Gautam Anand

Homework 4

1e)

Test case three fails because there is only 5x4 bytes of storage for 5 integers in the array. So when the 6th element is pushed back onto the vector, then the computer needs to find a new place for the vector to have continuous bytes of information. This leaves the p pointer dangling and pointing to where the vector used to be before it got bigger, and this leads to the test case failing.

3)

The one argument form causes a complilation error because, the one argument form of sequence compares values in order to determine where the item should be placed in the sequence. Because Coord doesn’t have a comparison operator overloaded to it, the code produces an error.

4b)

There is a need for two parameters when recursively calling this function because there needs to be an argument that stores the current path, and each recursive call allows us to add one more element to the path because that parameter is passed into the function.

5a)

This algorithm is O(N^3) because there are 3 for loops and they are all nested. This means that in every for loop, all N elements are traversed and this makes it N\*N\*N, or O(N^3) complexity.

5b)

This algorithm is still O(N^3) because there are still three for loops and they are all nested. So the outer one repeats N times, and the max amount that i can reach is N-1, so j’s max value is N-2. The inner most for loops goes through N elements. So there is still an N^3 complexity, even if the second loop goes through a little less times in total.

6a)

The time complexity of this algorithm is O(N^2). The first for loop goes through N times, where N is the size of the sequence, and then inserts and gets the item, which is time complexity N. The second for loop is not nested, and it only goes through also N times, and also inserts and gets in time complexity N. So the time complexity multiplies in this case and is O(N^2).

6b)

The time complexity of this algorithm is O(N). The first for loop goes through N times, where N is size of the sequence. The item is then inserted with constant time. The other for loop does the same, and also inserts in constant time. So the complexity is O(N) in this case. However, this algorithm’s time complexity is better than in part a, because a seperate sequence does not need to be created in order to insert, and the inserting is done direclty on the sequence using the insertBefore function. Also, insert in this function is done in constant time, instead of having to use other functions inside the for loop that have time complexity O(N).