Permutation Parity

Lucas Bachmann

February 12, 2024

1 Intro

A warning to all: while I will try my best to be as clear and simple as possible, I will leaning into mathematical and programming notation where I feel is appropriate. I highly suggest getting yourself familiar with this, as it will make any future endeavours into mathematics and computer science a lot more bearable.

Furthermore, this document is an extension of a blog post on my website, luckys-corner.com, titled "Impossible! An Introduction to Invariants". As such, there will be references to it in this article, although this document should still be self-contained and readable as is.

2 Permutation Parity

Throughout this section, I will write down a permutation as follows:

Definition 1. *A permutation of* $\{1, 2, \ldots, n\}$ *is a list of* n *numbers such that each number from* 1 *to* n *appears in the list exactly once.*

If l is one such list, I will explicitly write down it's content using square brackets (e.g. $[3, 4, 6, 1, 5, 2]$), and I will use $l[k]$ to refer to the k^{th} element of the list (e.g. if $l = [3, 4, 6, 1, 5, 2]$, then $l[3] = 6$ and $l[6] = 2$).

In the context of balls and pedestals from the blog post, we say that $l[k]$ is the number of the ball sitting on pedestal k.

In particular, we'll define the initial permutation I_n to be the list of n numbers that satisfies $I[k] = k$ for every k i.e. all of the balls are on the pedestal with the same number. For instance, $I_6 = [1, 2, 3, 4, 5, 6]$.

Next, we'll need to define a swap:

Definition 2. *A swap of elements i and j (for $i \neq j$) is an operation that takes a list, and returns the list with the numbers at position i and j swapped. We'll denote the swap of elements i and j over list l as $(i, j)l$.*

So for instance $(2,4)[3,4,6,1,5,2] = [3,1,6,4,5,2]$ and $(5,6)[3,4,6,1,5,2] = [3,4,6,1,2,5]$. In particular, we'll allow ourselves to chain these swap operations together, so $(2,4)(5,6)(2,5)[3,4,6,1,5,2] = (2,4)(5,6)[3,5,6,1,4,2]$
$= (2,4)[3,5,6,1,2,4] = [3,1,6,5,2,4]$.

We'll also say that a swap is a neighbour swap if it swaps elements that are next to each other (e.g. $(1,2)$ or $(5,6)$).

Finally, we'll define when a permutation is even or odd:

Definition 3. *A permutation l of n numbers is generated by swaps $a_1, a_2, \ldots, a_k$ if $l = a_k \ldots a_2 a_1 I_n$. A permutation l is even if it is generated by an even number of swaps. A permutation l is odd if it is generated by an odd number of swaps.*

For instance, $[3,4,6,1,5,2] = (1,4)(2,6)(3,4)(4,6)[1,2,3,4,5,6]$, thus $[3,4,6,1,5,2]$ is generated by the swaps $(4,6), (3,4), (2,6), (1,4)$ and so it is an even permutation. Similarly, $[3,4,6,1,2,5] = (1,4)(2,5)(3,4)(4,5)(5,6)[1,2,3,4,5,6]$, thus $[3,4,6,1,2,5]$ is generated by the swaps $(5,6), (4,5), (3,4), (2,5), (1,4)$ and so it is an odd permutation.

Note that the way even and odd permutations were defined does not make them mutually exclusive; it could be the case that a permutation is generated by both an even number of swaps, and an odd number of swaps. We'll show now that this cannot be the case, by considering a calculable quantity.

For $i < j$, define

$$out_{i,j}(l) = \begin{cases} 0 \text{ if } l[i] < l[j] \\ 1 \text{ if } l[i] > l[j] \end{cases}$$

which indicates if the list is 'decreasing'. For instance, if $l = [3,4,6,1,5,2]$, then $out_{1,3}(l) = 0$ since $l[1] = 3 < 6 = l[3]$, and $out_{2,4}(l) = 1$ since $l[2] = 4 > 1 = l[4]$.

We'll furthermore define

$$tout(l) = \sum_{i<j} out_{i,j}(l)$$

to be the sum of $out_{i,j}(l)$ over all possible values of i and j. Below is the table for permutations of $\{1,2,3\}$:

l	$tout(l)$
$[1,2,3]$	0
$[1,3,2]$	1
$[2,1,3]$	1
$[2,3,1]$	2
$[3,1,2]$	2
$[3,2,1]$	3

Our first results is as follows:

Lemma 1. *Let l be a permutation, and consider some neighbour swap $s = (i, i+1)$. If $tout(l)$ is even, then $tout(sl)$ is odd. Similarly, if $tout(l)$ is odd, then $tout(sl)$ is even.*

Proof. We have the following equalities:

- if m, k is not equal to i or $i+1$, and $m < k$ then $out_{m,k}(l) = out_{m,k}(sl)$.
- if $k < i$, then $out_{k,i}(l) = out_{k,i+1}(sl)$ and $out_{k,i+1}(l) = out_{k,i}(sl)$.
- if $k > i+1$, then $out_{i,k}(l) = out_{i+1,k}(sl)$ and $out_{i+1,k}(l) = out_{i,k}(sl)$.

therefore, we have

$$\begin{aligned} tout(l) - out_{i,i+1}(l) = &\sum_{\substack{k<m \\ k,m\neq i,i+1}} out_{k,m}(l) + \sum_{k<i} (out_{k,i}(l) + out_{k,i+1}(l)) \\ &+ \sum_{i+1<k} (out_{i,k}(l) + out_{i+1,k}(l)) \\ &= \sum_{\substack{k<m \\ k,m\neq i,i+1}} out_{k,m}(sl) + \sum_{k<i} (out_{k,i+1}(sl) + out_{k,i}(sl)) \\ &+ \sum_{i+1<k} (out_{i+1,k}(sl) + out_{i,k}(sl)) \\ &= tout(sl) - out_{i,i+1}(sl) \end{aligned}$$

Therefore, $tout(sl) = tout(l) + (out_{i,i+1}(sl) - out_{i,i+1}(l))$. Now, since we are swapping $l[i]$ and $l[i+1]$, therefore if $out_{i,i+1}(l) = 0$, then $out_{i,i+1}(sl) = 1$, and if $out_{i,i+1} = 1$, then $out_{i,i+1}(sl) = 0$, thus $out_{i,i+1}(sl) - out_{i,i+1}(l) = \pm 1$. Therefore, if $tout(l)$ is even, then $tout(sl)$ is odd, and if $tout(l)$ is odd, then $tout(sl)$ is even. □

Technically, this proof tells us a little more, namely that the value of $tout(l)$ after a neighbour swap changes by exactly 1.

This result allows us to build up to the case when s is a general swap. This is because we can decompose the swap (i, j) into a sequence of neighbour swaps as follows (let $a = l[i]$ and $b = l[j]$):

1. If a and b are separated by n spaces, we perform $n-1$ neighbour swaps to bring b next to a.
2. We swap a and b.
3. We perform another $n - 1$ neighbour swaps to bring a back to where b began.

for a total of $2 \cdot n - 1$ neighbour swaps. For instance, we have $(1, 4)l = (3, 4)(2, 3)(1, 2)(2, 3)(3, 4)l$ for any permutation l with at least 4 elements. The fact that $2 \cdot n - 1$ is odd is very important, because it gives us the following general statement.

Lemma 2. *Let l be a permutation, and consider some general swap $s = (i, j)$. If $tout(l)$ is even, then $tout(sl)$ is odd. Similarly, if $tout(l)$ is odd, then $tout(sl)$ is even.*

Proof. Suppose $tout(l)$ is even. Then, from the above remark, we can decompose s into an odd number of neighbour swaps $t_1, t_2, t_3, \ldots, t_k$ i.e. $sl = t_k \ldots t_3 t_2 t_1 l$. We thus get by repeatedly applying Lemma 1 that, if $tout(l)$ is even, then $tout(t_1 l)$ is odd, which implies $tout(t_2 t_1 l)$ is even, which implies $tout(t_3 t_2 t_1 l)$ is odd etc. Since we have an odd number of neighbour swaps, we are flipping the 'evenness' of the value of $tout$ an odd number of times, which means $tout(t_k \ldots t_3 t_2 t_1 l) = tout(sl)$ is odd. A similar argument can be made if $tout(l)$ is odd. □

All of this brings us to the result that is permutation parity:

Theorem 1. *If a permutation l is even, then $tout(l)$ is even. Similarly, if a permutation l is odd, then $tout(l)$ is odd. In particular, since $tout(l)$ can't be both even and odd, this means a permutation l can't be both even or odd.*

Proof. We perform a similar proof to that of Lemma 2: namely if l is of length n, we write $l = s_k \ldots s_2 s_1 I_n$. We note that $tout(I_n) = 0$ because the list is in order, so all of the components $out_{i,j}(l) = 0$ for all $i < j$. Therefore, since $tout(I_n)$ is even, we get $tout(s_1 I_n)$ is odd, so $tout(s_2 s_1 I_n)$ is even etc. If l is even, this means that the sequence $s_1, s_2, \ldots, s_k$ has an even number of swaps, thus $tout(l) = tout(s_k \ldots s_2 s_1 I_n)$ is even. Similarly, if l is odd, then $tout(l) = tout(s_k \ldots s_2 s_1 I_n)$ is odd. □

3 Further Thoughts

If you are familiar with some Group Theory, then you might be more familiar with permutations in the form of cycles e.g. $[2, 4, 6, 1, 5, 3] = (1, 2, 4)(3, 6)$. In this case, the equivalent formulation of permutation parity is as follows: a permutation is even if it has an even number of even-length cycles, and odd otherwise. So, for instance $(1, 2)(3, 4)$ is even, while $(1, 2, 3)(4, 5)$ is odd. In this case, the difficulty would be proving that applying a swap (or a 2-cycle) changes the 'evenness' of the number of even-length cycles (which is true; in fact, applying a 2-cycle changes the number of even-length cycles by exactly 1).