Quick search yielded this question on [stack-exchange](https://math.stackexchange.com/questions/2183520/number-of-permutations-of-matrices-with-unique-rows-and-columns). which guided me to read on Burnside's lemma. Since row exchanges and column exchanges are commutative, it’s enough to find the Cycle index for both separately then multiply the answers.

Swapping rows is equivalent of having a permutation. A group where all permutations are symmetric is known as the symmetric set . This is already solved, but I dug deeper into how the formulas work.

So let’s think about the order of the rows, for example, for , the possibilities are: 123, 132, 213, 231, 312, 321. All of these are considered the same, and we need to calculate the cycle index if we want to apply Burnside’s lemma. If we draw arrows from each node to where it ends up, we can find the cycle decomposition of each permutation. I’ll denote to a cycle of length by , such that a cycle of size two repeated three times is denoted by: .

We can extract the following decomposition from :

The Cycle index is the average of these decomposition .

So once we extract for both columns and rows, we can combine them using the rule

Notice that this combines a singular with a singular , to combine:

Once we do that, we have the cycle index for the whole operation, and plugging in number of colors to every will give us the needed answer according to Burnside’s lemma.

This is a straightforward algorithm where is the number of unique elements in the cycle index. Notice that is just the number of possible partitions of into non-negative integers, aka partition function ([A000041 - OEIS](https://oeis.org/A000041)) Why? Observe that sum of cycle sizes must be , so the 3 terms above correspond to the partitions: , each of them has a coefficient that we don’t know, but we’ll get by combining all terms from the preprocessing step. Our upper limit for is , so clearly, is not the bottleneck, but rather is.

We need a faster way to calculate , we know that the number of unique terms is so we need to know how much each term contributes.

To do that, we simply need to know how many ways we can organize the elements such that they form the expected cycles.

Let’s partition by the following partition we’ll combine the into a tuple , so , for example or

We need to find the coefficient of this partition. We will split into subsets each of which will correspond to a cycle, so we’ll split into subsets of size . Reminder that multinomial is used to split into subsets of sizes :

So, we need to do this: , there are ways to organize a cycle of size (so multiply by ) and any of the cycles can be interchanged (so divide by )

This leads to:

Putting everything together we get:

is such that .

This has the complexity , so just .