**REPORT ON BM25 IMPLEMENTATION**

To implement the bm25 algorithm, I have initially read the stemmed file “tccorpus.txt” line by line. By splitting each line on spaces, I have obtained the document number that is the next string after “#”. The rest of the tokens have been stored in a list. I have then iterated through this list to check the valid tokens, which must not be purely integers. The valid tokens have been stored in a hashmap that has document number as the key and then another hashmap as its mapping. The inner hashmap stores the token string and the number of times the token appears in the document. Concurrently I have created a hashmap storing the document number and the, number of valid tokens in that document. A count of total tokens has also been kept.

I have further read the query file in the same manner, line by line and created a similar hashmap with a query id and another hashmap as its mapping. The inner hashmap stores the query word and the number of times the toke occurs in the query.

These mappings have then be used to further calculate the bm25 scores for each query.

The variables required for bm25 score have been calculated for every document separately and the formula applied on the values. Document scores have been incremented as and when another query word is found in that document and the final results are then passed to sort.java

A separate java file-sort.java has been created to which a map storing the query id and the bm25 scores for all queries has been passed. This java file sorts the document scores by values by overriding the comparator method. Out of these results I have iteratively printed the top 100 ranked documents for all the separate queries in queries.txt