Brooks Jackson
April 10th, 2023
Week 10 Journal

1a. The two operations that I see as most similar are the “push” operation from the ADT Stack Operations and the “insert” operation from the ADT Sorted List Operations. “Push” adds a new item to the top of the stack, while “insert” adds a new item in the sorted list in the correct order. The two operations are similar because they involve adding a new item to a collection, however, the sorted list operation varies slightly because of it needing to be inserted in a specific position on its value.

1b. The two operations that I see as least similar are the “pop” operation from the ADT Stack Operations and the “remove” operation from the ADT Sorted List Operations. “Pop” removes the very top item from the stack, while “remove” must remove a specific item in the list. The two operations are different because “pop” can only ever be the top item, while “remove” has no limits on which item it can remove, as well as the fact that it might have to shift items around to fill the gap, whereas “pop” does not have to do that.

2a. Array-based implementation and reference-based implementation are both different ways to implement a stack data structure. Firstly, the array-based implementation uses a fixed-size array to store elements in the stack. The top of the stack is an index variable which points to the last element in the stack. If a new element gets added, the top gets incremented, and the new element goes into the next available slot in the array. If an element gets removed, the top gets decremented, and the element at the top position of the array is removed. On the other hand, the reference-based implementation uses a linked list to store elements, each one being represented by a node. Each node has a reference to the next node in the list, and the top of the stack is a reference to the first node in the list. If a new element gets added, a new node is created, and the reference is set to the current top node, which becomes the new top of the stack. If an element gets removed, the top reference is updated to point to the next node, and the removed node is discarded. One way in which the two are similar is that they both have stack operations such as push, pop, peek, and isEmpty. One way in which the two are different is that the array-based needs a fixed size, while the reference-based can have a dynamically growing size.

2b. The greatest advantage of the array-based implementation is the simplicity and efficiency. The array allows direct access to the elements, which making pushing and popping very quick and easy. The fixed-size array is also very convenient for memory efficiency. The greatest advantage of the reference-based implementation is its flexibility. The linked list can grow as needed, so it can handle stacks of any size. There is also some more convenience in scenarios like when the size of the stack is not known beforehand or if it needs to be modified.

3a. Infix: 3 + 4 * 2 / (1 – 5) ^ 2
Postfix: 3 4 2 * 1 5 – 2 ^ / +

3b. “To convert an infix expression to postfix form, we use a stack to keep track of operators and parentheses and to determine the order of precedence and associativity among the operators.” (pg. 351) I chose this quote from the textbook because it does a good job of summarizing the main role of the stack with infix and postfix conversions. The stack is used to keep track of the operators and parentheses, as well as making sure the operators have the correct order.