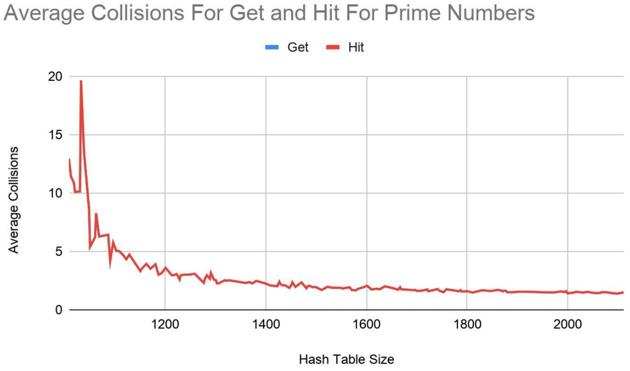
Alex Yuk | January 11th, 2020

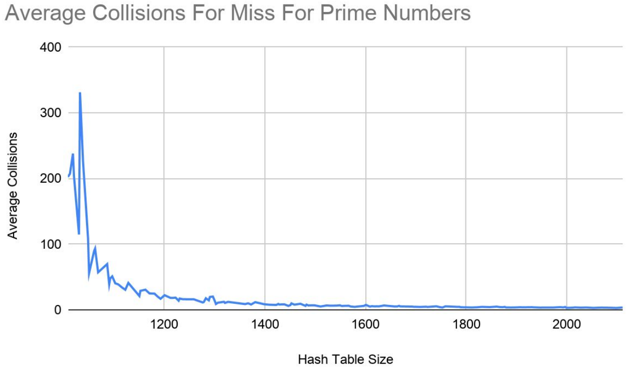
Assignment 8 - Hash Tables Write Up

Graph Observations:

The put method and get method for all hits has the exact same number of collisions. This is because both methods use the same for loop for linear probing when searching for a key after collision. The only difference between the two methods is that put adds a new value using the key into the table whereas get returns it the value of the hashed key index. The "midpoint" of the graph seems to be around 1200, averging 3.6 collisions. So the for storing 1000 words, the best prime number size that I should use is around 1350.

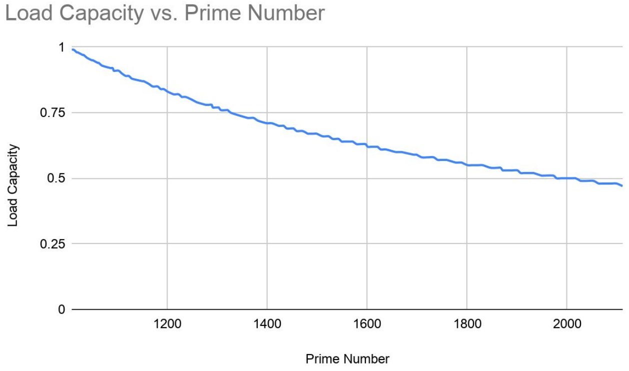


Using the same method to find the average collisions above. The average collision for misses are much higher than the hit counterparts. It starts at around 340 when the hash table size is 1009. At 1201, the average collision is 22.34.



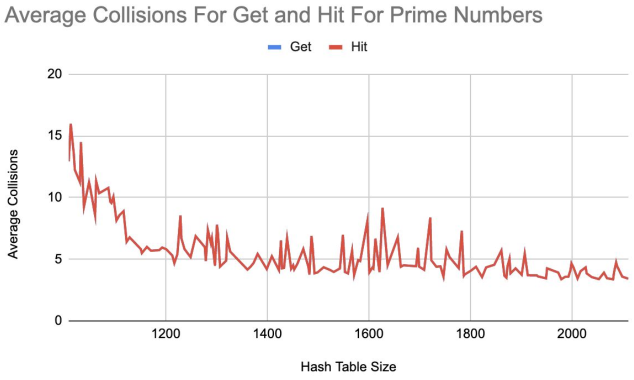
Analysis:

The load capacity is the ratio between the number of items in the hash table and the size of the hash table. Since the earlier graphs showed that the most optimal size should be around 1300, we can see from this graph that the ratio is around 0.75. This ratio is the most optimal for this hash function so to generalize it, I have to use other hash functions. So for the concordance problem in this assignment, I found the number of words in each text file, then divided them by 0.75 and found the nearest prime number as that is technically the most optimal use of memory.



Notes:

The first hash function I used had an extremely unstable result. This is because I multiplied the ACII total and divided by the true size of the hash table at the end instead of doing it every step of adding the ACII value of



characters