BIT 2 (1962), 228–231

ON THE GENERATION OF PERMUTATIONS AND COMBINATIONS

MOK-KONG SHEN

Abstract.

In this paper a new method of generating permutations in lexicographical order is developed. Analysis shows that the method is superior to the known method of generation by addition. A simple method of generating combinations is also described.

Introduction.

In cetain fields of numerical investigations, e.g. in number theory, it is sometimes necessary to have procedures to generate permutations and combinations systematically. In the general case there are n given numbers or symbols to be operated upon. However, since any set of n elements may be mapped to the set of the first n natural numbers, the general problem is solved if one succeeds in generating permutations and combinations of the set of the n integers $1, 2, \ldots, n$ in lexicographical order.

The known method of generating permutations by addition has been improved by Howell [1] in a recent paper. In the following we develop an alternative method of generation by actually permuting the n integers (considered to be the digits of a n-base number) according to the results of applying a few simple criteria. This turns out to be a process requring a far less number of computing operations than the method of Howell. The resulting saving of computing time is of special importance in problems involving fairly large values of n.

The parallel problem of generating combinations is a relatively straight forward task. Nevertheless, as the method of generation has so far not been dealt with in the literature, we shall also treat it here in some detail.

Permutations.

Let us first consider a concrete example with five digits 1, 2, 3, 4 and 5. The first ten permutations in lexicographical order are:

No.	Permutation	No.	Permutation
1	1,2,3,4,5	6	1,2,5,4,3
2	1,2,3,5,4	7	1,3,2,4,5
3	1,2,4,3,5	8	1,3,2,5,4
4	1,2,4,5,3	9	1,3,4,2,5
5	1,2,5,3,4	10	1,3,4,5,2

Denoting the digits from left to right by k_1, k_2, k_3, k_4, k_5, we notice that in the change-over from No. 6 to No. 7 the following conditions are satisfied by No. 6:

$$k_3 > k_4 > k_5 ,$$
$$k_2 < k_3 .$$

These conditions point out that No. 6 is the largest permutation with 1, 2 as the first two digits. To obtain the next permutation we have therefore to increase k_2, i.e. to replace it by another digit. Since k_1 should remain unchanged in this stage, a choice is to be made from among the three digits k_3, k_4, k_5. Remembering that we are generating the permutations lexicographically, we have obviously to select the least of these three digits that is greater than k_2. In the present case $k_5=3$ meets our requirement. Interchanging thus k_2 with k_5 we obtain the permutation:

No. 6′ 1,3,5,4,2 .

This is however not the No. 7 desired. Noting that No. 6′ also satisfies the first of the two conditions satisfied by No. 6 above, we reverse the order of the digits k_3, k_4, k_5 of No. 6′ and obtain the permutation:

1,3,2,4,5 ,

which is exactly No. 7.

It is easy to see how the arguments may be applied to the general case. The procedure to generate permutations of n digits $1,2,\ldots,n$ lexciographically may be stated as follows:

a. Find the largest i such that $k_{i-1} < k_i$.
b. Find the largest j such that $k_{i-1} < k_j$.
c. Interchange k_{i-1} and k_j.
d. Reverse the order of the digits $k_i, k_{i+1}, \ldots, k_n$.

Combinations.

Suppose we have twelve digits $1,2,\ldots,12$ to be taken five at a time. The first ten combinations in lexicographical order are:

No.	Combination	No.	Combination
1	1,2,3,4,5	6	1,2,3,4,10
2	1,2,3,4,6	7	1,2,3,4,11
3	1,2,3,4,7	8	1,2,3,4,12
4	1,2,3,4,8	9	1,2,3,5,6
5	1,2,3,4,9	10	1,2,3,5,7

Since we are generating the combinations lexicographically, it is evident that:

$$k_5 \leqq 12, \ k_4 \leqq 11, \text{ etc.}$$

Observe the change-over from No. 8 to No. 9. We notice that No. 8 satisfies the condition:

$$k_5 = 12 .$$

This condition points out that No. 8 is the largest combination with 1, 2, 3, 4 as the first four digits. To obtain the next combination we have therefore to increase k_4 by unity, if the condition $k_4 \leqq 11$ stated above is thereby not violated. In the present case we change k_4 to 5 and put the next larger digit, 6, in k_5, obtaining the combination:

$$1,2,3,5,6 ,$$

which is exactly No. 9.

The procedure to generate combinations of n digits $1,2,\ldots,n$ taken m at a time in lexicographical order may be stated as follows:

a. Find the largest i such that $k_i < n-m+i$.
b. Add 1 to k_i.
c. Perform the substitutions:

$$k_{i+j} = k_i + j \quad \text{with } j = 1,2,\ldots,(m-i) .$$

Conclusions.

The procedures described above are believed to require the least possible numbers of operations for achieving their respective purposes. In order to convince ourselves in the case of permutations we note that Howell's method consists in the repeated addition of $n-1$ to the n-base number

$$(0,1,2,\ldots,n-1) = 0\cdot n^{n-1} + 1\cdot n^{n-2} + 2\cdot n^{n-3} + \ldots + (n-1)\cdot n^0 .$$

Since among the numbers generated successively only those not containing common digits represent the permutations desired, it is necessary to cast out the irrelevant numbers having two or more identical digits through a comparison digit by digit. The number of comparison operations involved differs from case to case, being dependent upon the number and the distribution of the identical digits, if any; the maximum number of such operations, $n(n-1)/2$, is required for every n-base number which contains no identical digits and thus proves to represent a permutation. In our method there are at most $2n-2$ comparison operations and $(n+1)/2$ exchange operations for each permutation generated, there being no counterpart of the irrelevant numbers generated in Howell's method. Thus compared, our method is evidently superior to the method of Howell. Even if we entirely ignored the computing time necessary for casting out the irrelevant numbers, there would still remain a significant saving of the computing time by employing our method in place of that of Howell, the saving being greater the greater the value of n is.

Let us further investigate how many irrelevant numbers are to be cast out in Howell's method. We note that the first and the last permutation are represented in Howell's method by the n-base numbers $(0, 1, \ldots, n-1)$ and $(n-1, n-2, \ldots, 0)$ respectively and that among the numbers generated by the addition process $n!$ of them represent the permutations desired. Hence there are

$$Z = \{(n-1)\cdot n^{n-1} + (n-2)\cdot n^{n-2} + \ldots + 0\cdot n^0 - \\ - [0\cdot n^{n-1} + 1\cdot n^{n-2} + \ldots + (n-1)\cdot n^0]\}/(n-1) + 1 - n!$$

irrelevant numbers generated in that method. It suffices for the reader to try a few values of n to see just how rapidly Z and hence the time involved to case out the irrelevant numbers grow with increasing values of n.

The procedures here described lend themselves very well to ALGOL-programming, the n digits being represented by an integer array.

REFERENCE

1. Howell, J. R. *Generation of Permutations by Addition*, Math. Comp., v. 16, 1962, p. 243–244.

POSTFACH 74,
MÜNCHEN 34, GERMANY