2. The one argument form of insert, compares the value passed to it with the values at the nodes that already exist in the sequence in order to evaluate. However, when trying to use the one argument form of insert with Complex, for the complex object to be successfully inserted, it needs to have overloaded the ‘>’ operator, which the class doesn’t, and this is why we receive an error.

3. b) For us to be able to write a recursive solution the MenuItem\* m parameter needs to be broken down into smaller parts (i.e. make the problem smaller), but at the same time we also need to change the string path of the MenuItem, so to solve this problem recursively we need at least 2 parameters.

4. a) O(N^3) as we have 3 for loops that run up to N nested in each other.

4. b) It remains O(N^3) because we calculate the worst case for the time complexity. The worst case would be that i will be N, making this similar to 4.a).

5. a) O(N2). In the first for loop k goes from 0 to N. Then within this for loop, we use get(int pos, ItemType& value) that goes through the linked list k to N/2 times. Although the second for loop in the function also has time complexity O(N2), we are ignoring coefficients while calculating the overall time complexity.

5. b) O(N). This is better than 5.a) interleave. The for loop loops runs both seq1 and seq2 N times, however insertBefore is a constant time O(k) function. Although in the worst case, the second for loop in the function and the swap also have time complexity O(N), we are ignoring coefficients while calculating the overall time complexity.