ALGORITHM 81
ECONOMISING A SEQUENCE 1
Brian H. Mayoh
Digital Computer Laboratory, University of Illinois,
Urbana, Ill.

procedure ECONOMISER 1 (desired property, costs, n, C);
\[\text{array costs; integer } n;\]
\[\text{Boolean procedure desired property;}\]
\[\text{Boolean array } C;\]
begin comment Given a finite, monotonely increasing
sequence of positive numbers, looked upon as prices, ECONOMISER 1 selects
the cheapest subsequence with a given property. The formal parameters are:
\textit{Desired property}, a function designator to answer the question: Does the
subsequence held in array } C \text{ possess the required property? } n \text{ is (number of
elements in the sequence) } + 1. \textit{Costs} is an array of size } [1:n].
\text{Costs}[1] \text{ to costs}[n-1] \text{ hold the numbers of the sequence and}
costs[n] \text{ is any arbitrary number greater than the sum of all
other elements of costs. } C \text{ is an array of the same size and indicates
a subsequence by the rule: } C[i] = \text{ element } i \text{ of the original
sequence is in the subsequence. At exit from ECONOMISER 1,
} C \text{ indicates the cheapest subsequence. It is supposed that the
original sequence has the desired property.}\]
\[\text{integer } d, j, k, \ell; \text{ real } i;\]
for \(j := 1 \text{ step 1 until } n \text{ do } C[j] := j = 1; \cdot d := 0;\]
reenter: \(d := d+1;\)
INSIDE: begin own real array prices [1:d];
\[\text{own Boolean array alternatives}[1:d, 1:n];\]
procedure ENTER SUCCESSORS;
begin \(k := n-1;\)
A: if \(\neg C[k]\) then
begin \(k := k-1;\) go to A end; \(i := 0;\)
for \(j := 1 \text{ step 1 until } n \text{ do }\)
begin alternatives[\(\ell, j]\)
\(:= j \neq k \land j \neq k-1 = C[j];\)
if alternatives[\(\ell, j]\) then
\(i := i + \text{costs}[j]\) end;
B: \(k := k-1;\)
go to if \(k = 0\) then find cheapest
\(\text{else if } C[k]\) then (if \(k = 0\) then
\(\text{find cheapest else B)}\)
\(\text{else if } k = 0\) then E
\(\text{else if } C[k-1]\) then D
\(\text{else find cheapest};\)
D: \(C[k-1] := \text{false};\)
E: \(C[k] := \text{true};\) go to reenter
end of ENTER SUCCESSORS;
i := 0; for \(j := 1 \text{ step 1 until } d \text{ do }\)
begin alternatives[\(\ell, j]\) := \(C[j];\) if \(C[j]\) then
\(i := i + \text{costs}[j]\) end;
prices[\(d]\) := \(i;\)
find cheapest: \(i := 0;\) for \(j := 1 \text{ step 1 until } d \text{ do }\)
begin if \(\text{prices}[j] < i\) then
begin \(\ell := j;\) \(i := \text{prices}[\ell]\) end
end;