Inverted Index Algorithm

Leonardo Moya Rosales 152037
Nelson Manuel González Pichardo 151685
Arturo Emmanuel Ortíz González 151564

For this algorithm we used Python since it's scripting language and already has built in functions as to split, lines, open files, etc. easily, also we used an iPython Jupyter Notebook since it makes the whole ordeal easer to debug and when running it you can see the functions alongside their results.

First we created a structure that contains the word and the files its contained in, this is in order to handle the index as a list of these structures, with a nested list within. For adding the words we used sets as we knew that there couldn't be any duplicates. Once the words are done being added we have our inverted index.

The next big issue was how to split the queries, so what we did was create a search function and an inverted search function that handled one word queries that is a query that yields the list of documents it is contained in or the opposite.

Next, we implemented a function to handle intersections, for we will use it later on to intersect between lists consisting of a mix of both earlier functions.

The final step was to convert a human readable search into a query our system could understand, for this we created a syntax where we can decide to search for something based on the operators "&" and "?" meaning "and" and "not" respectively, we then evaluated each of these terms individually and placed the results in a pool which we passed to a recursive function that intersects all of these results until no more sets than one exists.