

CSCI 2275 – Programming and Data Structures
Instructor: Hoenigman
Assignment 7
Due Wednesday, October 31 by 11 am

Binary Search Trees and Heaps

An online movie service needs help keeping track of their stock. You should help them by developing a program that stores the movies in a Binary Search Tree (BST) ordered by the first letter in the movie title, and then a heap array for each node in the BST that includes the information for the movie. For each of the movies in the store's inventory, the following information is kept:

- IMDB ranking
- Title
- Year released
- Quantity in stock

Your program will have a menu similar to previous assignments from which the user could select options. In this assignment, your menu needs to include options for finding a movie, renting a movie, printing the inventory between 2 letters, deleting a movie, and quitting the program.

Insert all the movies in the tree.

When the user starts the program they will pass it the name of the text file that contains all movie information. Each line in the file shows the IMDB ranking, title, year released, and quantity in stock. Your program needs to handle that command line argument, open the file, and read all movie data in the file. The heap is based on the quantity of the movies.

From this data, build the BST ordered by the first letter in the movie title. For each of the nodes in the BST, there should be a heap of the actual movie data. *Note: the nodes should be added to the heap in the order they are read in.* The name of the file that contains the movie data is Assignment7Movies.txt.

Rent a movie.

When the user selects this option from the menu, they should be prompted for the name of the movie. If the movie is found in your data structure, your program should update the Quantity in stock property of the movie and display the new information about the movie.

If the movie is not found, your program should display, "Movie not found."
You need to rebalance the heap if needed when the quantity is reduced. When the quantity reaches 0, the movie won't be deleted from the array but will be just

pushed down the array. Keep a count of the number of nodes in the heap and if the number becomes 0 you delete the BST node.

Print the entire inventory.

When the user selects this option from the menu, your program should display all movie titles and the quantity available. You can just print the heap as it is. (It won't be sorted by title or quantity in the heap array)

Delete a movie.

When the user selects this option, they should be prompted for the title of the movie to delete. Your code should then search the tree for the first letter of that movie and then search the heap linearly for the title. If the title is found, set the quantity to 0 and rebalance the heap and decrease the count of movies in the heap. If it was the only title left for that letter in the BST, you also need to delete the node in the BST and re-assign the parent and child pointers to bypass the deleted node and free the memory assigned to the node. If the movie is not found in the search process, print "Movie not found" and do not attempt to delete.

Count movies in the tree.

When the user selects this option, your program should traverse the tree and heap and count the total movie nodes in the tree and print the count.

Print top 10 movies between 2 characters.

Given 2 characters, print the top 10 movies by quantity for each character between the given characters. Eg: Given A and H- Print top 10 movies for A, top 10 for B, top 10 for C and so on till top 10 till H.

To get top 10 for each characters you will have to pop 10 times from heap. Print those elements and push those movies back again.

Quit the program.

When the user selects this option, your program should delete the nodes in the tree and exit the program.

When the user selects quit, the destructor for the MovieTree class should be called and in the destructor, all of the nodes in the tree and singly linked lists should be deleted. You need to use a postorder tree traversal for the delete or you will get segmentation fault errors.

Implementation details

Your BST and Heap should be implemented in a class. Each BST node will have an instance of the heap class. You have to write test cases for the functions you write. Those test cases will be in main and you can write your own dataset of movies (preferable small) and write test cases for that small dataset.