Cycle Structure of Permutations

August 31, 2001

1 Introduction

Consider an n-permutation p, start with an element $i \in [n]$ and walk through the elements of $[n]$ in the order

$$i, p(i), p(p(i)), p(p(p(i))), \ldots$$

Since p is one-one this walk takes us through distinct elements in $[n]$ until we return to i, thus forming a cycle. If there are any elements in $[n]$ not already visited, we can repeat this process by starting from an element not already visited. In this manner, p induces a partition of $[n]$ into disjoint cycles. This set of disjoint cycles is an alternate representation of p and is called the *cycle structure* of p.

For example, the permutation $(5, 3, 4, 2, 6, 1)$ has cycle structure $((1, 5, 6), (2, 3, 4))$. *Combinatorica* provides two functions `ToCycles` and `FromCycles` that translate a permutation into its cycle structure and back. The following example shows a 10-permutation transformed into its cycle structure via the function `ToCycles`. The resulting cycle structure contains 4 cycles, including two singletons. When `FromCycles` is applied to the constructed cycle structure, we get the original permutation back. This happens because a cycle structure has a unique corresponding permutation.

```
In[1]:= ToCycles[{10, 2, 4, 5, 7, 6, 3, 1, 8, 9}]

Out[1]= {{10, 9, 8, 1}, {2}, {4, 5, 7, 3}, {6}}

In[2]:= FromCycles[%]

Out[2]= {10, 2, 4, 5, 7, 6, 3, 1, 8, 9}
```

Here are a two more examples that provide more insights into the cycle structure of permutations and also a few more illustrations of *Combinatorica* functions.

- Performing a swap on an n-permutation corresponds to multiplying it by a permutation that has one 2-cycle and $(n-2)$ 1-cycles. In fact, a permutation with one 2-cycle and $(n-2)$ 1-cycles is called a *swap* or a *transposition*.

  ```
  In[3]:= Permute[{8, 1, 9, 2, 4, 3, 6, 5, 7, 10},
                    FromCycles[{{1}, {2}, {3}, {4}, {5, 8}, {6}, {7}, {9}, {10}}]
          ]

  Out[3]= {8, 1, 9, 2, 5, 3, 6, 4, 7, 10}
  ```

- A permutation with maximum cycle length 2 is called an *involution*. Equivalently, an involution is a permutation that is its own inverse.

```
In[4]:= Permute[FromCycles[p = {{1, 7}, {2}, {3, 5}, {4}, {6}}],
                FromCycles[p]
        ]

Out[4]= {1, 2, 3, 4, 5, 6, 7}
```

You should ponder over the typical questions (at least for this course) one can ask about involutions: how many size-n involutions are there? how can we enumerate all size-n involutions? how can we select, uniformly at random a size-n involution?

2 Application to Sorting

The cycle structure of a permutation plays a crucial role in unexpected places, as this example will show.

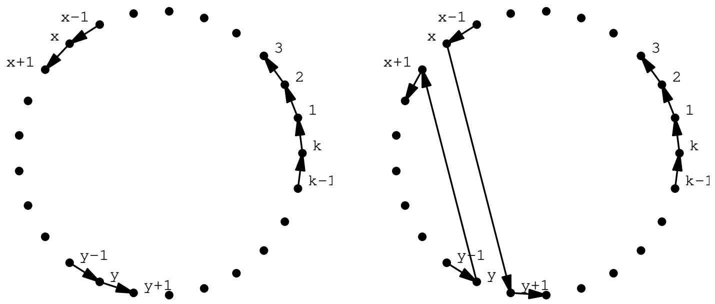
Given an n-permutation p what is the minimum number of swaps needed to sort p? Another way of asking this question is: what is the smallest sequence of swaps $q_1, q_2, \ldots$ such that

$$p \times q_1 \times q_2 \times \cdots = I.$$

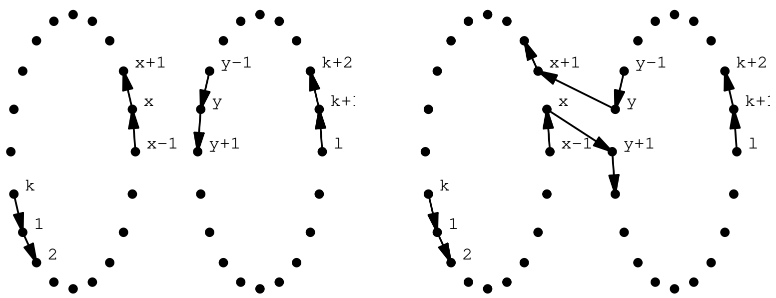
As mentioned earlier, if only adjacent transpositions are allowed, exactly as many transpositions are needed as the number of inversions in the permutation. In arbitrary swaps are allowed, inversions are not the right measure because a single swap can change the total number of inversions by a large amount.

We will now show that if p has c cycles then p can be sorted in $n - c$ swaps and this is the minimum needed. To see this let us first understand the effect that a swap has on the cycle structure of a permutation. There are two cases depending on whether the elements being swapped belong to the same cycle or not.

(i) Suppose that p contains the cycle $(1, 2, 3, \ldots, k)$ and let x and y, $1 \leq x < y \leq k$, be the elements being swapped. Swapping x and y splits the cycle into two: $(1, 2, \ldots, x, y + 1, \ldots, k)$ and $(y, x + 1, \ldots, y - 1)$. The "before" and "after" versions of the cycle are shown below pictorially.



(ii) Suppose that p contains the cycles $(1, 2, \ldots, k)$ and $(k+1, k+2, \ldots, \ell)$ and let x, $1 \le x \le k$ and y, $k < y \le \ell$, be the elements being swapped. Swapping x and y joins the two cycles into one and the result is $(1, 2, \ldots, x, y+1, \ldots, \ell, k+1, k+2, \ldots, y, x+1, \ldots, k)$. The two separate cycles and the single cycle that results from the swap are shown below.



Thus swapping changes the number of cycles by 1. The goal of sorting can be view as increasing the number of cycles in p from c to n. It follows that at least $n - c$ swaps are needed to sort p. The fact that we can find $n - c$ swaps that sort p follows from the fact that as long as p is not the identity, it has at least one cycle of size 2 or more and from such a cycle a pair of elements to swap can be arbitrarily chosen. Swapping such a pair of elements increases the number of cycles by 1.

In the following example we start with a 10-permutation with 4 cycles and swap the elements 3 and 4 to get a permutation with 5 cycles.

```
In[5]:= ToCycles[
          Permute[
             FromCycles[{{10, 9, 8, 1}, {2}, {4, 5, 7, 3}, {6}}],
             FromCycles[{{1}, {2}, {3,4}, {5}, {6}, {7}, {8}, {9}, {10}}]
          ]
        ]

Out[5]= {{10, 9, 8, 1}, {2}, {5, 7, 3}, {4}, {6}}
```

3 Stirling Numbers of the First Kind

The number of n-permutations with exactly k cycles is denoted $\left[{n \atop k}\right]$. The numbers $\left[{n \atop k}\right]$ for different values of n and k are called the *Stirling numbers of the first kind*, named after the Scottish mathematician, James Stirling (1692-1770). As you might expect, there are combinatorial numbers called the *Stirling numbers of the second kind* as well. These will show up in the context of set partitions. Stirling numbers, along with *binomial numbers*, *harmonic numbers*, and *Eulerian numbers*, are important combinatorial numbers that have the propensity to turn up in unexpected places.

For certain values of n and k Stirling numbers of the first kind are easy to determine. Here are some examples.

- For any positive integer n $\left[{n \atop 1}\right] = (n-1)!$ because a single cycle is just like a permutation except that it is invariant under cyclic shifts.

- For any positive integer n, $\left[{n \atop n}\right] = 1$ because a permutation with n cycles is just the identity permutation.

- For any positive integer n, $\left[{n \atop n-1}\right] = \binom{n}{2}$ because a permutation with $n-1$ cycles has one 2-cycle and $(n-2)$ 1-cycles. It is then just a matter of choosing the elements in the 2-cycle and that can be done in $\binom{n}{2}$ ways.

- Since a permutation has at least one cycles $\left[{n \atop 0}\right] = 0$ for any positive integer n. Also, a permutation with 0 elements has no cycles and so $\left[{0 \atop k}\right] = 0$ for any positive integer k. To be complete we just decree that $\left[{0 \atop 0}\right] = 1$.

```
In[6]:= Table[StirlingFirst[n, k], {n, 0, 10}, {k, 0, n}] // ColumnForm

Out[6]= {1}
        {0, 1}
        {0, 1, 1}
        {0, 2, 3, 1}
        {0, 6, 11, 6, 1}
        {0, 24, 50, 35, 10, 1}
        {0, 120, 274, 225, 85, 15, 1}
        {0, 720, 1764, 1624, 735, 175, 21, 1}
        {0, 5040, 13068, 13132, 6769, 1960, 322, 28, 1}
        {0, 40320, 109584, 118124, 67284, 22449, 4536, 546, 36, 1}
        {0, 362880, 1026576, 1172700, 723680, 269325, 63273, 9450, 870, 45, 1}
```

It is possible that by staring at this table you are able to observe that, like in the case *Pascal's triangle*, each entry depends on two other entries in the table, the one to its north and the one to its northeast. The precise relationship is given by the beautiful recurrence below.

$$\left[{n \atop k}\right] = \left[{n-1 \atop k-1}\right] + (n-1)\left[{n-1 \atop k}\right].$$

This recurrence allows us to compute $\left[{n \atop k}\right]$ in general and it is used in the implementation of the *Combinatorica* function `StirlingFirst`. To prove the recurrence partition the set of all n-permutations with k cycles into a set S_1 which contains permutations in which n occurs in a singleton cycle and a set S_2 which contains permutations in which n occurs in a cycle of size 2 or more. We now show that the size of S_1 is $\left[{n-1 \atop k-1}\right]$ by showing a bijection between S_1 and the set of all $(n-1)$-permutations with $(k-1)$ cycles. The bijection is simply obtained: from each permutation $p \in S_1$ remove n. This gives an $(n-1)$-permutation with $(k-1)$ cycles. We now show that the size of the set S_2 is $(n-1)\left[{n-1 \atop k}\right]$. We try the same trick and remove n from each permutation $p \in S_2$ and we get an $(n-1)$-permutation with k cycles. So we have defined a mapping from S_2 to the set of all $(n-1)$-permutations with k cycles. However, this mapping is not a bijection. In fact, it is an onto function which maps exactly $(n-1)$ permutations in S_2 to each element in the range. To see this observe that n can be inserted into an $(n-1)$-permutation with k cycles in exactly $(n-1)$ ways to get distinct n-permutations with k cycles. So the size of S_2 is $(n-1)$ times the size of the set of all $(n-1)$-permutations with k cycles. This proves the recurrence. The boundary cases of the recurrence occur when $n = 0$ or $k = 0$.

The above proof is also a recursive algorithm for generating all n-permutations with exactly k cycles. The related questions you should give some thought to are:

(i) How do we select a random n-permutation with exactly k cycles, uniformly at random?

(ii) Can we generate all n-permutations with exactly k cycles in "minimum change order" — an order in which each permutation is one swap away from the previous?

4 Hiding and Revealing Cycles

The parentheses in the cyclic representation of a permutation turn out to be unnecessary, since the cycles can be unambiguously represented without them. Before dropping the parentheses we need to bring the cycle structure into a *canonical form*. Rotate each cycle so that the minimum element is first and then sort the cycles in decreasing order of the smallest element. We can easily and unambiguously go back and forth between a canonical cycle structure and a permutation obtained by dropping the parentheses. The choice of a canonical cycle structure is not unique and alternate ways of doing this appear in the literature. *Combinatorica* has a function `HideCycles` that drops parentheses in from a cycle structure and gets a permutation.

In the following example, we start with a permutation $(6, 2, 1, 5, 4, 3)$ and compute its cycle structure $((6, 3, 1), (2), (5, 4))$. The cycles are first rotated to bring the smallest element first and we get the equivalent cycle structure $((1, 6, 3), (2), (4, 5))$. We then order the cycles in descending order by their first elements to get the canonical cycle structure $((4, 5), (2), (1, 6, 3))$. Dropping the parentheses from this results in the permutation $(4, 5, 2, 1, 6, 3)$. Notice that we started with a 6-permutation, applied `ToCycles` followed by `HideCycles` and we ended up with another 6-permutation. So the composition `HideCycles` $\circ$ `ToCycles` is a mapping from the set of n-permutations to the set of n-permutations.

```
In[7]:= ToCycles[{6, 2, 1, 5, 4, 3}]

Out[7]= {{6, 3, 1}, {2}, {5, 4}}

In[8]:= HideCycles[{{6, 3, 1}, {2}, {5, 4}}]

Out[8]= {4, 5, 2, 1, 6, 3}
```

We could drop the parentheses from the cycle structure of a permutation with impunity because we know that we can reconstruct the cycle structure. In particular, given a permutation $pi = (p_1, p_2, \ldots, p_n)$ we want to construct a cycle structure c such that `HideCycles` applied to c returns p. In going from c to p via `HideCycles`, we first reworked c into canonical form and hence each element p_i in p that is the minimum among the first i elements of p denotes the beginning of a new cycle. We will call such elements p_i *left-to-right minima*. For example, in the permutation $(4, 5, 2, 1, 6, 3)$, the elements 4, 2, and 1 are left-to-right minima and each of these is the beginning of a new cycle. *Combinatorica* has a function `RevealCycle` that takes a permutation p and reveals the cycles in it and returns the cycle structure c.

```
In[9]:= RevealCycles[{4, 5, 2, 1, 6, 3}]

Out[9]= {{4, 5}, {2}, {1, 6, 3}}
```

So `ToCycles` and `RevealCycles` are inverses of each other. When we start with a cycle structure c and pass it successively through `ToCycles` and `RevealCycles`, we get c back (except that c is now in canonical form). More importantly `FromCycles` $\circ$ `RevealCycles` is the inverse of `HideCycles` $\circ$ `ToCycles` implying that each of these is a bijection from the set of n-permutations to itself.

This bijection has several applications. For example, it implies that the number of n-permutations with k left-to-right minima is $\left[{n \atop k}\right]$. A more interesting implication is that the average number of cycles in an n-permutation is $\Theta(\log n)$. We show this by showing that the average number of left-to-right minima in an n-permutation is $\Theta(\log n)$. This analysis uses a technique quite common in the analysis of randomized algorithms. Pick an n-permutation $p = (p_1, p_2, \ldots, p_n)$ uniformly at random. For each $i \in [n]$, let a binary random variable X_i denote whether p_i is a left-to-right minima. In other words, $X_i = 1$ if and only if p_i is left-to-right minima. Now note that $X = \sum_{i=1}^{n} X_i$ is the random variable that denotes the total number of left-to-right minima p contains. Our goal is to calculate $E[X]$. By linearity of expectation, $E[X] = \sum_{i=1}^{n} E[X_i]$. Now let r_i denote the probability that p_i is a left-to-right minima. It is easy to see that $r_1 = 1/i$. Furthermore, $E[X_i] = 1 \cdot r_i + 0 \cdot (1 - r_i) = r_i = 1/i$. It follows that

$$E[X] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} r_i = \sum_{i=1}^{n} 1/i = H_n,$$

where H_n is the nth Harmonic number. It is well known that $H_n = \Theta(\log n)$ and the claim follows. The following experiment shows that the above prediction is quite accurate. The 120 5-permutations are scanned, the number of cycles in each is calculated, and the mean of these numbers is reported.

```
In[10]:= Mean[ Map[Length[ToCycles[#]]&, Permutations[5]]] // N

Out[10]= 2.28333

In[11]:= HarmonicNumber[5] // N

Out[11]= 2.28333
```

In the following experiment, we consider 100-permutations. There are too many of them to scan so we resort to random sampling. We pick a random sample of size 1000 and find the mean cycle length of these. The result is close to H_{100}.

```
In[12]:= Mean[ Map[ Length[ ToCycles[#]]&, Table[RandomPermutation[100], {1000}]]] // N

Out[12]= 5.222

In[13]:= HarmonicNumber[100] // N

Out[13]= 5.18738
```

The fact that the average number of cycles in an n-permutation is H_n can be equivalently stated as the identity:

$$\sum_{k=1}^{n} \frac{k}{n!} \cdot \left[{n \atop k} \right] = H_n.$$

In effect, what we have done is provide a probabilitic proof of this identity.