**Lab 5 – MST – Approach – Jacob Greer and Jonathan Wilson**

**Preparation:**

* Research Minimum Cost Spanning Trees online and in the textbook. This led to learning of Prim’s algorithm and Kruskal’s algorithm (we settled on Prim) used in MCST
* Copy the matrix onto paper and graph the resulting MST (for visual representation)
* Make sure that we understand what constitutes an adjacency matrix than what we already know
* Construct a project with base classes and a main

**Process:**

* First constructed GraphADT.h which was found on page 387 of the textbook, little to no issues here. We added another setEdges2 function to make printing the “add edge” parts a bit easier for us
* We did use smart pointers in our program.
* Constructed functions that can read, write, and print the graphs to the console
* From here we started to test out the functions we had implemented and decided that the best case for using smart pointers would be in main storing the graphs and utilizing the functions
* The previous step facilitated during the testing of our MCST class. This class was implemented without much difficulty except for a few hiccups here and there, such as how to print where each edge is being added to and updating whether a node had been visited or not.

**Conclusion:**

* We learnt a lot about how to code a matrix, manipulate said matrix, and storing/printing graphs from a text file.
* We also have an increased level of understanding of Prim’s algorithm practically, as well as a theoretical understanding of Kruskal’s algorithm