1e.

The problem is that the iterator variable is not redefined after the push back so when push back is called, the vector allocates more memory in a new location and the iterator is not changed to match this location. Therefore, the iterator pointer does not increment to the right location and is incrementing to where the smaller vector used to be.

3.

There is an argument to the function call of the one-argument form of Sequence<Coord>::insert because of the > comparison operator that is in the single argument call of insert. For the struct type of Coord, there is no real logical way for comparison and the operator is overloaded, so there will be an error when trying to determine where to insert; however, this is not a problem for the two argument call since the position is given and it does not use the > operator.

4b.

The recursive call needs two parameters because the second parameter is needed to store the information about the path. If there was only one parameter, then the path will be reset each time. The parameter is needed to be able to append the new menu item to a preexisting path and print it.

5a.

The time complexity is O(N^3) because there is a triple nested for loop that goes to N each time. Therefore, each loop will run a N amount of time, but since it is nested two other times it becomes N \* N \* N

5b.

The time complexity is O(N^3). The j loop runs N times Since j runs up to i, that means that the value can only reach up to i, equating to around N-1. There is also the k loop that still runs to N, so the time is N \* N \* N

6a.

O(N^2) In terms of the number of nodes visited, the Concat Reverse function visits O(N) nodes. The loops run to N-1, or N nodes and inside each loop, it visits 2 nodes. So it becomes a total of 2(N-1)\*2, which simplifies to O(N^2)

6b.

O(N) the function visits each sequence once in the function. For the two sequences, thai would be 2\*N. Then the swap just visits each value in the sequences once more for 2\*N. In the end, the value would be 2\*N + 2\*N = 4\*N, or O(N)