Vamshi Jandhyala

Can You Swap the Cups?

✠

September 2026

A puzzle from Fiddler on the Proof, in which a random walk on six states turns out to be two lotteries taking turns.

Problem

Three cups labelled A, B and C stand in a row. A friend picks two of the three positions at random and swaps the cups there, and keeps doing so, a fresh random pair each time, until the cups are back in the order A, B, C. On average, how many swaps does that take?

The shape of the problem

There are six orders of three cups, and a swap moves between them. The useful thing is to see which orders a single swap can reach.

Call an order even or odd according to how many pairs of cups stand the wrong way round, and every swap flips it. Three orders are even: ABC itself and its two rotations, BCA and CAB. The other three, ACB, BAC and CBA, are odd, since each is a single swap away from ABC.

Now fix any order and try the three possible swaps. They give three different orders, all of the other parity, and there are only three of those. So from every order the three swaps lead to the three orders of the opposite parity, one each. In the language of graphs the swaps form $K_{3,3}$: every even order is joined to every odd one.

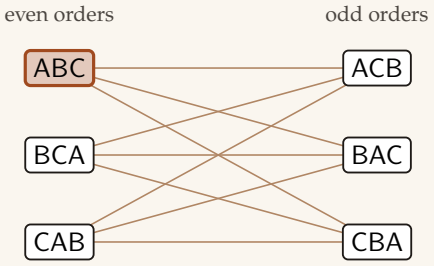

Figure 1: *The six orders, with a line wherever one swap turns one into the other. Every even order is one swap from every odd order.*

That is the whole puzzle. After every swap the cups stand in a uniformly random

order of the new parity, whatever happened before, so the walk forgets its past completely at every single step and each swap is a fresh draw.

Solution

Look only at the even-numbered swaps: the second, fourth, sixth and so on. Before each of them the cups are in an odd order, and after it they are in one of the three even orders, chosen uniformly and independently of everything earlier. So each even-numbered swap restores ABC with probability $\frac{1}{3}$, independently, and the odd-numbered swaps never can.

The number of even-numbered swaps needed is therefore geometric with success probability $\frac{1}{3}$, which takes 3 attempts on average, and each attempt costs two swaps:

$$\boxed{\mathbb{E}[\text{swaps}] \;=\; 2 \times 3 \;=\; 6.}$$

There is a second route to the same number that needs none of the structure. In the long run every one of the six orders is equally likely, and for a random walk of this kind the average time to come back to a state is the reciprocal of the long-run share of time spent there. One order in six gives a return time of six. It is reassuring that the two agree, and the first explains why.

Extra Credit

Including the starting order, there are six orders. On average, how many swaps does it take until all six have appeared?

Two lotteries taking turns

The structure above says more than the main puzzle used. The odd-numbered swaps draw an odd order uniformly at random, each draw independent of the others, and the even-numbered swaps do the same for the even orders. So the walk is two separate lotteries, run in alternation, and the task is to see every prize in both.

On the odd side all three orders are new. Collecting three equally likely coupons takes $1 + \frac{3}{2} + 3 = \frac{11}{2}$ draws on average. The kth odd draw happens at swap $2k - 1$, so the odd side is complete at swap $A = 2N_o - 1$, where N_o is the number of draws, and

$$\mathbb{E}[A] \;=\; 2 \cdot \tfrac{11}{2} - 1 \;=\; 10.$$

On the even side ABC is already seen and a draw of it is wasted, so only two prizes are wanted from three. The first of them comes in $\frac{3}{2}$ draws on average and the second in 3 more, so $\frac{9}{2}$ draws, and the kth even draw is swap $2k$. The even side is complete at swap $B = 2N_e$, with

$$\mathbb{E}[B] \;=\; 2 \cdot \tfrac{9}{2} \;=\; 9.$$

All six have appeared once both sides are complete, at swap $\max(A, B)$. The two sides run at the same time, so the answer is well short of $10 + 9$, and the identity

$$\max(A, B) \;=\; A + B - \min(A, B)$$

isolates exactly how much the overlap saves.

The overlap

The two lotteries use separate draws, so A and B are independent and

$$\mathbb{E}[\min(A,B)] \;=\; \sum_{t\ge 0} \Pr(A > t)\,\Pr(B > t).$$

Both tails are short inclusion-exclusion counts. At least one of the three odd orders is still missing after $k \ge 1$ draws with probability

$$\Pr(N_o > k) \;=\; 3\left(\tfrac{2}{3}\right)^k - 3\left(\tfrac{1}{3}\right)^k,$$

and at least one of the two wanted even orders is still missing after k draws with probability

$$\Pr(N_e > k) \;=\; 2\left(\tfrac{2}{3}\right)^k - \left(\tfrac{1}{3}\right)^k.$$

After $t = 2m$ swaps each side has had m draws, and after $t = 2m+1$ the odd side has had one more. Multiplying the tails gives terms in $\left(\frac{4}{9}\right)^m$, $\left(\frac{2}{9}\right)^m$ and $\left(\frac{1}{9}\right)^m$, each a geometric series. The even values of t contribute $\frac{1009}{280}$ and the odd values $\frac{891}{280}$, so

$$\mathbb{E}[\min(A,B)] \;=\; \tfrac{1900}{280} \;=\; \tfrac{95}{14},$$

and

$$\boxed{\mathbb{E}[\text{swaps to see all six}] \;=\; 10 + 9 - \tfrac{95}{14} \;=\; \tfrac{171}{14} \;\approx\; 12.21.}$$

The 7 in the denominator comes from the cross term, where the ratio $\frac{2}{3}\cdot\frac{1}{3} = \frac{2}{9}$ leaves $1 - \frac{2}{9} = \frac{7}{9}$.

A comparison puts the number in context. If instead of swapping, the friend shuffled the cups into a uniformly random order every time, seeing the five new orders would take $6\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) = 13.7$ shuffles on average. Swapping does better, by about one and a half steps. A swap can never leave the cups where they were, and it can never land on an order of the parity just visited, so fewer draws are wasted than with a full shuffle.

The computation

Three checks, each independent of the others and of the argument above. The first solves the full Markov chain on (orders seen so far, current order) in exact fractions, knowing nothing about parity or $K_{3,3}$. The second sums the tail series of the two-lottery argument numerically. The third swaps cups.

```
def swap(p, t):
    q = list(p); q[t[0]], q[t[1]] = q[t[1]], q[t[0]]; return tuple(q)

T = [(0, 1), (0, 2), (1, 2)]
start = (0, 1, 2)
for _ in range(R):
    p, seen, k = start, {start}, 0
    while len(seen) < 6:
        p = swap(p, T[rng.integers(3)]); k += 1; seen.add(p)
    cover.append(k)
```

The exact chain gives 6 and $\frac{171}{14}$. The series gives 12.2142857 … Four hundred thousand simulated games give 6.009 ± 0.008 and 12.221 ± 0.009, and every one of the simulated return times is even.