# **Lab 7\_1 Graphs**

# **Week beginning Monday 13/11/2023**

We will look at code to implement an unweighted graph.

We have discussed two different implementations of a graph: adjacency matrix and adjacency lists. To allow for these two implementations we will use an interface Graph.

Both representations of a graph will have data fields numVertices (number of vertices) and directed (a boolean to indicate whether graph is directed) so will have an abstract class AbstractGraph with these two properties.

We will do both implementations: adjacency lists and adjacency matrix.

We have looked at code for MatrixGraph which is adjacency matrix implementation.

(1) Complete the code for this class. Use JUnit to test the methods. Also write a driver program to use/test this class.

(2) Then complete the code for class ListGraph - adjacency lists implementation.

An adjacency list representation of a graph uses an ArrayList of lists - one list for each vertex. We will use an ArrayList of LinkedLists.

Use JUnit to test the methods of ListGraph class. Also write a driver program to use/test this class.

Use the following graphs as examples to test the code:

**directed graph**

**undirected graph**