5SENG003W Algorithms
Week 09 Tutorial Exercises: Graphs

These exercises cover: representations of graphs, DFS and BFS traversal algorithms.

You will also make use of David Galles' web-based *Data Structure Visualizations* tools. During its use it may help to slow the animation right down so that you can follow the steps.

Given the following definition of a *directed* graph G:

V = {0, 1, 2, 3, 4, 5, 6, 7}

E = {(0,1), (1,2), (1,3), (1,4), (2,5), (3,1), (3,5), (3,6), (4,6), (5,7), (6,3), (6,5), (6,7)}

Use the definition of **G** in the following exercises.

Exercise 1.

(a) Draw the *directed* graph G, using either pen and paper or e.g. Draw.io

(b) Create the **adjacency matrix** for graph G, using either pen and paper or a software tool.

(c) Create the **adjacency list** for graph G, using either pen and paper or a software tool.

Exercise 2.

Complete the Java or C++ program implementing matrix- and list-based graphs. This involves completing the functions addEdge and toString for both versions.

Exercise 3.

Use David Galles' visualisation tools of the DFS and BFS algorithms explore how they work in practice. They are available at:

https://www.cs.usfca.edu/~galles/visualization/DFS.html

https://www.cs.usfca.edu/~galles/visualization/BFS.html

Using either the matrix- or list-based implementations (or both if you have enough time), implement the BFS and DFS traversal algorithms and apply it to the graph **G** (using example1.txt). The implementation should print out the traversal paths.