HW 4 Problem 3

joelgrimaldi728

March 2021

1

My representation is a collection of edges as mentioned above. I create an array of nodes represented as strings and an array of edges represented as a String startNode and a String endNode showing the path of the node along with a String label which holds the weight of the edge. My advantage was simplicity in writing the code I think. I also think that the computational time of my code might be a little longer than the adjacency list and matrix because it is easier to access specific values. The checkRep especially takes a long time for me because I have multiple nested for loops to track down duplicate nodes and invalid edges.

2

Edge:
// Abstraction Function:
// Edge, e, represents a directed edge from
start(node) to end(node) with weight label:
//
// Representation Invariant for every Edge e:
// start != null
// end != null
//
// In other words:
// * If an edge is created it must have a starting
// and ending node and will have some weight

Graph:

```
// Abstraction Function:
// Graph, g, represents a graph that contains nodes and edges.
// Nodes are held in an array of Strings, nodes.
// Edges are held in an array of Edges, edges.
// If there are no nodes, g represents an empty graph.
// Otherwise g represents a graph containing node(s) and edge(s).
// Duplicate edges are allowed, duplicate node names are not.
//
// Representation Invariant for every Edge e:
// nodes != null && end != null &&
// if(nodes.length == 0):
// edges.length == 0; &&
// foreach i, 0 <= i < nodes.length:
//      foreach j, 0 <= j < nodes.length:
// nodes[i] != nodes[j]; &&
//      foreach i, 0 <= i < edges.length:
// edges[i].start() && edges[i].end must be in nodes
//
// In other words:
// * Array of strings representing nodes cannot be null
// * Array of edges cannot be null
// * There cannot be 2 nodes with the same name within a graph
// * There cannot be an edge if there are not nodes
```

3 q5

I did not need any extra tests as it satisfied the 75%.