Zdeněk Režný; Evžen Kindler

Algorithms. 31. PERMUT. Simple algorithm generating all permutations

Aplikace matematiky, Vol. 18 (1973), No. 3, 211–212

Persistent URL: http://dml.cz/dmlcz/103471

ALGORITMY

31. PERMUT

SIMPLE ALGORITHM GENERATING ALL PERMUTATIONS

Ing. ZDENĚK REŽNÝ, CSc., Dr. EVŽEN KINDLER, CSc.,
Biofysikální ústav Fakulty všeobecného lékařství Karlovy university,
Praha 2, Salmovská 3

The procedure *PERMUT* generates all permutations of the sequence $a_m, a_{m+1}, \ldots, a_{m+n-1}$. In every step of its work, it generates a permutation, assigned for the variables $a[m], \ldots, a[m+n-1]$. The first permutation is the original one, entering into the algorithm; after generating all the permutations, the procedure *PERMUT* restitutes the original order of values in the array a. Any permutation can be processed by the procedure P immediately after it has been generated. Besides the permutations, the number of their inversions is generated and assigned for the variable r*). If we do not need it, we can simplify the algorithm by omitting all statements where the variable r occurs.

```
procedure PERMUT(a, m, n, p);
value m, n;
array a; integer m, n; procedure P;
begin
        integer i, k, r; integer array j[1 : n − 1]; real b;
        for i := 1 step 1 until n − 1 do j[i] := i + m; r := 0;
    G : P; i := n − 1;
    H : if i ≦ 0 then go to D;
        if j[i] = m then go to F;
        b := a[j[i]]; a[j[i]] := a[j[i] − 1]; j[i] := j[i] − 1; a[j[i]] := b;
        r := r + 1; go to G;
    F : j[i] := j[i] + i; b := a[m];
        for k := m step 1 until m + i − 1 do a[k] : = a[k + 1];
        a[m + i] := b; r := r − i; i := i − 1; go to H;
    D:
end PERMUT;
```

*) The number of inversions in a permutation $a_{j_1}, a_{j_2}, \ldots, a_{j_n}$ is defined commonly as the number of all pairs of indices (i, k), $1 \leqq i, k \leqq n$, where $i < k, j_i > j_k$.

The array a may be of course of the type integer, in which case the variable b is to be declared integer too.

Example. Calling $PERMUT(a, 1, 4, p)$ where $a_i = i$ and P prints the generated permutation, followed by the number of its inversions in parentheses:

1234(0), 1243(1), 1423(2), 4123(3), 1324(1), 1342(2), 1432(3), 4132(4),
3124(2), 3142(3), 3412(4), 4312(5), 2134(1), 2143(2), 2413(3), 4213(4),
2314(2), 2341(3), 2431(4), 4231(5), 3214(3), 3241(4), 3421(5), 4321(6).

An interesting property of the algorithm may be seen which holds not only for the given example, but generally: the sequence of generated permutations has certain symmetrical attributes, namely that the sum of the pair of the numbers of the inversions of the i-th and $(n! - i + 1)$-th permutations is equal to $\binom{n}{2}$, and that the order of the i-th permutation is exactly inverse to the order of the $(n! - i + 1)$-th one $(i = 1, \ldots, n!)$.

The algorithm has been programmed and tested in the small computer ODRA 1013 [1] in the programming language MOST [2].

References

[1] *Černý, V., Půr J.:* Programmer's Manual on Automatic Computer ODRA 1013 (in Czech). Kancelářské stroje n. p., Hradec Králové 1967.

[2] *Szczepkowicz, J.:* Programming in the autocode MOST 1 (in Polish). ELWRO Publication 03—VI—1, Wroclaw.