Management Science/Operations Research

Reflection-Free Permutations, Rosary Permutations, and Adjacent Transposition Algorithms

Mohit Kumar Roy
Jadavpur University

Key Words and Phrases: permutation, permutation generation, scheduling, combinatorial analysis
CR Categories: 5.39

Introduction

Several algorithms for the systematic generation of the permutations of n marks have been published. Recently Ord-Smith [1] has reviewed them. However, almost all of them are intended to generate all the $n!$ arrangements.

Corresponding to each member of the set of $n!$ possible arrangements, there is another member such that one read from left-to-right and the other read from right-to-left are identical. We shall call them "reflections" of each other. The set of $n!/2$ permutations of n marks with the property that the reflection of any member is not included in the set may be called a "reflection-free" set of permutations of the marks. The generation of such reflection-free sets of permutations is fundamental to some problems such as scheduling problems.

In this note we show how the adjacent transposition algorithms [2, 3, 4] can be used to generate reflection-free permutations and rosary permutations very efficiently. Rosary permutations have been discussed in detail by Harada [5].

Generation of Reflection-Free Permutations

The adjacent transposition algorithm of Trotter [2] or its variation, the algorithm of Johnson [3, 4], have the property that the first $n!/2$ arrangements generated using either of these algorithms form a reflection-free set. A proof of this is given in the Appendix. Therefore, to generate reflection-free permutations, either of these algorithms may be modified to produce only the first $n!/2$ arrangements.

It can be verified easily that no other standard method for generating permutations has this property. Moreover, the adjacent transposition algorithms are useful in problems in which there is a cost attached to moving the marks [4]. They are also quite fast [1]. These facts coupled with the said property make them very efficient algorithms for generating reflection-free permutations.

Author's address: Computer Centre, Jadavpur University, Calcutta 32, India.

Generation of Rosary Permutations

Rosary permutations are, in fact, equivalence classes of reflection-free permutations of n marks, where two permutations are equivalent if they differ only by a cyclic order. Therefore rosary permutations can be generated as follows: (1) place the first mark in first position; and (2) generate the $(n-1)!/2$ reflection-free permutations of the remaining $(n-1)$ marks using the adjacent transposition algorithm.

This algorithm appears to be more efficient than those presented by Harada [5] and Read [6].

Appendix

Let the permutations be serially numbered (starting from 0) in the order they are generated by either of the adjacent transposition algorithms. Let N^0_{n-1} denote the serial number of a particular arrangement of $(n-1)$ marks and N^1_{n-1} that of its reflection. The process of generating the permutations of n marks can be considered as taking the permutations of $(n-1)$ marks one-by-one and generating n arrangements from each of them, by placing the new mark either at the extreme left or at the extreme right, and then passing it through the other marks. Let $N_n{}^0$ be the serial number of one of the arrangements of n marks produced from N^0_{n-1}. Then

$$N_n{}^0 = nN^0_{n-1} + d_n \text{ where } 0 \leq d_n \leq n-1. \tag{1}$$

It is easy to observe that the reflection of $N_n{}^0$, denoted by $N_n{}^1$, will be one of those n arrangements produced from N^1_{n-1} and

$$N_n{}^1 = nN^1_{n-1} + e_n \tag{2}$$

where

$$e_n = \begin{cases} d_n & \text{if } N^0_{n-1} \text{ and } N^1_{n-1} \text{ are both not odd or even at the same time,} \\ (n-1) - d_n & \text{if } N^0_{n-1} \text{ and } N^1_{n-1} \text{ are both odd or both even.} \end{cases}$$

By definition $N_1{}^0 = N_1{}^1 = 0$, and therefore $e_2 = 1 - d_2$. Repeated uses of (1) and (2) give respectively

$$N_n{}^0 = n![d_2/2!+d_3/3!+\cdots+d_n/n!] \tag{3}$$
$$N_n{}^1 = n![e_2/2!+e_3/3!+\cdots+e_n/n!] \tag{4}$$

where $0 \leq d_j \leq j-1$ and $0 \leq e_j \leq j-1$; $j = 2, 3, \ldots, n$.

Now for $0 \leq N_n{}^0 < n!/2$, from (3) we have $d_2 = 0$, and therefore $e_2 = 1$. So from (4), $N_n{}^1 \geq n!/2$.

This proves the property that the set of first $n!/2$ permutations are reflection-free.

Received June 1972; revised January 1973

References
1. Ord-Smith, R.J. Generation of permutation sequences: Pt. 1. *Comp. J. 13*, 2 (May 1970), 152–155.
2. Trotter, H.F. Algorithm 115, Perm. *Comm. ACM 5*, 8 (Aug. 1962), 434–435.
3. Johnson, S.M. Generation of permutations by adjacent transposition. *Math. Comp. 17* (1963), 282–285.
4. Lehmer, D.H. The machine tools of combinatorics. In *Applied Combinatorial Mathematics*, E.F. Beckenback (Ed.), Wiley, New York, 1964, pp. 5–31.
5. Harada, K. Generation of rosary permutations expressed in Hamiltonian circuits. *Comm. ACM 14*, 6 (June 1971), 373–379.
6. Read, R.C. A note on the generation of rosary permutations. *Comm. ACM 15*, 8 (Aug. 1972), 775.

Scientific Applications

Concerning Music and Computer Composition in Computational Linguistics

Clyde H. McMorrow
Control Data Corporation

Key Words and Phrases: artificial intelligence, heuristic programming, models of cognitive processes, computer music, computer composition, music theory
CR Categories: 3.44, 3.65

The discussions of both Moorer and Smoliar (*Comm. ACM 15*, 11 (Nov. 1972), 1000–1001) seem to concern more the emotional criterion involved in the generation of music than the practical implementation of machine-oriented music generators. Assuming that the mechanics of a given piece of music can be fairly well defined by rules of counterpoint, conventional harmony, random walks, or some other functional system, the only problem is the introduction of "emotional content." It is this constraint that is missing in Moorer's analysis. By having a system with a missing constraint, it is fairly easy to see how that might make a system "sacred" or "unattainable by mechanical means." It is just this criterion that a composer uses to "break the rules, to compose in erratic manners, or to add those nuances to the music that make the piece outstanding." Indeed it is by the composer's judicious use of the emotional content that we subjectively rate him. Therefore, while it is necessary to continue research on the mathematical analysis of music, it is also important to work in the area of the emotional composition of music. I think it is a bit unfair of Smoliar to completely discount the work of Moorer merely for the sake of a concept not completely understood and almost completely disregarded by present-day computer composers. The examples presented in Smoliar's paper are, as pointed out by Moorer, examples of the mechanical composition of music in which the emotional theme tends to be constant throughout a given composition and throughout the work of a given composer (i.e. Vivaldi Concerti, Canto, etc.).

Author's address: Eagle Signal Corp., 736 Federal Road, Davenport, IA 52802.

Now the remaining question. How does one numerically define the emotional content of a composition? Unfortunately, or fortunately for future Ph.D. candidates, I don't have this worked out. I have done some research in this area, but it was a couple of years back and has given way to work in speech recognition and making a living. But if someone were interested in pursuing this, my work was in the analysis of the coincidence of overtones and the possible relationship to subjective analysis of musical passages. This work is incomplete and inconclusive and may be inaccurate, but it is available to anyone who is interested.

Received November 1972

Another Comment on Computer Music

R.L. Wexelblat
Bell Laboratories

I would like to add a comment or two to the short communications on computer music in the November Communications.

It would appear that both Smoliar and Moorer have bypassed the essential difference between the "human" and the "mechanical" composer which lies not in the technique used to generate the music but in the ability to discriminate between "good" and "bad" melodies or harmonies and to adapt or modify one that is unsatisfactory. This discrimination is not the musical taste mentioned in the penultimate paragraph of Moorer's original article, but is an absolute value judgment exercised by the composer on his output.

Until such time as the software composer can image the action of its meatware rival in this area, I hold out little hope for computer compositions, either classical or pop, that will advance the state of musical art.

One other topic of minor relevance. Prof. Moorer

Author's address: Bell Laboratories, Holmdel, NJ 07733.

raised the question of how long he, Smoliar, and Illiac had been working on computer music. I don't know about Moorer and Illiac, but Smoliar first wrote a program to generate "Pseudo-Bach" melodies when he was employed at the Moore School of Electrical Engineering, University of Pennsylvania, during the summer of 1964.

Education

A Comment on the Practical Aspects of Computer Science Education

Robert G. Estell

Key Words and Phrases: education, computer engineering, computer sciences curriculum, systems design

CR Categories: 1.52, 6.0

Louis Fein (Communications, Jan. 1973, pp. 45-46) is correct in the charge that "the industry" does not know what its educational needs are. May I suggest a reason, and a remedy.

Spokesmen for the industry tend to be its managers; they in turn tend to have one of three backgrounds: (1) up through the ranks of computing; or (2) graduates of sophisticated computer science curricula; or (3) outstanding histories of management successes in other areas.

In the first class, these managers tend to be overawed by those of superior academic achievement; in the second class, they tend to be enamored with higher education per se; in the third class, they are frequently intimidated by computing jargon.

An obvious and valid long-term solution is to develop managers who have experience in at least two, and preferably all three, of these areas.

For the immediate future, some leading universities might offer short courses (i.e. average of one week) in a broad spectrum of computer science subjects—according to needs suggested by the schools and the industry. The value of each course will, over a decade or so, be directly proportional to the number of attendees—especially for those courses where employers have sent others before.

Undoubtedly, this value curve will be biomodal—reflecting the different needs of theory and application.

Author's address: 1227 Tourmaline Street, San Diego, CA 92109.

Operating Systems — B. Randell, Editor

WYLBUR: An Interactive Text Editing and Remote Job Entry System

Roger Fajman and John Borgelt
Stanford University Computation Center

WYLBUR is a comprehensive system for manipulating all kinds of text, such as computer programs, letters, and manuscripts, using typewriter terminals connected to a computer. It has facilities for remote job entry and retrieval as well as facilities for text alignment and justification. A powerful method for addressing text by content is provided. This paper describes the external appearance of WYLBUR as well as its internal structure. A short description of the major features of ORVYL, a general purpose time-sharing system which operates in conjunction with WYLBUR, is also included.

Key Words and Phrases: text editing, time-sharing, online text editing, interactive text editing, terminal, remote terminal, terminal system, interactive terminal, remote job entry, remote job retrieval, program preparation, document preparation, data entry, content addressing

CR Categories: 3.70, 3.79, 4.10, 4.19, 4.41, 4.43, 4.49

Authors' addresses: Roger Fajman, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213; John Borgelt, WYLBUR Systems Inc., 1520 Page Mill Rd., Palo Alto, CA 94304.