Matthew Bramer

January 23, 2022

CS-260 Data Structures and Algorithms

3-2 Programming Activity: Lists and Searching Code Reflection and Pseudocode

**Code Reflection:** A brief explanation of the code and its purpose, and a brief discussion of your experience in developing it, including any issues that you encountered while completing the exercise and what approaches you took to solve them.

**Bid** **Search**(**string** bidId);

Here the Bid is a structure that was defined before this line of code. Bid has been appended to bid (notice the lower case “b”). The Search operation returns item if found, else returns 0. Learning the linked list has been fun! I’ve learned a lot, especially with the ADT. The Common Operations for a List ADT, table 3.1.1 in ZyBooks, seems like it is going to be very helpful in the developing field. Additionally, I was amazed by the simplicity of pointers! Specifically, the reference operator as in: “&variableName” and the dereference operator: “\*variableName”. A very understandable process.

**Pseudocode:** A pseudocode or flowchart description of the code that is clear and understandable and captures accurate logic to translate to the programming language.

Bid, being the structure being used for this LinkedList, opens with a node in tempPtr (temp pointer) that equaling the headPtr (head pointer). Next, a return bid is required, so Bid returnBid. Then an if/then is needed. If there were no search started, then print out “there are no nodes in this list, search aborted”, and then return a returnBid. Else, while tempPtr is not null, add another if tempPtr utilizing the member access operator (->) to bid.bidId equaling bidId, that returns returnBid equaling tempPtr to bid. Then tempPtr equaling tempPtr to nextNodePtr. And finally returning returnBid.