210CT Week 7 Coursework Tasks
Dr. Diana Hintea

LEARNING OUTCOMES

1. Understand the graph data structure, together with their traversal operations.

BASIC/INTERMEDIATE TASKS

1. Write the pseudocode for an unweighted graph data structure. You either use an adjacency matrix or an adjacency list approach. Also, write a function to add a new node and a function to add an edge. Following that, implement the graph you have designed in the programming language of your choice. You must use Object Oriented Programming for this task. You may use your own linked list from previous labs, the STL LL, or use a simple fixed size array (10 elements would be fine).
2. Implement BFS and DFS traversals for the above graph. Save the nodes traversed in sequence to a text file.

ADVANCED TASK

1. Implement the structure for an unweighted, undirected graph G, where nodes consist of positive integers. Also, implement a function isPath(v, w), where v and w are vertices in the graph, to check if there is a path between the two nodes. The path found will be printed to a text file as a sequence of integer numbers (the node values). 13. Using the graph structure previously implemented, implement a function isConnected(G) to check whether or not the graph is strongly connected. The output should be simply 'yes' or 'no‘.

READING

Felzenszwalb, P. and Zahib, R. (2010). Dynamic Programming and Graph Algorithms in Computer Vision.