Project Part 1: Applying the Pillars of Computational Thinking

Purpose: Counting the number of occurrences of a word and its synonyms in a corpus of text documents.
Assignment location: https://www.coursera.org/learn/computational-thinking-problem-solving/peer/dRgUj/project-part-1-applying-the-pillars-of-computational-thinking

Title: Word Repetition in Text Documents

Assignment questions:

1. Using decomposition, what are the primary sub-problems that need to be solved in solving the overall problem?

The two primary sub-problems consist of matching the keyword to the corpus and finding synonyms in the thesaurus and matching them to the corpus.

- Finding matches of a keyword to determine how many times it is used in the corpus.
- Finding matches of synonyms to the keyword based on a thesaurus and determine how many times they are used in the corpus.

2. Using pattern recognition, what patterns do you see in the solution, i.e., what processes need to be repeated?

Pattern recognition approach for matching the keyword for frequency based on documents located in the corpus. The same process could continue with synonyms found in the thesaurus.

- Determine the keyword to match.
- Setup the corpus and associated documents for searching for the keyword.
- Conduct comparison of the keyword with words in the corpus documents.
- Display a count of frequency when the keyword is found in the corpus

documents.
- Pattern should be replicated with synonyms as found in the thesaurus.

3. Using data abstraction and representation, how would you represent the thesaurus, the corpus, and each of the documents in the corpus?

Data includes the keyword to be matched, synonyms found in the thesaurus, corpus documents, and frequency counts of the keyword and associated synonyms. Data not needed includes rationale for picking the keyword and anything outside of the basic data elements identified here.

- keyword, thesaurusSynonyms, corpusDocument, frequencyCount (thinking in terms of having a database, there may need to be a documentID to keep track of different documents, would be easier to have a flat file like a json).

4. Using the results of the first three pillars, what is the algorithm that you would use to solve this problem? Describe it in as much detail as possible.

This algorithm will focus on matching the keyword to the text of the documents located in the corpus. Further iterations can be done with determining synonyms and matching them to the text of the documents located in the corpus. A frequency count of the keyword and synonyms found in the text would be provided.

1. Determine the keyword for matching in the documents.
2. Gather the documents of the corpus.
3. Are the document data in text that is searchable (i.e., simple text for an array/object or standard notation like a JSON)?
    a. Yes, proceed to #4.
    b. No, turn documents into standard data format.
4. Create a python script for using logical operations to compare the keyword to the document text.
5. Determine frequency count and display to the user.
6. Repeat for each keyword that is a synonym after matching original keyword to a thesaurus.

5. Describe a problem that you may face -- either in your career or in everyday life -- that involves determining the number of occurrences of a word and its synonyms in a corpus of documents. The problem you face may be much bigger than that and require that calculation as only a small part of the solution, but should involve looking through some collection of text and looking for certain words.

This seems like a standard computer script where you have one value and want to match against a collection to look at trends in the text. I do something similar with open source engagement metrics at work. I often want to know who is engaging with our code repositories and what are the most common thoughts about the code. This would be a great way to get into the details of text associated with engagement.