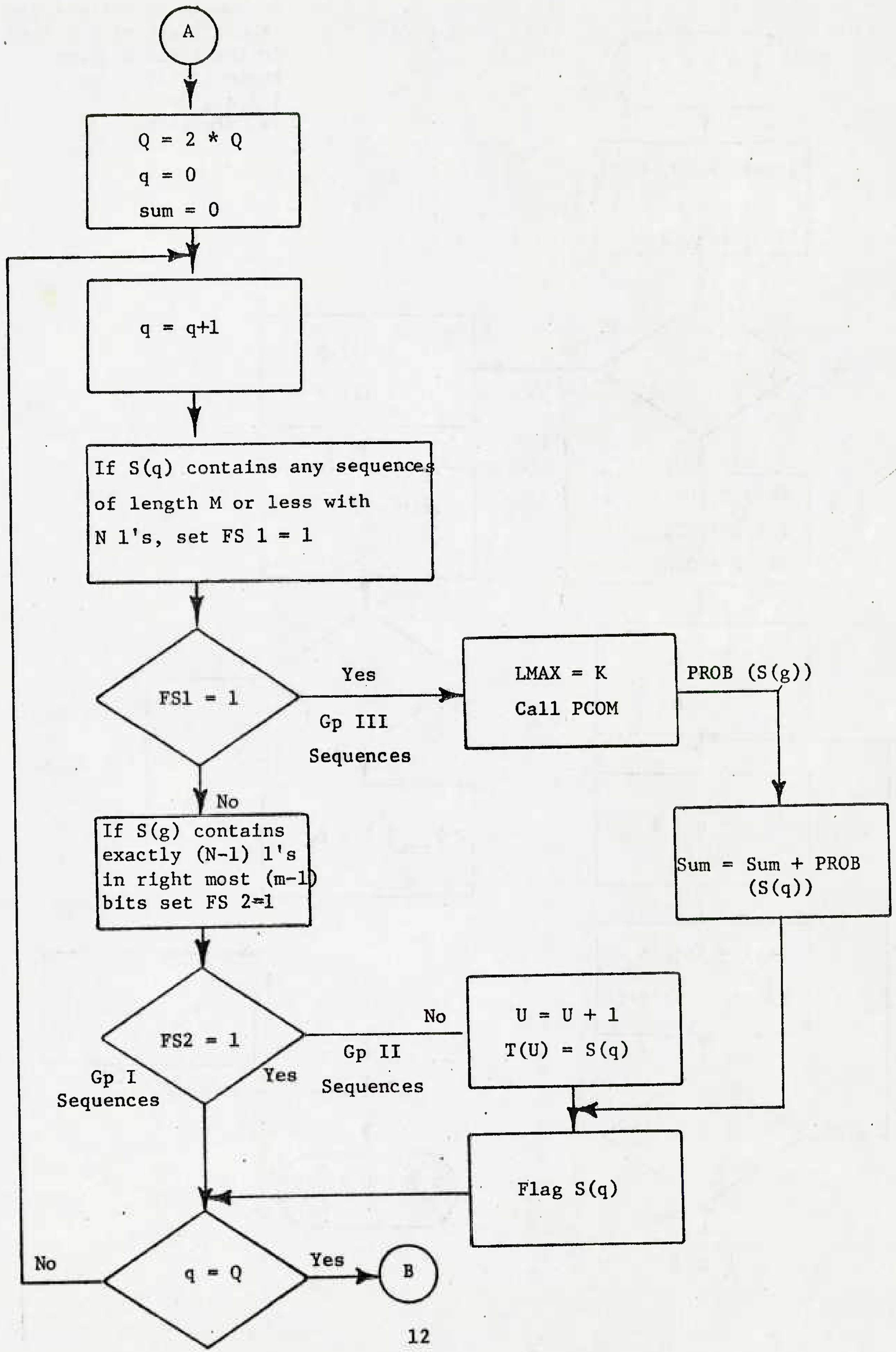
T is an array of length V containing the sequences in Group II.



SUBROUTINE INITIALIZE

Initialize generates the sequences in Group I and Group II and stores them in the S & T a rrays respectively. For 1 $1 \leq k \leq 2m-1$, $P_{cum}(k)$ is calculated.

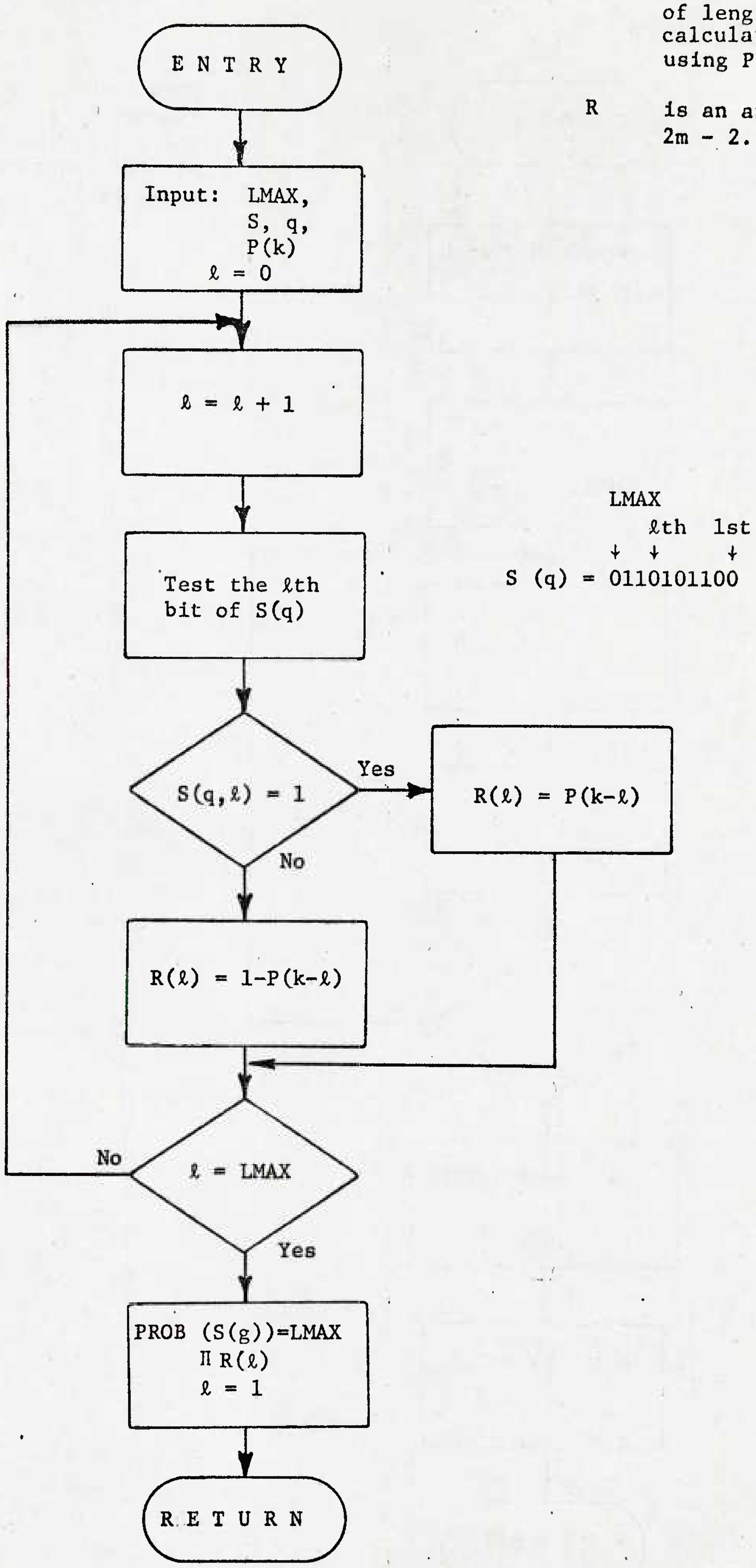



A
Q = 2 * Q
q = 0
sum = 0
q = q+1
If S(q) contains any sequences of length M or less with N 1's, set FS 1 = 1
FS1 = 1
Yes
Gp III Sequences
LMAX = K
Call PCOM
PROB (S(g))
No
If S(g) contains exactly (N-1) 1's in right most (m-1) bits set FS 2=1
Sum = Sum + PROB (S(q))
FS2 = 1
No
Gp II Sequences
Yes
Gp I Sequences
U = U + 1
T(U) = S(q)
Flag S(q)
q = Q
No
Yes
B

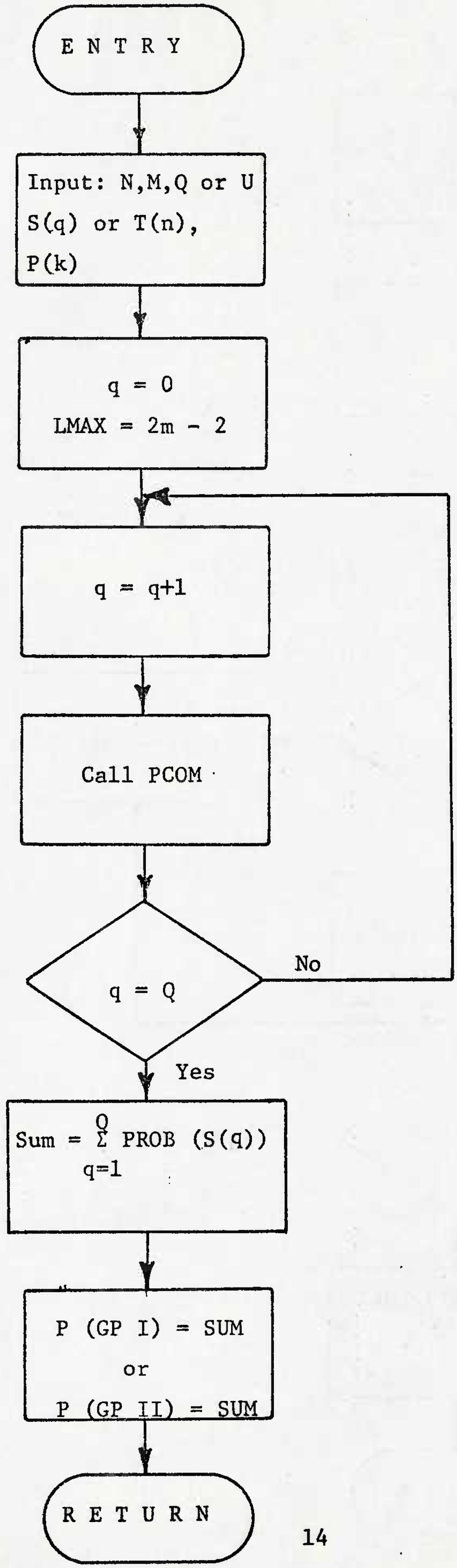
SUBROUTINE PCOM

PCOM accepts a binary number of length LMAX and calculates its probability using P(r).

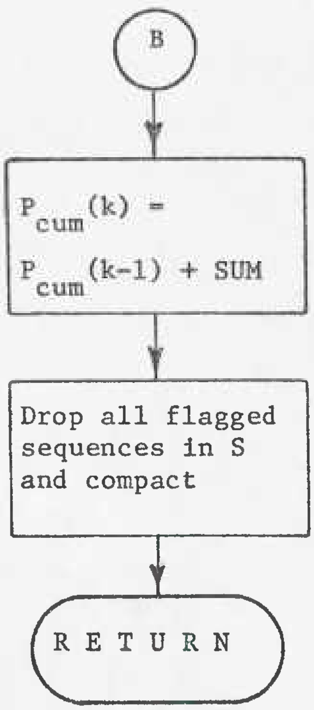
R is an array of length 2m - 2.



SUBROUTINE PGEN




PGEN accepts an array sequence (either S or T) and calculates the sum of the probability of all the sequences in the array using P(k).


B
Pcum(k) =
Pcum(k-1) + SUM
Drop all flagged sequences in S and compact
R E T U R N