ADD ITEM TO CHAIN-LINKED LIST

ALGORITHM 100

PHILIP J. KIVIAT

United States Steel Corp., Appl. Research Lab., Monroeville, Penn.

procedure inlist (t,info,m,list,n,first,flag,addr,listfull);
integer n,m,first,flag,1; integer array info,list,addr;
comment inlist adds the information pair (t,info) to the chain-
link structured matrix list (i,j), where t is an order key \( \geq 0 \), and
info[k] an information vector associated with t. info(k) has di-
nension m, list(i,j) has dimensions \( (n \times (m+3)) \). flag denotes
the head and tail of list(i,j), and first contains the address of the
first (lowest order) entry in list(i,j). addr(k) is a vector con-
taining the addresses of available (empty) rows in list(i,j).
Initialization: list(i,m+2) = flag, for some i \( \leq n \). If list(i,j) is
filled exit is to listfull;

begin integer i, j, link1, link2;
0: if addr[i] = 0 then go to listfull; i := 1;
1: if list[i,1] \( \leq 1 \)
   then begin if list[i,2] \( \neq 0 \) then begin link1 := m+2;
         link2 := m+3; go to 2 end; else begin if
list[i,m+2] = flag then begin i := flag;
link1 := m+3; link2 := m+2; go to 3 end;
else begin i := i+1; go to 1 end end end;
else begin link1 := m+3; link2 := m+2 end;
2: if list[i,link2] \( \neq \) flag
   then begin k := 1; i := list[i,link2];
if (link2 = m+2 \( \wedge \) list[i,1] \( \leq t \) \( \vee \)
(link2 \( \neq \) m+2 \( \wedge \) list[i,1] \( > t \)) then go to 4;
else go to 1 end;
   else begin list[i,link2] := addr[i] end;
3: j := addr[i]; list[j,link1] := i;
list[j,link2] := flag; if link2 = m+2 then
first := addr[i]; go to 5;
4: j := addr[i]; list[j,link1] := list[i,link1];
list[j,link1] := list[k,link2] := addr[i];
list[j,link2] := i;
5: list[j,1] := t; for i := 1 step 1 until m do
list[j,i+1] := info[i]; for i := 1 step 1 until n-1 do
addr[i] := addr[i+1]; addr[n] := 0
end inlist