2.

The one argument form of the call to the insert function would cause a compilation error because that insert function relies on comparing the elements (inserting at the smallest index where the parameter <=). Because the coordinates class does not currently a way to compare coordinates against each other, a custom comparison operator would need to be defined in order to successfully compare within the single parameter insert function.

4b.

If there was only a one-parameter listAll function, this problem cannot be solved recursively with the same output as in part a, because in part a, the second parameter keeps track of the path successively in each recursive call. This is the only way to output the entire path from each directory to each further file.

5a.

O(N^3) because in the first for loop, I is compared with N, N times. In the second for loop, j is compared with N also N times, but since it’s nested in the first for loop, it’s being compared N^2 times. In the third for loop, k is compared with N, N times, but since it’s nested in the other two loops, the comparison is occurring N^3 times and that’s the most frequently occurring operation in this algorithm.

5b.

O(N^3) because the first for loop is still occurring N times, and then the second for loop runs N\*(N-1)/2 iterations, and the innermost for loop still runs N times. Since these loops are all nested, the innermost comparison occurs N\*(N-1)\*N/2 times and the most significant term is N^3. Thus, the big-o of this algorithm is still O(N^3)

6a.

If seq1 and seq2 are assumed to be of size N, then nmin is N and all the statements within the first for loop are executed N times, and the time complexity to insert into a linked list is O(1). However, the time complexity to get elements of a linked list is O(n), which occurs when accessing elements in the middle of a linked list. Accessing elements within the first for loop is the most often occurring operation in this algorithm. The outer loop is executed N times and within that loop, there are N executions in each access. Thus the time complexity is O(N^2)

6b.

The time complexity for this function is O(N) because the first for loop occurs N times and within that for loop, each insertion is an insertion at the beginning of the linked list and the time complexity for inserting at the beginning is O(1). The contents of the loop contain only 1 execution and so the time complexity for the function overall is O(N), as the statements within the first for loop are the most often occurring statements. This is better than the implementation in part a.