Let us describe a permutation group code, based on sorting using coset leaders, but employing longer cycles rather than just transpositions. The code consists of permutations \( \langle x_0, \ldots, x_{n-1} \rangle \) of \( x_0 = \langle 0, 1, \ldots, n-1 \rangle \).

We encode by writing a permutation in a canonical form. For \( 2 \leq k \leq n \), let \( \sigma_k \) be the permutation that cycles the first \( k \) entries of \( x \) to the right, \( \sigma_k(x) = \langle x_{k-1}, x_0, x_1, \ldots, x_{k-2}, x_k, \ldots, x_{n-1} \rangle \).

It is easy to see how to write an arbitrary permutation \( x = \pi x_0 \) as \( \pi = \sigma_{m_2} \cdots \sigma_{m_n} \) with each \( 0 \leq m_j < j \). Recursively use the \( \sigma_j \)'s to cycle the \( j \)-th entry of \( x \) into position.

As messages, we take integer sequences of the form \( m_2 \ldots m_n \) with each \( 0 \leq m_j < j \). This message is encoded as \( \pi x_0 \) with \( \pi = \sigma_{m_2} \cdots \sigma_{m_n} \). This can be quickly done.

Now we turn to decoding. Suppose that we receive the permutation \( x \). Set \( z_2 = x \). Inductively, for \( j < n \), assume that we have \( m_2 \ldots m_{j-1} \) such that \( z_j = \sigma_{j-1}^{-m_{j-1}} \cdots \sigma_2^{-m_2} x \) has its first \( j \) entries \( t_0 \ldots t_{j-1} \) in the correct cyclic order. Compute the differences \( d_0 = t_j - t_0, \ldots, d_{j-1} = t_j - t_{j-1} \).

Note that these are also in cyclic order. If any \( d_i \) is negative, choose \( m_j \) such that \( d_{m_j} \) is the least negative; otherwise choose \( m_j \) such that \( d_{m_j} \) is the most positive. Then set \( z_{j+1} = \sigma_{j-1}^{-m_j} z_j \).

Finally, given \( z_n = \langle t_0 \ldots t_{n-1} \rangle \), which will have its entries in the correct cyclic order, choose \( m_n \) such that \( t_{m_n} = 0 \). It follows that \( x_0 = \sigma_n^{-m_n} z_n \), as desired.

An example is illuminating. To send the message 1032 with \( n = 5 \), we encode thusly:

\[
\begin{align*}
x_0 &= \langle 0 \ 1 \ 2 \ 3 \ 4 \rangle \\
\sigma_2^2 x_0 &= \langle 3 \ 4 \ 0 \ 1 \ 2 \rangle \\
\sigma_3^3 \sigma_2^2 x_0 &= \langle 4 \ 0 \ 1 \ 3 \ 2 \rangle \\
\sigma_3^0 \sigma_3^3 \sigma_2^2 x_0 &= \langle 4 \ 0 \ 1 \ 3 \ 2 \rangle \\
\sigma_3^1 \sigma_3^0 \sigma_3^3 \sigma_2^2 x_0 &= \langle 0 \ 4 \ 1 \ 3 \ 2 \rangle
\end{align*}
\]

and send the last vector. Decoding takes us backward through the same sequence.

If \( n \) is large, then decoding can be sped up using the fact that initial segments are always cyclically ordered.

\date{February 16, 2009.}