

## HydratedHakofos Write-Up

---

Isaac Abboudi

The logic of the code is based on the idea that if one gets dehydrated at a certain point, there is no table before that point that could be the starting table. Let's say I start at table[0] and begin my hakafot, when I engage some middle table[i] and I am dehydrated, that means that no previous table supplied me with enough water to make it this far. Further, any subsequent table that MIGHT provide enough water will need to come before this table in the loop. So the next step is to begin the loop from table[i+1].

My code is around  $O(n)$  I believe. First beginning at table[0] to test for a valid hakafa, and if it returns 'dehydrated', then we start the next loop from the next dehydrated table.

I believe the worst case scenario is if at every table started out dehydrated until the last table was the only valid option. In which case we've only looped over the whole array twice.

I found it very difficult to implement the array wrap-around. I tried some stuff with modulo, but couldn't crack it. I settled with using two counters.

Another difficulty was my sephardic heritage and remembering to write hakofos with o's and an s.