Homework Submission Guidelines

1. **Due date: 05.12.2024**
2. The assignment can be done in pairs
3. Answers can be submitted either in English or Hebrew
4. HW submission should be done via moodle in the corresponding area (by **only** one of the students)
5. Late submission penalty (**5% a day**) for submitting after the assignment's due date
6. Questions / clarifications and more in the dedicated discussion sub-forum in Piazza

Dry Part – Positional Index (20%)

1. Consider the following documents:

   Doc 1: I am a scientist, and I am currently enrolled to IR course. I am a student at Reichman University.
   Doc 2: I was a student and scientist.

   Let's say we want to find documents in which **"I"** and **"scientist"** are at most 2 words apart, but in the same sentence.
   How would you modify the positional index to support queries that demand the terms to be in the same sentence? You can assume that there is a pre-parsing step that identifies sentences in documents. Describe the structure of the index you would construct. Write the modified postings list for the words "I" and "scientist".(10%)

2. Malware indexing and retrieval: malware is a malicious software program with several characteristics. Malware has a **type** (virus, worm, ransomware..), **author** and a short **description**. A malware search engine retrieves malware information in response to a malicious activity (query). Describe the structure of the index you would construct for malware search given a collection of malware objects. (10%)

Note – Use a pdf format for submission and name the file: **Dry_part.pdf**

Wet Part – Inverted Index (80%)

In this programming assignment, you will construct an inverted index for the AP collection and use Boolean queries to retrieve documents.

The files for the assignment are located in model under Assignment_1/
Inside the folder you will find the following files and directories:

a. **"data" folder containing AP_Coll_Parsed_1 -- AP_Coll_Parsed_9**
A directory containing 9 .Zip files of **242,918** documents from the AP dataset in a **trectext** format. Each file contains several documents, separated by <DOC> tags. Each document has a unique document ID, specified by the <DOCNO> tag, which comes right after the opening <DOC> tag. **The text of the document to be indexed is contained within <TEXT> tags**. **(Several documents contain several <TEXT> tags.)**

Note1: The text was lowercased, and the following punctuation marks were removed:
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

Note2: No other pre-processing steps are required.

Here is an example document:

```
<DOC>
<DOCNO> AP900101-0002 </DOCNO>
<FILEID>AP-NR-01-01-90 0005EDT</FILEID>
<FIRST>r w PM-SocialSecurity-Glance     01-01 0304</FIRST>
<SECOND>PM-Social Security-Glance,290</SECOND>
<HEAD>New Year Brings Social Security Changes</HEAD>
<HEAD>With PM-Social Security Bjt</HEAD>
<DATELINE>WASHINGTON (AP) </DATELINE>
<TEXT>
here are some changes in social security
benefits and taxes that take effect with the new year
benefits monthly benefit checks increase 47 percent to offset
the effects of inflation the average retired workers social
security check will rise from 541 to 566
…
</TEXT>
</DOC>
```

b. **BooleanQueries.txt** – a query file with 5 Boolean queries

Boolean Query Structure

A Boolean Query is composed of terms and the following logical operators:
AND, OR, NOT.

We Consider an example Boolean Query:
southwest airlines OR africa NOT

Queries are represented using the Reverse Polish Notation.
See the link for more details: https://www.programcreek.com/2012/12/leetcode-evaluate-reverse-polish-notation/

For this query, we want to retrieve all the documents containing the term "**southwest**" or "**airlines**" but not the term "**africa**".
**The operator "NOT" is treated as "AND NOT"

Part 1 (30%) – Inverted Index

Your first task is to write a function/class that creates an inverted index for the AP collection. As you have seen in the lectures, the inverted index is a data structure that supports efficient access to documents allowing the lookup of all the documents that contain a specific term. Your program will take the AP corpus as input and produce an inverted index. Before implementing the function, please read the following notes carefully.

- During index construction, specifically, for building the postings lists you should use successive integers as document internal identifiers (IDs) for optimizing query processing, as taught in class, but you still need to be able to get the original document ID when required.
- Name your function/class "**InvertedIndex**".
- Document your code.

An example of an inverted index

'the' -> 1 (AP880219-0002) -> 2 (AP880314-0254)

'sanctions' -> 2 (AP880314-0254) -> 4 (AP880221-0077)

'african' -> 3 (AP880222-0029)

Part 2 (30%) – Boolean Retrieval Model

Your second task is to write a function that retrieves a set of matching documents given an inverted index and a Boolean query.
Before implementing the function, please read the following notes carefully.

- Keep the information of the retrieved documents as follows:
  - Each line (5 lines total = 5 queries) contains the original IDs (not internal IDs) of the retrieved documents separated by space.
  - Keep the same line order as in the "BooleanQueries.txt" file.
  - Name the result file "**Part_2.txt**".
  - Name your function/class "**BooleanRetrieval**".
  - Document your code.
  - See an example format below.
- **Important!** In your solution you **must** use the fact that the postings lists are sorted by their internal IDs. **For example – using the "set" data structures for intersection is not allowed.**

Part_2.txt

```
AP880219-0002 AP880314-0254 AP880404-0200 ….
AP880503-0228…
AP880221-0077 AP880222-0029…
.
.
.
```

Part 3 (20%) – Collection Statistics

Your third task is to write the following statistics to a file "**Part_3.txt**".

1. Write the top 10 terms with the highest document frequency (10%).
2. Write the top 10 terms with the lowest document frequency (10%).
3. Explain the different characteristics of the above two sets of terms (3%).

Submission Instructions:

1. Zip all files together (Dry + Web parts) and name your submission as follows:
HW1_Student_1_ID_Student_2_ID.zip

2. The Zip file should contain the following files:
    1. Dry_part.pdf
    2. "Wet_part" directory
        a. invertedIndex (code)
        b. booleanRetrieval (code)
        c. Part_2.txt
        d. Part_3.txt