

Notes on Permutations: A Chalkboard Algorithm

Tyler Neylon

422.2022

[Formats: [html](#) | [pdf](#)]

This is a continuation of notes about permutations; this series began with [some notes on cycles and transpositions](#).

This short article describes a chalkboard method for multiplying (that is, composing) permutations written in cycle notation. When I call it a *chalkboard method*, I mean that it's useful for humans doing simple math by hand — akin to long-hand addition taught in school — and as opposed to an algorithm designed for a computer.

This article can be read as an extension to an earlier note about permutations, although this one can be read independently, assuming that you know both what permutation multiplication (synonymously, composition) is, and what cycle notation is. (If not, you can learn by reading [the previous note in this series](#).)

The chalkboard algorithm I'll describe is something I derived for fun. I haven't put effort into discovering if someone else has come up with it before me.

Let's get started. As an example, consider

$$\sigma = (1\ 3\ 7\ 2\ 5\ 4) \quad \text{and} \quad \tau = (1\ 2)(4\ 6). \tag{1}$$

We can define the product $\sigma \cdot \tau$ as the permutation μ where

$$\mu(x) = (\sigma \cdot \tau)(x) = \tau(\sigma(x)). \tag{2}$$

I've chosen to work with the intuition that the element on the left (σ) “happens first;” this notation seems more natural to me. Some authors choose the opposite ordering because it avoids the awkward reversal of symbols in (2).

In our example,

$$\mu = (1\ 3\ 7)(2\ 5\ 6\ 4),$$

as can be verified by asking, for each $x \in \{1, 2, \dots, 7\}$, what is $y = \sigma(x)$? followed by asking what is $\tau(y)$?

Let's informally call this a 2-lookup method, in the sense that a human will perform a scan of σ to find $\sigma(x)$ and then another scan of τ to find $\tau(y)$ for each

input x to μ . This article improves on this by providing an algorithm that is a 1-lookup method. In some informal sense we can say it's twice as fast for a human to execute.

A computer can multiply permutations even more efficiently. We could encode a permutation as a hash map (a `dict` in Python, or an `object` in JavaScript, for example) where the keys are the input numbers and the values are what those values map to. In this case, each lookup of a value $\mu(x)$ takes constant time, so that computing all of μ requires $O(n)$ time, where n is the size of the domain of the permutations. I don't consider a human working on a chalkboard to have access to constant-time lookups, so this algorithm is only for computers.

1 A Worked Example

I'll illustrate the method by working through example (1), and then I'll summarize the general process.

Say that a permutation is *single-cycle* when its cycle notation has one cycle: $\sigma_1 = (1\ 4\ 2)$ is single-cycle, whereas $\sigma_2 = (1\ 2)(3\ 5)$ is not.

- Step 1. [Write the problem.] Write $\sigma \cdot \tau$ in cycle notation; include single-element cycles in τ , but not of σ . Write a dot over $\tau(a)$, where a is the left-most element in σ 's cycle:

$$\sigma \cdot \tau = \frac{(1\ 3\ 7\ 2\ 5\ 4)}{\sigma} \frac{(1\ \dot{2})(3)(4\ 6)(5)(7)}{\tau}.$$

$$a = 1 \quad \tau(a) = 2.$$

- Step 2. [Process the problem.] Repeat this until σ is exhausted: Draw a line over element \bar{x} in σ , where x is the right-most, not-yet-overlined element of σ (above, we have $x = 4$).

Call the dotted element in τ as \dot{y} (above, we have $y = 2$). In τ , insert a new \hat{y} after x and add a dot over $z = \tau(x)$, like so:

$$x\ \hat{y}\ \dot{z}.$$

Cross out the previous dot: \acute{y} .

Here is how step 2 applies to our example:

$$\sigma \cdot \tau = (1\ 3\ 7\ 2\ 5\ \bar{4})\ (1\ \acute{2})(3)(4\ \dot{6})(5)(7)$$

$$\sigma \cdot \tau = (1\ 3\ 7\ 2\ \bar{5}\ \bar{4})\ (1\ \acute{2})(3)(4\ \dot{\hat{6}})(\dot{\hat{5}})(7)$$

$$\sigma \cdot \tau = (1\ 3\ 7\ \bar{2}\ \bar{5}\ \bar{4}) (\overset{\hat{1}}{1}\ \overset{\hat{2}}{2}) (\overset{\hat{3}}{3}) (\overset{\hat{4}}{4}\ \overset{\hat{6}}{6}) (\overset{\hat{5}}{5}) (\overset{\hat{7}}{7})$$

$$\sigma \cdot \tau = (1\ 3\ \bar{7}\ \bar{2}\ \bar{5}\ \bar{4}) (\overset{\hat{1}}{1}\ \overset{\hat{2}}{2}) (\overset{\hat{3}}{3}) (\overset{\hat{4}}{4}\ \overset{\hat{6}}{6}) (\overset{\hat{5}}{5}) (\overset{\hat{7}}{7})$$

$$\sigma \cdot \tau = (1\ \bar{3}\ \bar{7}\ \bar{2}\ \bar{5}\ \bar{4}) (\overset{\hat{1}}{1}\ \overset{\hat{2}}{2}) (\overset{\hat{3}}{3}) (\overset{\hat{4}}{4}\ \overset{\hat{6}}{6}) (\overset{\hat{5}}{5}) (\overset{\hat{7}}{7})$$

$$\sigma \cdot \tau = (\bar{1}\ \bar{3}\ \bar{7}\ \bar{2}\ \bar{5}\ \bar{4}) (\overset{\hat{1}}{1}\ \overset{\hat{2}}{2}) (\overset{\hat{3}}{3}) (\overset{\hat{4}}{4}\ \overset{\hat{6}}{6}) (\overset{\hat{5}}{5}) (\overset{\hat{7}}{7})$$

At the end of that last equation, we've drawn a line over all the elements of σ , indicating that we're done with step 2.

- Step 3. [Write the answer.] Repeat this until all the elements are used (meaning they are included in the answer).
 - [Step 3a.] Write “ a ” where a is the least unused element appearing in σ or τ . If all the elements are used, you're done.
 - [Step 3b.] Let b be the right-most element in the answer so far. Let c be the element after b in our final notation for τ : If the notation is

$$b\ \hat{c}\ d,$$

then c is the inserted element; otherwise $c = \tau(b)$. If $c = a$, then write “)” and go to step 3a; otherwise write “ c ” and repeat this step (step 3b).

To finish our example:

$$\sigma \cdot \tau = (1\ 3\ 7)(2\ 5\ 6\ 4).$$

The first time you read about this method, it may seem like a lot, but after working a few examples, it feels natural and easy.

2 Why It Works

In this section, I'll define μ as the permutation produced by this method, and it will be my job to show that $\mu = \sigma \cdot \tau$.

At the start of each iteration of step 2, we're going to insert element y after x in our notation of τ , where

- x = the right-most un-overlined element of σ , and
- y = the dotted element in τ .

Effectively, we're recording that $\mu(x) = y$.

In each case, y is chosen as, informally, the element in τ after [the element in σ after x]. In other words,

$$y = \tau(\sigma(x)). \tag{3}$$

whenever we write $x\hat{y}$.

At the end of an iteration of step 2, the newly-dotted element z will satisfy $z = \tau(x)$. This sets us up for the next iteration of step 3, in which we'll begin looking at a new x' with $\sigma(x') = x$, preparing us to continue fulfilling (3) for the next value $y' = z$.

When we write out the answer in step 3, either element x in τ has an inserted element after it or not. If element y is inserted, then we have

$$\mu(x) = y = \tau(\sigma(x)) \tag{4}$$

by (3). If x has no inserted element after it, then x never appeared in σ 's cycle notation, meaning $\sigma(x) = x$. Then

$$\mu(x) = \tau(x) = \tau(\sigma(x)). \tag{5}$$

Since either (4) or (5) holds for all inputs x to μ , we must have $\mu = \sigma \cdot \tau$, concluding the proof of correctness.

3 The General Process

The steps written in §1 will work for any single-cycle permutation σ . It's not difficult to extend this method to work for any σ .

Below is the general method, simplified with some shorthand notation that I believe will be clear in the context from §1. This is the same as above, with some changes in step 2 to accommodate multi-cycle σ s.

- Step 1. [Write the problem.] Write $\sigma \cdot \tau$ in cycle notation; include single-element cycles (x) in τ whenever $\sigma(x) \neq x$; exclude single-element cycles in σ .
- Step 2. [Process the problem.] Repeat until σ is overlined:
 - [2a.] Set $x :=$ the right-most non-overlined element in σ , or “”.
 - [2b.] If $x =$ “” then let $a :=$ the left-most element in this cycle of σ . Write a dot over $\tau(a)$; go to step 2a.
 - [2c.] In τ , write $x\hat{y}z$, where y is the dotted element of τ . Cross out y 's dot in τ .
- Step 3. [Write the answer.] Repeat this until all the elements are used (meaning they are included in the answer).

- [3a.] Write “(a” where a is the least unused element appearing in σ or τ . If all the elements are used, you’re done.
- [3b.] Let b be the right-most element in the answer so far. Let c the element after b in our final notation for τ : If the notation is

$$b \hat{c} d,$$

then c is the inserted element; otherwise $c = \tau(b)$. If $c = a$, then write “)” and go to step 3a; otherwise write “ c ” and repeat this step (step 3b).

It’s straightforward to apply essentially the same proof of correctness to this general version of the method.