Graphs++ Technical Report

Ana Balcells, Ye Eun Jeong, Lucie Randall for CS230, Fall 2014

A. ADTs

Graphs: Adjacency Matrix implementation. Hold vertices in 1D array of T objects and edges in 2D boolean array representation.
Vectors: Holds sequences of nodes visited in a circuit or path.
LinkedList: Holds vertices in graph traversal and retrieving a vertex's successors.
Stack: Used in the algorithm for finding Hamiltonian circuits and paths.

B. Classes

MainPanel class:
Constructs the home panel that appears when program is opened.

InputPanel class:
Constructs the panel where graphs are created by adding nodes/edges or by file upload. This class is involved in the manipulation of the adjacency matrix which is being used in this project to represent our graph ADT.

OutputPanel class:
This class constructs the panel which will display the results of a Eulerian/Hamiltonian circuit/path search. It will confirm (or not) if there is a circuit/path, as well as explain why. If a graph traversal sequence exists, this will be displayed as well.

GraphPlusPlusGUI class:
Driver class that combines the 3 above panels and a GraphPlusPlus object. Each panel is an instance variable that can be retrieved through getter methods, for the purpose of switching panels when buttons are clicked. A GraphPlusPlus object is also an instance variable, and is passed into the panels as a constructor parameter.

GraphPlusPlus class:
Responsible for constructing the graphs created in the InputPanel, as well as running the calculations necessary to display the circuits/paths in OutputPanel. Uses Graphs, LinkedLists, Stacks, and Vectors.

C. Methods & Primary Functions

MainPanel:

- public static int getSource() - Determine which button clicked
- public void actionPerformed() - ButtonListener

InputPanel:

- public void buttonUpdate() - Reflects the user's choice of input for calculation: from custom graph or tgf, but not both.
- public static GraphPlusPlus getGraph() - Returns the graph object created in InputPanel for use in OutputPanel.
+ public void actionPerformed() - Included are several element-specific listeners for responding to user input.

OutputPanel:

- public void updateEC() - Updates the result labels when user chooses to calculate an Euler circuit.
- public void updateEP() - Updates the result labels when user chooses to calculate an Euler path.
- public void updateHC() - Updates the result labels when user chooses to calculate a Hamiltonian circuit.
- public void updateHP() - Updates the result labels when user chooses to calculate a Hamiltonian path.
- public void actionPerformed() - Listener for switchPanel to return to MainPanel.

GraphPlusPlusGUI:

- public static MainPanel getMainPanel() - Retrieves MainPanel
- public static InputPanel getInputPanel() - Retrieves InputPanel
- public static OutputPanel getOutputPanel() - Retrieves OutputPanel
- public static void main() - Creates the three panels and displays them
- public static void switchPanel() - Allows the program to traverse panels in a set sequence by changing panel contents, rather than showing 3 tabbed panes.

GraphPlusPlus:

See individual method comments in the .java file for details and algorithm explanations.

- public int vertexDegree() - Returns an int value of the degree of a given vertex
- public boolean isAllEvenDegrees() - Checks to make sure that all vertices in a graph have even degrees. Returns false if all are not even.

- public boolean isEulerian()- Checks the two conditions required for a graph to be eulerian: whether it is connected and all vertex degrees are even.
- public boolean hasTwoOddDegrees() - Checks whether a graph has exactly two vertices of odd degrees. Used to check whether a graph has an euler path.
- public boolean hasEulerPath() - Checks whether a graph is connected and has exactly two vertices of odd degree.
- private void traverseEdges() - Recursive helper method used to find an euler circuit.
- public Vector<T> getEulerCircuit() - Creates a vector with a node path followed to complete an Euler circuit. If not Eulerian, returns null.
- private void traverseEdgesEP() - Recursive helper method used to find an euler path.
- public Vector<T> getEulerPath() - Finds an euler path and returns a vector that contains a node sequence of an euler path. Returns null if a path is not found.
- public boolean isConnected() - Uses dfsTraversal to make a LinkedList of all nodes in a single component. Compares the list length to the number of total nodes in the graph. If these numbers do not match, graph is not complete and returns false.
- public LinkedList<T> dfsTraversal() - Returns a LinkedList containing a depth-first search traversal of the graph starting at the given vertex. The resulting list should contain all vertices visited during the traversal in the order they were visited.
- private void traverseVertices() - Recursive helper method used to find a hamiltonian circuit.
- private void traverseVerticesHP() - Recursive helper method used to find a hamiltonian path.
- public Vector<T> getHamiltonianCircuit() - Calculates a hamiltonian circuit and returns a vector containing the node sequence of the circuit.
- public Vector<T> getHamiltonianPath() - Finds a hamiltonian path and returns a vector that contains a node sequence of a hamiltonian path.