Raymond Chen

Professor Smallberg

CS32 Spring 2023

CS 32 Homework 4

**1e.** Test case 3 fails because the vector iterator has been invalidated in the process of adding items. This occurs when k == 2 and the line “v2.push\_back(MAGIC)” is run.

**3.** The call to the one form argument of Sequence<Coord>::insert causes at least one compilation error because there is no overloading of the comparison operator for the “Coord” type. This is needed as the nature of the one form argument insert function is that it compares the to be inserted item with items already in the sequence. Meanwhile, the call to the insert function with two parameter works because the user is determining the position of insertion.

**4b.** We cannot solve the problem with recursion if we only had the one parameter listAll function because we would not be able to pass down and print the corresponding menu items. The path parameter is what allows for us to keep track of the menu items when traversing through the menu.

**5a.** The time complexity of this algorithm is O(N^3). Since there are 3 for loops nested inside one another, the time complexity would be N + N^2 + N^3. We then keep the most significant term of the function and end up with N^3.

**5b.** The time complexity of this algorithm should still be O(N^3). Despite the second for loop limit being i and not N, we have to account for the fact that the maximum number of iterations (worst case scenario) for the second for loop could be N. Thus, the time complexity of this tweaked algorithm is found by N\*N\*N = N^3.

**6a.** The time complexity of this algorithm is O(N^2). This is because concatReverse is implemented with linked lists. In each iteration of the first for loop, concatReverse executes the get function to get a value and insert it at the end of the result sequence . The time complexity of this would be N(N+1) which we can say is N^2. The same thing is done in the second for loop so the time complexity would also be N^2. We add these together and get 2N^2 and end up with O(N^2) when dropping the coefficient.

**6b.**

There are two independent for loops. Each for loop runs N times so it would be 2N but we drop the coefficient so the time complexity is O(N). This is better than the part a implementation as it has a lower time complexity thereby running faster.