DEFENSIVE PUBLICATION

UNITED STATES PATENT AND TRADEMARK OFFICE

Published at the request of the applicant or owner in accordance with the Notice of Dec. 16, 1969, 869 O.G. 687. The abstracts of Defensive Publication applications are identified by distinctly numbered series and are arranged chronologically. The heading of each abstract indicates the number of pages of specification, including claims and sheets of drawings contained in the application as originally filed. The files of these applications are available to the public for inspection and reproduction may be purchased for 30 cents a sheet.

PUBLISHED MARCH 2, 1976

944 O.G. 40

T944,002

END-OF-PATH INSERTION SYSTEM

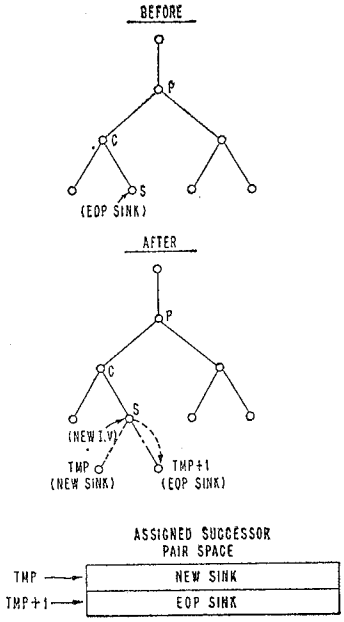
Luther Jay Woodrum, Poughkeepsie, N.Y., assignor to International Business Machines Corporation, Armonk, N.Y.

Continuation of application Ser. No. 415,085, Nov. 12, 1973, which is a continuation of abandoned application Ser. No. 203,573, Dec. 1, 1971. This application Sept. 30, 1974, Ser. No. 510,811

Int. Cl. G06f *13/00, 13/06*

U.S. Cl. 444—1

12 Sheets Drawing. 41 Pages Specification



Electronically generating and inserting each entry always at the end of a path in a binary tree directory. Each new insertion entry directly displaces an existing sink found by a search of the directory with a search argument (which may be an input key). The input sequence of search arguments may be unsorted. The ascending (descending) path property is not used in a directory constructed with this invention. As a result, there is no sorted relationship among the sinks contained within this directory as occurs, on the other hand, when generating or inserting into directories constructed with the inventions using the ascending path property in prior related applications having Ser. Nos. 136,902 and 136,951 filed Apr. 23, 1971, by the same inventor as the subject application. The end-of-path insertion location is found using the search method invention in prior application Ser. No. 136,686 filed Apr. 23, 1971 by the same inventor. The inner vertices in this directory may contain edges having the absolute, offset, or invertible form.

FIG. 1A

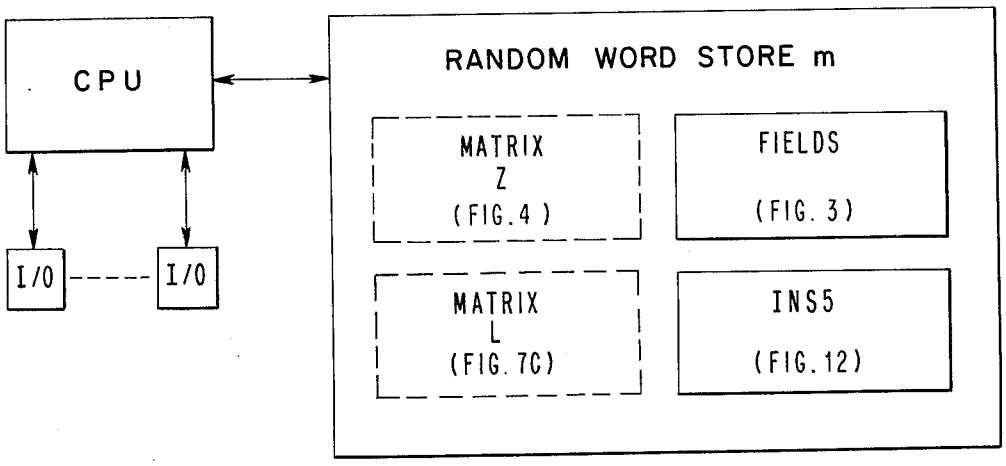

CPU
I/O
I/O
RANDOM WORD STORE m
MATRIX Z (FIG. 4)
FIELDS (FIG. 3)
MATRIX L (FIG. 7C)
INS5 (FIG. 12)

FIG. 1B

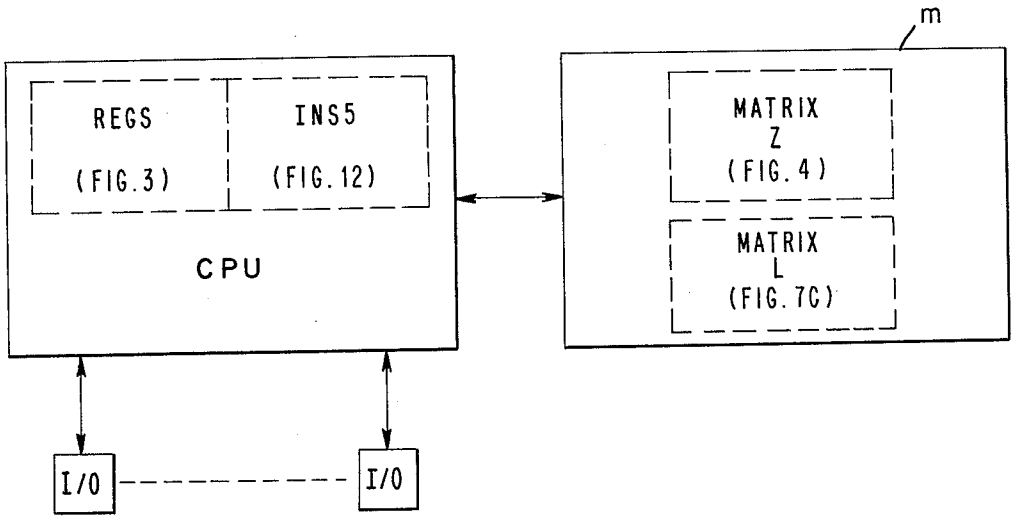
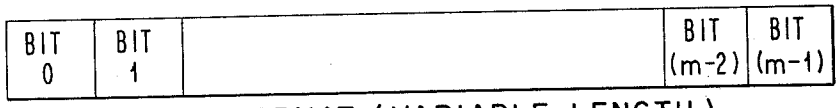
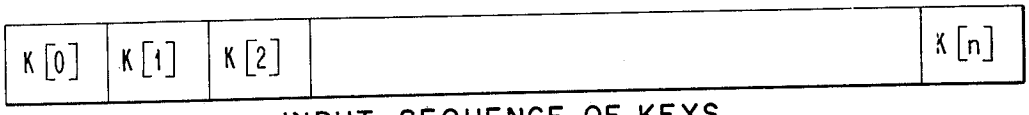

m
REGS (FIG. 3)
INS5 (FIG. 12)
CPU
MATRIX Z (FIG. 4)
MATRIX L (FIG. 7C)
I/O
I/O

FIG. 2A



KEY FORMAT (VARIABLE LENGTH)

FIG. 2B



INPUT SEQUENCE OF KEYS

FIG. 3

	FIELD / REGS	
DAR	ADDRESS OF DIRECTORY	
P	PREDECESSOR VERTEX ADDRESS	
C	CURRENT VERTEX ADDRESS	
S	SUCCESSOR VERTEX ADDRESS	
KEY0	LENGTH	END-OF-PATH KEY
ADR0	ADDRESS OF END-OF-PATH RECORD	
KEY1	LENGTH	CURRENT INPUT KEY
I	CURRENT L INDEX OF KEY1	
D	NEW D-INDEX	
BIT	VALUE OF BIT AT D-INDEX IN KEY1	
TMP	TEMPORARY REGISTER	
ALT	ADDRESS OF NEW LOCATION FOR EOP SINK	
SNK	ADDRESS OF NEW SINK ROW	

FIG. 4

MATRIX Z

ADDRESS IN M	# SINKS FIELD			SPACE CHAIN FIELD		
DAR + 0	0	1	2	3	4	5
DAR + 1	D	t_0	c_0	t_1	c_1	± EDGE
TMP						
TMP+1						
DAR + (2N-1)						

NOTE:

TMP IS ROW ADDRESS FOR LEFT SUCCESSOR AND TMP +1 IS ROW ADDRESS FOR PAIRED RIGHT SUCCESSOR IN ASSIGNED SUCCESSOR PAIR SPACE.

N IS MAXIMUM NUMBER OF SINKS IN MATRIX Z

FIG. 5A

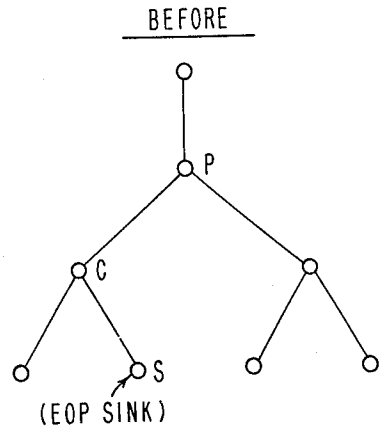


FIG. 5B

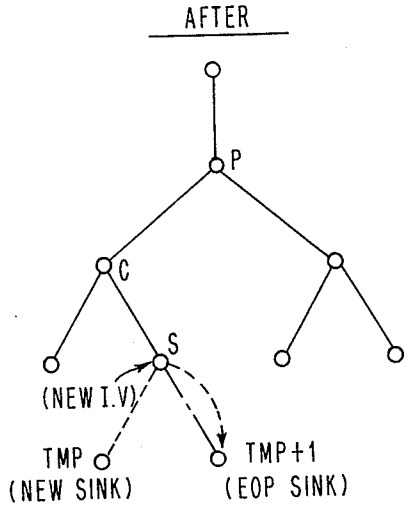


FIG. 6

FIG. 7A

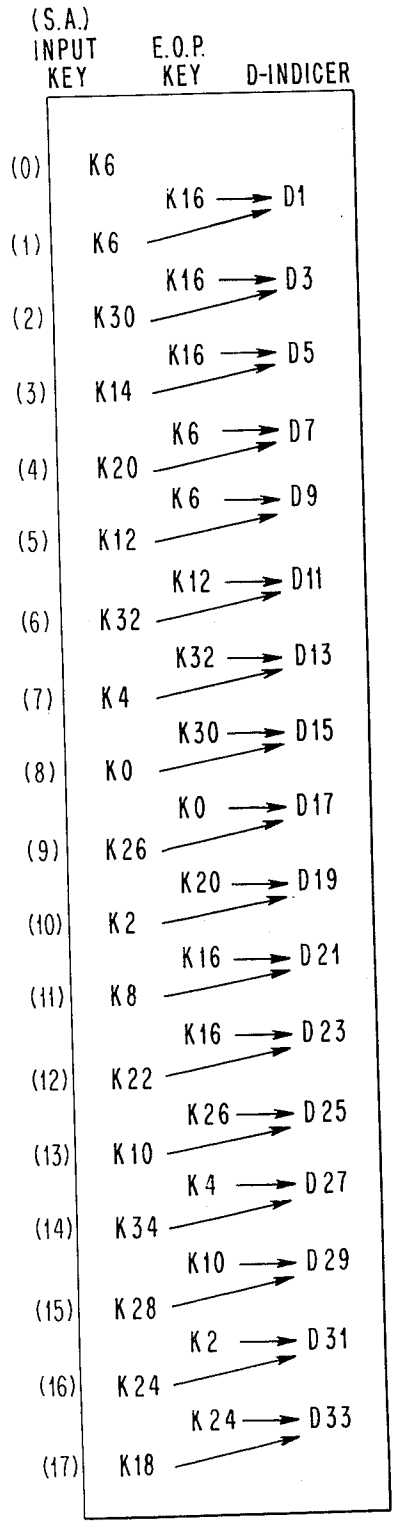

(S.A.) INPUT KEY
E.O.P. KEY
D-INDICER
(0) K6
K16 → D1
(1) K6
K16 → D3
(2) K30
K16 → D5
(3) K14
K6 → D7
(4) K20
K6 → D9
(5) K12
K12 → D11
(6) K32
K32 → D13
(7) K4
K30 → D15
(8) K0
K0 → D17
(9) K26
K20 → D19
(10) K2
K16 → D21
(11) K8
K16 → D23
(12) K22
K26 → D25
(13) K10
K4 → D27
(14) K34
K10 → D29
(15) K28
K2 → D31
(16) K24
K24 → D33
(17) K18

FIG. 7B

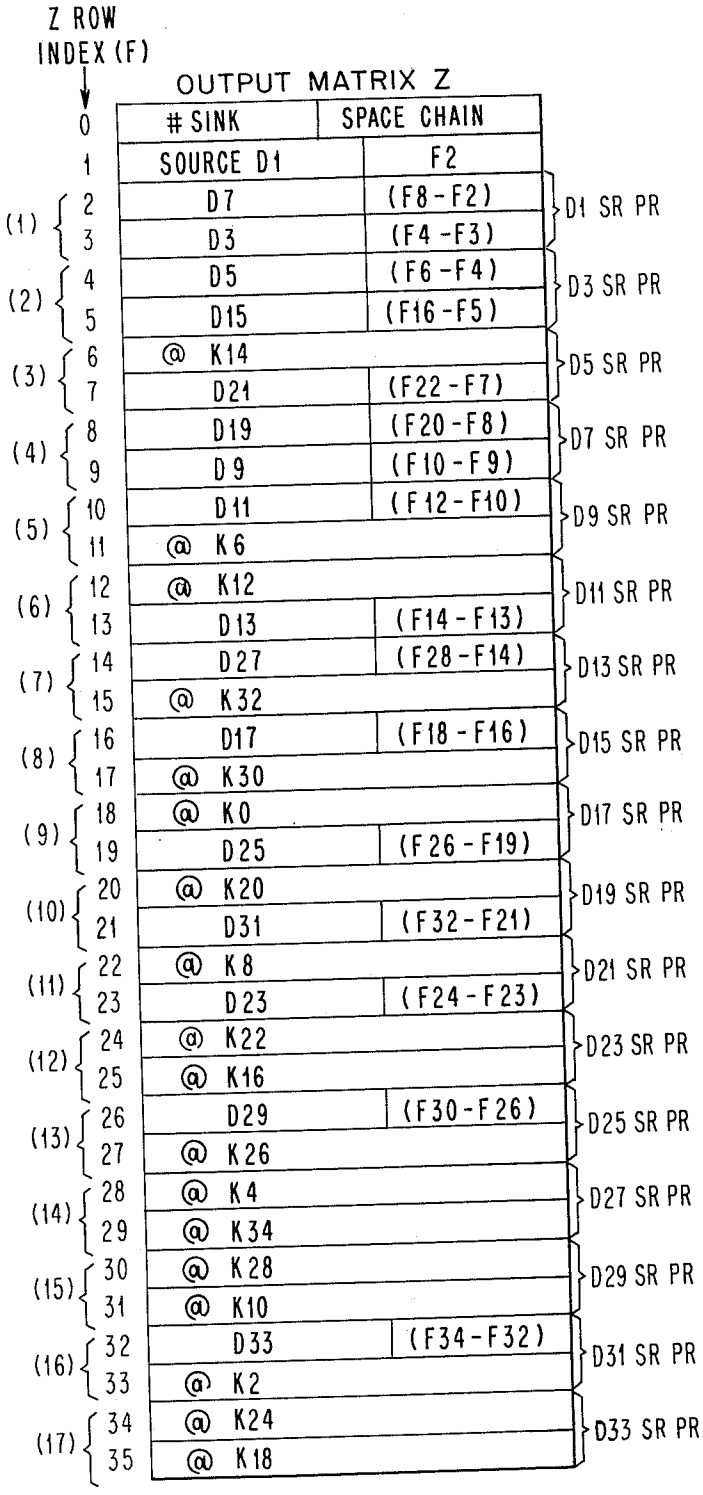

Z ROW INDEX (F)
OUTPUT MATRIX Z
SINK
SPACE CHAIN
SOURCE D1
F2
D7 (F8-F2)
D3 (F4-F3)
D5 (F6-F4)
D15 (F16-F5)
@ K14
D21 (F22-F7)
D19 (F20-F8)
D9 (F10-F9)
D11 (F12-F10)
@ K6
@ K12
D13 (F14-F13)
D27 (F28-F14)
@ K32
D17 (F18-F16)
@ K30
@ K0
D25 (F26-F19)
@ K20
D31 (F32-F21)
@ K8
D23 (F24-F23)
@ K22
@ K16
D29 (F30-F26)
@ K26
@ K4
@ K34
@ K28
@ K10
D33 (F34-F32)
@ K2
@ K24
@ K18
D1 SR PR
D3 SR PR
D5 SR PR
D7 SR PR
D9 SR PR
D11 SR PR
D13 SR PR
D15 SR PR
D17 SR PR
D19 SR PR
D21 SR PR
D23 SR PR
D25 SR PR
D27 SR PR
D29 SR PR
D31 SR PR
D33 SR PR

FIG. 7C

MATRIX L

I	INPUT KEY	I/O RECORD ADDRESS
0	K16	R16
1	K6	R6
2	K30	R30
3	K14	R14
4	K20	R20
5	K12	R12
6	K32	R32
7	K4	R4
8	K0	R0
9	K26	R26
10	K2	R2
11	K8	R8
12	K22	R22
13	K10	R10
14	K34	R34
15	K28	R28
16	K24	R24
17	K18	R18

FIG. 8A

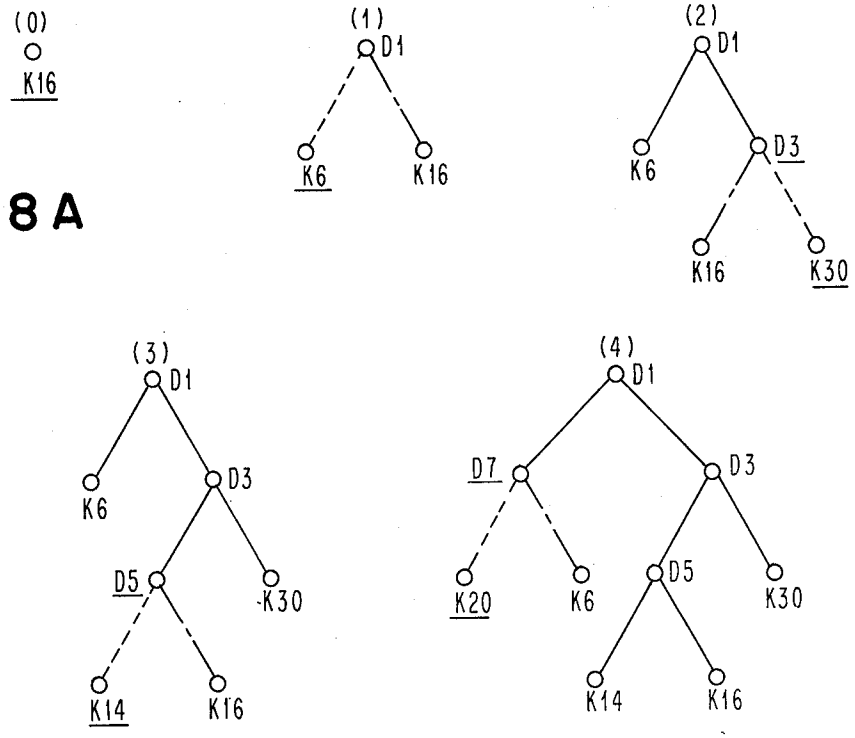


FIG. 8B

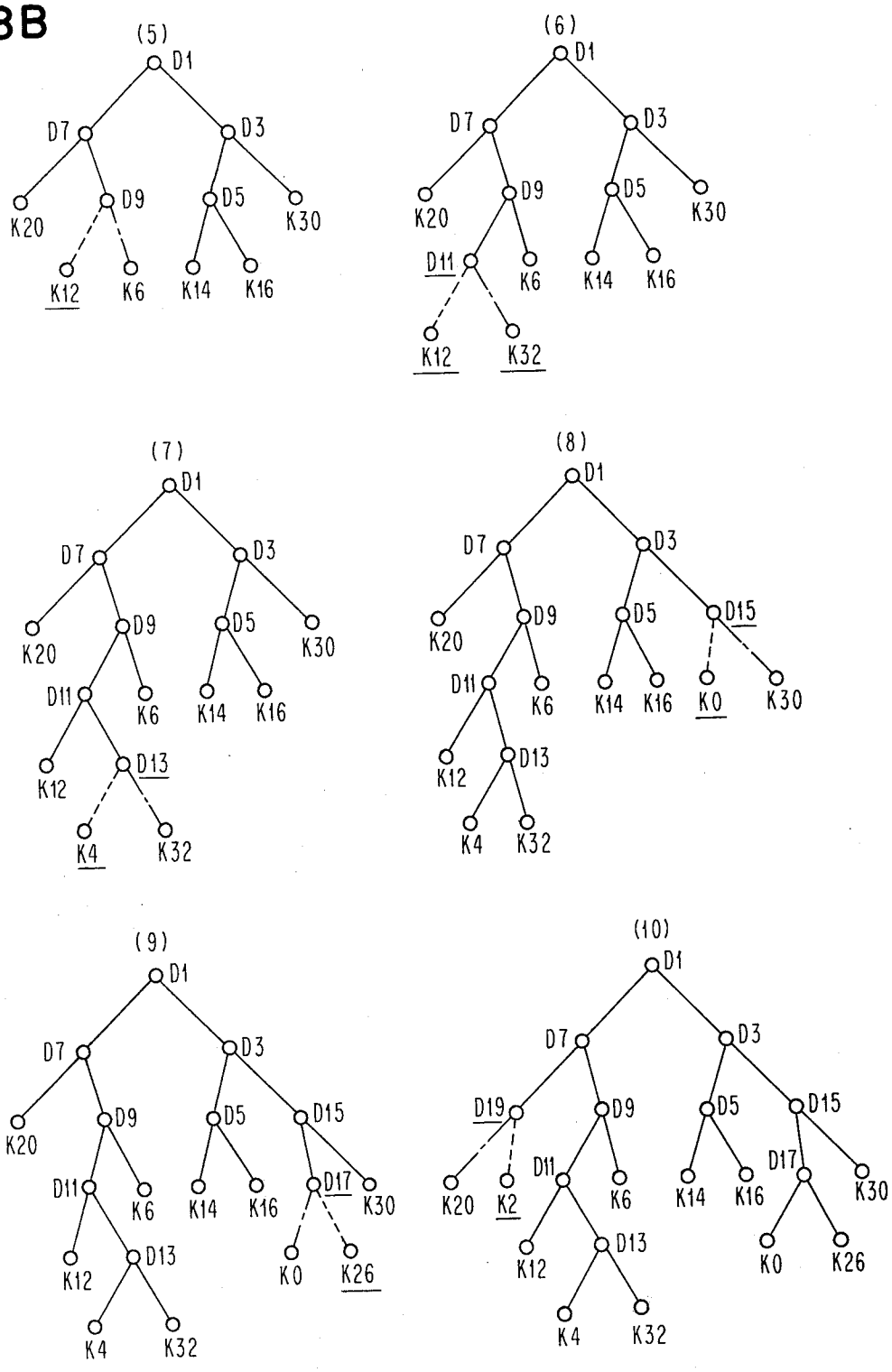
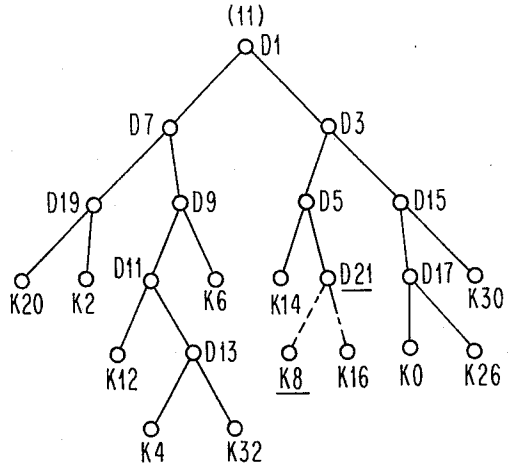
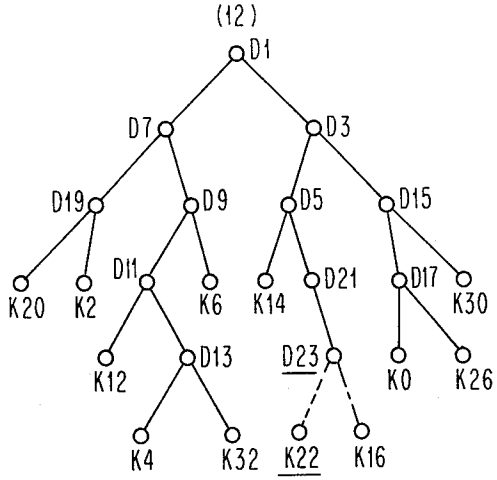
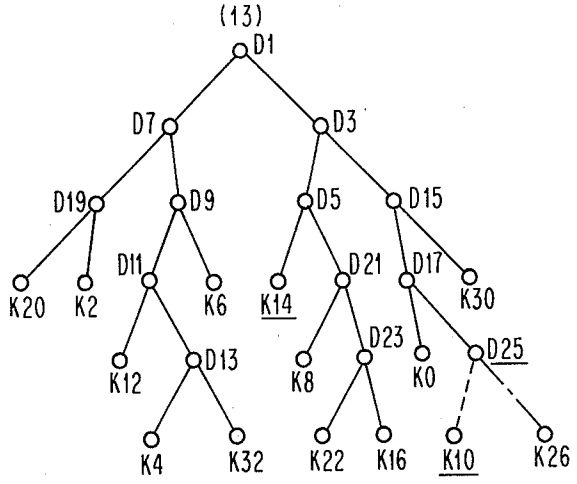

(5)
D1
D7
D3
K20
D9
D5
K30
K12
K6
K14
K16
(6)
D1
D7
D3
K20
D9
D5
K30
D11
K6
K14
K16
K12
K32
(7)
D1
D7
D3
K20
D9
D5
K30
D11
K6
K14
K16
K12
D13
K4
K32
(8)
D1
D7
D3
K20
D9
D5
D15
D11
K6
K14
K16
K0
K30
K12
D13
K4
K32
(9)
D1
D7
D3
K20
D9
D5
D15
D11
K6
K14
K16
D17
K30
K12
D13
K0
K26
K4
K32
(10)
D1
D7
D3
D19
D9
D5
D15
K20
K2
D11
K6
K14
K16
D17
K30
K12
D13
K0
K26
K4
K32

FIG. 8C


(11)
D1
D7
D3
D19
D9
D5
D15
K20
K2
D11
K6
K14
D21
D17
K30
K12
D13
K8
K16
K0
K26
K4
K32


(12)
D1
D7
D3
D19
D9
D5
D15
K20
K2
D11
K6
K14
D21
D17
K30
K12
D13
D23
K0
K26
K4
K32
K22
K16


(13)
D1
D7
D3
D19
D9
D5
D15
K20
K2
D11
K6
K14
D21
D17
K30
K12
D13
K8
D23
K0
D25
K4
K32
K22
K16
K10
K26

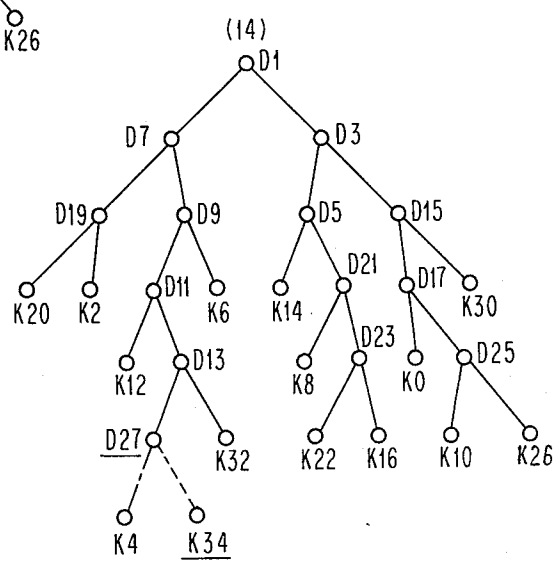

(14)
D1
D7
D3
D19
D9
D5
D15
K20
K2
D11
K6
K14
D21
D17
K30
K12
D13
K8
D23
K0
D25
D27
K32
K22
K16
K10
K26
K4
K34

FIG. 8D

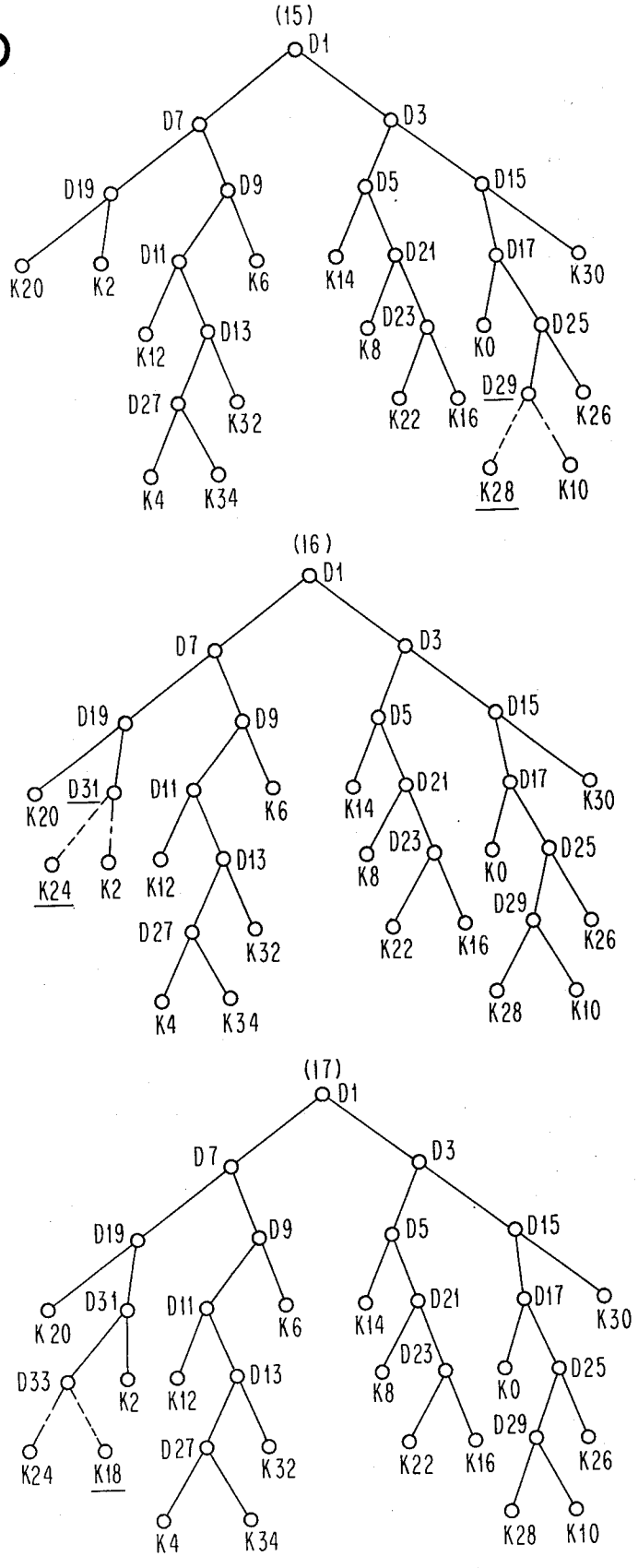

(15)
D1
D7
D3
D19
D9
D5
D15
K20
K2
D11
K6
K14
D21
D17
K30
K12
D13
K8
D23
K0
D25
D27
K32
K22
K16
D29
K26
K4
K34
K28
K10
(16)
D1
D7
D3
D19
D9
D5
D15
K20
D31
D11
K6
K14
D21
D17
K30
K24
K2
K12
D13
K8
D23
K0
D25
D27
K32
K22
K16
D29
K26
K4
K34
K28
K10
(17)
D1
D7
D3
D19
D9
D5
D15
K20
D31
D11
K6
K14
D21
D17
K30
D33
K2
K12
D13
K8
D23
K0
D25
K24
K18
D27
K32
K22
K16
D29
K26
K4
K34
K28
K10

FIG. 9

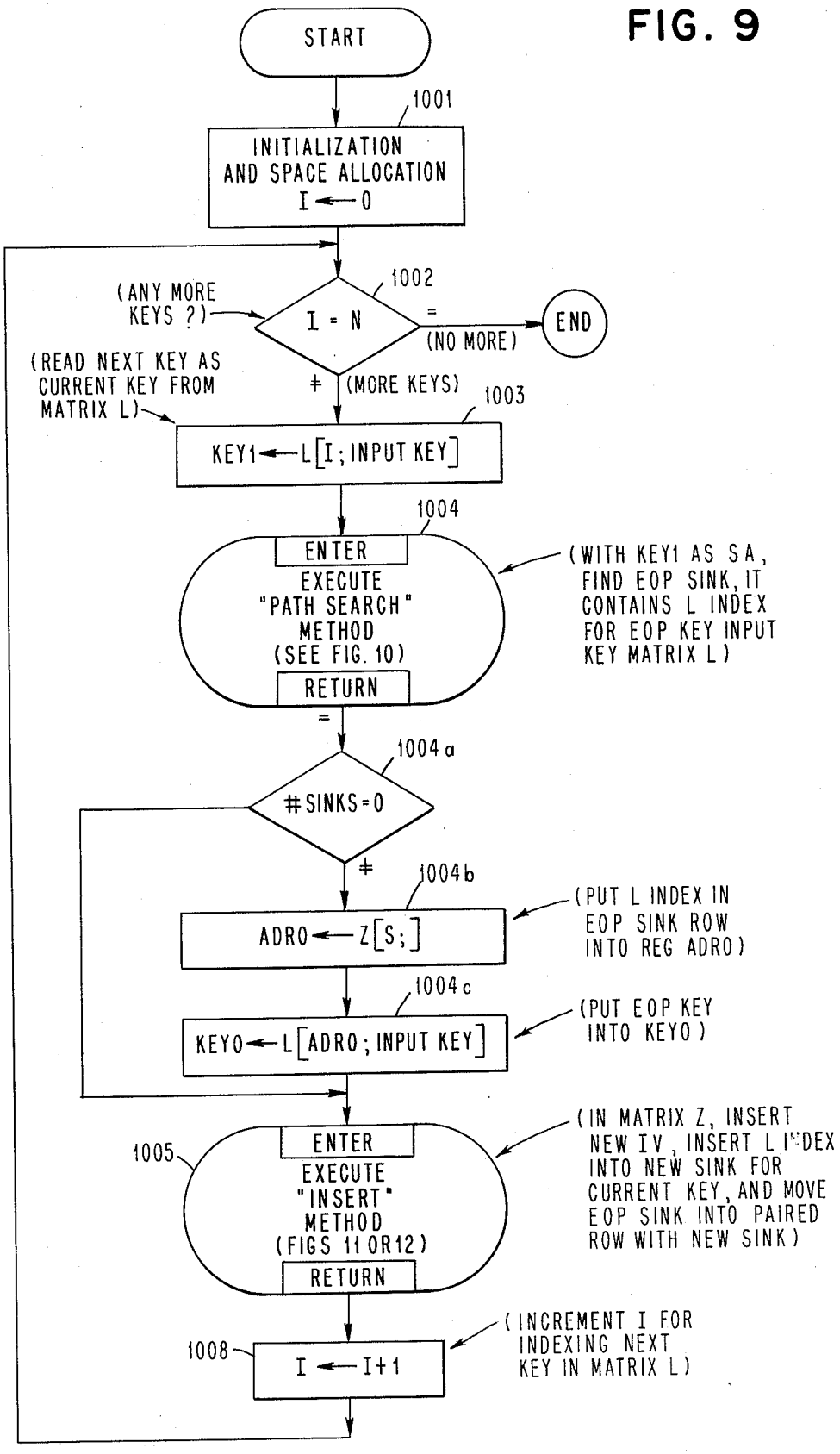

START
1001
INITIALIZATION AND SPACE ALLOCATION I ← 0
1002
(ANY MORE KEYS ?)
I = N
=
(NO MORE)
END
≠ (MORE KEYS)
(READ NEXT KEY AS CURRENT KEY FROM MATRIX L)
1003
KEY1 ← L[I; INPUT KEY]
1004
ENTER
EXECUTE "PATH SEARCH" METHOD (SEE FIG. 10)
RETURN
(WITH KEY1 AS SA, FIND EOP SINK, IT CONTAINS L INDEX FOR EOP KEY INPUT KEY MATRIX L)
=
1004a
#SINKS=0
≠
1004b
ADRO ← Z[S;]
(PUT L INDEX IN EOP SINK ROW INTO REG ADRO)
1004c
KEY0 ← L[ADRO; INPUT KEY]
(PUT EOP KEY INTO KEY0)
1005
ENTER
EXECUTE "INSERT" METHOD (FIGS 11 OR 12)
RETURN
(IN MATRIX Z, INSERT NEW IV, INSERT L INDEX INTO NEW SINK FOR CURRENT KEY, AND MOVE EOP SINK INTO PAIRED ROW WITH NEW SINK)
1008
I ← I+1
(INCREMENT I FOR INDEXING NEXT KEY IN MATRIX L)

FIG. 10

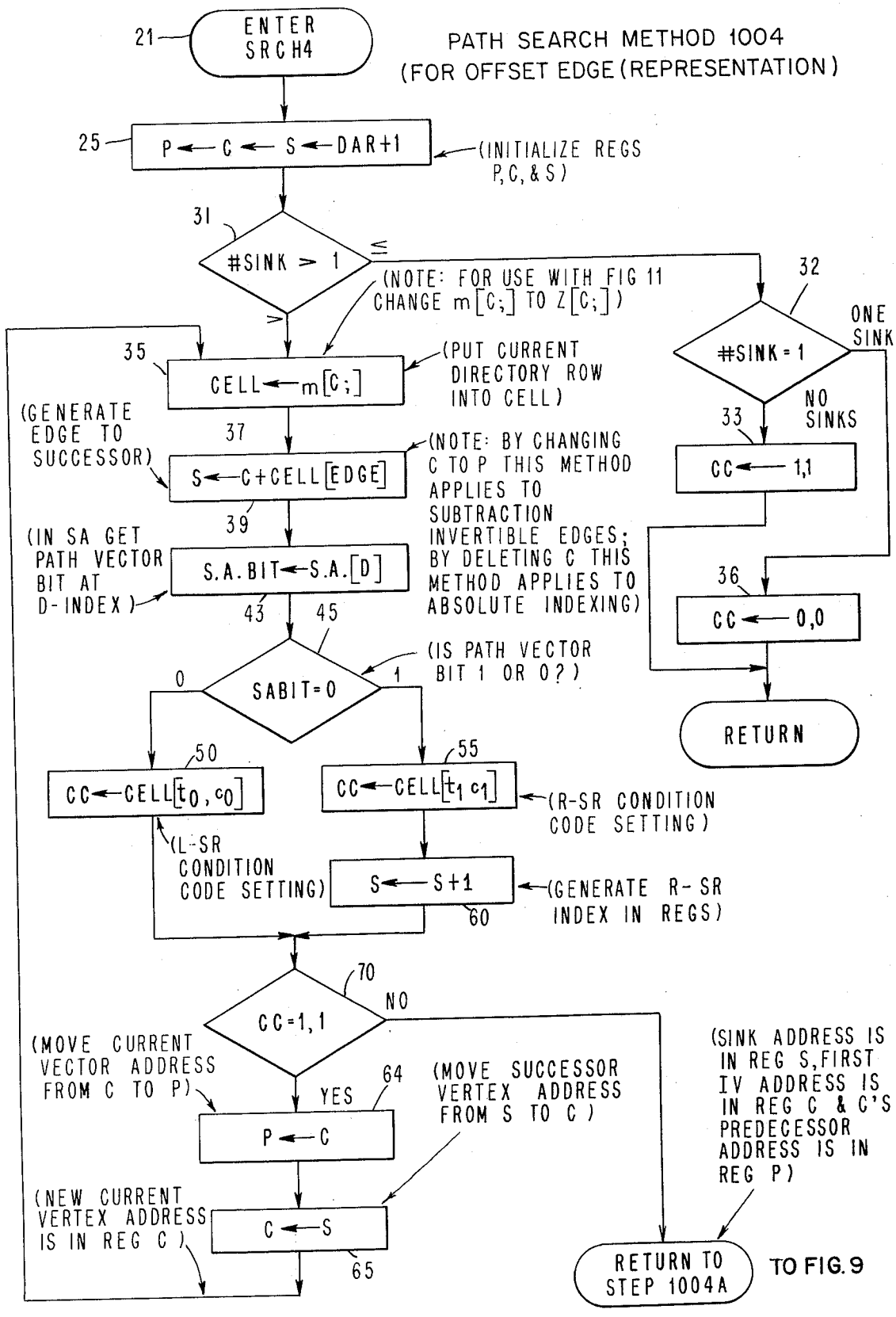

21
ENTER SRCH4
PATH SEARCH METHOD 1004
(FOR OFFSET EDGE (REPRESENTATION)
25
P ← C ← S ← DAR+1
(INITIALIZE REGS P, C, & S)
31
#SINK > 1
≤
>
(NOTE: FOR USE WITH FIG 11 CHANGE m[C;] TO Z[C;])
32
#SINK = 1
ONE SINK
NO SINKS
33
CC ← 1,1
35
CELL ← m[C;]
(PUT CURRENT DIRECTORY ROW INTO CELL)
(GENERATE EDGE TO SUCCESSOR)
37
S ← C+CELL[EDGE]
(NOTE: BY CHANGING C TO P THIS METHOD APPLIES TO SUBTRACTION INVERTIBLE EDGES; BY DELETING C THIS METHOD APPLIES TO ABSOLUTE INDEXING)
39
(IN SA GET PATH VECTOR BIT AT D-INDEX)
S.A.BIT ← S.A.[D]
36
CC ← 0,0
43
45
(IS PATH VECTOR BIT 1 OR 0?)
0
SABIT=0
1
RETURN
50
CC ← CELL[t0, c0]
55
CC ← CELL[t1 c1]
(R-SR CONDITION CODE SETTING)
(L-SR CONDITION CODE SETTING)
S ← S+1
(GENERATE R-SR INDEX IN REGS)
60
70
CC=1,1
NO
(MOVE CURRENT VECTOR ADDRESS FROM C TO P)
YES
64
(MOVE SUCCESSOR VERTEX ADDRESS FROM S TO C)
P ← C
(NEW CURRENT VERTEX ADDRESS IS IN REG C)
C ← S
65
(SINK ADDRESS IS IN REG S, FIRST IV ADDRESS IS IN REG C & C'S PREDECESSOR ADDRESS IS IN REG P)
RETURN TO STEP 1004A
TO FIG. 9

FIG. 11 (END-OF-PATH INSERT METHOD-1)

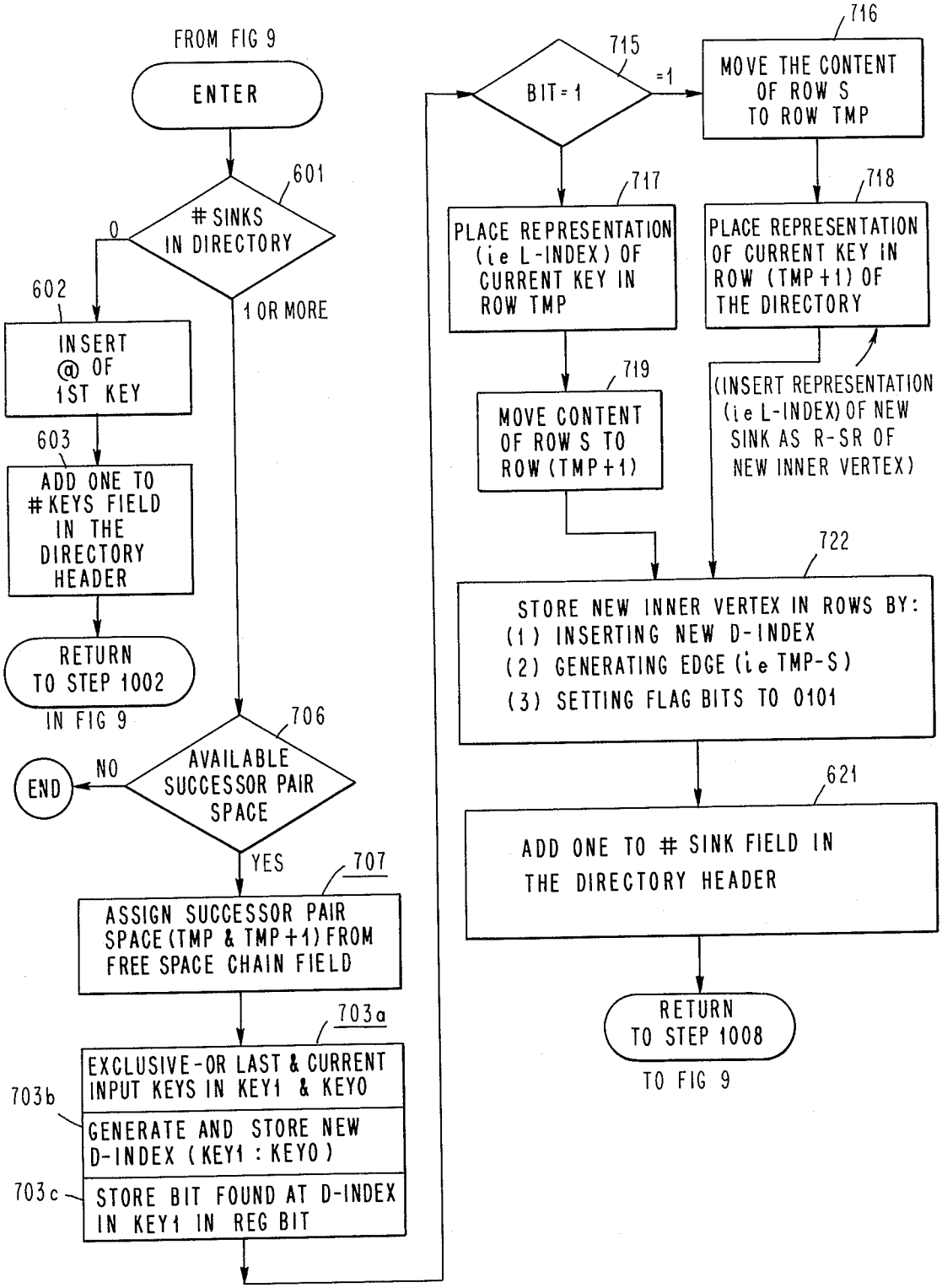
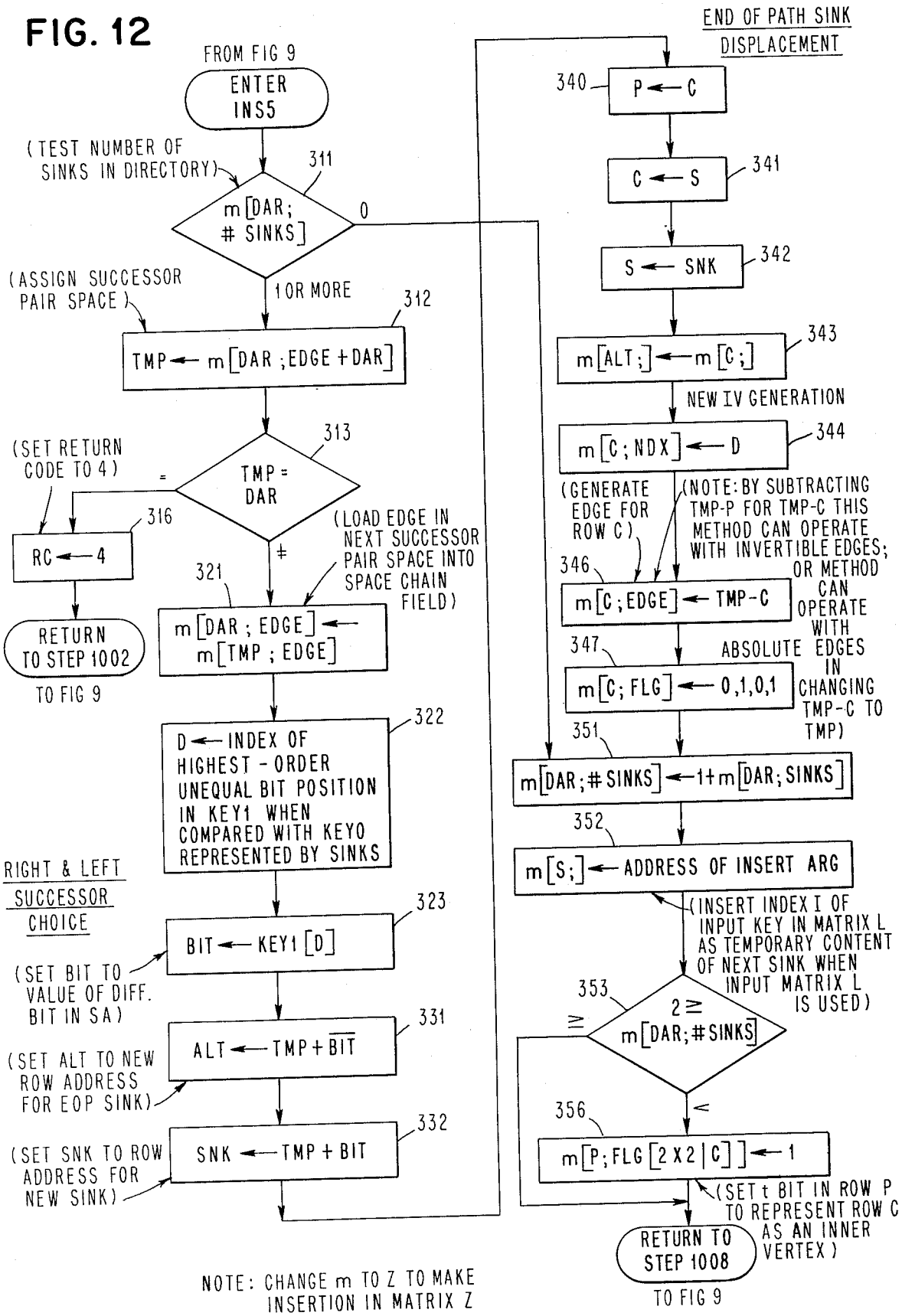


FIG. 12


FROM FIG 9
ENTER INS5
(TEST NUMBER OF SINKS IN DIRECTORY)
311
m[DAR; # SINKS]
0
1 OR MORE
(ASSIGN SUCCESSOR PAIR SPACE)
312
TMP ← m[DAR ; EDGE + DAR]
313
(SET RETURN CODE TO 4)
=
TMP = DAR
≠
316
RC ← 4
RETURN TO STEP 1002
TO FIG 9
321
(LOAD EDGE IN NEXT SUCCESSOR PAIR SPACE INTO SPACE CHAIN FIELD)
m[DAR ; EDGE] ← m[TMP ; EDGE]
322
D ← INDEX OF HIGHEST - ORDER UNEQUAL BIT POSITION IN KEY1 WHEN COMPARED WITH KEY0 REPRESENTED BY SINKS
RIGHT & LEFT SUCCESSOR CHOICE
323
BIT ← KEY1 [D]
(SET BIT TO VALUE OF DIFF. BIT IN SA)
331
ALT ← TMP + BIT
(SET ALT TO NEW ROW ADDRESS FOR EOP SINK)
332
SNK ← TMP + BIT
(SET SNK TO ROW ADDRESS FOR NEW SINK)
NOTE: CHANGE m TO Z TO MAKE INSERTION IN MATRIX Z
END OF PATH SINK DISPLACEMENT
340
P ← C
C ← S
341
S ← SNK
342
m[ALT;] ← m[C;]
343
NEW IV GENERATION
m[C; NDX] ← D
344
(GENERATE EDGE FOR ROW C)
(NOTE: BY SUBTRACTING TMP-P FOR TMP-C THIS METHOD CAN OPERATE WITH INVERTIBLE EDGES; OR METHOD CAN OPERATE WITH ABSOLUTE EDGES IN CHANGING TMP-C TO TMP)
346
m[C; EDGE] ← TMP-C
347
m[C; FLG] ← 0,1,0,1
351
m[DAR; # SINKS] ← 1 + m[DAR; SINKS]
352
m[S;] ← ADDRESS OF INSERT ARG
(INSERT INDEX I OF INPUT KEY IN MATRIX L AS TEMPORARY CONTENT OF NEXT SINK WHEN INPUT MATRIX L IS USED)
353
2 ≥ m[DAR; # SINKS]
≥
<
356
m[P; FLG[2 X 2 | C]] ← 1
(SET t BIT IN ROW P TO REPRESENT ROW C AS AN INNER VERTEX)
RETURN TO STEP 1008
TO FIG 9