Data Structures **Name:** _____
Prof. Bolton **Net ID:** _____
Assignment 3

This assignment contains 1 pages (including this cover page) and 5 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.

Include your name and Net ID. Follow submission instructions as indicated on Canvas.

---

1. (15 points) Consider matrices. Briefly describe the difference between a Sparse Matrix vs. a Matrix.

2. (15 points) Assume you plan to implement a Sparse Matrix Class in C++. Draw a UML class diagram(s) including any helper classes as needed. Include all members, parameters and types.

3. (15 points) Draw a flow diagram for the addition method for your Sparse Matrix class. State all assumptions about the state of the two sparse matrices prior to the addition. State postconditions related to the resulting sparse matrix.

4. (35 points) Consider operations on a matrix. Investigate the following question: *Is using a SparseMatrix (rather than a standard Matrix) more efficient in all cases?* Consider an nxn matrix. Observe: when there is only 1 non-zero entry, a SparseMatrix implementation is very efficient. However, when a matrix is completely filled with non-zero entries, is using a Sparse Matrix representation efficient as compared to a standard Matrix representation. Why? Consider the Addition and Transpose Operations when answering this question. Describe and explain an exemplar input case(s) to help support your discussion. Compare step and memory counts.

5. (20 points) Sparse Matrix. In class we discussed a fast transpose method for an array implementation of a Sparse Matrix. In this method a jump table is created using arrays name *RowStart* and *RowSize*.

   Given the following Matrix,

$$M = \begin{bmatrix} 1 & 0 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

   A. Illustrate its Sparse Matrix representation.

B. Assume the fast transpose method is applied to this Sparse Matrix. Illustrate the end state of the following two arrays *RowStart* and *RowSize*.