ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Discrete Mathematics ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

DISCRETE MATHEMATICS

www.elsevier.com/locate/disc

Gray code for permutations with a fixed number of cycles

Jean-Luc Baril

LE2I UMR-CNRS 5158, Université de Bourgogne, B.P. 47 870, 21078 DIJON-Cedex, France

Received 12 January 2006; received in revised form 22 August 2006; accepted 2 September 2006

Abstract

We give the first Gray code for the set of n-length permutations with a given number of cycles. In this code, each permutation is transformed into its successor by a product with a cycle of length three, which is optimal. If we represent each permutation by its transposition array then the obtained list still remains a Gray code and this allows us to construct a constant amortized time (CAT) algorithm for generating these codes. Also, Gray code and generating algorithm for n-length permutations with fixed number of left-to-right minima are discussed.

Keywords: Restricted permutations; Gray codes; Generating algorithms; Transposition array

1. Introduction

Various study have been made on generating algorithms for permutations and their restrictions (with given *ups* and *downs* [5,10], derangements [1], involutions [6], and fixed-point free involutions [16]) or their generalizations (multiset permutations [15]). See [4,12] for surveys of permutation generation methods. At [7] is given an implementation of Taylor and Ruskey [11] generating algorithm for n-length permutations with k cycles. However, the generating order is neither lexicographic nor Gray code order. On the other hand, these codes also called permutation codes or arrays have some applications for power line communication [2].

Let S_n be the set of all permutations of length $n \geqslant 1$. We represent permutations in one-line notation, i.e. if $i_1, i_2, \ldots, i_n$ are n distinct values in $[n] = \{1, 2, \ldots, n\}$, we denote the permutation σ by the sequence $(i_1, i_2, \ldots, i_n)$ if $\sigma(k) = i_k$ for $1 \leqslant k \leqslant n$. Moreover, if $\gamma = (\gamma(1), \gamma(2), \ldots, \gamma(n))$ is an n-length permutation then the *composition* (*or product*) $\gamma \cdot \sigma$ is the permutation $(\gamma(\sigma(1)), \gamma(\sigma(2)), \ldots, \gamma(\sigma(n)))$. In S_n, a *k-cycle* $\sigma = \langle i_1, i_2, \ldots, i_k\rangle$ is an n-length permutation verifying $\sigma(i_1) = i_2, \sigma(i_2) = i_3, \ldots, \sigma(i_{k-1}) = i_k, \sigma(i_k) = i_1$ and $\sigma(j) = j$ for $j \in [n]\backslash\{i_1, \ldots, i_k\}$; and the *domain* of σ is the set $\{i_1, \ldots, i_k\}$. In particular, a *transposition* (i.e. a 2-cycle) has domain of cardinality two. It is well known that each n-length permutation can be uniquely decomposed in a product of cycles with disjoint domains. For $1 \leqslant k \leqslant n$, we denote by $S_{n,k}$ the set of all n-length permutations which admit a decomposition in a product of k (disjoint) cycles and $\{S_{n,k}\}_{1 \leqslant k \leqslant n}$ forms a partition for S_n. The cardinality of $S_{n,k}$ is given by the signless Stirling numbers of the first kind $s(n, k)$ satisfying:

$$s(n,k) = (n-1) \cdot s(n-1,k) + s(n-1,k-1) \tag{1}$$

with the initial conditions $s(n, k) = 0$ if $n \leqslant 0$ or $k \leqslant 0$, except $s(0, 0) = 1$. See for instance [13,18].

E-mail address: barjl@u-bourgogne.fr.

0012-365X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2006.09.007

A *Gray code* for a sequence set S is an ordered list for S in which the Hamming distance between any two consecutive sequences in the list (the number of positions in which they differ) is bounded by a constant, independent on the size of the sequences. If this constant is minimal then the code is called *optimal*. A Gray code is *cyclic* if the Hamming distance between the first and the last sequence is also bounded by this constant.

Now we introduce some notations concerning lists.

If $\mathscr{S}$ is a list then $\overline{\mathscr{S}}$ is the list obtained by reversing $\mathscr{S}$, and first($\mathscr{S}$) (last($\mathscr{S}$), respectively) is the first (last respectively) element of the list, and obviously first($\mathscr{S}$) = last($\overline{\mathscr{S}}$) and first($\overline{\mathscr{S}}$) = last($\mathscr{S}$); $\mathscr{S}^{(i)}$ is the list $\mathscr{S}$ if i is even, and $\overline{\mathscr{S}}$ if i is odd; if $\mathscr{S}_1$ and $\mathscr{S}_2$ are two lists, then $\mathscr{S}_1 \circ \mathscr{S}_2$ is their concatenation, and generally $\bigcirc_{i=1}^{m} \mathscr{S}_i$ is the list $\mathscr{S}_1 \circ \mathscr{S}_2 \circ \cdots \circ \mathscr{S}_m$. Similarly, $\bigcirc_{i=m}^{1} \mathscr{S}_i$ is the list $\mathscr{S}_m \circ \mathscr{S}_{m-1} \circ \cdots \circ \mathscr{S}_1$.

In the following, $\mathscr{S}_{n,k}$ will denote our Gray code for the set $S_{n,k}$, $f_{n,k} = \text{first}(\mathscr{S}_{n,k})$ and $\ell_{n,k} = \text{last}(\mathscr{S}_{n,k})$.

Remark that if $\sigma \in S_n$ has k cycles in its (unique) decomposition in cycles with pairwise disjoint domains, and $\tau = \langle i, j \rangle$, $i \neq j$ a transposition in S_n, then the permutation $\sigma \cdot \tau$ has $k-1$ or $k+1$ cycles in its decomposition. Indeed, if i and j belong to the domain of the same cycle in σ then this cycle is splited into two cycles in $\sigma \cdot \tau$; otherwise two cycles of σ merge into a single cycle in $\sigma \cdot \tau$. This shows that there does not exist a Gray code for $S_{n,k}$ such that two successive permutations differ in less than three positions.

In this paper, we give the first Gray code for the set $S_{n,k}$. In this code successive permutations differ in three positions (or equivalently, by a product with a three-length cycle) and so, by the remark above it is optimal. By representing each permutation by its transposition array we provide an other Gray code which allows us to construct a constant amortized time (CAT) algorithm for generating these codes. Actually, this second Gray code lists n-length permutations with exactly k left-to-right minima in inversion array representation.

2. Preliminaries

In this section we show how $S_{n,k}$ can be recursively constructed from $S_{n-1,k}$ and $S_{n-1,k-1}$ which also gives a constructive proof of the counting relation (1). We also give three lemmas, crucial in our construction of the code.

Let $\gamma \in S_{n-1,k}$ be an $(n-1)$-length permutation with k cycles, $n \geqslant 2$, $n-1 \geqslant k \geqslant 1$; let also i be an integer, $1 \leqslant i < n$. If we denote by σ the permutation in S_n obtained from γ by replacing the entry in position i by n and appending this entry in the nth position, then σ is an n-length permutation with k cycles.

Similarly, if $\gamma \in S_{n-1,k-1}$ is an $(n-1)$-length permutation with $(k-1)$ cycles, $n \geqslant k \geqslant 2$, and if σ denotes the permutation in S_n obtained from γ by appending n in the nth position, then σ is an n-length permutation with k cycles. Moreover, each permutation in $S_{n,k}$, $n \geqslant 2$, can uniquely be obtained by one of these two constructions.

For $n, k \geqslant 1$, let $S'_{n,k}$ be the set of n-length permutations with k cycles where n is a fixed point (i.e. $\gamma(n) = n$) and $S''_{n,k} = S_{n,k} \backslash S'_{n,k}$ is its complement. The next definition formalizes the two constructions above.

Then the functions ϕ_n and ψ_n defined below induce a bijection between $S_{n-1,k-1}$ and $S'_{n,k}$ on one hand and between $[n-1] \times S_{n-1,k}$ and $S''_{n,k}$ on the other.

Definition 1.

(1) For $1 \leqslant k < n$, an integer $i \in [n-1]$ and a permutation $\gamma \in S_{n-1,k}$, we define an n-length permutation $\sigma = \psi_n(i, \gamma)$ by

$$\sigma(j) = \begin{cases} \gamma(i) & \text{if } j = n, \\ n & \text{if } j = i, \\ \gamma(j) & \text{otherwise.} \end{cases}$$

(2) For $n \geqslant k \geqslant 2$ and a permutation $\gamma \in S_{n-1,k-1}$, we define an n-length permutation $\sigma = \phi_n(\gamma)$ by

$$\sigma(j) = \begin{cases} n & \text{if } j = n, \\ \gamma(j) & \text{otherwise.} \end{cases}$$

Remark that with i and γ as above, it is easy to see that

- if $\gamma \in S_{n-1,k}$, $\psi_n(i, \gamma) \in S''_{n,k}$ and $\psi_n : [n-1] \times S_{n-1,k} \to S''_{n,k}$ is a bijection; and
- if $\gamma \in S_{n-1,k-1}$, $\phi_n(\gamma) \in S'_{n,k}$ and $\phi_n : S_{n-1,k-1} \to S'_{n,k}$ is a bijection.

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

So, the cardinality of $S_{n,k}$ is given by $s(n,k)=\text{card}(S'_{n,k})+\text{card}(S''_{n,k})=s(n-1,k-1)+(n-1)\cdot s(n-1,k)$ which is a combinatorial proof of (1).

In the following, we will omit the subscript n for the functions ϕ_n and ψ_n, and it should be clear from context. Also, we extend the functions ϕ and ψ in a natural way to sets and lists of permutations. Moreover for $i\in[n-1]$ and $\mathscr{S}$ a list of $(n-1)$-length permutations we have $\psi(i,\overline{\mathscr{S}})=\overline{\psi(i,\mathscr{S})}$, $\psi(i,\text{first}(\mathscr{S}))=\text{first}(\psi(i,\mathscr{S}))$, and $\psi(i,\text{last}(\mathscr{S}))=\text{last}(\psi(i,\mathscr{S}))$. If we do not consider the parameter i, we obtain similar results for the function ϕ.

Now, we give some elementaries results which are crucial in the construction of our Gray code.

Lemma 2. *Let γ be an $(n-1)$-length permutation, if $n\geqslant 3$ and $1\leqslant i,j\leqslant n-1$, $i\neq j$ then $\psi(i,\gamma)=\psi(j,\gamma)\cdot\langle i,j,n\rangle$.*

Proof. For each $(n-1)$-length permutation $\gamma=(\gamma_1,\gamma_2,\ldots,\gamma_{n-1})$, we have $\psi(j,\gamma)=(\gamma_1,\ldots,\gamma_{j-1},n,\gamma_{j+1},\ldots,\gamma_{n-1},\gamma_j)$. Thus, we obtain

$$\begin{aligned}\psi(j,\gamma)\cdot\langle i,j,n\rangle&=(\gamma_1,\ldots,\gamma_{i-1},n,\gamma_{i+1},\ldots,\gamma_{n-1},\gamma_i)\\&=\psi(i,\gamma).\qquad\square\end{aligned}$$

Lemma 3. *Let γ and σ be two $(n-1)$-length permutations satisfying $\sigma=\gamma\cdot\langle i,j,k\rangle$ with i,j,k pairwise different, $1\leqslant i,j,k\leqslant n-1$, $n\geqslant 3$* then $\psi(j,\sigma)=\psi(k,\gamma)\cdot\langle i,j,k\rangle$.

Proof.

$$\begin{aligned}\psi(j,\sigma)&=\psi(j,\gamma\cdot\langle i,j,k\rangle)\\&=\psi(j,(\gamma_1,\ldots,\gamma_{i-1},\gamma_j,\gamma_{i+1},\ldots\gamma_{j-1},\gamma_k,\gamma_{j+1},\ldots,\gamma_{k-1},\gamma_i,\gamma_{k+1},\ldots,\gamma_{n-1}))\\&=(\gamma_1,\ldots,\gamma_{i-1},\gamma_j,\gamma_{i+1},\ldots\gamma_{j-1},n,\gamma_{j+1},\ldots,\gamma_{k-1},\gamma_i,\gamma_{k+1},\ldots,\gamma_{n-1},\gamma_k)\end{aligned}$$

and

$$\begin{aligned}\psi(j,\sigma)\cdot\langle i,k,j\rangle&=(\gamma_1,\ldots,\gamma_{i-1},\gamma_i,\gamma_{i+1},\ldots\gamma_{j-1},\gamma_j,\gamma_{j+1},\ldots,\gamma_{k-1},n,\gamma_{k+1},\ldots,\gamma_{n-1},\gamma_k)\\&=\psi(k,\gamma).\qquad\square\end{aligned}$$

Lemma 4. *Let γ be an $(n-1)$-length permutation and $\sigma=\gamma\cdot\langle i,j\rangle$ with $i\neq j$, $1\leqslant i,j\leqslant n-1$, $n\geqslant 3$ then $\psi(i,\sigma)=\phi(\gamma)\cdot\langle i,n,j\rangle$.*

Proof.

$$\begin{aligned}\psi(i,\sigma)&=\psi(i,\gamma\cdot\langle i,j\rangle)\\&=\psi(i,(\gamma_1,\ldots,\gamma_{i-1},\gamma_j,\gamma_{i+1},\ldots,\gamma_{j-1},\gamma_i,\gamma_{j+1},\ldots,\gamma_{n-1}))\\&=(\gamma_1,\ldots,\gamma_{i-1},n,\gamma_{i+1},\ldots,\gamma_{j-1},\gamma_i,\gamma_{j+1},\ldots,\gamma_{n-1},\gamma_j)\end{aligned}$$

and

$$\begin{aligned}\psi(i,\sigma)\cdot\langle i,j,n\rangle&=(\gamma_1,\ldots,\gamma_{i-1},\gamma_i,\gamma_{i+1},\ldots,\gamma_{j-1},\gamma_j,\gamma_{j+1},\ldots,\gamma_{n-1},n)\\&=\phi(\gamma).\qquad\square\end{aligned}$$

3. The gray code

From the remark following Definition 1 results that the set $S_{n,k}$ can be written as:

$$S_{n,k}=\bigcup_{i=1}^{n-1}\psi(i,S_{n-1,k})\cup\phi(S_{n-1,k-1})\tag{2}$$

with $\phi(S_{n,0})$ and $\psi(i,S_{n,n+1})$ empty.

If $\mathscr{S}$ is a list of permutations where any two consecutive permutations differ in three positions then so is the image of $\mathscr{S}$ by ψ or ϕ. Therefore, it is natural to look for a Gray code for the set $S_{n,k}$ of the form

$$\mathscr{S}_1\circ\mathscr{S}_2\circ\cdots\circ\mathscr{S}_\ell\circ\mathscr{T}\circ\mathscr{S}_{\ell+1}\circ\cdots\circ\mathscr{S}_{n-1},\tag{3}$$

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

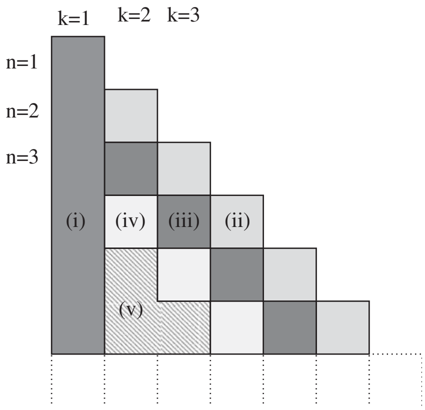

Fig. 1. The five cases for our Gray code.

where

- $\mathscr{T}$ is the list $\phi(\mathscr{S}_{n-1,k-1})$ or its reverse, and
- $\mathscr{S}_i$ is $\psi(j, \mathscr{S}_{n-1,k})$ or its reverse, for some j.

Notice that (3) is the ordered counterpart of (2).

Primarily, there are three difficulties to construct such a Gray code:

- for each $\mathscr{S}_i$, as in the last point above, we must determine a $j \leqslant n-1$ to apply ψ to $\mathscr{S}_{n-1,k}$,
- decide for each list if it is reversed or not,
- find the place where $\mathscr{T}$ must be inserted.

In the next we construct a Gray code of the form given by (3) according to the following cases (see Fig. 1):

(i) $k = 1 \leqslant n$,
(ii) $2 \leqslant k = n$,
(iii) $2 \leqslant k = n-1$,
(iv) $2 \leqslant k = n-2$,
(v) other cases,

and computer tests enable us to think that there is not simpler expression of such a Gray code.

For each case above, we give a recursive definition for $\mathscr{S}_{n,k}$, an ordered list for the set $S_{n,k}$ and we provide its first $f_{n,k}$ and last element $\ell_{n,k}$. $\mathscr{S}_{n,k}$ is the concatenation of n lists as in (3) and we prove that it is a Gray code by showing that there is a 'smooth' transition between successive sublists, that is, the last permutation of a sublist and the first one of the next sublist differ by a product with a three-length cycle. Recall that, by the remark in introduction, the Gray code will be optimal.

3.1. *The case $k = 1$, $n \geqslant 1$*

For $n \geqslant 1$ we define,

$$\mathscr{S}_{n,1} = \begin{cases} (1) & \text{if } n = 1, \\ (2,1) & \text{if } n = 2, \\ (3,1,2), \underline{(2,3,1)} & \text{if } n = 3, \\ \psi(1, \mathscr{S}_{3,1}) \circ \psi(2, \overline{\mathscr{S}_{3,1}}) \circ \psi(3, \overline{\mathscr{S}_{3,1}}) & \text{if } n = 4, \\ \psi(1, \mathscr{S}_{n-1,1}) \circ \bigcirc_{i=n-1}^{4} \psi(i, \mathscr{S}_{n-1,1})^{(i+1)} \circ \psi(2, \mathscr{S}_{n-1,1}) \circ \psi(3, \overline{\mathscr{S}_{n-1,1}}) & \text{if } n \geqslant 5. \end{cases} \qquad \text{(a)}$$

Remark that the function ϕ does not appear in relation (a).

The lemma below gives the first and the last permutation of the list $\mathscr{S}_{n,1}$.

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

Lemma 5. *If* $n \geqslant 4$ *then*

(1) $f_{n,1} = (n, 1, 2, 3, \ldots, n-2, n-1)$,
(2) $\ell_{n,1} = (n-1, 1, n, 3, \ldots, n-2, 2)$.

Proof.

(1) $\mathcal{S}_{2,1}$ is the list $(2, 1)$ and more generally relation (a) gives $f_{n,1} = \psi(1, f_{n-1,1})$, for $n \geqslant 3$. The recurrence on n completes the proof.
(2) Similarly, we have for $n \geqslant 4$, $\ell_{n,1} = \text{last}(\overline{\psi(3, \mathcal{S}_{n-1,1})}) = \psi(3, f_{n-1,1}) = (n-1, 1, n, 3, \ldots, n-2, 2)$. □

Proposition 6. *The list* $\mathcal{S}_{n,1}$ *defined by* (a), $n \geqslant 1$, *is a Gray code.*

Proof. By Lemma 2, we have for $n \geqslant 3$ and $i \neq j$:

$$\begin{aligned}\text{last}(\psi(i, \mathcal{S}_{n-1,1})) &= \psi(i, \text{last}(\mathcal{S}_{n-1,1})) \\ &= \psi(j, \text{last}(\mathcal{S}_{n-1,1})) \cdot \langle i, j, n\rangle \\ &= \text{last}(\psi(j, \mathcal{S}_{n-1,1})) \cdot \langle i, j, n\rangle \\ &= \text{first}(\psi(j, \overline{\mathcal{S}_{n-1,1}})) \cdot \langle i, j, n\rangle.\end{aligned}$$

Similarly, $\text{last}(\psi(i, \overline{\mathcal{S}_{n-1,1}})) = \text{first}(\psi(j, \mathcal{S}_{n-1,1})) \cdot \langle i, j, n\rangle$.

By Lemma 3,

$$\begin{aligned}\text{last}(\psi(1, \mathcal{S}_{n-1,1})) &= \psi(1, \text{last}(\mathcal{S}_{n-1,1})) \\ &= \psi(1, (n-2, 1, n-1, 3, \ldots, n-3, 2)) \\ &= \psi(1, \text{first}(\mathcal{S}_{n-1,1}) \cdot \langle 1, n-1, 3\rangle) \\ &= \text{first}(\psi(n-1, \mathcal{S}_{n-1,1})) \cdot \langle 1, n-1, 3\rangle.\end{aligned}$$

Notice that the transition between $\psi(1, \mathcal{S}_{n-1,1})$ and $\overline{\psi(n-1, \mathcal{S}_{n-1,1})}$ is given in the first point of the proof by setting $i = 1$ and $j = n-1$. □

3.2. The case $k = n$, $n \geqslant 2$

$$\text{Obviously, } \mathcal{S}_{n,n} \text{ is the single element list } (1, 2, \ldots, n-1, n) \text{ for } n \geqslant 2, \tag{b}$$

and in this case, there is nothing to do.

3.3. The case $k = n-1$, $n \geqslant 3$

For $n \geqslant 3$ we define

$$\mathcal{S}_{n,n-1} = \begin{cases} \psi(1, \mathcal{S}_{2,2}) \circ \psi(2, \mathcal{S}_{2,2}) \circ \phi(\mathcal{S}_{2,1}) & \text{if } n = 3, \\ \psi(1, \mathcal{S}_{n-1,n-1}) \circ \bigcirc_{\substack{i=n-1 \\ i \neq n-2}}^{2} \psi(i, \mathcal{S}_{n-1,n-1}) \circ \psi(n-2, \mathcal{S}_{n-1,n-1}) \circ \phi(\overline{\mathcal{S}_{n-1,n-2}}) & \text{otherwise.} \end{cases} \tag{c}$$

Lemma 7. *If* $n \geqslant 3$ *then*

(1) $f_{n,n-1} = (n, 2, 3, \ldots, n-2, n-1, 1)$,
(2) $\ell_{n,n-1} = (n-1, 2, 3, \ldots, n-2, 1, n)$.

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

Proof.

(1) By relation (c), $\mathscr{S}_{3,2}$ is the list (3, 2, 1), (1, 3, 2), (2, 1, 3) and more generally $f_{n,n-1} = \text{first}(\psi(1, \mathscr{S}_{n-1,n-1}))$. So, we obtain $f_{n,n-1} = \psi(1, \overline{f_{n-1,n-1}}) = (n, 2, 3, \ldots, n-2, n-1, 1)$ with a recurrence on n.
(2) Similarly, $\ell_{n,n-1} = \text{last}(\overline{\phi(\mathscr{S}_{n-1,n-2})}) = \phi(f_{n-1,n-2}) = (n-1, 2, 3, \ldots, n-2, 1, n)$. □

Proposition 8. *The list $\mathscr{S}_{n,n-1}$ defined by* (c), $n \geqslant 3$, *is a Gray code.*

Proof. By Lemma 2, and since $\psi(i, \mathscr{S}_{n-1,n-1})$ has only one element, the transition between $\psi(i, \mathscr{S}_{n-1,n-1})$ and $\psi(j, \mathscr{S}_{n-1,n-1})$, $1 \leqslant i, j \leqslant n-1$ is a product with a three-length cycle. So, it remains to verify that $\text{last}(\psi(n-2, \mathscr{S}_{n-1,n-1}))$ differs from $\text{first}(\phi(\overline{\mathscr{S}_{n-1,n-2}}))$ by a three-cycle:

$$\begin{aligned}\text{last}(\psi(n-2, \mathscr{S}_{n-1,n-1})) &= \psi(n-2, (1, 2, 3, \ldots, n-1)) \\ &= \psi(n-2, (n-2, 2, 3, \ldots, n-3, 1, n-1) \cdot \langle 1, n-2 \rangle),\end{aligned}$$

and by Lemma 4,

$$\begin{aligned}\text{last}(\psi(n-2, \mathscr{S}_{n-1,n-1})) &= \phi((n-2, 2, 3, \ldots, n-3, 1, n-1)) \cdot \langle 1, n, n-2 \rangle \\ &= \phi(\text{first}(\overline{\mathscr{S}_{n-1,n-2}})) \cdot \langle 1, n, n-2 \rangle \\ &= \text{first}(\phi(\overline{\mathscr{S}_{n-1,n-2}})) \cdot \langle 1, n, n-2 \rangle. \quad \square\end{aligned}$$

3.4. The case $k = n-2$, $n \geqslant 4$

In this case we define for $n \geqslant 4$,

$$\mathscr{S}_{n,n-2} = \psi(1, \mathscr{S}_{n-1,n-2}) \circ \bigcirc_{i=3}^{n-1} \psi(i, \mathscr{S}_{n-1,n-2})^{(i)} \circ \phi(\mathscr{S}_{n-1,n-3}) \circ \psi(2, \overline{\mathscr{S}_{n-1,n-2}}). \tag{d}$$

Lemma 9. *If $n \geqslant 4$ then*

(1) $f_{n,n-2} = (n, 2, 3, \ldots, n-2, 1, n-1)$,
(2) $\ell_{n,n-2} = (n-1, n, 3, 4, \ldots, n-2, 1, 2)$.

Proof.

(1) By relation (d), $\mathscr{S}_{4,2}$ is the list (4, 2, 1, 3), (4, 3, 2, 1), (4, 1, 3, 2), (2, 1, 4, 3), (1, 3, 4, 2), (3, 2, 4, 1), (3, 1, 2, 4), (2, 3, 1, 4), (2, 4, 3, 1), (1, 4, 2, 3), (3, 4, 1, 2) and more generally $f_{n,n-2} = \text{first}(\psi(1, \mathscr{S}_{n-1,n-2}))$. So, $f_{n,n-2} = \psi(1, f_{n-1,n-2})$ and the recurrence on n completes the proof.
(2) Similarly,

$$\begin{aligned}\ell_{n,n-2} &= \text{last}(\psi(2, \overline{\mathscr{S}_{n-1,n-2}})) \\ &= \psi(2, f_{n-1,n-2}) \\ &= (n-1, n, 3, \ldots, n-2, 1, 2). \quad \square\end{aligned}$$

Proposition 10. *The list $\mathscr{S}_{n,n-2}$ defined by* (d), $n \geqslant 4$, *is a Gray code.*

Proof. By Lemma 2, the transition between $\psi(i, \mathscr{S}_{n-1,n-2})$ and $\psi(j, \overline{\mathscr{S}_{n-1,n-2}})$, $1 \leqslant i, j \leqslant n-1$, (or respectively between $\psi(j, \overline{\mathscr{S}_{n-1,n-2}})$ and $\psi(i, \mathscr{S}_{n-1,n-2})$) is a product by a three-length cycle. It remains to examine the transitions between $\psi(n-1, \mathscr{S}_{n-1,n-2})^{(n-1)}$ and $\phi(\mathscr{S}_{n-1,n-3})$, and between $\phi(\mathscr{S}_{n-1,n-3})$ and $\psi(2, \overline{\mathscr{S}_{n-1,n-2}})$

$$\text{last}(\psi(n-1, \mathscr{S}_{n-1,n-2})^{(n-1)}) = \begin{cases} (n-1, 2, 3, \ldots, n-3, n-2, n, 1) & \text{if } n \text{ even}, \\ (n-2, 2, 3, \ldots, n-3, 1, n, n-1) & \text{if } n \text{ odd} \end{cases}$$

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

and

$$\mathrm{first}(\phi(\mathscr{S}_{n-1,n-3})) = (n-1, 2, 3, \ldots, n-3, 1, n-2, n) = \begin{cases} \mathrm{last}(\psi(n-1, \mathscr{S}_{n-1,n-2})^{n-1}) \cdot \langle n-2, n-1, n\rangle & \text{if } n \text{ even},\\ \mathrm{last}(\psi(n-1, \mathscr{S}_{n-1,n-2})^{n-1}) \cdot \langle 1, n-1, n\rangle & \text{if } n \text{ odd}. \end{cases}$$

Moreover,

$$\begin{aligned}\mathrm{last}(\phi(\mathscr{S}_{n-1,n-3})) &= (n-2, n-1, 3, 4, \ldots, n-3, 1, 2, n)\\ &= (n-2, n, 3, 4, \ldots, n-3, 1, n-1, 2) \cdot \langle 2, n-1, n\rangle\\ &= \psi(2, \mathrm{last}(\mathscr{S}_{n-1,n-2})) \cdot \langle 2, n-1, n\rangle\\ &= \mathrm{first}(\psi(2, \overline{\mathscr{S}_{n-1,n-2}})) \cdot \langle 2, n-1, n\rangle. \qquad \square\end{aligned}$$

3.5. The case $2 \leqslant k \leqslant n-3$

If $2 \leqslant k \leqslant n-3$, we define

$$\mathscr{S}_{n,k} = \psi(1, \mathscr{S}_{n-1,k}) \circ \bigcirc_{i=n-1}^{k+1} \psi(i, \mathscr{S}_{n-1,k})^{(i)} \circ \phi(\mathscr{S}_{n-1,k-1})^{(k)} \circ \bigcirc_{i=k}^{2} \psi(i, \mathscr{S}_{n-1,k})^{(i-1)}. \tag{e}$$

Lemma 11. *If $2 \leqslant k \leqslant n-3$ then*

(1) $f_{n,k} = (n, 2, 3, \ldots, k, 1, k+1, k+2, \ldots, n-1)$

(2) $\ell_{n,k} = \begin{cases} (n-1, n, 3, \ldots, k, 1, k+1, k+2, \ldots, n-2, 2) & \textit{if } k \neq 2,\\ (n-1, n, 1, 3, \ldots, n-2, 2) & \textit{otherwise}. \end{cases}$

Proof. By recurrence on n, we have $f_{n,k} = \mathrm{first}(\psi(1, \mathscr{S}_{n-1,k})) = \psi(1, f_{n-1,k})$ anchored by $f_{k+2,k} = (k+2, 2, 3, \ldots, k, 1, k+1)$. Similarly, $\ell_{n,k} = \mathrm{last}(\psi(2, \overline{\mathscr{S}_{n-1,k}})) = \psi(2, f_{n-1,k})$ also anchored by $f_{k+2,k} = (k+2, 2, 3, \ldots, k, 1, k+1)$. $\square$

Proposition 12. *The list $\mathscr{S}_{n,k}$ defined by* (e), $2 \leqslant k \leqslant n-3$ *is a Gray code.*

Proof. We use Lemma 2 for the transitions from $\psi(i, S_{n-1,k})$ to $\psi(j, \overline{S_{n-1,k}})$ and from $\psi(j, \overline{S_{n-1,k}})$ to $\psi(i, S_{n-1,k})$. It remains to examine the three transitions (i) $\psi(1, S_{n-1,k})$ and $\psi(n-1, S_{n-1,k})$; (ii) $\psi(k+1, \mathscr{S}_{n-1,k})^{(k+1)}$ and $\phi(\mathscr{S}_{n-1,k-1})^{(k)}$; (iii) $\phi(\mathscr{S}_{n-1,k-1})^{(k)}$ and $\psi(k, \mathscr{S}_{n-1,k})^{(k-1)}$.

Case (i): If $k \neq 2$,

$$\begin{aligned}\mathrm{last}(\psi(1, \mathscr{S}_{n-1,k})) &= \psi(1, \mathrm{last}(\mathscr{S}_{n-1,k}))\\ &= \psi(1, (n-2, n-1, 3, \ldots, k, 1, k+1, \ldots, n-3, 2))\\ &= \psi(1, (n-1, 2, 3, \ldots, k, 1, k+1, \ldots, n-3, n-2) \cdot \langle 1, n-1, 2\rangle).\end{aligned}$$

By Lemma 3,

$$\begin{aligned}\mathrm{last}(\psi(1, \mathscr{S}_{n-1,k})) &= \psi(n-1, (n-1, 2, 3, \ldots, k, 1, k+1, \ldots, n-3, n-2)) \cdot \langle 2, n-1, 1\rangle\\ &= (n-1, 2, 3, \ldots, k, 1, k+1, \ldots, n-3, n, n-2) \cdot \langle 2, n-1, 1\rangle\\ &= \psi(n-1, \mathrm{first}(\mathscr{S}_{n-1,k})) \cdot \langle 2, n-1, 1\rangle.\end{aligned}$$

The case $k=2$ is similar.

Case (ii): Let $\sigma = \mathrm{first}(\mathscr{S}_{n-1,k})$ (resp. $\mathrm{last}(\mathscr{S}_{n-1,k})$) and $\gamma = \mathrm{first}(\mathscr{S}_{n-1,k-1})$ (resp. $\mathrm{last}(\mathscr{S}_{n-1,k-1})$). We have $\sigma = \gamma \cdot \langle k, k+1\rangle$. By Lemma 4,

$$\begin{aligned}\psi(k+1, \sigma) &= \psi(k+1, \gamma \cdot \langle k, k+1\rangle)\\ &= \phi(\gamma) \cdot \langle k+1, n, k\rangle.\end{aligned}$$

Case (iii): The proof is similar as for the case (ii). $\square$

Table 1
The lists $\mathscr{S}_{3,k}$, $1 \leqslant k \leqslant 3$, and $\mathscr{S}_{4,k}$ for $1 \leqslant k \leqslant 4$

$\mathscr{S}_{3,1}$		$\mathscr{S}_{3,2}$		$\mathscr{S}_{3,3}$		$\mathscr{S}_{4,1}$		$\mathscr{S}_{4,2}$		$\mathscr{S}_{4,3}$		$\mathscr{S}_{4,4}$	
1	**312**	1	**321**	1	**123**	1	**4123**	1	**4213**	1	**4231**	1	**1234**
2	*231*	2	*132*			2	**4312**	2	**4321**	2	*1243*		
		3	**213**			3	*2413*	3	**4132**	3	**1432**		
						4	*3421*	4	*2143*	4	*2134*		
						5	**2341**	5	*1342*	5	*1324*		
						6	**3142**	6	*3241*	6	*3214*		
								7	**3124**				
								8	**2314**				
								9	*2431*				
								10	*1423*				
								11	*3412*				

For instance, in $\mathscr{S}_{4,2}$ the sublists of relation (d), $\psi(1, \mathscr{S}_{3,2})$, $\overline{\psi(3, \mathscr{S}_{3,2})}$, $\phi(\mathscr{S}_{3,1})$ and $\overline{\psi(2, \mathscr{S}_{3,2})}$, are alternatively in bold-face and italic.

Note that our Gray code $\mathscr{S}_{n,k}$ is cyclic and optimal for all n, k. See Table 1 for some examples.

Definition 13. For a list of permutations $\mathscr{L}$ let denote $\mathscr{L}^{-1}$ the list obtained from $\mathscr{L}$ by replacing each permutation in $\mathscr{L}$ by its group theoretical inverse.

By a simple calculation we have:

Theorem 14. *The list $\mathscr{S}_{n,k}^{-1}$ is also an optimal cyclic Gray code for n-length permutations with k cycles; i.e. two successive permutations differ by a product with a three-cycle.*

4. Transposition array representation

Here we show that replacing each permutation in the list $\mathscr{S}_{n,k}$ by its transposition array (defined below) the obtained list still remains an optimal Gray code. Its interest is twofold. First, the recursive definition of $\mathscr{S}_{n,k}$ and the resulting generating algorithm in the next section express subsequently permutations in transposition array representation. Secondly, this shows how a map, which generally does not preserve closeness, transforms a Gray code into another one. See for instance [3] for more about closeness preserving bijections.

Any permutation $\pi \in S_n$ has a unique decomposition as a product of transpositions

$$\pi = \langle p_1, 1\rangle \cdot \langle p_2, 2\rangle \cdot \langle p_3, 3\rangle \cdot \ldots \cdot \langle p_n, n\rangle = \prod_{i=1}^{n} \langle p_i, i\rangle \tag{4}$$

with $1 \leqslant p_i \leqslant i$ and obviously any such decomposition provides a permutation. So (4) yields a bijection $S_n \longrightarrow T_n$ with T_n the product set $[1] \times [2] \times \cdots \times [n]$, and a string $p_1 p_2 p_3 \ldots p_n \in T_n$ is yet another way to represent a permutation called *transposition array*. Alternatively, a string $p_1 p_2 p_3 \ldots p_n \in T_n$ can be viewed as the *inversion table* of a permutation $\sigma \in S_n$: p_i is the number of entries $\sigma_j > \sigma_i$, $j < i$, plus 1.

The relation between the two permutations, one obtained by regarding a given string in T_n as its transposition array and the other as its inversion table, has never been studied and this might be an interesting direction of research.

Let now $T_{n,k}$ be the set of strings in T_n with exactly k fixed points, that is, k entries p_i with $p_i = i$. Since the number of 'identity transpositions' $\langle i, i\rangle$ in (4) equals the number of cycles in π, $T_{n,k}$ is the set of transposition arrays of permutations in $S_{n,k}$ and relation (4) induces a bijection $S_{n,k} \longrightarrow T_{n,k}$.

On the other hand, $T_{n,k}$ is the set of inversion tables of permutations in S_n having k *left-to-right minima*; a left-to-right minimum in a permutation is an entry less than all the entries to its left. As a consequence, we obtain the following enumerative result: the number of permutations in S_n with k left-to-right minima is the signless Stirling number $s(n, k)$. See also [13].

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

Table 2
The lists $\mathscr{S}_{4,2}$ and $\mathscr{T}_{4,2}$

$\mathscr{S}_{4,2}$	$\mathscr{T}_{4,2}$
4213	1211
4321	1221
4132	1131
2143	1133
1342	1223
3241	1213
3124	1114
2314	1124
2431	1132
1423	1222
3412	1212

Each string in $\mathscr{T}_{4,2}$ is the transposition array of the corresponding permutation in $\mathscr{S}_{4,2}$.

Generally, a bijection between two sets can magnify small changes between objects and this is the case with the bijection $S_n \longrightarrow T_n$ defined in (4). For example, if π and π' are two permutations such that $\pi' = \pi \cdot \langle a, b, c\rangle$ then the decomposition of π' differ from π by possibly many transpositions: take $\pi = (7, 1, 2, 3, 4, 5, 6) = \langle 1, 1\rangle \cdot \langle 1, 2\rangle \cdot \langle 1, 3\rangle \cdot \langle 1, 4\rangle \cdot \langle 1, 5\rangle \cdot \langle 1, 6\rangle \cdot \langle 1, 7\rangle$ and $\pi' = (2, 1, 4, 3, 7, 5, 6) = \pi \cdot \langle 1, 3, 5\rangle = \langle 1, 1\rangle \cdot \langle 1, 2\rangle \cdot \langle 3, 3\rangle \cdot \langle 3, 4\rangle \cdot \langle 5, 5\rangle \cdot \langle 5, 6\rangle \cdot \langle 5, 7\rangle$. Conversely, if two decompositions differ by at most two transpositions, then the corresponding one-line permutations can possibly differ by many entries: consider $\pi = (2, 3, 4, 5, 6, 1) = \langle 1, 1\rangle \cdot \langle 1, 2\rangle \cdot \langle 2, 3\rangle \cdot \langle 3, 4\rangle \cdot \langle 4, 5\rangle \cdot \langle 5, 6\rangle$ and $\pi' = (2, 4, 1, 6, 3, 5) = \langle 1, 1\rangle \cdot \langle 1, 2\rangle \cdot \langle 2, 3\rangle \cdot \langle 2, 4\rangle \cdot \langle 4, 5\rangle \cdot \langle 4, 6\rangle$.

Let now $\mathscr{T}_{n,k}$ be the list for the set $T_{n,k}$ obtained by replacing each permutation in the list $\mathscr{S}_{n,k}$ by its transposition array. Surprisingly, in $\mathscr{T}_{n,k}$ two consecutive sequences differ in at most two positions and so it is a Gray code (see Table 2). This result is shown in the next theorem and despite its very similarity with Theorem 14 they are not a simple consequence each other.

Definition 15. With the notations above, the permutation

$$\tau = \prod_{i=n-j+1}^{n} \langle p_i, i\rangle$$

in S_n is called the *jth* characteristic of π. The nth characteristic of an n-length permutation is the permutation itself, and it is convenient to consider that the 0th characteristic of any permutation is the identity.

Remark 16. Let τ be the *jth* characteristic of π as above.

(1) The number of cycles in τ is $(n - \ell)$, where ℓ is the number of p_i, $n - j + 1 \leqslant i \leqslant n$, with $p_i < i$,
(2) If τ and π have the same number of cycles then $\tau = \pi$.

Remark 17. For π and σ, the following are equivalent:

(1) π and σ have the same *jth* characteristic,
(2) π and σ have the same *ith* characteristic for i from 0 to j,
(3) π and σ have the entries from $n - j + 1$ to n in the same respective positions,
(4) π^{-1} and σ^{-1} have the same entries for each position from $n - j + 1$ to n.

Now let consider the transformations ψ and ϕ in Definition 1. If γ is a permutation in S_{n-1} and $\pi = \psi(i, \gamma) \in S_n$ then the first characteristic of π is $\langle i, n\rangle$; and if $\pi = \phi(\gamma) \in S_n$ then the first characteristic of π is the identity (actually, the 'transposition' $\langle n, n\rangle$). Similar results hold if we replace γ and π by sets (or lists) of permutations. Generally, if $\mathscr{A}$ is a list of permutations in S_{n-j} and

$$\mathscr{B} = \alpha_1(\alpha_2(\ldots \alpha_j(\mathscr{A}) \ldots)) \in S_n$$

with $\alpha_i(\cdot)$ of the form $\phi(\cdot)$ or $\psi(x_i, \cdot)$ then each permutation in $\mathscr{B}$ has the jth characteristic

$$\prod_{i=n-j+1}^{n} \langle p_i, i\rangle,$$

where $p_i = x_{n-i+1}$ if $\alpha_i(\cdot) = \psi(p_i, \cdot)$, and $x_i = i$ if $\alpha_i(\cdot) = \phi(\cdot)$.

By recursivity of relations (a)–(e) we have:

Remark 18. For any $\tau \in S_{n,k}$, all permutations with the same *jth* characteristic as τ, $0 \leqslant j \leqslant n$, form a contiguous sublist of the list $\mathscr{S}_{n,k}$ defined by the appropriate relation (a)–(e). Actually, j is the depth of recursivity involved to reach this sublist.

Theorem 19. *The list $\mathscr{T}_{n,k}$ is a cyclic Gray code for n-length permutations with k cycles*; *i.e. two successive permutations differ by one or two transpositions in their decompositions*.

Proof. Let π and σ be two successive permutations in $\mathscr{S}_{n,k}$. We will show that their transposition arrays differ in at most two positions. π and σ belong to the same sublist of $\mathscr{S}_{n,k}$ as in relation (3) iff they have the same first characteristic. By induction, it is enough to show that for two successive sublists $\alpha(\mathscr{A})$ and $\beta(\mathscr{B})$ in the definition of $\mathscr{S}_{n,k}$ (with α, β as $\phi(\cdot)$ or $\psi(i, \cdot)$) the transposition arrays of last$(\mathscr{A})$ and first$(\mathscr{B})$ differ in at most one position. Indeed, we give below the transposition array representations of the first element $f_{n,k}$ and the last element $\ell_{n,k}$ of the list $\mathscr{S}_{n,k}$ and with relations (a)–(e) we can verify that last$(\mathscr{A})$ and first$(\mathscr{B})$ differ in at most one position

$$f_{n,k} = \begin{cases} 1\ 1 \ldots 1 & \text{if } k = 1,\\ 1\ 2 \ldots (n-1)n & \text{if } k = n,\\ 1\ 2 \ldots (n-1)1 & \text{if } k = n-1,\\ 1\ 2 \ldots (n-2)11 & \text{if } k = n-2,\\ 1\ 2 \ldots k11 \ldots 1 & \text{otherwise,} \end{cases}$$

$$\ell_{n,k} = \begin{cases} 11 \ldots 13 & \text{if } k = 1,\\ 12 \ldots (n-1)n & \text{if } k = n,\\ 12 \ldots (n-2)1n & \text{if } k = n-1,\\ 12 \ldots (n-2)12 & \text{if } k = n-2,\\ 12 \ldots k11 \ldots 12 & \text{otherwise.} \end{cases} \qquad \square$$

Remark 20.

- $\mathscr{T}_{n,k}$ is at the same time a Gray code for the set of n-length permutations with k cycles in transposition array representation, and for the set of n-length permutations with k left-to-right minima in inversion table representation.
- $\mathscr{T}_{n,k}$ is suffix partitioned and satisfies Walsh's desiderata, so looplessly implementable [17] .

5. Algorithmic considerations

In this part, we explain how the recursive definitions (a)–(e) can be implemented into an efficient algorithm, i.e. in a CAT algorithm. Such algorithms already exist for derangements or involutions [1,9,10], so we will just give here the main difficulties to implement our one.

Before the main call of procedure *gen_up(n,k)* given in Appendix, τ is initialized by the identity, so it has n cycles and is the 0th characteristic of the permutations in $\mathscr{S}_{n,k}$. Before each recursive call of $gen\_up(n, k)$ or $gen\_down(n, k)$, τ is, for some j, the jth characteristic of a sublist of permutations in $\mathscr{S}_{n,k}$.

(1) If τ has k cycles, then τ is printed, and no recursive call is produced. This corresponds to point 2 of Remark 16 and to the recursive definition (b).

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

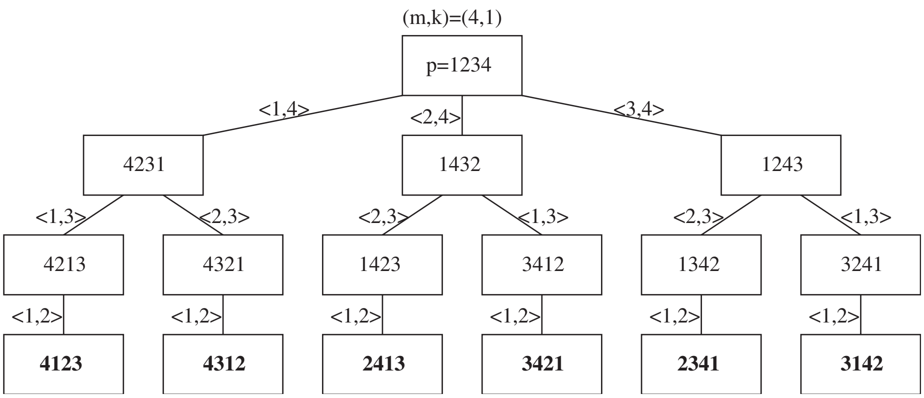

Fig. 2. The generating tree for the list $\mathscr{S}_{4,1}$ with the decomposition in a product of transpositions of each element. The list $\mathscr{S}_{4,1}$ appears in bold.

(2) If τ has $(k + (n - j))$ cycles then $(n - j - 1)$ calls are produced and before each of them τ is updated as $\tau := \langle i, n - j\rangle \cdot \tau$, with $i = 1, 2, \ldots, n - j - 1$ (not necessarily in this order). Each τ updated like that is a $(j + 1)$th characteristic obtained by a premultiplication by a (not dummy) transposition of the previous jth characteristic. The number of cycles in τ decreases by one. This corresponds to the recursive definition (a) which gives the order of calls.

(3) Similarly, if the number of cycles in τ is more than k but less than $(k + (n - j))$, then $(n - j)$ calls are produced, $(n - j - 1)$ of them are those of the point above. Before the additional call, τ, the new $(j + 1)$th characteristic is unchanged (τ is 'updated' as $\tau := \langle n - j, n - j\rangle \cdot \tau$). This corresponds to the recursive definitions (c)–(e), and again they give the order of calls.

In our algorithm j is the depth of the recursive call and the order of successive calls directly produced by a given call is determined by the appropriate definition (a), (c) and (d). See Fig. 2 for the generating tree of $\mathscr{S}_{4,1}$.

This algorithm enable us to ensure that this transforms an object into its successor in CAT. Indeed, excepted for the calls where $n = 2$, $k = 1$, all calls have a degree 0 or at least 2 (see Fig. 2 for an example). Between two recursive calls, we execute two transpositions, and moreover at least one permutation is generated in each recursive call. This means that the total amount of computation divided by the number of objects is bounded by a constant (see [14]). Thus the complexity of this algorithm is $\mathcal{O}(s_{n,k})$. A Java implementation of our algorithm is available from the author on request.

6. Uncited reference

[8].

Acknowledgments

I thank V. Vajnovszki and the anonymous referees for their constructive remarks which have greatly improved this paper.

Appendix A.

The call of *gen_up(n,k)* generates the list $\mathscr{S}_{n,k}$. In order to produce $\overline{\mathscr{S}_{n,k}}$ we consider also the procedure *gen_down(n,k)* which has the same instructions of *gen_up(n,k)* in the inverse order. The notation $gen\_up/down(n - 1, k)$ means that we use *gen_up(n,k)* or *gen_down(n,k)* according to the sense of each sublist in relations (a)–(e). To multiplicate (on the left) a jth characteristic σ by a transposition $\langle x, n\rangle$, we conserve the inverse σ^{-1} at each level of the recursivity, and we calculate the product $\sigma \cdot \langle \sigma^{-1}(x), \sigma^{-1}(n)\rangle$, i.e we just need to exchange the positions $\sigma^{-1}(x)$ and $\sigma^{-1}(n) = n$ in σ.

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

```
procedure gen_up(n, k)
var i,j;
begin
if n = k
then print(σ);
else if n = 4 and k = 1
    then σ := ⟨1, n⟩ · σ; gen_up(n − 1, k); σ := ⟨1, n⟩ · σ;
         σ := ⟨2, n⟩ · σ; gen_down(n − 1, k); σ := ⟨2, n⟩ · σ;
         σ := ⟨3, n⟩ · σ; gen_down(n − 1, k); σ := ⟨3, n⟩ · σ;
    else
        if k = n − 1
        then σ := ⟨1, n⟩ · σ; gen_up(n − 1, k); σ := ⟨1, n⟩ · σ;
             if n − 1 > 1 then
               σ := ⟨n − 1, n⟩ · σ; gen_up(n − 1, k); σ := ⟨n − 1, n⟩ · σ;
             endif
             for j = n − 3 downto 2 do
                σ := ⟨j, n⟩ · σ; gen_up/down(n − 1, k); σ := ⟨j, n⟩ · σ;
             enddo
             if n − 2 > 1 then
               σ := ⟨n − 2, n⟩ · σ; gen_up(n − 1, k); σ := ⟨n − 2, n⟩ · σ;
             endif
             if k ⩾ 2 then
               σ := ⟨n, n⟩ · σ; gen_down(n − 1, k − 1);
             endif
        else
            if k = n − 2
            then σ := ⟨1, n⟩ · σ; gen_up(n − 1, k); σ := ⟨1, n⟩ · σ;
                 for j = 3 to n − 1 do
                    σ := ⟨j, n⟩ · σ; gen_up/down(n − 1, k); σ := ⟨j, n⟩ · σ;
                 endo
                 if k ⩾ 2 then
                   σ := ⟨n, n⟩ · σ; gen_up(n − 1, k − 1);
                 σ := ⟨2, n⟩ · σ; gen_down(n − 1, k); σ := ⟨2, n⟩ · σ;
                 endif
            else
                if k = 1
                then σ := ⟨1, n⟩ · σ; gen_up(n − 1, k); σ := ⟨1, n⟩ · σ;
                     for j = n − 1 downto 4 do
                        σ := ⟨j, n⟩ · σ;gen_up/down(n − 1, k); σ := ⟨j, n⟩ · σ;
                     endo
                     σ := ⟨2, n⟩ · σ; gen_up(n − 1, k); σ := ⟨2, n⟩ · σ;
                     σ := ⟨3, n⟩ · σ; gen_down(n − 1, k); σ := ⟨3, n⟩ · σ;
                else
                     σ := ⟨1, n⟩ · σ; gen_up(n − 1, k); σ := ⟨1, n⟩ · σ;
                     for j = n − 1 downto k + 1 do
                        σ := ⟨j, n⟩ · σ; gen_up/down(n − 1, k); σ := ⟨j, n⟩ · σ;
                     endo
                     if k ⩾ 2 then σ := ⟨n, n⟩ · σ; gen_up/down(n − 1, k − 1); endif
                     for j = k downto 2 do
                        σ := ⟨j, n⟩ · σ;  gen_up/down(n − 1, k); σ := ⟨j, n⟩ · σ;
                     endo
                endif
            endif
        endif
    endif
end
```

References

[1] J.L. Baril, V. Vajnovszki, Gray code for derangements, Discrete Appl. Math. 140 (1–3) (2004) 207–221.

[2] W. Chu, Ch. Colbourn, P. Dukes, Constructions for permutation codes in powerline communications, Designs, Codes and Cryptography 32 (2004) 51–64.

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007

[3] A. Juarna, V. Vajnovszki, Isomorphism between classes counted by Fibonacci numbers, Words 2005, September 2005, pp. 51–62.
[4] D.E. Knuth, The Art of Computer Programming, Combinatorial Algorithms, vol. 4, Pre-fascicle 2b, Generating all permutations, 2002.
[5] J.F. Korsh, Loopless generation of up–down permutations, Discrete Math. 240 (1–3) (2001) 97–122.
[6] D. Roelants van Baronaigien, Constant time generation of involutions, Congr. Numer. 90 (1992) 87–96.
[7] F. Ruskey, ⟨http://www.theory.cs.uvic.ca/∼cos/gen/perm.html⟩.
[8] F. Ruskey, Simple combinatorial Gray codes constructed by reversing sublists in ISAAC conference, Lecture Notes in Computer Science, vol. 762, Springer, Berlin, 1993, pp. 201–208.
[9] F. Ruskey, S. Effler, A CAT algorithm for generating permutations with a fixed number of inversions, Inform. Process. Lett. 86 (2) (2003) 107–112.
[10] F. Ruskey, D. Roelants van Baronaigien, Generating permutations with given ups and downs, Discrete Appl. Math. 36 (1) (1992) 57–67.
[11] F. Ruskey, U. Taylor, Fast generation of restricted classes of permutations, Manuscript, 1995.
[12] R. Sedgewick, Permutation generation methods, Comput. Surveys 9 (2) (1977) 137–164.
[13] R. Stanley, Enumerative Combinatorics, vol. 1, Cambridge University Press, Cambridge, England, 1997.
[14] V. Vajnovszki, Gray visiting Motzkins, Acta Inform. 38 (2002) 793–811.
[15] V. Vajnovszki, A loopless algorithm for generating the permutations of a multiset, Theoret. Comput. Sci. 307 (2003) 415–431.
[16] T. Walsh, Gray code for involutions, J. Combin. Math. Combin. Comput. 36 (2001) 95–118.
[17] T. Walsh, Generating Gray codes in O(1) worst-case time per word, Lecture Notes in Computer Science, vol. 2731, 2003, pp. 73–88.
[18] R.M. Wilson, J.H. van Lint, A Course in Combinatorics, Cambridge University Press, Cambridge, 2002.

Please cite this article as: Jean-Luc Baril, Gray code for permutations with a fixed number of cycles, Discrete Mathematics (2006), doi: 10.1016/j.disc.2006.09.007