COMP3009J Information Retrieval

Worksheet 2

Download the file named "index.txt" from Brightspace. This file contains postings lists for a document collection. In this worksheet, you will write functions to merge postings lists using common operations: AND, OR, NOT. First you must read the lists from the file into an appropriate data structure. You should write all of your code in a single .py file. Do not use any 'import' statements.

File format:

The "index.txt" file contains a postings list for one term on each line. Each line has the following format:

- All tokens in the file are separated by a single space.
- The first token on each line is the term.
- All other tokens are document IDs that make up the postings list for this term. These are already in the correct order, according to their integer values.

Q1. Write a program that reads the contents of "index.txt" line by line into a suitable data structure that you can use for processing later.

Hint: In your data structure, you will need to be able to find a postings list quickly, based on the term that you are searching for.
Hint: It is important to keep the ordering of the postings lists correct, so that they can be merged efficiently.
Hint: The document IDs must be stored as int values. If not, they will not be compared correctly.

Q2. Write a function with the following definition:

```
def mergeAND(list1, list2):
```

This function should merge two postings lists (which should be Python lists) and return another list that contains every document ID that is in both list1 AND list 2. Pseudocode for this was provided in the lecture.

Make sure to test your function by using two postings lists from your dictionary and checking that the outcome is correct.

Q3. Write a function with the following definition:

```
def mergeOR(list1, list2):
```

This function is similar to mergeAND(), except that the returned list contains all document IDs that are either in list1 OR list2. This should not contain any duplicate values.

Remember to test that this function works correctly.

Q4. Write a program that allows a user to search for one term, or for two terms using the AND or OR operators. The program should use the appropriate function (mergeAND() or mergeOR()) depending on the user input, and print the merged list of document IDs.

Hint: You can use Python's input() function for getting user input. Read more here: https://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/io.html

Advanced

If you finish the above exercises early, here are some more you can try.

Q5. Write a `mergeNOT(list1, list2)` function that returns a list of document IDs that appear in list1 but do not appear in list2.

Q6. Adapt the program to allow users to enter more complex queries, such as:

(data OR information) AND (radiation OR energy) AND (environment OR area)

Q4. Copy “q3.py” to a file named “q4.py” and modify it so that it counts the number of times each word appears in the text. For each word, print the word and the number of times it appears.

Sample output (the order of your output might be different):

```
--------------------
2048: is
128: answer
4096: this
...
--------------------
```

Hint: You will need to choose an appropriate data structure for this task.

Q5. Copy “q4.py” to a file named “q5.py” and modify it so that it prints the words (and their frequencies) in order of frequency. The most common word should be first, followed by the others in descending order.