Run the Code Example

Instructions

1. Download one of the 3 jar files in the following drive folder: https://drive.google.com/drive/folders/1UjWqVZGhx3iKatM2-cwR1G9J1ZF28PuD?usp=sharing. In the following, EX stands for the files associated with the EX.jar, where X is 4, 5 or 6 (E4 is with the smallest input size).
2. Extract one of the EX.jar files on your machine (this will create a directory with the name 'EX'):
   `jar xf EX.jar`
3. Enter the directory created:
   `cd EX`
4. Compile the source code:
   `javac dynamic_index/*.java`
5. Run:
   `java -Xmx1G -Xms1G dynamic_index/Main`

More About Running the Code

1. Standard output: The program will run and print a lot to the standard output. These are less important details about the run of the experiment such as: what words are queried and what reviews get deleted in each insertion and so on (In short: you may ignore).
2. Log file: On the other hand, the program will create a LOG.txt file. This file main purpose is to hold all the results of the experiments as raw data (space separated numerical values to be easily imported to say, Excel). The graphs in the project's written component used the data in the LOG.txt file.
3. Disk "Requirement": The program assumes there is at least 3GB of disk space to run the largest input (E6).
4. About the jar files: The E4.jar is also submitted to the Moodle. Each jar file has almost the same source code ('dynamic_index') and differs mainly by its input size in total number of reviews: E4 has 10,000 reviews, E5 has 100,000 reviews and E6 has a million reviews. The instructions are the same for each jar file.
    a. The differences in source code is only 2 lines in main which don't run the million reviews input (E6) on all the temporary index sizes. These lines have been commented out.
5. Watching the Indexes Files: The directory in which all indexes are created is deleted in the end of the run (I assumed this because of the exercises and because it is not so much the theme of the project). If needed, this could be disabled. There are two options depending on what level you want to examine the index files:
    a. Low level: Stopping mid-run and looking at the files, or while the program is running in the bigger input cases.
    b. High level: changing 5 lines of code before compiling (there probably is a better easier way but this works):
        i. For the simple experiment, go to: EX/dynamic_index/index_experiments/SimpleMergeExperiment.java, and comment out the line `removeIndex();` in the `runExperiment()` method.

ii. For the log experiments this can't be done right away because the same directory name is used for all the different temporary index sizes.
    1. Go to the `Main.java` file, comment out the `testTempSize(localDir);` line, and uncomment the next three lines. This will run only one iteration with the log experiment.
    2. Now, like the simple experiment we go to EX/dynamic_index/index_experiments/LogMergeExperiment.java, and comment out the line `removeIndex();` in the `runExperiment()` method.
    3. Now we compile and run in the same way as instructions 4 and 5.