Faster Generation of Shorthand Universal Cycles for Permutations

Alexander Holroyd[1] Frank Ruskey[2] Aaron Williams[2]

[1]Microsoft Research
Redmond, Washington, USA.
[2]Department of Computer Science
University of Victoria, CANADA.

Talk at COCOON Conference,
Nha Trang, Vietnam, July 20, 2010

The underlying problem

- An example with $n = 3$: Consider the circular string

$$321312$$

- Its length 2 substrings are

$$32, 21, 13, 31, 12, 23.$$

  These are exactly the 2-permutations of a 3-set.
- Question: Given n and k, can you construct a circular string of length $(n)_k := n(n-1)\cdots(n-k+1)$ such that every k-permutation of $[n] := \{1, 2, \ldots, n\}$ occurs (uniquely) as a substring?
- Yes, existence shown by Brad Jackson: "*Universal cycles of k-subsets and k-permutations*", Discrete Mathematics, 149 (1996) 123–129, for all $k < n$.
- No such string for permutations ($k = n$) if $n \geq 3$.
- In this talk, we are only concerned with the case $k = n - 1$.

Knuth's challenge for the $k = n - 1$ case

- The problem for $k = n - 1$ is discussed by D.E. Knuth, *The Art of Computer Programming, Volume 4, Generating All Tuples and Permutations*, Fascicle 2, in Exercise 112 of Section 7.2.1.2. On page 121 we find the following quote:

  > *"At least one of these cycles must almost surely be easy to describe and to compute, as we did for de Bruijn cycles in Section 7.2.1.1. But no simple construction has yet been found."*

- This challenge was answered in Ruskey and Williams, *An explicit universal cycle for the $(n-1)$-permutations of an n-set*, ACM Transactions on Algorithms, June 2010.
- We present in this talk two additional constructions, both of which are better in some respect.

As a result of that TALG paper:

Dear Frank,

I finally have gotten Section 7.1.4 to the point where I could take a small breath and look at the mail that has come in since last summer about the other fascicles and prefascicles.

One of the most exciting things, of course, was to learn about your nice explicit universal cycles of permutations. In the next printing of Volume 4 Fascicle 2 I shall replace exercise 7.2.1.2–112 by two exercises, 112 and 113; 112 asks for (and gives hints towards) your explicit construction, while 113 is the former 112.

These updates will be posted in the TAOCP errata listing all4f2.ps, later this week. I also stuck in a very brief mention of the multiset case, although you have apparently not yet written that paper.

Beautiful: stringology is really coming of age!

...

Thanks again for keeping me informed.

Best regards, Don

Obviously, $(n)_k = n \cdot (n-1) \cdots 3 \cdot 2 = n!$

- By adding the missing numbers, a $(n-1)$-permutation of $[n]$ becomes a permutation of $[n]$.
- $$\begin{array}{ccc} 32 & \to & 321 \\ 21 & \to & 213 \\ 13 & \to & 132 \\ 31 & \to & 312 \\ 12 & \to & 123 \\ 23 & \to & 231 \end{array}$$
- A Ucycle := a universal cycle for the $(n-1)$-permutations of $[n]$.
- $\underline{3}71526\,4$ $\nearrow$ $715263\underline{4}$ rotate first $n-1$ symbols left; $\searrow$ $715264\underline{3}$ rotate all symbols left

Obviously, $(n)_k = n \cdot (n-1) \cdots 3 \cdot 2 = n!$

- By adding the missing numbers, a $(n-1)$-permutation of $[n]$ becomes a permutation of $[n]$.
- $$\begin{array}{lcl} 32 & \rightarrow & 321 \\ 21 & \rightarrow & 213 \\ 13 & \rightarrow & 132 \\ 31 & \rightarrow & 312 \\ 12 & \rightarrow & 123 \\ 23 & \rightarrow & 231 \end{array}$$
- A Ucycle := a universal cycle for the $(n-1)$-permutations of $[n]$.
- In a Ucycle for the 6-permutations of [7]:

$\underline{3}71526\,4$ $\nearrow$ $715263\underline{4}$ rotate first $n-1$ symbols left.

$\searrow$ $715264\underline{3}$ rotate all symbols left.

Obviously, $(n)_k = n \cdot (n-1) \cdots 3 \cdot 2 = n!$

- By adding the missing numbers, a $(n-1)$-permutation of $[n]$ becomes a permutation of $[n]$.
- $$\begin{array}{ccc} 32 & \to & 321 \\ 21 & \to & 213 \\ 13 & \to & 132 \\ 31 & \to & 312 \\ 12 & \to & 123 \\ 23 & \to & 231 \end{array}$$
- A Ucycle := a universal cycle for the $(n-1)$-permutations of $[n]$.
- In a Ucycle for the 6-permutations of [7]:

$$\begin{array}{lcll} & \nearrow & 71526\underline{3}4 & \text{rotate first } n-1 \text{ symbols left.} \\ \underline{3}715264 & & & \\ & \searrow & 715264\underline{3} & \text{rotate all symbols left.} \end{array}$$

Obviously, $(n)_k = n \cdot (n-1) \cdots 3 \cdot 2 = n!$

- By adding the missing numbers, a $(n-1)$-permutation of $[n]$ becomes a permutation of $[n]$.
- $$\begin{array}{ccc} 32 & \to & 321 \\ 21 & \to & 213 \\ 13 & \to & 132 \\ 31 & \to & 312 \\ 12 & \to & 123 \\ 23 & \to & 231 \end{array}$$
- A Ucycle := a universal cycle for the $(n-1)$-permutations of $[n]$.
- As permutations:

$$\underline{3}71526\mathbf{4} \quad \begin{array}{ll} \nearrow & 71526\underline{3}\mathbf{4} \quad \text{rotate first } n-1 \text{ symbols left.} \\ \searrow & 715264\underline{\mathbf{3}} \quad \text{rotate all symbols left.} \end{array}$$

Obviously, $(n)_k = n \cdot (n-1) \cdots 3 \cdot 2 = n!$

- By adding the missing numbers, a $(n-1)$-permutation of $[n]$ becomes a permutation of $[n]$.
- $$\begin{array}{ccc} 32 & \to & 321 \\ 21 & \to & 213 \\ 13 & \to & 132 \\ 31 & \to & 312 \\ 12 & \to & 123 \\ 23 & \to & 231 \end{array}$$
- A Ucycle := a universal cycle for the $(n-1)$-permutations of $[n]$.
- As permutations:

$$\underline{3}71526\textcolor{green}{4} \quad \begin{array}{lll} \nearrow & 71526\underline{3}\textcolor{green}{4} & \text{rotate first } n-1 \text{ symbols left.} \\ \searrow & 715264\underline{\textcolor{green}{3}} & \text{rotate all symbols left.} \end{array}$$

The Cayley Graph Connection

- $$3715264 \quad \begin{array}{ll} \nearrow & 7152634 \quad \sigma_6 = \sigma_{n-1} \\ \searrow & 7152643 \quad \sigma_7 = \sigma_n \end{array}$$
- Define the directed Cayley graph

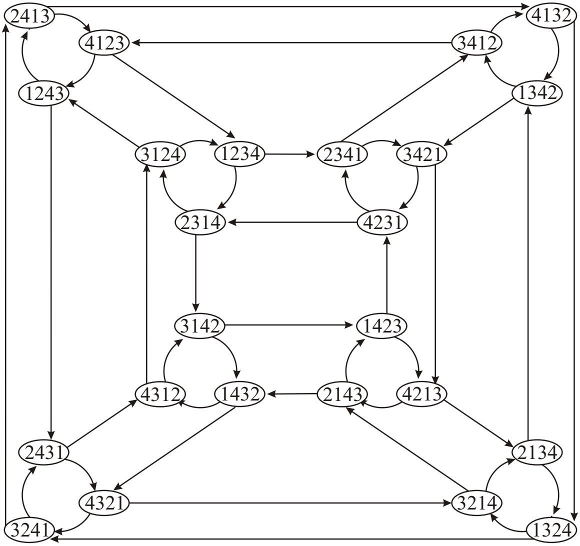
  $$\Xi_n := \overrightarrow{\mathrm{Cay}}(\{\sigma_n, \sigma_{n-1}\}; \mathbb{S}_n)$$

  In this graph the vertices are the permutations, $\mathbb{S}_n$, and edges are of the form $\pi \longrightarrow \sigma_j(\pi)$ for $j \in \{n-1, n\}$.
- Observe that Ξ_n is a 2-in 2-out digraph.
- The problem of finding a Hamilton cycle in Ξ_n is equivalent to finding a Ucycle of $(n-1)$-permutations of an n-set.
- (Which is equivalent to finding an Eulerian cycle in $J_{k,n}$, the graph that Jackson used in his existence proof.)

Objective: maximize σ_n edges

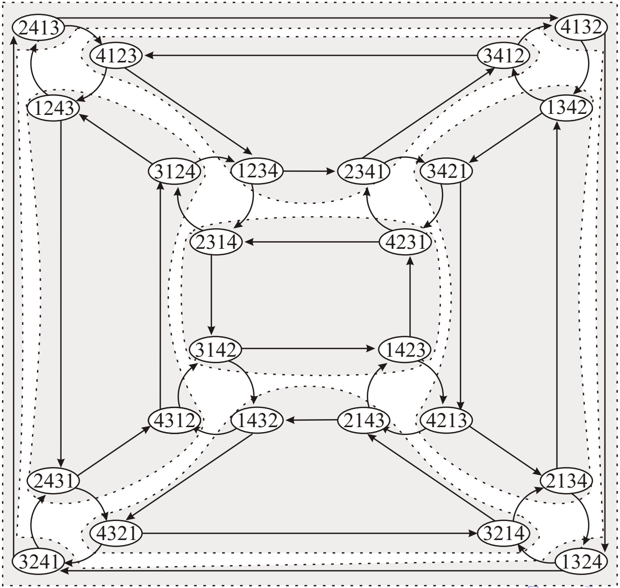
Consider the Cayley graph Ξ_4:

Straight edges are σ_4, circular arc edges are σ_3.



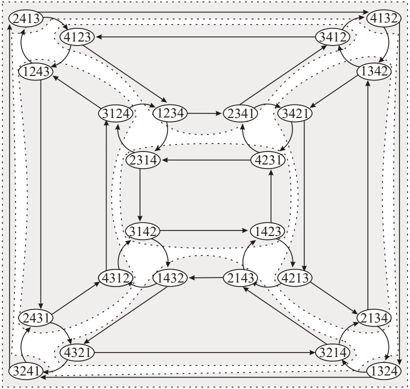
Objective: maximize σ_n edges

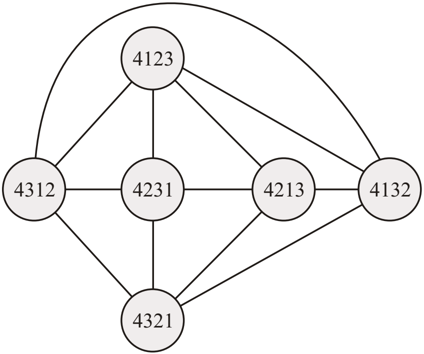
The cosets (equivalence classes) induced by $\sigma_n = \sigma_4$ are shaded.



Objective: maximize σ_n edges

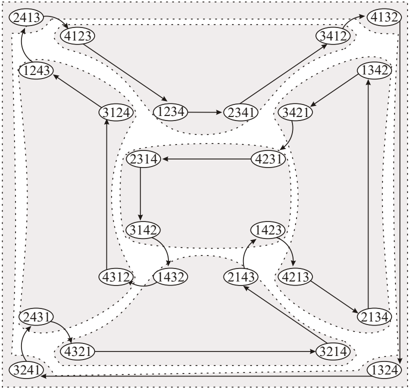
- Note that $\sigma_n\sigma_{n-1}^{-} = (n\ n{-}1)$ and so $(\sigma_n\sigma_{n-1}^{-})(\sigma_n\sigma_{n-1}^{-}) = id$.
- Thus the Cayley coset graph (shrink the cosets to vertices) is *undirected*, call it $\mathcal{C}_n$. (The graph $\mathcal{C}_4$ shown below.)
- Each vertex of $\mathcal{C}_n$ is labeled by the unique permutation in the coset that starts with n.

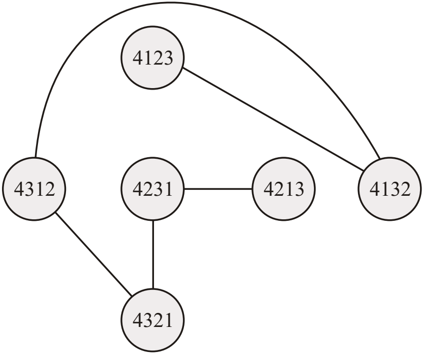




Spanning trees and Hamilton cycles

There is a one-to-one correspondence between spanning trees of $\mathcal{C}_n$ and Hamilton cycles in Ξ_n that minimize the number of σ_{n-1}s used.





Recall our objective...

- Using the spanning tree idea will minimize the number of σ_{n-1}s and therefore maximize the number of σ_ns.
- The number of nodes in $\mathcal{C}_n$ is $n!/n = (n-1)!$, so there are $(n-1)! - 1$ edges in the spanning tree.
- Each spanning tree edge gets used in both directions in Ξ_n and it follows that the number of σ_ns used is

$$n! - 2(n-1)! + 2.$$

- The proportion of σ_ns is about $(n-2)/n$; so asymptotically all of them are σ_ns.

What are natural spanning trees of $\mathcal{C}_n$?

Two natural *parent rules*:

- For both rules $n(n-1)\cdots 21$ is the root.
- For both rules the parent is obtained by swapping two adjacent elements (and so uses edges of $\mathcal{C}_n$).
- **Decrementing rule:**

$$\begin{aligned}\text{parent}(\ n(n-1)\cdots(n-s+1)x_1x_2\cdots x_{t-1}\ x_t(n-s)\ \gamma\ ) \\ =\quad n(n-1)\cdots(n-s+1)x_1x_2\cdots x_{t-1}\ (n-s)x_t\ \gamma\end{aligned}$$

Example: $987\underline{24}613 \rightarrow 9872\underline{64}13$

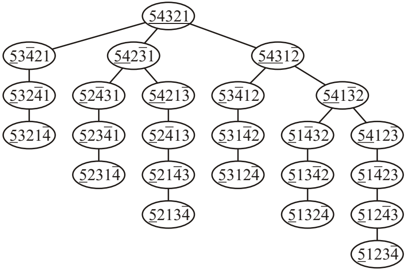
Example: 98724613 → 98726413

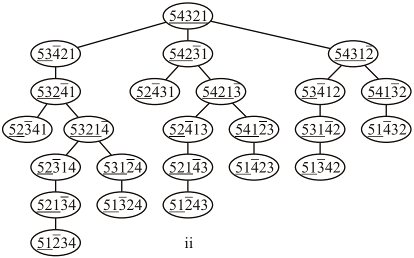
- **Decreasing rule:**

$$\begin{aligned}\text{parent}(\ n > \pi_2 > \cdots > \pi_{s-1} > \ \pi_s < x\ \gamma\ ) \\ =\quad n > \pi_2 > \cdots > \pi_{s-1} \lessgtr\ x > \pi_s\ \gamma\end{aligned}$$

Example: 98642713 → 98647213

The decrementing and decreasing trees for $n = 5$





An unexpected property

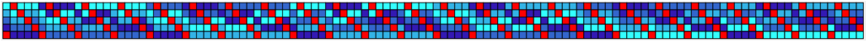
- Universal cycle from the decrementing rule:
  4 321 4 213 4 231 4 312 4 123 4 132
- Universal cycle from the decreasing rule:
  4 321 4 213 4 123 4 231 4 312 4 132
- Observe that in both: $n = 4$ occurs in every n-th position. This is true in for both rules for any n (but not for every spanning tree).
- Natural question(s): Is there an elegant way to describe the order of the permutations that are between the ns? We call those permutations of $[n-1]$, *sub-permutations*.
- YES!

Sub-permutations from the decrementing rule

- We call this order of permutations 7_n-order.
- Recursive rule: Each permutation $\pi = \pi_1\pi_2\cdots\pi_{n-1}$ in 7_{n-1}-order is expanded into the following list of n permutations.

$$
\begin{array}{c}
n\pi_1\pi_2\cdots\pi_{n-1} \\
\pi_1\pi_2\cdots\pi_{n-1}n \\
\pi_1\pi_2\cdots n\pi_{n-1} \\
\vdots \quad \cdots \quad \vdots \\
\pi_1 n\pi_2\cdots\pi_{n-1}
\end{array}
$$

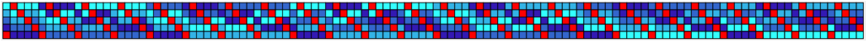
- See how the path of the n is like a 7?
- The graphic below: 7-order for $n = 5$.



Examples of 7-order

21	321	4321 4132 4123
12	213	3214 1324 1234
	231	3241 1342 1243
	312	3421 1432 1423
	123	4213 4312 4132
	132	2134 3124 1324
		2143 3142 1342
		2413 3412 1432

$n = 5$



Sub-permutations from the decreasing rule

- Amazingly, these are exactly the permutations arising from cool-lex order as applied to permutations, recently presented in Aaron Williams, *Loopless Generation of Multiset Permutations Using a Constant Number of Variables by Prefix Shifts*, SODA 2009, pp. 987–996.
- However, the decrementing/7-order results are easier to understand and will be the main focus of the rest of this talk.

Ranking $\mathcal{T}$-order

Suppose $\pi = \pi_1\pi_2\cdots\pi_{k-1}n\pi_{k+1}\cdots\pi_n$

$$R(\pi) = \begin{cases} 0 & \text{if } n = 1, \\ n \cdot R(\pi_2\cdots\pi_n) & \text{if } k = 1, \\ (n-k+1) + n \cdot R(\pi_1\pi_2\cdots\pi_{k-1}\pi_{k+1}\cdots\pi_n) & \text{if } k > 1. \end{cases}$$

If π is a permutation let $inv_\pi(i)$ denote the number of pairs $i > j$ such that $\pi_i^- < \pi_j^-$ (inversions). For $i = 1, 2, \ldots, n$ define

$$a_i = \begin{cases} 0 & \text{if } inv_\pi(i) = i-1, \\ 1 + inv_\pi(i) & \text{if } inv_\pi(i) < i-1. \end{cases}$$

Then we can iterate our ranking recursion to obtain

$$R(\pi) = \sum_{j=0}^{n-1} a_{n-j} \cdot (n)_j.$$

This expression can be evaluated using $O(n \log n)$ operations.

Consequences and Further Results

- Given a permutation π in the Ucyle (in either order), its successor can be determined in time $O(n)$.
- Ranking of the 7-order Ucycle can be done in time $O(n + \nu(n))$, where $\nu(n)$ is the time required to compute the inversion vector inv_π. It is known that $\nu(n) = O(n \log n / \log \log n)$.
- Note: Ranking of Ucycles has applications in robotics.
- Setting $0 = \sigma_n$ and $1 = \sigma_{n-1}$, a Hamilton cycle in Ξ_n becomes a binary string of length $n!$.
- The underlying recursive structure of 7-order is similar to that of counting with multi-radix numbers from $(n-1) \times \cdots \times 3 \times 2$.
- Blocks of n bits can be output for each multi-radix number.
- In fact, we can generate those blocks of bits by a loopless algorithm.

Consequences and Further Results

- Given a permutation π in the Ucyle (in either order), its successor can be determined in time $O(n)$.
- Ranking of the 7-order Ucycle can be done in time $O(n + \nu(n))$, where $\nu(n)$ is the time required to compute the inversion vector inv_π. It is known that $\nu(n) = O(n \log n / \log \log n)$.
- Note: Ranking of Ucycles has applications in robotics.
- Setting $0 = \sigma_n$ and $1 = \sigma_{n-1}$, a Hamilton cycle in Ξ_n becomes a binary string of length $n!$.
- The underlying recursive structure of 7-order is similar to that of counting with multi-radix numbers from $(n-1) \times \cdots \times 3 \times 2$.
- Blocks of n bits can be output for each multi-radix number.
- In fact, we can generate those blocks of bits by a loopless algorithm.

Consequences and Further Results

- Given a permutation π in the Ucyle (in either order), its successor can be determined in time $O(n)$.
- Ranking of the 7-order Ucycle can be done in time $O(n + \nu(n))$, where $\nu(n)$ is the time required to compute the inversion vector inv_π. It is known that $\nu(n) = O(n \log n / \log \log n)$.
- Note: Ranking of Ucycles has applications in robotics.
- Setting $0 = \sigma_n$ and $1 = \sigma_{n-1}$, a Hamilton cycle in Ξ_n becomes a binary string of length $n!$.
- The underlying recursive structure of 7-order is similar to that of counting with multi-radix numbers from $(n-1) \times \cdots \times 3 \times 2$.
- Blocks of n bits can be output for each multi-radix number.
- In fact, we can generate those blocks of bits by a loopless algorithm.

The loopless algorithm

$a_n a_{n-1} \cdots a_1 \leftarrow 0\ 0\ \cdots\ 0;$

$d_{n-1} \cdots d_1 \leftarrow 1\ 1\ \cdots\ 1;$

$f_{n-1} \cdots f_1 \leftarrow n\ n{-}2\ \cdots\ 1;$

repeat

  $j \leftarrow f_1;\ f_1 \leftarrow 1;$

  $a_j \leftarrow a_j + d_j;$

(L1) **if** $d_j = -1$

(L2)   **if** $a_j = n{-}j{-}2$ **then** output(001^{n-2})

(L3)   **else** output($001^{j-1}0^{a_j+1}10^{n-j-a_j-3}$)

(L4) **else**

(L5)   **if** $a_j = 1$ **then** output(001^{n-2})

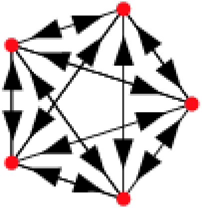
(L6)   **else** output($001^{j-1}0^{n-a_j-j}10^{a_j-2}$)

  if $a_j = 0$ **or** $a_j = n{-}j{-}1$

    then $d_j \leftarrow -d_j;\ f_j \leftarrow f_{j+1};\ f_{j+1} \leftarrow j+1;$

until $j \geq n;$

An "Application"



Task: list all $n!$ Hamilton paths in a weighted directed complete graph to solve some Traveling Salesperson style problem, so as to minimize the total number of changes between successive directed paths.

- **Transpositions**, central and extremal: new arcs = 3 or 2

$$\begin{array}{rll} \text{Before:} & \cdots abcd \cdots & bcd \cdots \\ \text{After:} & \cdots acbd \cdots & cbd \cdots \\ \text{New arcs:} & +\overrightarrow{ac} + \overrightarrow{cb} + \overrightarrow{bd} & +\overrightarrow{cb} + \overrightarrow{bd} \end{array}$$

- **Rotations**, σ_n or σ_{n-1}: new arcs = 1 or 2

$$\begin{array}{rll} \text{Before:} & abcd \cdots xyz & abcd \cdots xyz \\ \text{After:} & bcd \cdots xyza & abc \cdots xyaz \\ \text{New arcs:} & +\overrightarrow{za} & +\overrightarrow{ya} + \overrightarrow{az} \end{array}$$

- Conclusion: rotations are better than transpositions.
- Conclusion: σ_ns are better than σ_{n-1}s.

Final thoughts

- Is there a way to generalize these results to k-permutations of an n-set where $k < n - 1$?
- Are there other natural rules for obtaining spanning trees of $\mathcal{C}_n$ besides the decrementing and the decreasing rules?

The end

Thanks for coming! I hope you enjoyed the talk. Any questions?

Photo of smiling Buddha taken in Da Lat.