Graph GUI Help

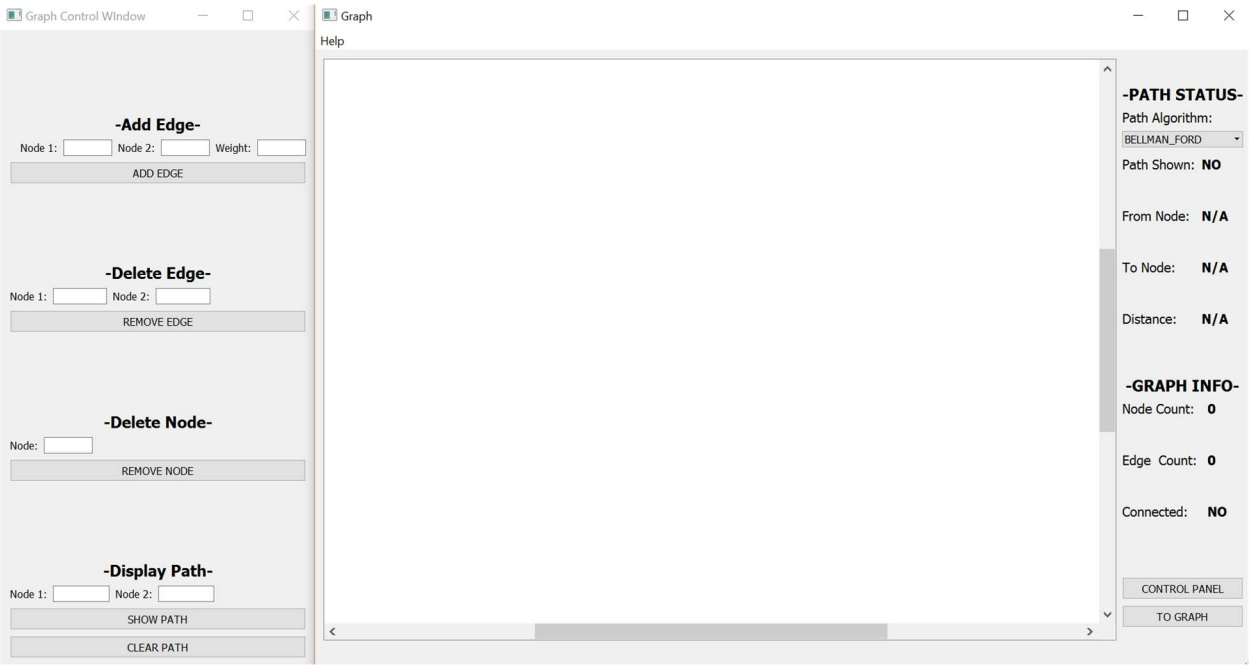

Figure 1: *Graph GUI Layout*

LAYOUT:

When first opening the project two windows appear. The left window is a control panel and the right is split into two sections, one empty white section for the graph itself and another section for displaying data about the graph.

Control Panel

The control panel on the left allows the user to edit numerous elements of the graph, however it is not necessary for controlling the graph as all the actions on the panel can be implemented using key commands in the graph area.

Graph Area

The white graph area is where the graphical representation of the graph appears. Nodes and edges are added to and deleted from this area. The user interacts directly with this window in order to add nodes and all other operations can be performed using special key commands within the graph area.

Graph Data Panel

The panel to the far right is the data panel. It contains data about the graph currently being displayed including the number of nodes and edges in the graph as well as whether the graph is connected. The panel also contains information about the current graph algorithm being utilized. The combo box in the panel indicates the current algorithm being used and the user can switch algorithms by selecting different options from the box. The section titled **-PATH STATUS-** shows some path information such as whether or not a path is currently displayed, what nodes the path connects if applicable and the length of the shown path. Close to the bottom of the panel is a button labeled **CONTROL PANEL** that can be used to open the control panel to the left if it was previously closed. Finally, the last item on the panel is a button that is either labeled **TO GRAPH**

or **TO DIGRAPH**. Hitting this buttons changes the type of graph being used between an undirected graph and a directed one.

GRAPH COMMANDS:

Add Node

To add a node simply left click somewhere within the white graph area. After left clicking, a dialog box will appear requesting the name of the node. Input a node name between 1 and 4 characters in the dialog's text area and either press OK in the box or simply hit ENTER on the keyboard. An orange node will now appear at the location that was originally clicked with the input name.

Figure 2: *Node*

Select/Deselect Node

Some commands require that nodes are selected. To select a node simply move the cursor over an existing node and right click on the mouse. When the node is selected it will turn red. To deselect a node simply right click over a selected node and it will turn back to orange.

Figure 3: *Selected node*

Add Edge

Control Panel: To add an edge from the control panel go to the text boxes directly under the label **-Add Edge-**. In the first two text boxes input the names of the existing nodes that the added edge will connect. In the third box input the numerical weight value of the desired edge. To add the described edge click the **ADD EDGE** button.

Key Command: To add an edge without the control panel simply select two nodes in the graph by right clicking over them and then hit ENTER. A dialog will appear requesting a numerical weight value. Input a number and either click OK or hit ENTER again. An edge of the given weight will then appear between the selected nodes.

Figure 4: *Directed edge from* **X** *to* **Y**

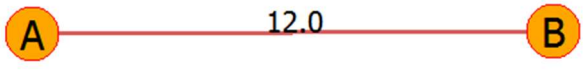

Figure 5: *Undirected edge between* **A** *and* **B**

Remove Node

Control Panel: To remove a node from the control panel go to the text box directly under the label **-Delete Node-**. Input the name of the node to be deleted and then click the **REMOVE NODE** button.

Key Command: To remove a node without the control panel simply right click select all the nodes to be deleted and hit the BACKSPACE key. The nodes and any connected edges will then be deleted.

Remove Edge

Control Panel: To remove an edge from the control panel go to the text boxes directly under the label **-Delete Edge-**. In the two text boxes input the names of the existing nodes between which the edge exists. To remove the described edge click the **REMOVE EDGE** button.

Key Command: To remove an edge without the control panel simply select two nodes in the graph by right clicking over them and then hit the DELETE button on the keyboard. As long as an edge existed between the two selected nodes it will be removed from the graph.

Display Path

To display a path using a path algorithm, first make sure that the algorithm shown in the combo box in the rightmost panel is the desired algorithm. The default algorithm selected is DIJKSTRA's shortest path algorithm. Click the combo box to view and select another algorithm.

Control Panel: When the desired algorithm is selected from the right panel go to the control panel underneath the **-Display Path-** label. If the selected algorithm requires node references input the node values into the text boxes. To display a path using the algorithm, click the **SHOW PATH** button. If graph conditions are valid a path will then be displayed by coloring the edges and nodes of the discovered path green.

Key Command: To display a path without the control panel simply right click select the nodes in the graph that are required for the algorithm if selected nodes are needed. Hit the UP ARROW button on the keyboard and a highlighted green path will be displayed if the graph conditions are valid.

Clear Path

Control Panel: To clear a highlighted path simply click the **CLEAR PATH** button on the control panel.

Key Command: To clear a path without the control panel simply hit the DOWN ARROW key on the keyboard.

DIRECTED AND UNDIRECTED GRAPHS:

In a directed graph, or digraph, an edge between 2 nodes has an associated direction meaning it can be traversed from one way and not the other, whereas in an undirected graph, or simply a graph, edges between

nodes have no direction and can be traversed from either side. This application supports both graphs and digraphs.

The default setting is to a digraph, meaning edges are directed. To change back and forth between graph types, simply hit the bottom button on the far right panel labeled either **TO GRAPH** or **TO DIGRAPH** depending on the current setting.

2 Edges between 2 Nodes

Unlike in an undirected graph, directed graphs use arrows to indicate in what direction an edge is going, as shown in Figure 4 above. When a directed edge goes from one node to another and a different edge goes between the same nodes but in the opposite direction a single edge with two arrows is shown and each direction can have a different associated weight.

Figure 6: *Directed edges from* **X** *to* **Y** *and* **Y** *to* **X**

As is indicated by Figure 6, in this scenario there are two weights shown, one for each direction. The weight values in a digraph are always followed by a carrot symbol to indicate towards which node value this edge is directed; to make things clearer still, the weight always appears closer to the node at which it is directed.

Highlighted Edges

When a path is being displayed on a digraph the edge and arrow are both highlighted. If two directed edges are between nodes and a highlighted path goes through one of the edges only the arrowhead in the direction of the path is highlighted.

Figure 7: *Highlighted path from* **X** *to* **Y**

Alternatively, with an undirected graph there can only be one edge between two nodes and therefore only one weight that applies to both directions. For this reason when an edge of a graph is highlighted it indicates a valid path in either direction.

Working with Directed Edges

Because the direction of edges matters, the way nodes are selected is important in digraphs. The direction of an edge added to or deleted from a graph is determined by the node you select first or the node input into the leftmost text box on the control panel. The edge is always viewed as being **from** the first node selected **to** the second node selected whether adding or deleting an edge. Likewise, when finding a path, the path searched for will be from the first node selected to the second node selected.

GRAPH ALGORITHMS:

Dijkstra

Dijkstra's algorithm can be used to find the single source shortest path from a single node to all other nodes in the graph that can be reached from the selected node. Dijkstra is implemented in this program to only find the shortest path between two selected nodes. To find the shortest path between two nodes using Dijkstra, first ensure DIJKSTRA is selected in the combo box. Select two nodes by right clicking nodes or typing node values into the text boxes below the **-Display Path-** label. When the UP ARROW is hit on the keyboard or the **SHOW PATH** button is pressed on the control panel a path with the lowest possible weight value will appear connecting the two selected nodes.

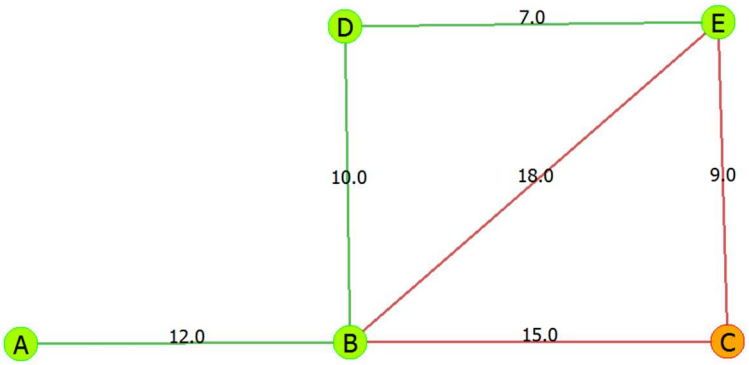

Figure 6: *Dijkstra between nodes* **A** *and* **E**

REQUIREMENTS: Two selected nodes must be connected by edges and weights of the connecting edges must be nonnegative.

Bellman Ford

Like Dijkstra, Bellman Ford's algorithm is used to find the single source shortest path to other nodes in the graph, however this algorithm can support negative edges in directed graphs. If a digraph contains negative edges Dijkstra's algorithm does not function properly, however Bellman Ford will continue to work as long as there are no negative weight cycles in the digraph (cycles in a graph for which the sum of the edge weights add to a negative value). To find the shortest path between two nodes using Bellman Fords, first ensure BELLMAN FORD is selected in the combo box. Select two nodes by right clicking nodes or typing node values into the text boxes below the **-Display Path-** label. When the UP ARROW is hit on the keyboard or the **SHOW PATH** button is pressed on the control panel a path with the lowest possible weight value will appear connecting the two selected nodes.

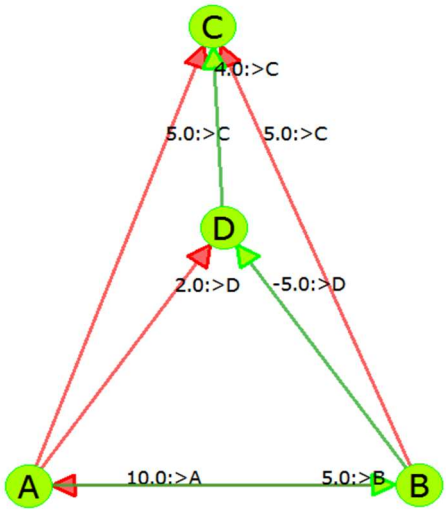

Figure 9: *Bellman Ford*

REQUIREMNTS: Two selected nodes must be connected by edges and graph cannot contain a negative cycle.

Prim's

Prim's algorithm finds the Minimum Spanning Tree of a graph or the path through the graph that touches all the nodes by traversing the edges with the lowest possible weights. To find the MST of the graph using Prim's, first ensure PRIMS is selected in the algorithm combo box and then either hit the UP ARROW on the keyboard or the **SHOW PATH** button on the control panel. All the nodes in the graph will then be highlighted and the least costly edges connecting those nodes will also be highlighted.

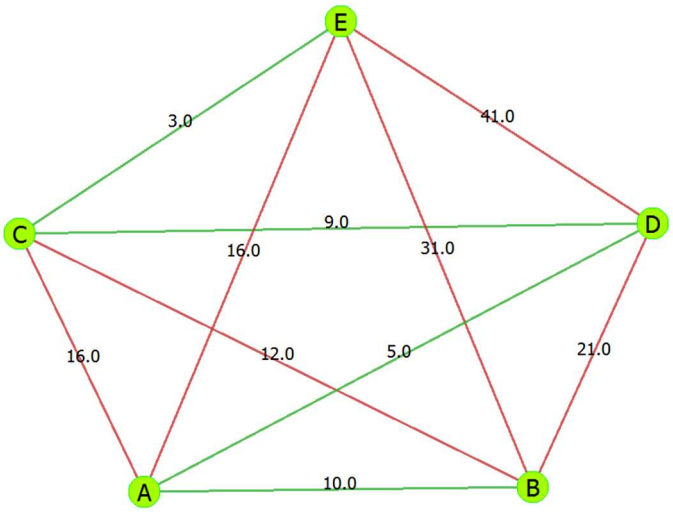

Figure 7: *Prim's*

REQUIREMENTS: Prim's will only work if the graph in the graph window is connected.