

Faculty of Engineering

Department of Management Sciences

MSCI 541/720 –HW1

Name: Rui Zhang

Student ID: 20599896

Date: January 21st, 2018

Problems (1-3)

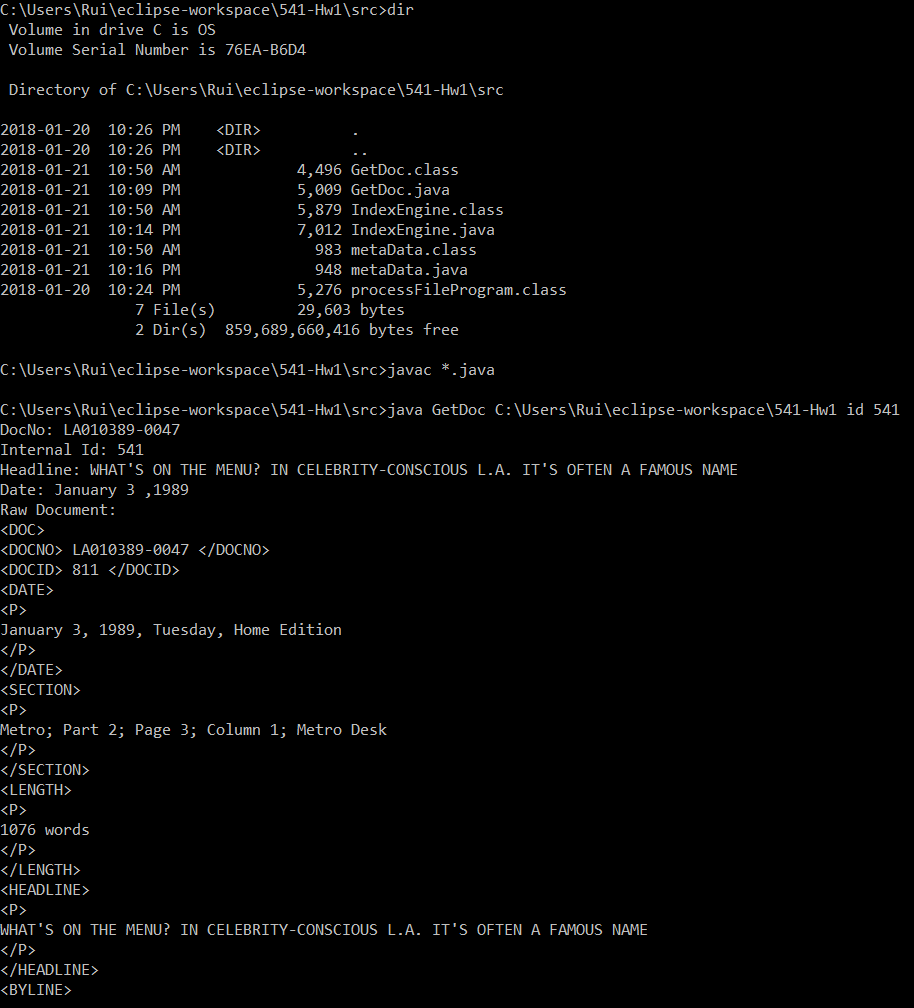
1. A) What is meant by precision enhancing and recall enhancing in context of search engine
   1. To begin, precision in terms of search engine means to retrieve the fraction of items that are relevant to the item being searched. Recall on the other hand is the fraction of items in the collection that were actually being retrieved. For example, if 2 items were retrieved and only one was relevant, then the precision would only be 0.5. If 3 relevant items were in the collection and only one was retrieved, then the recall would be 1/3. Therefore, precision enhancing means to try and maximize the number of relevant items that are already retrieved. Recall enhancing means to only retrieve the relevant items available in the collection.
   2. In the case of an alphabetical, subject card catalog, if users were to search up the catalog of Food and recipes, a precision enhancing technique would be to increase the number of items retrieved such that every item had the word “Food” or other words that go under the category of food and recipes but negates the items that use the words for other purposes. A recall enhancing technique would be to load up every single item that contained the words food and recipes.
2. 1. Databases and search engines are similar due to their capability in their own way of organizing data. Databases for example would use tables and columns whereas search engines would index the data. However, when a user interacts with a database, the database would always output the exact answer of what the user is looking for. As a result, the user must also know exactly what they’re looking for. As for search engines, the users may or may not know what they’re looking and therefore the engine would output the relevant information that the users may or may not need. To dive deeper into the technical side, relational databases have tables that use foreign keys to connect between on another if the tables have a connection. In search engines, the information among them are not linked in anyway and it’s the search engine’s responsibility to determine whether there is a relationship or not.
3. A disadvantage of down casing all terms in a search engine is that the words that could be referred to as a company or a name is down cased such that they could just mean ordinary English names. An example would be the company Apple, if a user was to search up Apple without the down casing, functionality, he or she will get the result that was wanted about the company. However, if the down casing functionality was activated, then the user would only be prompted with the fruit apple. An advantage of down casing items would be vice versa of what was described above. For example, if users were searching up the word Lamps, the results would not return items related to “lamps” because “Lamps” and “lamps” are different from the capital “L” at the front. Therefore, down casing can result in the correct output of “lamps” if a user searching “Lamps”.

Problem 4:

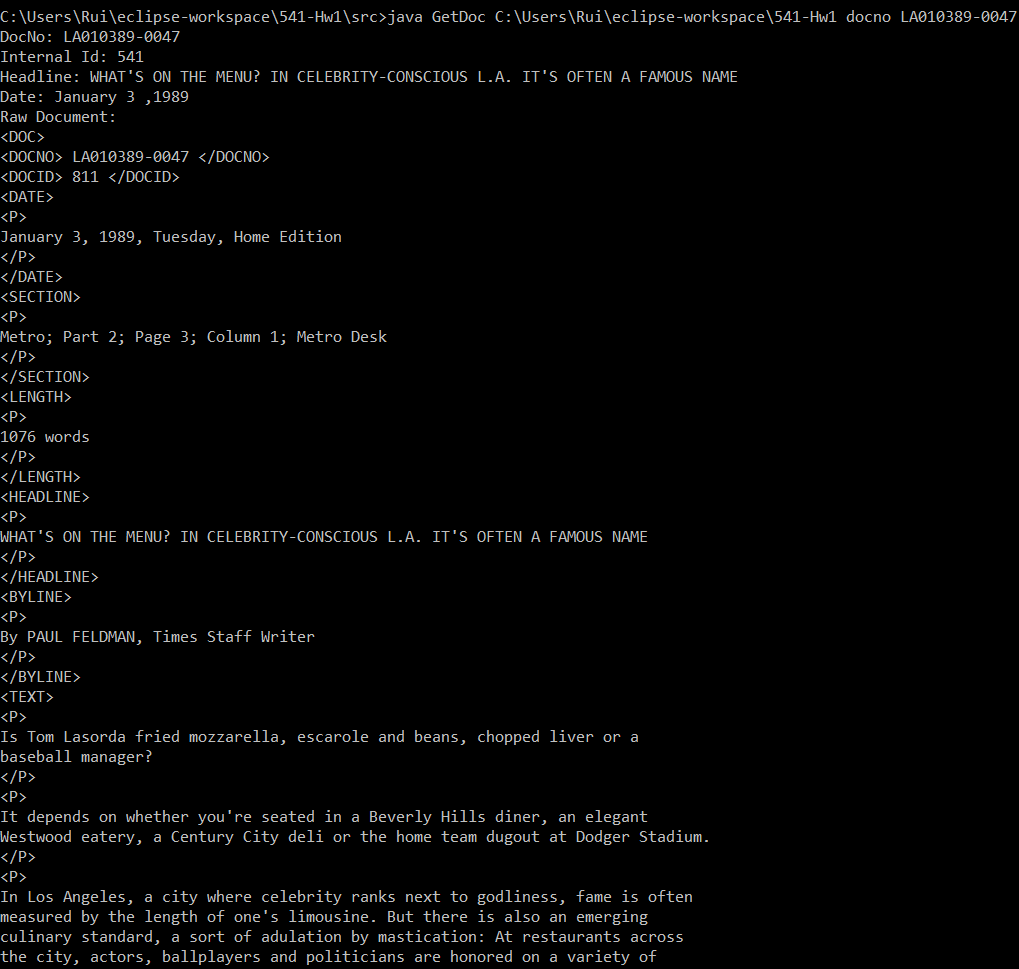
Test plan:

|  |  |  |
| --- | --- | --- |
| Test Case # | Description | Compliance |
| 1 | Using command line to activate the GetDoc class and search using internal id: 541 | The meta data and raw document appears for that specific internal id |
| 2 | Using command line to activate the GetDoc class and search using  **DOCNO: LA010389-0047** | **The raw document with the DOCID of 811 shows up** along with the corresponding raw text |
| 3 | While activating the Index engine class, the user did not input any arguments | System should stop and help the user to what should be put in |
| 4 | While activating the GetDoc engine class, the user did not input any arguments | System should stop and help the user to what should be put in |
| 5 | User inputs an invalid Docno in GetDoc | System should exit and throw a string out of bounds error |
| 6 | User inputs an invalid Docno but with the same length as a valid Docno | System should exit and throw a file not found error |
| 7 | User inputs an invalid internal id | System should exit and throw a number format error |
| 8 | **User activates the Index Engine but the directory already exists** | System should exit and tell the user that the directory already exists |
| 9 | User activating the Index Engine without the an existing final directory | Program should start reading the gzip file, process the data, and make files |
| 10 | User inputs wrong url address arguments for the paths for Index Engine | Program should stop and throw a file not found error |

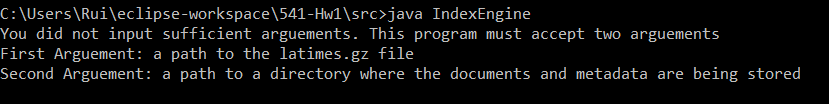
Test Case 1.



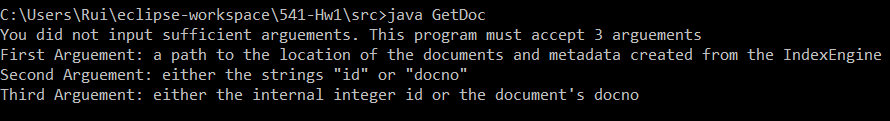
Test case 2:



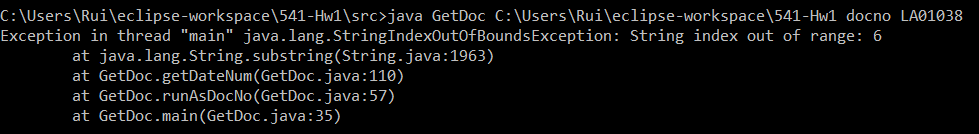
Test case 3:



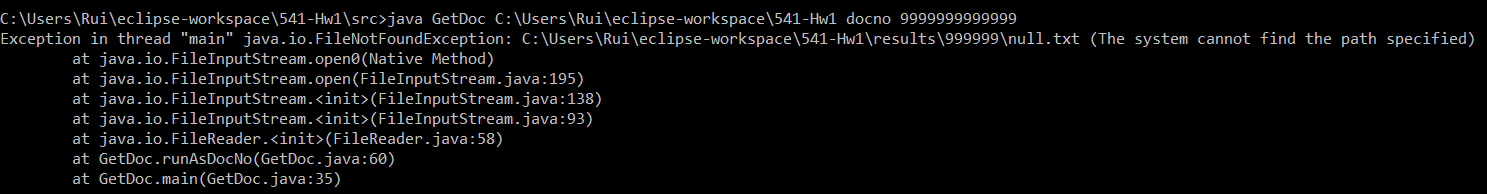
Test case 4:



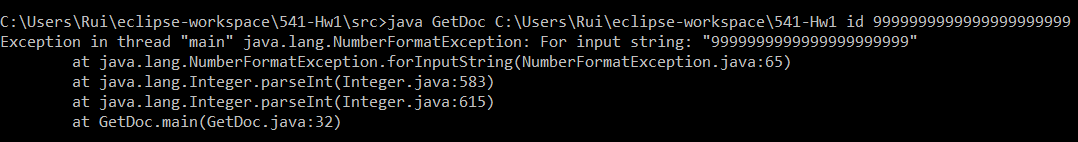
Test case 5:



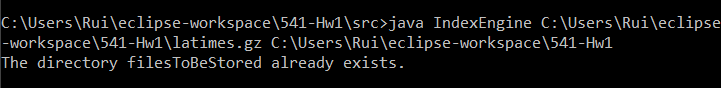
Test case 6:



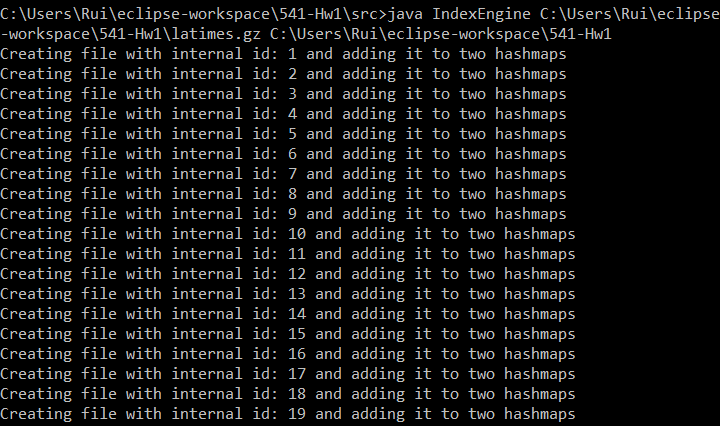
Test case 7:



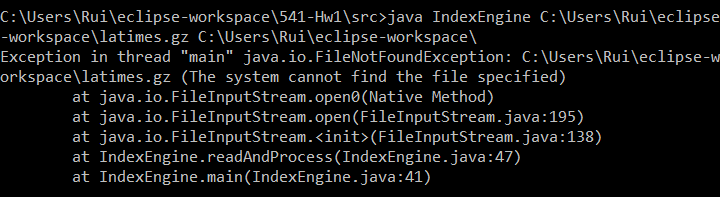
Test case 8:



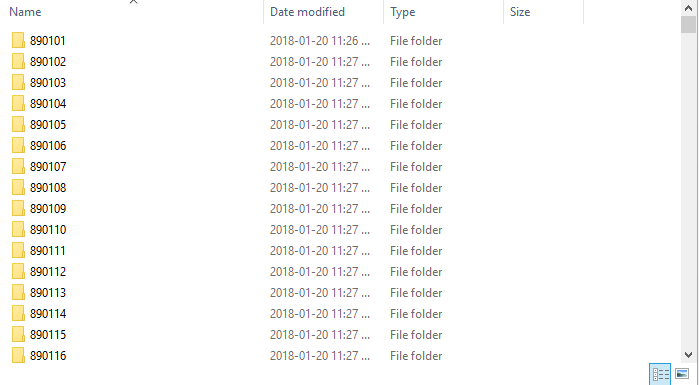
Test case 9:

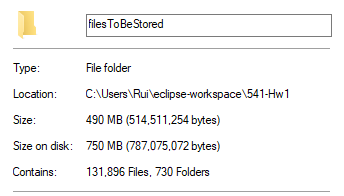


Test case 10:

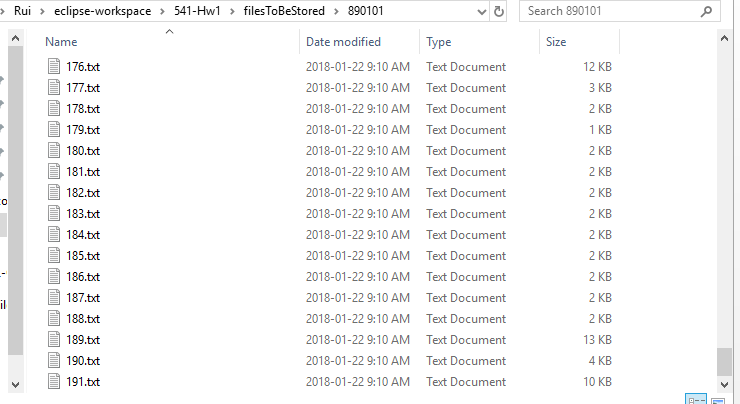


Proof of Directories and Files

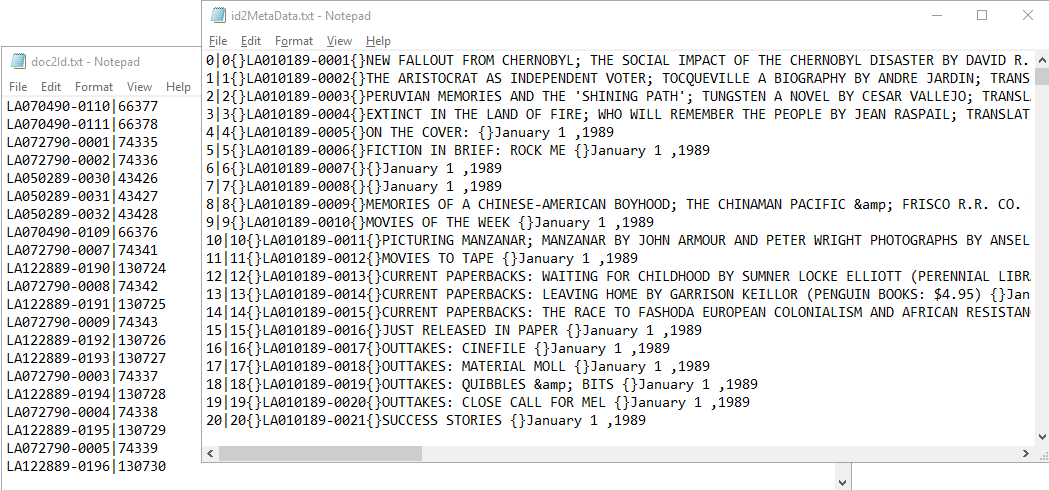




First directory dated Janauary 1st 1989 has 192 files named after their internal id starting with “0”.



Proof that the two hashmaps are saved to file:



An example of the 4th document with internal id as 3

