ALGORITHM 100
ADD ITEM TO CHAIN-LINKED LIST

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,t; 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 ≧ 0, and
   info(k) an information vector associated with t. info(k) has di-
   mension m, list(i,j) has dimensions (n × (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 ≦ n. If list(i,j) is
   filled exit is to listfull;
begin integer i, j, link1, link2;
0:  if addr [1] = 0;  then go to listfull;  i := 1;
1:  if list [i,1] ≦ t
       then begin if list [i,2] ≠ 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] ≠ flag
       then begin k := i;  i := list [i,link2];
         if (link2 = m+2 ∧ list [i,1] ≦ t) ∨
            (link2 ≠ m+2 ∧ list [i,1] > t) then go to 4;
         else go to 1 end;
       else begin list [i,link2] := addr [1] end;
3:  j := addr [1];  list [j,link1] := i;
    list [j,link2] := flag;  if link2 = m+2 then
    first := addr [1];  go to 5;
4:  j := addr [1];  list [j,link1] := list [i,link1];
    list [i,link1] := list [k,link2] := addr [1];
    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
```