CSC 228 Data Structures and Algorithms, Fall 2018
## Instructor: Dr. Natarajan Meghanathan

Quiz 7: Breadth First Search on a Graph: Test for Bipartivity

Due: Nov. 15th, 2018; 11.59 PM (submission through Canvas)

The objective of this quiz is to determine whether an input graph is bipartite or not. We will do this by running the Breadth First Search (BFS) algorithm on the graph, starting from vertex 0. Assume the vertices are labeled with IDs ranging from 0 to N-1, where N is the number of vertices in the graph.

You are given the code for a graph implementation wherein the adjacency list of the graph is stored as an array of singly linked lists (the code for the List class and the Node class is given). The BFS algorithm is run using a queue that is in turn implemented as a dynamic array.
The graph is input as a set of edges (the format of the edge list txt file for a sample graph is shown below) and the number of vertices. A sample output (featuring the result of bipartite test and the level numbers of the vertices) is also shown.


```
0  1
0  2
1  2
1  3
2  4
3  4
3  5
4  5
4  6
5  6
```
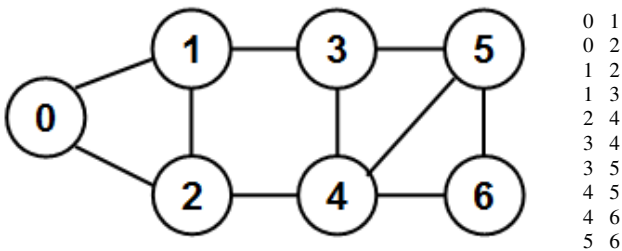
```
Enter the file name for the edges of the graph: graphEdges_1.txt
Enter number of nodes: 7
The graph is not bipartite
Level # of node 0 : 0
Level # of node 1 : 1
Level # of node 2 : 1
Level # of node 3 : 2
Level # of node 4 : 2
Level # of node 5 : 3
Level # of node 6 : 3
```

Your tasks in this quiz are as follows:

(a) Modify the code the BFS function to do two things: (1) It should determine the level numbers of the vertices, which is already setup as a member variable (as an array of integers); (2) It should determine whether the graph is bipartite or not and return a boolean (bool) accordingly.

(b) Extend the code in the main function to do two things: (1) Call the BFS function with '0' as the Start Node ID to determine whether the graph is bipartite or not and print the result; (2) After calling the BFS function and printing the result of the bipartite test, add a *for* loop to print the level numbers of the vertices. Note that it is expected that the level numbers of the vertices will be computed along with the test for bipartivity conducted by running BFS on the graph.

(c) Come up with two graphs of at least 10 vertices (with the degree of each vertex being at least 2 and at most 9): (1) A graph that is indeed bipartite and (2) another graph that is not bipartite. Draw both the graphs in your document with vertices labeled from 0 to N-1, where N is the number of vertices in the

graph. Manually, run the BFS algorithm on your two graphs starting from vertex '0' and determine the level numbers of the vertices, identify the tree edges and cross edges, and finally decide whether the graphs are bipartite or not. Show all the work in the document/quiz report.

(d) Develop the edgeList.txt file for each of the above two graphs (that you came up with for part-c) and input this to your code of parts a-b to print the level numbers of the vertices and whether the graph is bipartite or not. The output of your code should match with the results that you got by manually working on the two graphs for part-c.

What to submit (Submit as one Word or PDF document):
(1) The complete code for your quiz, encompassing parts-a and b.
(2) The test graphs that you came up with for part-c and the results of running BFS on the two graphs (see the description for part-c).
(3) The screenshots of your execution of the test graphs of part-c on the code (see the description for part-d).