CS32 HW4 Answers

Benjamin He

UID: 804962948

2) The reason the error occurs is because there is no definition of the comparison operator for comparing Coordinates, so the program doesn’t know how to find the proper location to insert the new Coordinate. The two-parameter insert works because the position to insert is explicitly indicated so no Coordinate comparison occurs, and the one-parameter insert for a type int also works because there is a default comparison operator already defined for type ints.

4b) Implementing a one-parameter listAll function would not be possible given the constraints because there would be no way of tracking the total path of the current file because whatever path changes that occur in the function would not be recorded without some sort of global/class member variable keeping track of it or without being able to pass in the current path like what has been done in the two parameter listAll function that we had to implement.

5a) The time complexity is **O(N^3)** because there are three nested loops, and the outermost loop occurs N times, and each time the second outer loop must occur N times, then for each of the second outer loops iterations, the innermost loop also occurs N times, so the algorithm takes approximately N\*N\*N actions to complete.

5b) The time complexity is still **O(N^3)** because to analyze this algorithm, we can substitute the i in the second outer loop’s condition with the maximum i can be, which is approximately N, then the algorithm is basically the same as before, taking approximately N\*N\*N actions to complete.

6a) The time complexity is **O(N^2)** where N is the size of the larger of the two Sequences. This is because there are two loops in series, where one goes from 0 to nmin, then the next goes from nmin to n (where n is the size of the larger Sequence), and all the actions that occur outside each loop is O(1), but the get and insert functions both call the nodeAtPos function, which has a loop that will, in the worst case scenario, traverse the entire Sequence and thus execute approximately N times. So the main factors of the time complexity are these two loops, both with loops within their function calls, thus the algorithm takes approximately n^2 actions to complete in the worst case scenario.

6b) The time complexity of this algorithm is **O(N)** where N represents the size of the larger Sequence of the two that are passed into this function. This is O(N) because there are two loops in series, where the first loop goes through both Sequences until it hits the end of the smaller sequence, then the second loop goes through the rest of the second sequence. Since every action in the code other than the loops are O(1), the complexity would just be the sum of the two loops which is O(N). Thus, this algorithm is the better in terms of time complexity as the one in part a.