CMSC420 PROJECT 3 REPORT

For this project, I decided to use three different hash functions to insert each value into the table. The three hash functions are all different as well; hash_1 is a linear hash function, with a = 47 and b = 23; hash_2 is a simpler multiplicative hash function, with just a = 137; finally, hash_3 is a universal hash function, where a = 67, b = 3, and p = 3007, a prime number larger than 3000, because we know in the test cases that the maximum value m can be is 3000.

In order for my find operation to work, it first relied on the correctness and functionality of my insert operation. As a result, my insert function works very simply; based on the k value, in this case only being either 2 or 3, the function would use either 2 of the hash functions or all 3 of them, changing the entries at those indices to a 1 instead of a 0. Since the maximum size of the table is 3000 but we are only intending on inserting 200 values, there should be minimal collisions. As a result, my pseudocode for my find operation worked based on the indices:

```
Def find( x):
    If k == 2:
        If arr[hash_1] == 1 and arr[hash_2] ==1
            Return true
        Return false
    Else:
        If arr[hash_1] == 1 and arr[hash_2] == 2 and arr[hash_3] == 3
            Return true
        Return false
```

Test Answer Analyses:

When k = 2, q ~= 6.57, or about 6-7 false positives
When k = 3, q ~= 1.19124856 or about 1-2 false positives

Test Answer 1 False Positives: 2
Test Answer 2 False Positives: 0

Discussion: Although not exact, my numbers are below the expected value of nq false positives, particularly for the first test. This surprises me because I didn't expect to be this far below the expected number, although it makes sense after thinking about it, due to the different hash functions I used.

Test Answer 3 False Positives: 1
Test Answer 4 False Positives: 0

Discussion: Similar to the first two test answers, I was surprised to find that the amount of false positives I got was much lower than the expected number. However, what was even more surprising for me was that the random T sets gave me lower numbers. I had expected them to be slightly larger than before, as they were randomized, meaning that it wouldn't consistently be 2 and 0 every time.