2. The error is because whoever creates the Complex class forgets to define a comparison operator for the class. Computer knows how to compare two int, and the insert function that takes two parameters doesn’t need to compare two ItemType, so they are fine. However, in the insert function that takes only one parameter, there is a comparison between the send in ItemType, so an operator> should be defined for Complex class.

3b. One parameter is not enough because there would be no way to know the name of the MenuItem that opens the current MenuItem. In this case the output would only be the name of this MenuItem and the name of MenuItems in the vector it holds. There is no way to find the third level of MenuItem’s name.

4a. O(N3) Because there are three loops, one inside another, from 0 to N to find the mutual friends.

4b. O(N3) Because there are still three loops, one inside another. Although the second loop is only from 0 to i, the worst case is i = N, so it is still from 0 to N for the three loops.

5a. O(N2) Because for each node in seq1 and seq2, it finds the node, and insert it in res. It never touches the nodes in result. Totally there are 2N nodes, and for each one the function finds it in the list in N steps at most, and insert it in new list in 2N step at most. Therefore the actual time complexity is N\*(N+2N), and we get rid of the number.

5b. O(N) Because it only need to visit each node in seq1 and seq2 once. It has access to other sequence, so it does not need to start from the head node and try to find a node in the list each time. Neither does it need to start from head node to insert anything. Therefore, it is faster and better than the interleave above.