IR Assignment 1

Homework Submission Guidelines

1. **Due date: 10.5.22 at 23:59**
2. Homework must be done in your assigned groups
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 (**20% a day**) for submitting after the assignment's due date
6. Questions / clarifications and more in the dedicated discussion sub-forum.

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

The files for the assignment are located in home/data/HW1/
Inside the folder you will find the following files and directories:

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

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 either the term "**southwest**" or "**Airlines**" and do not contain the term "**Africa**".

Note that in this assignment, the NOT operator is a binary operator that has the semantics of set difference (all documents in set A that do not appear in set B). For example, the query "*Airlines Africa NOT"* will retrieve all the documents that contain the world "airlines" and do not contain the word "Africa".

Part 1 (33%) – Inverted Index

Your first task is to write a function/class that creates an inverted index for the AP collection. Your program will take the AP corpus as input and produce an index of all the words.
Before implementing the function, please read the following notes carefully.

- During index construction, specifically, for building the posting 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.

```
'the' -> 1 (AP880219-0002) -> 2 (AP880314-0254)
'sanctions' -> 2 (AP880314-0254) -> 4 (AP880221-0077)
'african' -> 3 (AP880222-0029)
```

Part 2 (34%) – Boolean Retrieval Model

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

- **Important!** Write the retrieval results to a file "Part_2.txt" as follows.

Each line (5 lines total) 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 your function/class “**BooleanRetrieval**”.
- Document your code.
- **Important!** In your solution you **must** use the fact that the posting lists are ordered by their internal IDs. For example – using the “set” data structures for intersection is not allowed.

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

Part 3 (33%) – 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 to a file named "Part_3a.txt" (10%).
2. Write the top 10 terms with the lowest document frequency to a file named "Part_3b.txt" (10%).
3. Explain the different characteristics of the above two sets of terms. Write your answer in a file named "Part_3c.txt" (3%).

The format for collection statistic files ("Part_3a.txt" and "Part_3b.txt") is the following:

```
<term1>: < document frequency of term1>
<term2>: < document frequency of term2>
 ……
<term10>: <document frequency of term 10>
```

Each line describes a single term, sorted from the term with the highest document frequency to the term with the lowest document frequency.

Submission Instructions:

1. Zip all files together and submit a file with name of the following format: **HW1_Student1ID_Student2ID.zip** (**20% grade penalty otherwise**).

2. The Zip file must contain the following files: **Part_2.txt**, **Part_3a.txt**, **Part_3b.txt**, **Part_3c.txt** and **all the code files you used.** The code should be fully documented and clear, and has to be able to reproduce your results.