Name: Daniel Ems
Date: 02/4/2017
Current Module: Data Structures and Algorithms II
Project Name: Ticker

Project Goals:

The projects objective is to store formatted file text, and allow users to create or modify data. Finally the program must print the data numerically from least to greatest in an identical format. The objective is to familiarize students with different data structures, architectures, algorithms. Finally, to demonstrate the importance of addressing the time complexities associated with the each.

Considerations:

- Maintaining data integrity as it is stored.
- Deciding which data structure to store by one value, and print by another
- Determining what properties to incorporate into the data structure that lends flexibility to being modified, or duplicated.
- Comparing Tickers
- Error handling user and file input
- Printing numerically.
-

Initial Design:

The initial design revolved around a splay tree, and identifying the best way to maintain data integrity. The splay tree was populated by fgets and strtok. This allowed for error checking with both the file and user input. Finally, once the user was ready to exit the program, deciding which method of resorting would be best for a numerical print.

Data Flow:

1. Fgets and strtok pull in a string at a time from the file.
2. Each token is checked to match the appropriate format and requirements.
3. If the requirements are met, a node is inserted to a BST sorted by ticker.
4. Fgets and strtok pull in a string at a time from stdin.
5. Each token is checked to match the appropriate format and requirements.
6. The ticker is searched and the tree is splayed.
7. If the ticker is not present, a node is inserted, other wise the price is adjusted.
8. Once the user finishes, the tree is duplicated and sorted by cost, not ticker.
9. The tree prints numerically from least to greatest.
10. The appropriate FILE’s, malloc’s, trees, are respectively closed, freed or destroyed

Potential Pitfalls

The most notable pitfall I found was the amount of error handling required to maintain the integrity of the data. Extensive hours, and test cases can go in to making sure that your program handles all cases appropriately,. However, it is difficult to be certain you have identified them all. This however, is merely a matter of time constraint.

Another potential pitfall is the implementation of a self taught data structure, the splay tree. It is merely an adaptation of a binary search tree. It requires certain functionality that has not been covered extensively. While it is believed that the data structure was implemented accurately, it is truly to ask if the splay tree was utilized appropriately.

Test Plan

User Test:
1. Reading in stock file.
2. Read in a string at a time.
3. Error check appropriately, and handle appropriately.
4. Insert into binary search tree and sort based off stock tickers.
5. Repeat steps 1-4 until entire file is inserted.
6. Read in user input.
7. Repeat steps 1-4 until they press Ctrl-d
8. Once ctrl-d is pressed, create a second binary search tree from the nodes in the first
9. Sort the second tree numerically
10. destroy the first binary search tree and print the second.

Test Cases:
1. Read in stock file and print out put that is stored
2. Modify the stock file and try to crash my program.
3. Once the program crashes, fix error.
4. Repeat steps 1-3 for user input.

Conclusion:

The most effective aspect of this project was developing, and implementing a good strategy for developing a splay tree. The plan was crucial in successfully building out the data structure. This was important because is allowed me to program outside of my comfort zone. This, I truly believe, is the cornerstone of learning. More so, than any other project, I feel confident in saying that I have learned something.

The most detrimental aspect of this project was the evil twin to the most effective aspect, the data structure. Developing with tools, data structures and algorithms foreign to you, certain learning curves are identified throughout the process. Unfortunately, when a deadline is in place, these learning curves often times can become sloppy patch work. However, I do believe being forced to face new problems adds to the learning experience.

The most important thing to carry forward is the approach to learning new concepts. This can be as simple as evaluating what potential down falls you may run into along the way. I do believe this project was a success. However complacency is cousin to laziness. Future exploration will be more effective due to the lessons learned from this project.