**Izak Bunda – UID 305783387 – Homework 4 – Due 5/24/22**

**1e.**

Test 3 eventually fails because when vector becomes “full” or reaches the capacity of the initial vector, it will reallocate a new chunk of memory and copy over the contents of the old memory into the new one. In the process, the iterator is not copied over. So, past item 5 (the original end of the vector), the iterator will be pointing to random parts of the memory.

**3.**

Like the example in class where two items of the type Chicken were being compared with the less than or greater than operators, we can’t compare Coords to each other using the less than or greater than operators which is what is being done in the for loop inside the one argument version of Sequence<Coord>::insert.

**4b.**

We would have no way to keep track of the label of current/root Domain. The subdomains would be lost.

**5a.**

O(N^3). The outer loop runs for N times, the inner loop runs for N times, and the innermost loop runs for N times. N\*N\*N = N^3

**5b.**

O(N^3). It is the same because the outer loop runs for N times, the inner loops runs for i times which is bounded by N times, so technically just N times again, and the inner loop runs N times.

**6a.**

O(N^2). Given that there are seq1, seq2, and the old value of result all contain equal amount of N elements, the first loop runs N times, and within the loop the get function is called which calls the nodeAtPos which traverses through the linked list which takes N amount of steps. This is repeated in the second for loop, but we drop the coefficients in the end.

**6b.**

O(N). The for loop runs for the entire “length” of the linked list ( starting from m\_head->next to m\_head ). The second for loop also runs N times, but we drop the coefficients in the end.