Ethan Raymond and Jay Harris Project Phase 1
20363147, 20362789

Group member contribution: 50% each

Note: We are only printing out the top five most frequent words for each document, as specified at the top of the specification (it's unclear in the part 1 deliverable).

Note: We don't print out parent links (though we can, and have at previous times) as the part 1 requirements don't mention them.

We have two inverted indexes to store the words extracted from the page body and the words extracted from the page title. Both inverted indexes are encapsulated within the Index class.

The Index class also stores wordIndex, linkIndex, docIdIndex, and wordCountIndex.

InvertedIndex maps a wordID to a posting list. Postings contain the document ID, term frequency, and the positions at which the word occurs. This information will be used when scoring document relevancy.

WordIndex maps words to their ID. We store the word IDs in the posting list to conserve space and make faster comparisons. This consists of two maps, one from word to word id, and one from word id to word.

LinkIndex maps links to their ID and stores parent and child relationships. It consists of four hashmaps; A link to document id map, a document id to link map, a map from document id to child documents and a map from document id to parent documents. We store map documents to integers to conserve space and we store the parent/child relationships so that we can easily reconstruct the web graph if needs be.

DocIDIndex is a map from document ID to web page metadata. We store the data in the WebPage class, which contains docID, url, latest modification date, size, and title. We use the information to determine when a page needs to be updated in the database.

WordCountIndex maps a document id to a map of words and their occurrences. This makes it faster for us to print out the top five words for each document during the printing stage, as reconstructing word counts from an inverted index is a somewhat slow process.