The provided code made this lab much easier to complete for me than some of the previous ones. This one felt more like a puzzle where I just had to fit the pieces together. I was able to understand cuckoo hashing pretty easily. In the add method, each value gets two possible locations determined by two hash functions. First, we try to place the value in the first table. If that spot is already full, it swaps the existing element out and tries to insert it into its second position in table two. If that spot is also full, the process repeats, swapping values back and forth until an open spot is found or the swap limit is reached. When the counter hits max swaps, the table resizes and rehashes all elements to make more space (I wonder if there may be a way to improve this by not having to refill the table each time?). The rest of the methods were accurately described in the lab sheet as easy. Nonetheless, I had a few bugs in the first couple submissions related to syntax and just not accessing the correct objects, but the logic was really straightforward. Removing and finding could be done in constant time because all we had to do was check the spots in both tables because the value would either be in the one of the tables or in none of them. Overall, this lab helped me actually see how cuckoo hashing keeps things organized.