2. Because in function "int insert(const ItemType& value);" there is a comparison operator '>', which is not declare and define in Complex class, if we want it work, we must define the comparison operator > in Complex class.

3.b. If we only use one parameter, there is no way to pass the current menu name to deeper level menu. In another way, we can say that if we only use one parameter, we can only print out name of one menu level each time, but not be able to print out the full path.

4.a. The time complexity of this algorithm is O(N3). Because there are 3 for-loops nesting together, the time complexity for the most inner loop is O(N) since it's body has time complexity O(1) and looping N times since k < N. Then the second loop with O(N) in the body will loop N times again since j < N, and the third loop will loop N times as well. Hence, the time complexity for this whole algorithm is O(N3).

4.b. The time complexity of this algorithm is still O(N3), since we always using the worse case to calculate time complexity. Which means we treat j < i in the second loop as j < N since i < N.

5.a. The time complexity of this function is O(N2), the "insert(resultPos, v)" function has time complexity of O(N) since it contain function "nodeAtPos(pos)" which iterate through all nodes. And each for loop is looping through every element in sequence which is N times. There for the time complexity of the algorithm is combination of two O(N2), which is same as O(N2).

5.b. It is better than implementation of part a. "insertBefore" function has time complexity of O(1), and the for loop is looping N times since each sequence has N elements. The second loop is looping 2N times since it's the combination of seq1 and seq2. Therefore, the total time complexity of the whole algorithm is O(N), which is better than O(N2) above, especially when N is a large number.