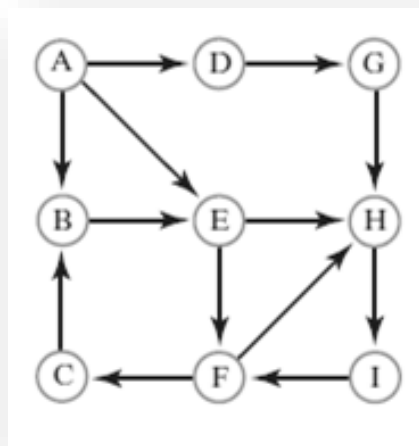


Assignment 9 (optional, +10 bonus points)

For this assignment, you are to modify the `UnweightedDirectedGraph.java` file attached as starter code and add a method to compute the **in-degree** and **out-degree** of each vertex in the graph. The in-degree of a vertex is the number of incoming edges to that vertex and the out-degree of a vertex is the number of outgoing edges from that vertex.

For example, in the directed graph shown below, The in-degree of vertex H is 3 (because there are three incoming edges to node H) and out-degree of vertex H is 1 (because there is only one outgoing edge from vertex H). Similarly, the in-degree of node A is 0 and its out-degree is 3.



What you need to do:

I have attached two files with this assignment:

- **Degree.java**: This is a class with two integer attributes: in-degree and out-degree used to store the in-degree and out-degree of a vertex, respectively.
- **UnweightedDirectedGraph.java** : This is a minimal code for an unweighted directed graph. The only attribute of this class is “adjacencyList” which is a `HashMap<LinkedList<V>>` where V is a parametric type for vertices in graph. For each vertex in the graph, the adjacencyList stores all the outgoing edges from that vertex. Please refer to the lectures and the practice problem in module 12 for more details on this adjacency list. The class also has an `addEdge` method which allows you to construct a graph by adding edges to it.

You need to implement the following method in the `UnweightedDirectedGraph.java`:

```
public Map<V, Degree> findInOutDegrees()
```

This method returns a Degree object for each vertex in the graph. The return type is a Map of <vertex, Degree> where the key is a vertex in graph and value is a Degree object containing the in-degree and out-degree for that vertex.

To test your method, you can make up a main method and create an instance of UnweightedDirectedGraph and add edges to it. Then call your findInOutDegrees method and print the map that is returned.

What you need to submit:

- Please submit your UnweightedDirectedGraph.java file containing your implementation of findInOutDegrees method.
- As before, you also need to submit a document explaining the efficiency of your method in terms of $|E|$ the number of edges in the graph, or $|V|$ the number of vertices in graph.

Grading Rubric:

Your method correctly and efficiently return the in-degree and out-degree for each vertex in the graph	7
Analysis of the running time of your algorithm	3
Total	10