CSC 228 Data Structures and Algorithms, Fall 2018
Instructor: Dr. Natarajan Meghanathan
Project 10: Degree Distribution of the Vertices in a Graph

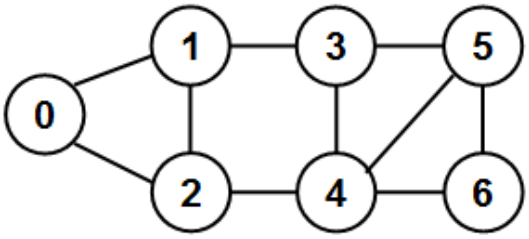
Due: Nov. 13th, 11.59 PM

You are given the code of graph implementation in which the information of the neighbors of the nodes (i.e., the adjacency list) is stored as an array of linked lists. The code for the List class, implemented as a Singly Linked List, is also given. Note that the implementation assumes the vertices are labeled from 0 to N-1, where N is the number of vertices in the graph.

Your tasks in this project are to: (i) add member functions to the List class and the Graph class so that we could find the degree (the number of neighbors) for any node in the graph (ii) after implementing the required member functions, add code in the main function to print the degrees of the vertices in the graph as well as compute and print the probability of finding vertices with a certain degree (ranging from 1 to N-1, where N is the number of vertices in the graph) and the average degree of the vertices in the given graph based on these probabilities (computed as the weighted average of the number of vertices with the possible degree values, as shown in the slides).

Test your code with a connected graph of 10 vertices or more such that the degree of any vertex is at least 1 and is at most N-1 (where N is the number of vertices in the graph).

The edgeList is passed as input to your program. The edges of a graph (edgeList) are stored as an ordered pair of vertices with the ID of the first vertex in the pair being always less than the ID of the second vertex in the pair. For example, the edgeList for the graph below is shown alongside.




```
0 1
0 2
1 2
1 3
2 4
3 4
3 5
4 5
4 6
5 6
```

A sample screenshot of the output for the above 7-vertex graph is shown below. Your code should generate a similar output for your test graph of 10 or more vertices.

```
Enter the file name for the edges of the graph: edgeList.txt
Enter number of nodes: 7

Degree of vertex 0 is 2
Degree of vertex 1 is 3
Degree of vertex 2 is 3
Degree of vertex 3 is 3
Degree of vertex 4 is 4
Degree of vertex 5 is 3
Degree of vertex 6 is 2

Probability of finding a vertex with degree 1 is 0
Probability of finding a vertex with degree 2 is 0.285714
Probability of finding a vertex with degree 3 is 0.571429
Probability of finding a vertex with degree 4 is 0.142857
Probability of finding a vertex with degree 5 is 0
Probability of finding a vertex with degree 6 is 0

Average Degree is 2.85714
```

What to submit (as one Word or PDF file, uploaded through Canvas):
(a) The complete code for the Graph, Node and List classes, including the implementation of the member functions to compute and retrieve the degree for any node in the graph.
(b) The complete code for the main function, including the computation of the probabilities of finding nodes with the different possible degree values and the average degree based on these probabilities.
(c) A diagram of the graph of 10 vertices or more (label the vertices with IDs 0 to N-1, where N is the number of vertices in the graph) and the manual computation of the probability of finding vertices with a certain degree (ranging from 1 to N-1, where N is the number of vertices in the graph) and the average degree based on these probabilities.
(d) Create an edge list of the graph of (c) and run it through your code of (a-b) and capture screenshots of the output for the degree of the vertices, probabilities of finding vertices with the different degree values and average degree based on these probabilities. The output should match with the values you obtained in (c).