Nahom Ogbazghi
Apriori PJ 1
CS 378

My implementation of the Apriori algorithm does not vary much from the pseudo code provided from the internet. The items are stored in a hash value and the data is placed into a sorted list of list for storage. The only variation I added to the algorithm was the for each item's value in the data structure, there was a list to hold the index locations of the value. I found this efficient because rather than going through the data file repeatedly to see if two items were in the same set, I could simply find the intersection of the lists. I would compare the size of the intersection against the minimum support to see if it was frequent.

Another additional action is when I put the candidate item keys into a list to iterate through to generate a new frequent items list, I would check the count of the item against the minimum support then remove it from the list if it is less than. This was helpful because it reduces the number of candidates to sort through.

This assignment allowed me to think outside of the box to find more efficient measures to gain an advantage. I took into consideration programming languages and data structures to use to improve my timing. This is something I would not have considered to do in other classes.

These are my results below:

K	Frequent items count	Time(seconds)
2000	155	14
1500	237	25
1000	385	46
500	1073	111