Data Structures **Name:** _____
Prof. Bolton **Net ID:** _____
Assignment 2

This assignment contains 2 pages (including this cover page) and 3 questions. Total of points is 100.

Conditions: All work must be completed individually. No outside resources are permitted. The only permitted resources are your texts and class notes.

Write your answers neatly and clearly on standard paper. Include your name and Net ID. Follow submission instructions as indicated on Canvas.

---

1. (40 points) Complete the following Exercises from Drozdek 4th ed : 1.11.4.a-b, 1.11.6, 3.10.11, 3.10.22.a-c.

2. (50 points) Recursive Sequential Search.

   A. Provide pseudocode for a recursive sequential search algorithm searching over a sequence. First, formalize a recurrence, then provide pseudocode. Always be sure to formally define the input with any preconditions, the output and any postconditions, and other desired conditions, assumptions, etc. Hint: ...

Algorithm 1

Require: Describe variables and define Domains: search for item $s' \in S$ in sequence $\{s_n\}$ of length n, where $s_i \in S$. S is some domain where equivalence relation is defined and index $i \in \mathbb{Z}^+$ is used for indexing. **Precondition** (assumption): Index $i = 1$. The searching algorithm will return the index where s' is found or will return -1. Formally the **Post Condition** is $(s_i = s') \vee (i = -1)$

function SEARCH(i, s', $\{s_n\}$)

...

return i

   B. Based on the pseudocode, derive a recurrence for a step count function, solve the recurrence, and determine the Big-Theta time complexity for the worst case.

   C. **Translate Theory into Practice.** Provide C++ code for **int search(item i)** a recursive member of a LinkedList class whick returns the index where item i is found. You may assume the == operator is overloaded so as to compare items. Feel free to make any other reasonable assumptions and state them as needed.

   D. Based on the C++ code provide a Big-O Space complexity analysis for the Worst Case. Justify via discussion.

3. (10 points) Compare and Contrast (via discussion, no proof necessary) the Time and Space complexities of member method **item removeFromBack()** for a SinglyLinkedList class vs a DoublyLinkedList class. You may wish to provide an implementation to support your discussion.