1. Database manager

Khang Nguyen

10/25/2018

1. Database manager is a software application that allows admin to access and manage data information. Data information is records that contains movie and actor and many fields associate with it. Admin can search, modify, delete existing record or add a new record into the system.
2. Overall Software Architecture:

Request Modified

DataBase

Csv file

Admin

Response

Added

Deleted

Response Request modify

Modify

Search

Request delete Search found

Search found

Request add

Search found

Delete

Add

1. Input:

Read actor-actress.csv and pictures.csv file into 2 BSTs. First BST (accTree) will contain actor-actress.csv data and second BST(picTree) will contain pictures.csv data. In Actress.csv

Field: Year(string) Award(string) Winner(bool) Name(string) Film(string)

In Pictures.csv

Field: Name(string) Year(string) Nomination(string) Rating(float) Duration(int) Genre1(string) Genre2(string) Release(string) Metacritic(int) synopsis(string)

1. Output Requirements

In Actress.csv

Modify existing record:

Field: Year(string) Award(string) Winner(bool) Name(string) Film(string).

Add new record:

Field: Year(string) Award(string) Winner(bool) Name(string) Film(string).

Delete existing record:

Just remove that existing record in file.

In Pictures.csv

Modify existing record:

Field: Name(string) Year(string) Nomination(string) Rating(float) Duration(int) Genre1(string) Genre2(string) Release(string) Metacritic(int) synopsis(string).

Add new record:

Field: Name(string) Year(string) Nomination(string) Rating(float) Duration(int) Genre1(string) Genre2(string) Release(string) Metacritic(int) synopsis(string).

Delete existing record:

Just remove that existing record in file.

6. Problem Solution Discussion

**R1:** You will read in actor-actress.csv which is formatted as a csv file (Ask the user for the file name). Then place the items in a Binary Search Tree, sorted by name.

Data will read into binary search tree. It is sorted by name. The algorithm is every new node add to BST need to compare to the parent node. If new node < parent node then inserts into the left of the tree, if new node > parent node then inserts into the right of the tree.

**R2:** Then you will read in information about the movies (called pictures by the Academy of Motion Pictures) that have won best picture award. Place these items in a Binary Search Tree, sorted by name.

Data will read into binary search tree. It is sorted by name. The algorithm is every new node add to BST need to compare to the parent node. If new node < parent node then inserts into the left of the tree, if new node > parent node then inserts into the right of the tree.

**R3/R4:** Choose either the movie or actor database and add a record

Search if new record already existing in database. If yes then response to admin as “Record already in database, you don’t need to add it in”. If no then response to admin as “Record is not yet in database, proceed to add it in”.

Add record into binary search tree and later add to database

**R5/R6:** Choose either the movie or actor database, search for a record, and modify the fields.

Search if record already existing in database. If yes then response to admin as “Record already in database, procced to modify”. If no then response to admin as “Record is not yet in database, do you want to add it in?”.

Modified record in database.

**R7/R8**: Choose either the movie or actor database, search for a record and delete the fields.

Search if record already existing in database. If yes then response to admin as “Record already in database, proceed to delete it”. If no then response to admin as “Record is not yet in database, you can’t delete it”

Remove that record in database.

**R9/R10:** Choose either the movie or actor database andsort by any single (sortable) field

I will copy every record from BST to vector. In order to copy from BST, I will visit every node from left node, parent node and right node. After getting every record from BST to vector, I will use selection sort to sort vector by any single field. Time complexity will be O(n^2). I could create many new BTS and sort it by many single field but heap could run out of memory and the program might choke since too many BTS in the memory.

**R11/R12:** Choose either the movie or the actor database and do a complete search on any “complete” searchable field. It is unlikely that you would have an exact match on an entire description, so that would not be listed to search.

Using binary search tree to do complete search. If search is true then output record, if search is false then output record not exist.

**R13/R14:** Choose either the movie or the actor database and do a partial search on any searchable field. A partial search is any substring within a field.

Using binary search tree to do partial search for name. If partial search is true then output records, if partial search is false then output record not exist.

**R15/R16:** Choose either the movie or actor database, ask for a file name, and print out a .csv file of the latest database (after adds, deletes or modifies). Remember that the first line of a .csv file lists the name of the fields separated with commas, ending in a newline. Then the following lines are the information from the fields separated with commas, ending in a newline.

Read data from the binary search tree and print out to the latest database.

7.Data structures

Data structure time complexity:

-Binary search tree: Search, Insert and Delete cost(O(log(n)), Already sort by comparison.

-Vector: Search (O(n)), Insert (O(1)), Delete(O(n)), Sort(O(n^2)).

-Array: Search(0(n)), Can’t insert because array is fixed size, Delete(O(n)), Sort(O(n^2)),

I choose vector over array to hold data because vector is dynamic size and I can insert new data into vector with only (O(1)). It means I can save a lot of memory by deleting vector after search.

8.User Interface Scheme

MenuDriver()

Welcome to main database manager

1.Go to manage actor-actress.csv file

2.Go to manage pictures.csv file

-1. Exit program

Actor-actressMenuDriver()

Welcome to actor-actress database manager

1.Search

2.Add

3.Modify

4.Delete

-1. Back to main database manager.

PicturesMenuDriver()

Welcome to pictures database manager

1.Search

2.Add

3.Modify

4.Delete

-1. Back to main database manager

SearchMenuDriver()

Welcome to search

1.Exact search

2.Partial search

-1. Back to database manager

9.Status of Application

The program runs successfully.

The program was developed and tested on CLion g++. It was

compiled, run, and tested on csegrid.ucdenver.pvt.