Cody Hubbard - HW4 - 004843389

2.

The call to the one-argument form of Sequence<Complex>::insert causes at least one compilation error because there is no comparison operator defined for the class Complex. Thus when the one-argument form of insert tries to compare the two Complex objects in the sequence it will have no idea what to do.

3(b).

I could not solve this problem by recursively calling the one parameter overload of listAll given the parameters in part a because every recursive call would completely forget the name of it’s parent. This completely removes the string concatenation required for correct output.

4(a).

The complexity of this algorithm is O(N^3) because there are three if loops, two of which are nested, and all 3 have N maximum possible iterations.

4(b).

Although this algorithm is a bit “simpler” than the first one the complexity is still O(N^3). Even though the first nested for loop only executes around N/2 times, coefficients are dropped when computing time complexity which results in a time complexity of O(N^3).

5(a).

Assuming that seq1, seq2, and the old value of result each have N elements the complexity of this algorithm will be O(N). This is because at the most the algorithm will go through all elements in seq1 plus all the elements in seq2 which would be 2N but coefficients are dropped.

5(b).

Assuming that seq1, seq2, and the old value of \*this each have about N elements the complexity of this new algorithm is O(N). Even though the data structure used for holding the sequences was presumably changed the speed at which elements are accessed is still largely the same. Additionally the logic did not change much and all iterations of length N are independent, additive, and ignorable in the context of O().