CURTIN UNIVERSITY (CRICOS number: 00301J)

Department of Computing, Faculty of Engineering and Science

Data Structures and Algorithms (COMP1002)

PRACTICAL 5 – GRAPHS

AIMS

- To create a general purpose Graph class
- To implement search algorithms in the Graph class

BEFORE THE PRACTICAL:

- Ensure you have completed the activities from previous practicals because this week will build on ideas and code developed in the previous classes
-

ACTIVITY 0: UML CLASS DIAGRAM

We will be using an Adjacency List implementation of graphs in this practical. Before you start coding, draw up the class diagrams for the DSAGraph and DSAGraphNode classes. Make sure to include any other Classes they make use of. Update this diagram as you work through the practical.

ACTIVITY 1: GRAPH IMPLEMENTATION

Now let's create a DSAGraph class using linked list to store the list of nodes, and linked lists within each node to store the adjacency list.

- Make sure you can display() the graph to the screen to help your testing
- Test it with a small graph at first, then display(), add more nodes/edges and display again.

ACTIVITY 2: READ GRAPH FROM FILE

Two input files for graphs have been uploaded to the Prac area. Create a FileIO class to read in the graph information and create a DSAGraph object. Work out manually what they should look like and check it against your display() output.

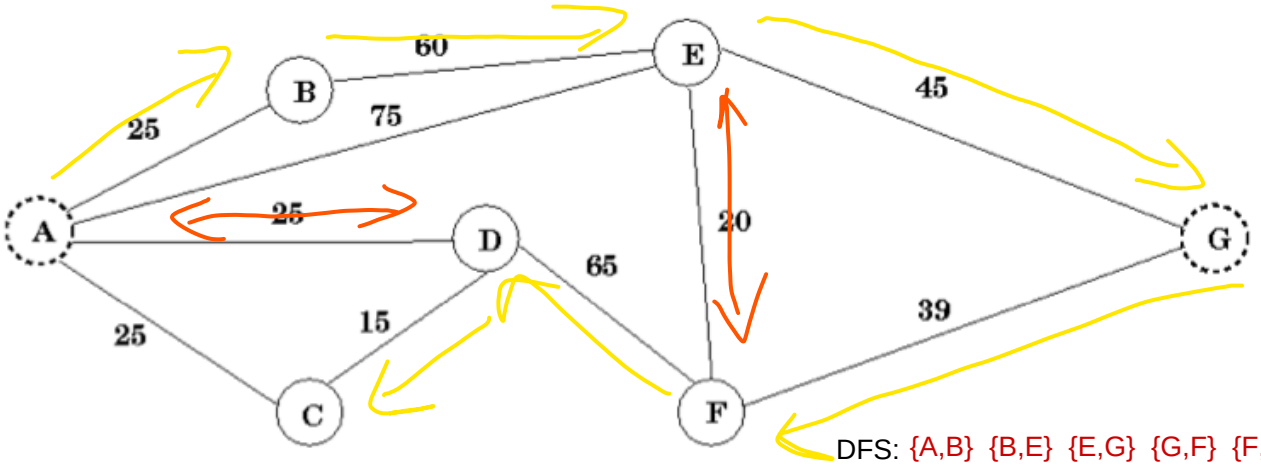
Format for the file is:

```
Edge_vertex1 Edge_vertex2 [Edge_weight]
Edge_vertex1 Edge_vertex2 [Edge_weight]
Edge_vertex1 Edge_vertex2 [Edge_weight]
```

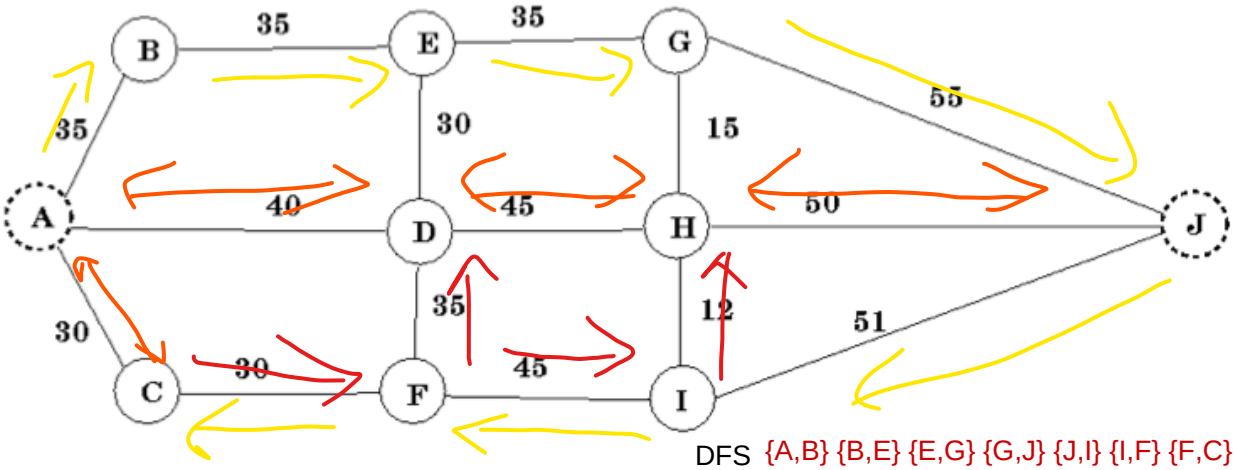
Where the Edge_weight is optional – ignore it for now.

ACTIVITY 3: MANUAL DEPTH FIRST SEARCH / BREADTH FIRST SEARCH

Consider the following graphs and carry out a breadth-first search and a depth-first search manually based on the algorithms in the lecture notes. (Assume they are sorted alphabetically - so you will choose vertices in alphabetical order)



DFS: {A,B} {B,E} {E,G} {G,F} {F,D} {D,C}



DFS {A,B} {B,E} {E,G} {G,J} {J,I} {I,F} {F,C} {F,D} {I,H}

ACTIVITY 4: IMPLEMENT DEPTH FIRST SEARCH / BREADTH FIRST SEARCH

Add methods for depthFirstSearch() and breadthFirstSearch() to your DSAGraph class. Test them against the graphs from Activity 2.

SUBMISSION DELIVERABLE:

Your UML and code (all that are required – see the marking guide) are due before the beginning of your next tutorial. **SUBMIT ELECTRONICALLY VIA BLACKBOARD**.

MARKING GUIDE

Your submission will be marked as follows:

- [2] UML diagram including all classes and “interesting” methods
- [2] Your DSAGraph is implemented properly – your test harness will show this.
- [2] You can read in a file and create a matching graph.
- [2] You have worked through the depth first search and breadth first search problems – submit a pdf, document or image file
- [2] You have implemented the depth-first search and breadth-first search