CSCI-1200 Computer Science II — Spring 2006
Lab 9 — MP3 Maps

This lab gives you practice working with maps and understanding and modifying code written by someone else. Please download the MP3 code we discussed in lecture yesterday:

```
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/09_maps/main.cpp
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/09_maps/computer.h
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/09_maps/computer.cpp
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/09_maps/mp3.h
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/09_maps/mp3_input.txt
http://www.cs.rpi.edu/academics/courses/spring06/cs2/labs/09_maps/mp3_output.txt
```

and then turn off your internet connection.

Checkpoint 1

Familiarize yourself with the code, then compile and run it with the sample input provided. To make sure you really understand how the code is working, draw a diagram (like we saw in lecture) of the data stored in the variable `mp3_system` as you walk through the processing of the sample input and output. Remember a *map* is a collection of *pairs*, each pair has a *key* and *value*. The pairs are stored in the map sorted by key. Give the order notation of the cost of each of the provided operations (adding a computer, printing the mp3s stored on a particular computer, and finding a song within the system). Assume there are n computers and m songs stored on each computer.

To complete this checkpoint: Show a TA your diagram and be prepared to explain the cost of each operation.

Checkpoint 2

Complete the implementation of the final operation, finding all songs by a particular artist. Use the diagram to help you write this code. Run the code on the provided input and make up your own interesting test to be sure your code is debugged. Again, give the order notation for the cost of this operation in terms of the number of computers and songs in the system.

To complete this checkpoint: Show a TA your debugged program. Be prepared to discuss why these operations aren't all "constant time" even though the maps are used. How could we re-engineer the system if the performance of finding all the songs by a particular artist was critical?