プログラムのページ

担当　一　松　　信

6505. 順列の生成

釜范真也（立教大学理学部数学科）

順列の生成については，すでに本誌3巻4号，プログラム No. 6212 に，米田信夫氏による巧妙なプログラムがあるが，ここには辞書式順序を作りだすプログラムをあげたい．a_1, ……, a_n をはじめに

$a_1 \leqq a_2 \leqq \cdots\cdots \leqq a_n$

であるように並べる．これらを順次辞書式に並べかえた置換を生成し，最後に $a_1 \geqq a_2 \geqq \cdots\cdots \geqq a_n$ の状態になったら完了する．算法は次の通りである．この方法のよいところは，a_i のうちに同じ値があるときには，同じ値のものをいれかえた組は一度しか作らない点である．たとえば，$a_1=1$, $a_2=1$, $a_3=2$ ではじめれば，112, 121, 211 の3種の順列だけを生成する．

算法　1° a_1, a_2, ……, a_n を後から順にみて

$a_j < a_{j+1} \geqq a_{j+2} \geqq \cdots\cdots \geqq a_n$

である j をさがす，もし $a_1 \geqq a_2 \geqq \cdots\cdots \geqq a_n$ となっていれば完了である（下記プログラムの GYAKUTEN の所）．

2° a_{j+1}, ……, a_n のうち，a_j より大なもののうち最小なもの a_i を求める．これには後から順にみて

$a_j < a_i$, $a_{i+1} \geqq a_j$

である所をとればよい（ŌUT 1 の所）．

3° a_j と a_i とを入れかえる（ŌUT 2 の所）．

4° a_{j+1},……, a_n 全体を逆の順に並べかえる．すなわち a_{j+i} と a_{n+1-i} ($i=1,2,\cdots\cdots$; $i \leqq (n-j+1)/2$) とを入れかえる（CHANGE の所）．

この算法については，大学院の佐久間紘一氏に負うところが大きい．同氏に厚く感謝する．

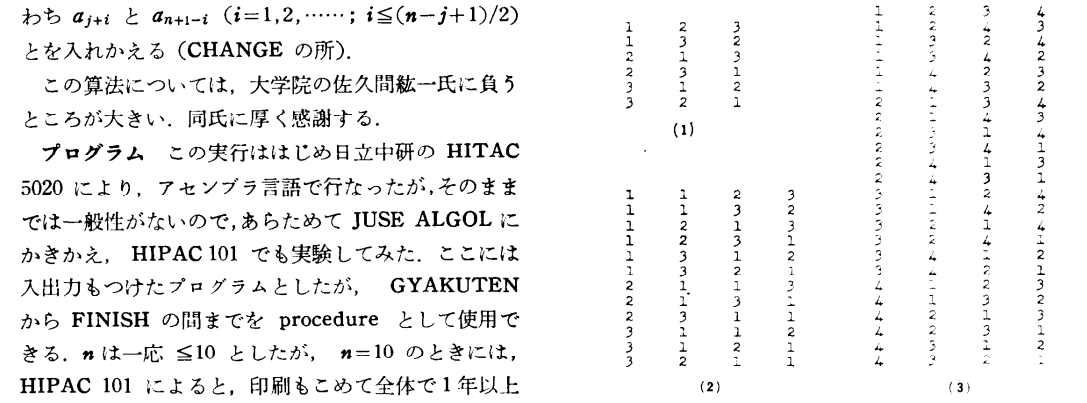
プログラム　この実行ははじめ日立中研の HITAC 5020 により，アセンブラ言語で行なったが，そのままでは一般性がないので，あらためて JUSE ALGOL にかきかえ，HIPAC 101 でも実験してみた．ここには入出力もつけたプログラムとしたが，GYAKUTEN から FINISH の間までを procedure として使用できる．n は一応 ≦10 としたが，$n=10$ のときには，HIPAC 101 によると，印刷もこめて全体で1年以上かかるという見積りになる．

順列生成プログラム

```
JUNRETSU: begin integer I, J, N, Q, W;
          integer array A[1 : 10];
          N:=Q:=0;
INPUT: for I:=1 step 1 until 10 do
       begin READINTEGER(W);
             if W<Q then go to INSATU;
             A[I]:=Q:=W; N:=N+1
       end;
INSATU: CRLF; for I:=1 step 1 until N do
        PRINTINTEGER(A[I]);
GYAKUTEN: for J:=N-I step -1 until 1 do
          if A[J]<A[J+1] then
          go to ŌUT 1;
          go to FINISH;
ŌUT 1: Q:=A[J]; for I:=N step -1 until J do
       if A[I]>Q then go to ŌUT 2;
ŌUT 2: A[J]:=A[I]; A[I]:=Q; I:=N;
CHANGE: J:=J+1; if J>=I then go to INSATU;
        W:=A[J]; A[J]:=A[I]; A[I]:=W;
        I:=I-1; go to CHANGE;
FINISH: CRLF
        end;
```

DATA: 1.2.3.0.　1.1.2.3.0.　1.2.3.4.0.

（注　0は区切りの記号）



順列生成プログラムの結果例

〔付記〕 このプログラムでは a_i を毎回新たに全部を並べかえているが，どこまですませたかを記録しておけば，多少能率を上げることもできる．

最後の CHANGE の部分は，5020 では前記の形でなく，じっさいには下記のような算法によった．⊕は bitwise addition (exclusive or) の演算である．この中の for の文は，いずれも 5020 の variable length 命令の repeat 機能を活用することにより，それぞれ 1 命令ですませえた．しかしこのままの形を JUSE ALGOL で書くと(たとえ⊕のサブルーチンを工夫してはさんでも)，さらに能率がおちるので，この方は前記の形にかきかえて実行した．

```
CHANGE: begin integer P;
        W:=N-J;Q:=W+2;
        if W≠Q* 2 then P:=A[N-Q];
        for I:=1 step 1 until W do
          A[J+I]:=A[J+I]⊕A[N+1-I];
        for I:=1 step 1 until Q do
          A[J+I]:=A[J+I]⊕A[N+1-I];
        if W≠Q* 2 then A[N-Q]:=P
        end;
```

6506. 辞書式順列 I

和田英一（東京大学工学部計数工学科）
浦部雍子（日本科学技術研修所）

0,1,2,……, $N-1$ の N 個の数字でつくった $N!$ 個の順列を辞書式にならべたときの M ($\leqq 0 M<N!$) 番目の順列 $P_M=(P_0, P_1, P_2, \cdots\cdots, P_{N-1})$ の P_I ($I=0,1,2,\cdots\cdots, N-1$) をもとめる．

まず M を $M=A_0\cdot 0!+A_1\cdot 1!+\cdots+A_{N-1}\cdot(N-1)!$ ただし $A_0=0$, $0\leqq A_1\leqq 1$, $0\leqq A_2\leqq 2, \cdots\cdots$, $0\leqq !A_{N-1}\leqq N-1$ とあらわす A_I をもとめ，添字の小さい方の A_I から順に，自分より大きい添字をもっている A_J について，添字の小さい方から順に比較し，それが自分に等しいか，自分より小さいとき，自分の数を一つふやすという操作をくりかえす．こうしてできた A の配列を反対にすると P_I がもとまる．$M\geqq N!$ のとき ERROR≠0 となり P_M はもとまらない．

ただし M は，この procedure を実行するとこわれる．以下のプログラムは ALGOLIP (OKITAC 5090 のための ALGOL) によるものである．array の上限が 99 になっているが，実際に必要な array の size は P[0: N−1], A[0: N−1] である．

(Beckenbach, E.F. 編 Applied Combinatorial Mathematics p. 21 参照)

```
begin integer N,M, ERROR;
  integer array P[0:99];
  procedure MTHPERMUTATION;
    begin integer I, J, Q;
    integer array A[0: 99];
      for I:=0 step 1 until N-1 do
        begin Q:=M÷(I+1);
              A[I]:=M-Q* (I+1);
              M:=Q
        end;
      ERROR:=M;
      if M=0 then
      for I:=0 step 1 until N-1 do
        begin
          for J:=I+1 step 1 until N-1 do
            if A[I]≧A[J] then A[I]:=A[I]+1;
          P[N-1-I]:=A[I]
        end
      end;
    comment driver program;
    begin integer I, PI;
      N:=READINTEGER;
L1:   M:=READINTEGER;
      PRINTINTEGER(M);
      MTHPERMUTATION;
      if ERROR=0 then
        begin
          PI:=0;
          for I:=0 step 1 until N-1 do
              PI:=PI* 10+P[I];
          PRINTINTEGER(PI)
        end
      else PRINTSTRING (' ERROR');
      CRLF (1); go to L1
  end
end
```

6507. 辞書式順列 II

和田英一（東京大学工学部計数工学科）
浦部雍子（日本科学技術研修所）

辞書式順列 I の逆である．P_I ($I=0, 1, \cdots\cdots, N-1$) であたえられた順列 (0, 1, 2, ……, $N-1$ よりなる) が辞書式順序にならべたとき，なん番目になるかを，NUMBER におく．もとの順列が正しければ ERROR =0 で，NUMBER がもとまり，正しくなければ，