Approach Document Lab 4

Lab 4 – Joyce Berdkan

# Assignment Objective

Implement the three self-organizing list heuristics: count, move-to-front, and transpose.

# Assignment Requirements

Your system should have the ability to

* Implement the three heuristics
* Compare the cost of each heuristic by keeping track of the number of compares required when searching the list.

# Approach

* I will first read the instructions in good detail and review any concept that I am unsure about prior to the start of my program.
* Next, I will thoroughly read the given files (selfOrderedListADT.h, book.h, llist.h, list.h, link.h)
* Subsequently, I will create a new Visual Studios program and implement the given files.
* I will determine whether I need to add any extra files.
* After concluding my understanding the assignment, I will begin the implementation fot the following SelfOrderedListADT.h functions:
* Bool find(const E& it);
* Void add(const E& it);
* Int getCompares() const;
* Int size() const;
* Void printlist();
* Void printlist(n);
* Void reorder(const E& it);
* I will likely need to add header files for count, move-to-front, and transpose
* I will determine if I need to add any private variables
* Once I complete the implementation of the functions in SelfOrderedListADT.h, I will determine the implementation of count, move-to-front, and transpose.
* I will modify the constructors for the three heuristic types.
* I will add comments where an explanation may be required.

# Build Log

11/01/2022 - I reviewed the lab instructions and other documents today and have I've already written my strategy document. I entered the specified I uploaded the data file and loaded the files into Visual Studios.

11/02/2022 - Today, I worked on the SelfOrderedListADT.h implementation. I created the SelfOrderedList.h header file and started working on the functions' implementation. I later learned that this header file was not required. In order to do this, I made three header files: count.h, moveToFront.h, and transpose.h. I moved these header files' functions from SelfOrderedList.h.

11/03/2022 - today I made classes for each of the header files and gave each heuristic type an inheritance from the SelfOrderedListADT.h class. I made a list in each class and an int comparison counter for each class. I worked on each class's constructors.

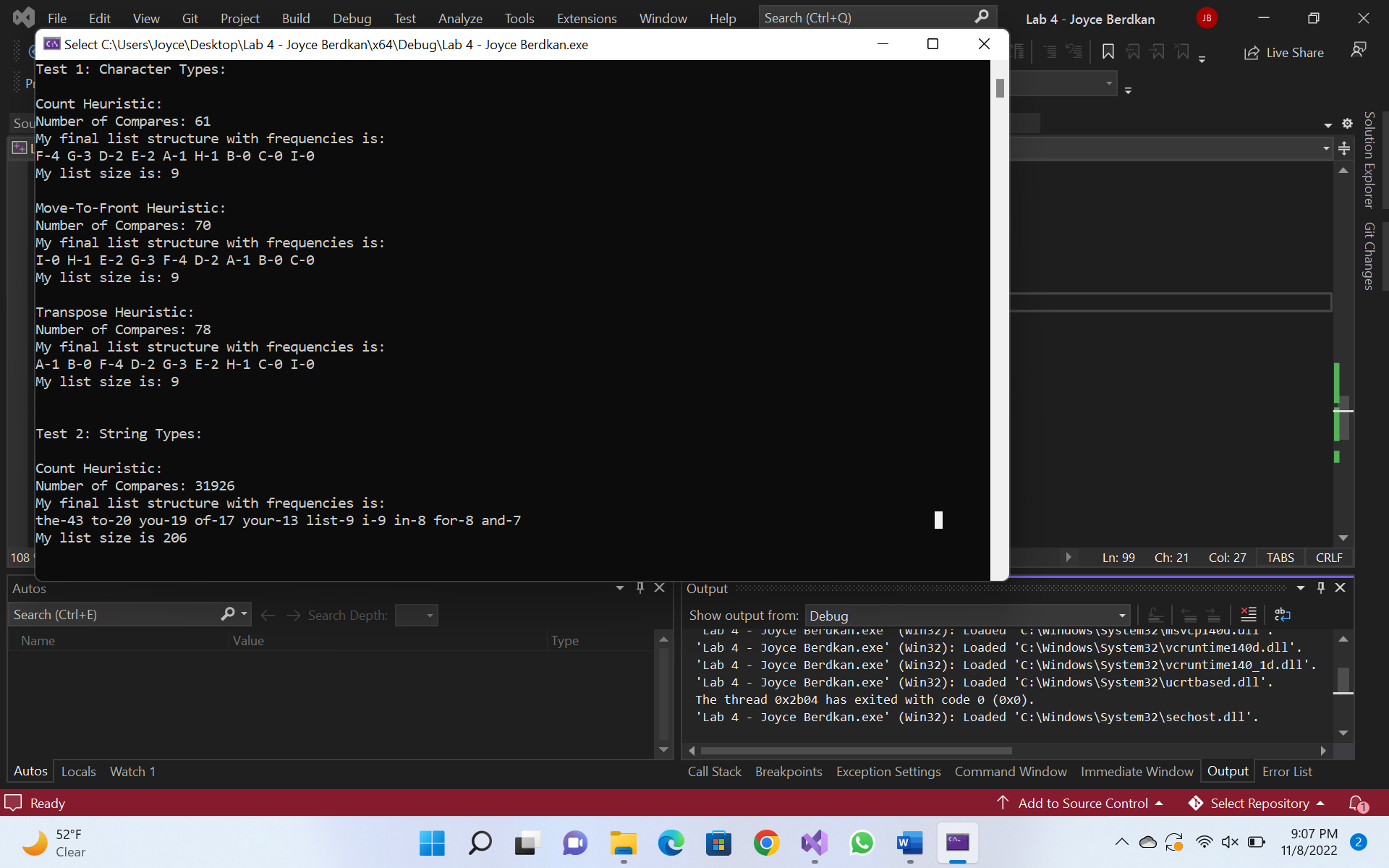
11/04/2022 - I developed a Boolean variable and an int variable for position while working on the moveToFront class's bool find(const E& it) method. After completing the locate function's implementation, I gave the other heuristic classes this function's implementation.

11/05/2022 - I've finished implementing the void add(const E& it), int getCompares(), and int size() functions. These were all really clear and easy tasks. I gave the other heuristic classes the implementation of these functions. On the printlist() method, I worked.

11/06/2022 - The printlist() and printlist(int n) functions were finished in my implementation. Implementing these features was simple. I then gave the other heuristic classes the implementation of these functions. I then added a frequency map along with an int variable to all the count, move-to-front and transpose header files.  I also finalized the implementation of the move-To-Front class's reorder method while working on it. I did, however, recognize that each heuristic class will have a separate set of ordering methods. I then started composing the count class's reorder method.

11/07/2022 - I completed the count reorder function's implementation and added a variable in the llink.h that is a Boolean that indicates if the frequency isn't set up correctly. In the count reorder function, I used this function. The development of the transpose reorder function was the next task I worked on, and I successfully finished it. Finally, I created objects for each of the heuristic classes in int main() and then added a number of character values that would work with the find(), add(), and printlist() functions that are located in each heuristic class. Then, I created a while loop that would read the text file associated with each condition and added code to int main() to open the text file  heuristic category. I made use of the printlist() and find() functions in  every course.

11/08/2022 – I added comments to any segment of code that needed explanation and finalized my program.



A screenshot of a computer

Description automatically generated with medium confidence