CSCI 2421 Final Project - Database
Tobby Lie
Student ID - 103385744
10/19/17

Problem Description:

- This program will utilize binary search trees as well as other data structures in order to implement a simple database system. The database will store it's information to a file. It will allow for addition and deletion of records, field modifications and it will allow users to sort records based on selected keys. The program will also allow the user to output reports according to their desired criteria. Records that are held in the data base have multiple fields, in order to encapsulate all of this data, classes will be made to keep the fields tied together with each record.

Overall Software Architecture:
(At bottom of document)

Input Requirements:

Input files:
actor-actress.csv
pictures.csv

- actor-actress.csv will have information about actors and actresses. This is a comma separated values file. The information for each record in this file will be:
  - year: int
  - award: string
  - winner: bool
    - Although this variable will be represented as a boolean, the file will provide input for true or false, represented by a 1 for true and 0 for false which can be held by an int variable.
  - name: string
  - film: string

- pictures.csv will have information about movies. This is a comma separated values file. The information for each record in this file will be:
  - name: string
  - year: int
  - nominations: string
  - rating: float

- duration: float
- genre2: string
- genre2: string
- release: string
- metacritic: string
- synopsis: string

User input: keyboard

- User input will mainly be stored in strings and chars. Since the program will be menu based, the input from the user will mainly be choices for menu navigation and strings when they wish to type information about records in the database in. For example the user may want to delete a record by a name in which they will need to provide a string input in order to identify which record they wish to delete. Another example may be that the user may want to modify a field of a record. In this case they may also need to input a string in order to provide their desired change.

Output Requirements:

Report:

- A report will allow a user to organize output in a way that they desire, meaning they can specify which fields are to be outputted and how this output should be sorted.

Report files:

- ActorReport.txt
- PictureReport.txt

Updated Actor-Actress/ Films:

- After data retrieval and modification, the information in either the actor-actress or film databases will need to be written to a csv formatted file in order to update the original files.

- Report should be output in ASCII text format as a table and tabs. Some sort of overloaded function may be used to specify which fields are to be outputted since an output may not include all fields and records.

Updated files:
- updatedFileActor.txt
- updatedFilePicture.txt

Problem Solution Discussion:

The actor records and picture records will both be read into separate binary search trees. The actor records will be implemented with a binary search tree solely while picture records will be read into both a binary search tree and vector and program implementation will be handled with a vector.

All functions will alter the data structures themselves, other than the sort for actors which will need to write out to an updated txt file. This is because a binary search tree cannot actually be sorted.

The program will be a text based menu implementation which means the user input will be used for either menu navigation or database record modification. There may at times, be the need for multiple levels of menus. For example if a user wanted to search a specific record, they would need to first specify their search key and then specify if they would like to do a complete or partial search.

The program will implement read from file functions in order to read the information from the databases into either a binary search tree or a vector, depending on the function used. This is important for each function as it will need to be done any time a function is called. For example if we want to call a sort function, we will use a vector rather than a binary search tree. This means that we would need to read the file into a vector instead of a binary search tree which is implemented slightly different. Once processing is complete, the newly updated database will be overwritten on an updated database txt file. Reading into a binary search tree is different than a vector because it will immediately sort records by name.

In order to add information to actors, information must be read into a binary search tree so that it can be ordered by name. A record can then be added and sorted appropriately. The newly updated database can now overwrite the database file. A picture record will need to be added to a vector by using push_back.

In order to delete information from a actors the binary search tree can be utilized to search for the desired record. Once found, that record can be removed. The updated database will then need to overwrite the database file. To delete information from pictures will need require a linear search to find the record to be deleted and then the erase function will be used to get rid of that record.

Sort functions will be implemented by using a vector, this is preferred over a binary search tree because it is simpler and less risky. The data base file will first need to be read into a vector. After that, the vector can perform a sort depending on sort key. The database file will then be overwritten with the newly sorted database.

For modification on actors, the findNode function will be used to find the node to be modified and then that record will then need to be reset. For modification on pictures, the node will need to be searched for in which the selected field can be set.

Outputting a report file is dependent on the user's needs. The user will specify which fields are to be outputted and how it will be sorted. After the user specifies, the function will read all records into a vector. The vector will then sort based on the user specification, and then output of the desired fields will be outputted based on the desired sort onto a report file. The user can also search for records which will be a separate type of report.

After any sort of data modification it is important that we output updated information and overwrite the data base so that it stays current.

Data Structures:

- Two data structures will be utilized for this assignment.
- The first being binary search trees, they prove to be useful as they have a good complexity for searches.
- The second data structure chosen is a vector, as this data structure will be able to perform things such as sorts that a binary search tree may be too complicated and cumbersome for. Since a vector also has random access via indexing, it is a good choice when a record modification is needed. This is because once the index of a record is found, we can then easily modify any fields we desire. Utilizing both of these data structures together is useful for operating on and implementing a database.

User Interface Scheme:

- User interface will be relatively simple. It will be based upon various levels of menus. Depending on what the user would like to do, the levels of menus will vary. For example adding a record to the database may only be one level of menus while search of a specific field in a specific record may result in three levels of menus. In any case, a menu will be available for the user at every point of program use. This will provide clear instruction on how to navigate the program.

Successfully compiled and tested on csegrid

Overall Software Architecture: