Valid node = not null and not already in the collection

Valid edge = connects 2 valid nodes

Public Boolean addEdge(e Edge)

Requires: That the edge is valid and connects 2 valid nodes

Modifies: graph

Effects: if(e == null) throws invalidEdgeException

Else if(e.getFirstNode != valid || e.getSecondNode != valid) throws invalidNodeException

Else adds edge to collection and returns true

Public Boolean addNode(n Node)

Requires: The input Node is valid and doesn’t already exist in the collection

Modifies: graph

Effects: if(inputNode == null) throws invalidNodeException

Else if(n already exists in Node collection) throws duplicateNodeException

Else adds Node to collection and returns true

Public Edge findPath(start Node, destination Node)

Requires: both start and destination nodes are valid and a path exists between them

Modifies: Nothing

Effects: if(start != an existing node || destination != an existing node) throws invalidNodeException

Else if(no path exists between nodes) throws invalidPathException

Else returns a sequence of edges where following the chain of nodes that leads from start to destination