1. Both C++ and Java have default “Collection” based data types. In C++ these are called Containers and are provided as part of the Standard Template Library (STL). In Java, these are provided in the default java.util package. As part of this course, we’ll be implementing some of these same data types from scratch. Using Java’s versions, answer the following questions. Note that you are not expected to understand the details or terminology yet. The goal is just to get a general idea and feel for a couple of collection-based data types. The answers to the questions below are found on the following webpages:

- docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/List.html
- docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Map.html

(a). Describe two of what you think are frequently used functions in the Java List interface. Explain.

(b). Using the Java documentation, briefly state the high-level difference between a Java ArrayList and a Java LinkedList in terms of how they implement the List interface.

(c). State the difference between a Java List and a Java Map.

(d). Describe two of what you think are frequently used functions in the Java Map interface. Explain.
2. Consider the following function where \(n\) is the size of array.

```c
1: // pre: array has at least one element
2: int find_max(const int array[], int n)
3: {
4:   int max = array[0];
5:   for (int i = 1; i < n; ++i) {
6:     if (max < array[i])
7:       max = array[i];
8:   }
9:   return max;
10: }
```

(a). In plain English give the function’s best and worst case.

(b). Give the detailed cost \(T(n)\) of the function for the best and worst case.

3. Mark whether the following statements are true or false.

   (a). If \(T(n)\) is \(O(n)\) then \(T(n)\) is also \(O(n^2)\).

   (b). If \(T(n)\) is \(\Omega(n)\) then \(T(n)\) must be \(O(n^2)\).

   (c). \(T(n)\) can be both \(\Theta(n^2)\) and \(\Omega(n)\).

   (d). The best case of a function is often when \(n\) is small.

   (e). A function that doesn’t have a best or worse case can still be slower as \(n\) increases.

4. Show that \(T(n) \leq 2n^2 + 3n - 1\) is \(O(n^2)\). (Note this means finding suitable values for \(k\) and \(n_0\) according to the definition of Big-O).

5. For \(T(n) \geq n^2 - 3n\) and \(T(n) \leq 3n^2 + 2n\), show that \(T(n)\) is \(\Theta(n^2)\). (Note this means finding suitable values for \(k_1, k_2,\) and \(n_0\) according to the definition of Big-\(\Theta\).)