General Overview

This application contains two phases. The user will first run phase1.py to construct a database which contains documents in a given file. Then, upon running phase2.py, they will be greeted by a menu with 5 options: search articles, search authors, list venues, add articles, and exit.

- Search article: user provides keywords to find articles matching those keywords. Then, the user can see more information for an article or exit to the menu.
- Search authors: user provides keywords to find authors matching those keywords. Then, the user can see more information for an author or exit to the menu.
- List venues: user provides a number of top venues to list. That number of venues will be found, sorted by most articles referencing articles from that venue.
- Add article: user provides fields for a new article, which is then added to the collection
- Exit: user exits the app.

Software Design

Phase 1: phase1.py: takes a given file and port, and uses os.system to run mongoimport. Connects to 291db, creating it if it doesn't exist. Drops the dblp collection if it exists, and repopulates it with the documents in the file provided.

Phase 2:

- phase2.py: contains functions for article search and author search. Contains the method to connect to the database from phase 1. Contains a method for the menu, which allows the user to query the database. See menu function calls below:
    - article_search_all: takes keywords from the user and then uses a regex to find articles matching those keywords. Displays the id, the title, the year and the venue fields. Users can select a found article to see more information (the abstract and the authors) or return to the menu.
    - author_search_all: takes keywords and finds authors similarly to articles. Users can select an author for more information(title, year and venue of all articles by that author) or return to the menu.
    - quit: exits the application.
- add_article.py: contains the functionality for adding articles. Is called by the menu in phase2.py.
    - add_article: Takes ID, authors, title, and year for a new article. Checks if ID is unique, and gets the number of authors and their names one by one. Adds a new article to the collection with the given fields and other fields empty.
- list_venues.py: contains the functionality for listing top venues. Is called by the menu in phase2.py.
    - list_venues: Takes a number n of venues to display. Finds n venues sorted by most cited venues descending, also displaying how many articles were in that venue.

Testing Strategy

We tested the various aspects of the application as we developed it, starting with Phase 1. The database was checked to ensure load() was adding documents to the collection correctly. The speed was also checked for different file sizes. The parts of Phase 2 were tested individually. Each function call from the menu was tested on its own in their own python modules. The overall menu functionality was also tested. In order to double check that things were working, we also checked over work that another member of the team did on our own machines just to make sure. The testing was done using the test data JSON files of various sizes provided on eclass. In testing part 3 of phase 2, listing venues, we assumed that venues with 0 references would not be listed.

Group work breakdown

Aagaman:

- 2 hours: implementing add_article for part 4 of phase 2
- 1 hour: testing app functionality
- 30 mins: writing up report and readme
-

Charlie:

- 8 hours: implementing list_venues for part 3 of Phase 2
- 1 hour: testing various app functions
- 20 mins: creating separate phase1/2 files, mongoimport command for phase 1
- 1 hour: Writing up report

Ilia:

- 10 hours: implementing both the search functions for phase 2
- 5 hours: implementing listing for phase 3
- 20 mins: writing up report
-