Weekly Report

# Supervisor Notes

Unfortunately I have been unable to meet my supervisor due to the strikes. I have added new objectives and removed complete tasks from the last week

This week’s checklist should cover the following 2 weeks

* Implement Unit tests for SDS
* Implement SDS to pass tests
* Create Requirements for DFS class
* Implement Unit tests for DFS class
* Pass all Unit tests for DFS class
* Pass all Unit tests for Agent class
* Create Gantt Chart

# This week’s progress

I have created the list of requirements for the DFS class that is used to check if a graph has cycles as well as if the graph is a spanning tree.

Requirements for DFS

* Able to identify cycles in graph
* Able to identify spanning trees
* Traversal is Depth First Search

I have not been able to implement the Unit tests for SDS and the SDS class itself. I have designed how each phase of SDS will work and the class is ready to be created once the Unit tests have been implemented and have created the skeleton class for SDS. Each phase is private to incorporate encapsulation. However, this makes testing each phase difficult as variables cannot be accessed by the test class.

This weeks’ time has been devoted to developing a method to generate random spanning trees using any given EWG. To accomplish this I have used a method similar to Krushkal’s algorithm. However I needed a method to detect if a EWG has a cycle and if it is a spanning tree. A Depth First Search traversal class has been created to accomplish these tasks.

I have completed the list of requirements and tests for the DFS class which means the implementation of the class has been completed. Using the DFS class the test for the Agent have now been passed thus completing the Agent class implementation