CSCE146 – Lab Report for Lab Assignment 3

1. (1.4 points) For each of the following methods written in the *LinkedList* class, give the best big O for the method's **average case** time and space complexities.

   a) add to beginning — Time: O(1) — Space: ? (1)

   b) add to end — Time: O(1) (with tail pointer) — Space: O(1)

   c) add after current — Time: O(1) — Space: O(1)

   d) move current forward — Time: O(1) — Space: O(1)

   e) remove current — Time: O(1) — Space: O(1)

   f) find — Time: O(n) — Space: O(1)

   g) toString — Time: O(n) — Space: O(n) (for the string builder storage)

2. (0.3 points) What challenges did you encounter when completing the lab assignment and how did you address them?

   One of the biggest headaches was making sure the linked list stayed intact while adding and removing nodes. It was tricky to keep all the pointers straight without messing things up. To tackle this, I tested the methods with all kinds of lists – empty, one node, and multiple nodes – to catch any issues. I also made sure to handle the tail pointer correctly when adding or removing nodes at the end to keep it efficient.

3. (0.3 points) What, if anything, is missing and/or could use improvement in the files that you submitted for the lab assignment?

   The current implementation could be improved by adding more comprehensive error handling and boundary checks, for example, ensuring that operations on the current node do not fail silently if the list is empty. Additionally, implementing unit tests for each method would further ensure the robustness and correctness of the linked list implementation. Lastly, adding documentation comments for each method would improve code readability and maintainability.