Proposal

Academic Reference:

https://epubs.siam.org/doi/epdf/10.1137/0206024

Algorithm Summary:

The KMP Algorithm finds a target pattern within data in O(n+m) where n is the length of data and m is the length of the target pattern. This is done by creating an array (LPS) that represents the longest proper prefix that matches as a suffix (ie. repeated at the beginning and end of the substring). When a mismatch occurs, we are able to identify where the next matching should begin based on the LPS array and the matching pattern we already identified. Therefore, each comparison gives information about where the next match could occur, which skips already matched substrings. This gives a linear time complexity of O(n) to search the dataset. Creating the LPS is also linear time complexity with O(m), therefore the total time complexity is O(n+m).

Function I/O:

The expected inputs for the algorithm are a dataset to be searched and a pattern to search for. In this project, we will write functions to handle strings. In testing, these two inputs would be two strings stored in a .txt file or pass directly to the function.

The expected output is a list of indexes that pattern occur in the dataset, which will be stored in another .txt file.

We will implement this algorithm by writing a class KMPsearch. It has four core functions:

1. KMPsearch, the constructor of the class, initializes all of the private variables.
    a. Takes in file name where first line is the pattern and the rest is data
    b. Takes in pattern and data as strings
2. convertData, this function takes in the file name and returns the converted data as a vector for later use. In the data, the first line of the txt file is the pattern to search for, and the rest is the dataset to be searched.
3. makeLPS, this function takes in the vector of the pattern and returns a vector of LPS. LPS stands for the longest proper prefix that matches a suffix. This is used during the search.
4. search, this function returns a vector of indexes where the pattern occurs in the dataset.

There are other trivial functions that set or get the private variables.

Data Description:

There are three txt files of data. The first one is a short example of searching a four letter pattern in a 11 letter dataset. The second one is a slightly longer example of searching for a word in a sentence. The third one is searching for a word in a long paragraph from the website.

For each .txt file, the first line is the pattern, and the rest is the dataset.