Asst2 - Reverse File Indexer
Colin Ackerley & Warren Ho

Design

Our design relies on linked lists of different structures to keep track of all the data we need.

We start off by creating a data structure a Record. A record node has 3 fields: A token, a count for that token, and a pointer to the next record.

We then create a filestruct linked list. The file struct has 3 fields: A filename, a list of records for that filename, and then a pointer to the next filestruct. This is our intermediate data structure. It has all of the information we need, but not in the form that we need it in.

After that, we create a listFileName. This has 3 fields. A filename, a single record instead of a list, and a pointer to the next listFileName node. This structure is similar to a filestruct, but using it made our code much simpler and easy to read and debug.

We also used an index struct. It has fields for: a token, a pointer to a listFileName node, a pointer to the next index. This is used by the indexer to add all of our data to a file. This gives us all of our information in sorted order.

Time & Space Analysis

The downside to our use of linked lists is their inefficiency. Any time we use a linked list, we have to go through it, which takes O(n) time. When we sort our lists, it takes O(n^2) time. Our overall program should run in O(n^2) time worst case since that is our slowest algorithm. The most time consuming part of this project is from the sorting we needed to do at various stages. However, our project is still quick to run. The only slowdown is when we try to open a huge file, and that slowdown is from the open function, not our code. The slowdown from a large file happens before any of our slower code begins to run. Therefor, I believe it is from the open function and not our code. We free all of our data as soon as we are done with it, so the space usage is not inefficient. In addition, we use pointers for all of our data, so there are not multiple copies being used and wasting space. Our program will use more space as it reads larger files, but that is to be expected.