Annie Wang

Professor Smallberg

CS 32

2 March 2021

HW 4

2. The call to the one argument form of Sequence<Coord>::insert causes a compilation error because there is no way to compare Coords, which is a user-defined type. To fix this error, the program would require that the user define a custom comparison operator such as the < operator.

4b. This problem could not be solved with only a one parameter listAll and we had to implement it as the recursive function because the second parameter is needed to keep track of the path up until the current point. With 2 parameters, the string tracks the path and produces the necessary output and without the second parameter, there is no way to track the path.

5a. The time complexity of this algorithm is O(N^3) because the innermost loop takes N steps to perform at most, the middle loop takes N steps as well, and the outer loop also takes N steps. Therefore, the time complexity is N\*N\*N which yields a big O of O(N^3)

5b. The time complexity of this algorithm is also O(N^3) because in the worst-case scenario, the innermost loop takes N steps, and the second loop takes i steps at most. The outermost loop takes N steps at most and if i reaches N in the worst-case scenario, the second loop also takes N steps, so N\*N\*N steps total, a big O of O(N^3)

6a. This function is O(N^2) time because the first for loop iterates and takes N steps to complete, assuming that both seq1 and seq2 have N items, it would take at most N steps to complete. The second loop also takes N steps to complete, but its calls to the get() function also takes N steps, since the get() calls the nodeAtPos(int pos) function, which iterates through all elements in the loop in the worst case scenario, and therefore, the second loop takes N\*N steps to complete at most. Because N\*N in the second loop overpowers the N in the first loop, the big O of this function is O(N^2).

6b. This function is O(N) time because both the first loop and the second loop iterate through all the elements of the sequence, which, in the worst-case scenario, would take O(N) steps to complete. Because both loops take O(N) time, the big O of the function would still be O(N).