Music Library Simulator

CSCI 2270: Data Structures Final Project
AJ Yang and Sebastian Brunet

For the final data structures project, we created a music library. Music is a huge part of both our lives and this project allowed us to explore the creation of an interface like something we use every day. To build the project, we used two different data structures, a binary search tree and a linked list.

BST (Binary Search Tree)
The binary Search Tree was crucial for our implementation as it is incredibly efficient for searching. The binary tree structure searches in average O(log n) time. For a library of data this allows for quick access to any "song" or node in the structure. An unsorted array would work in linear time which is far slower than log time. Our binary tree structure was built on a single struct and class. The struct nodes were comprised of the necessary right and left child pointers as well as data members for a song's title, artist, album, and year. Our tree class had member functions to build and search the tree.

Linked Lists
While the Binary Search Tree provided the backbone library of songs, we utilized multiple linked lists to store and create playlists for the user. Linked lists provide a quick method for creation of simple lists with a specific order. While the user might still search for songs, because the lists are not an entire library of songs the time complexity is a nonissue. The challenge in implementing the linked lists came from creating lists from nodes of the binary tree struct. It was necessary to create a convert Node function that took in the binary tree node and created a linked list node copy of that specific song. We also utilized the c++ std library to build our individual playlists of song nodes. These multiple std lists were stored in our own separate linked list of playlist nodes. This multilayer system allows the user to save, edit, and interact with multiple playlists. We used another instance of the std linked list to create a music queue to store the "playing songs".

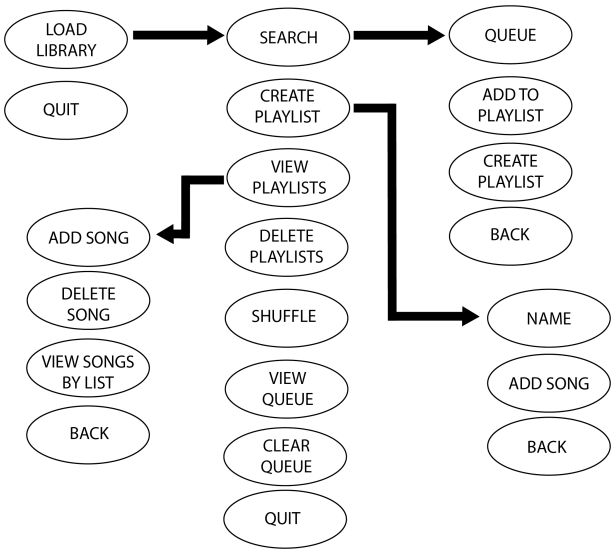
References
We referenced multiple assignments from this class including Assignment 6 which included a BST implementation as well as assignment 3 that focused on linked lists. The text file we used was from an online machine learning database.

Challenges and Task Division
The primary challenge we encountered was dealing with the nested linked lists of song nodes. Originally we thought we would store the playlist information in a hash table, but we ran into several issues accessing private class variables among three different classes. We resolved this by nesting an std list in each node of the playlistSet. In terms of task division, AJ created the tree implementation and text file loading, while Sebastian

created the linked list files. We wrote the main together, adding necessary components as we went.

Menu Flow

Menu flow incorporated through our switch menu in the main



Results

Screenshots of our program:

Menus

```
Sebastians-MacBook-Pro:Project baci$ g++ -std=c++11 MusicLibraryMain.cpp -o main

Sebastians-MacBook-Pro:Project baci$ ./main Library.txt
*********************************
MAIN MENU
 1 - Load Library.
 2 - Quit.
*********************************
 Enter your choice and press return:

 Enter your choice and press return:
1
Library Loaded.

*********************************
MENU
 1 - Search.
 2 - Create Playlist.
 3 - View Playlists.
 4 - Delete Playlist.
 5 - Shuffle.
 6 - View Queue.
 7 - Clear Queue
 8 - Quit.
*********************************
 Enter your choice and press return:

 Enter your choice and press return:
1

Search song by Title:
Title: Levon

Song Info:
***********
Title:Levon
Artist: Elton John
Album: Madman Across the Water
***********
*********************************
MENU
 1 - Add song to Queue.
 2 - Add song to Playlist.
 3 - Create Playlist.
 4 - Back to Main Menu.
*********************************
 Enter your choice and press return:
```

```
*********************************
MENU
 1 - Add song to Queue.
 2 - Add song to Playlist.
 3 - Create Playlist.
 4 - Back to Main Menu.
*********************************
 Enter your choice and press return:
3

Enter the name of your new playlist
Rock

 Enter your choice and press return:
2

What playlist would you like to add a song to?
Rock

*********************************
MENU
 1 - Add song to Queue.
 2 - Add song to Playlist.
 3 - Create Playlist.
 4 - Back to Main Menu.
*********************************
 Enter your choice and press return:

 Enter your choice and press return:
2

What playlist would you like to add a song to?
Rock

*********************************
MENU
 1 - Add song to Queue.
 2 - Add song to Playlist.
 3 - Create Playlist.
 4 - Back to Main Menu.
*********************************
 Enter your choice and press return:
```

```
*********************************
MENU
 1 - Search.
 2 - Create Playlist.
 3 - View Playlists.
 4 - Delete Playlist.
 5 - Shuffle.
 6 - View Queue.
 7 - Clear Queue
 8 - Quit.
*********************************
 Enter your choice and press return:
8

Quit Program.
*********************************
MAIN MENU
 1 - Load Library.
 2 - Quit.
*********************************
 Enter your choice and press return:
2
Quit Program.
Sebastians-MacBook-Pro:Project baci$
```