**CSCI 2270 Stage 3**

The Problem:

When writing research papers, many times students need to go through large pieces of text and obtain all the necessary information. But, these text files may have a lot of unimportant information therefore it would be nice to have a way to summarize these pieces of text. There are online text summarizers but they only allow a limited number of words/characters; so in some cases students need to separate their document into chunks which may not give an accurate summary.

The Design:

As a part of the project design we will be implementing a hash table, stack, and max heap in order to solve the problem aforementioned. In order to gather data and summarize a text file, the frequency and value (or weight) of each word must be taken into account using *term frequency - inverse document frequency* (TF-IDF). To do so, we implement a structure that contains the word as a *string* type variable and the “count” - the number of times it appears within the text file - as an *int* type variable. This structure type of data is then stored into a hash table by way of a hash function that will give us an index for each word - making it easier to access and use. While doing this, we also grab and store each sentence into a stack data structure for temporary storage. Once each sentence is collected and stored into the stack, we use the TF-IDF function in order to obtain the weights of each word based upon frequency in the text and rarity in reference documents. Thereby, each sentence will receive a “score” that will be the sum of these weights/values of each word using the TF-IDF values. We then use a max heap to sort each sentence by their “score” which will allow us to extract the “highest scored” sentences; which, hence, will hypothetically summarize the whole text document for the user. Finally, we will use the hash table and create a list of the most frequent words used within the document for use of the user.