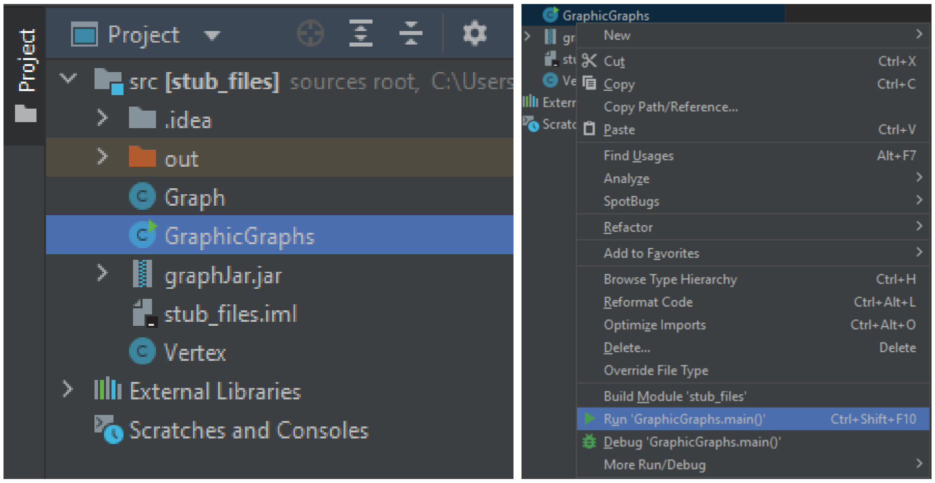
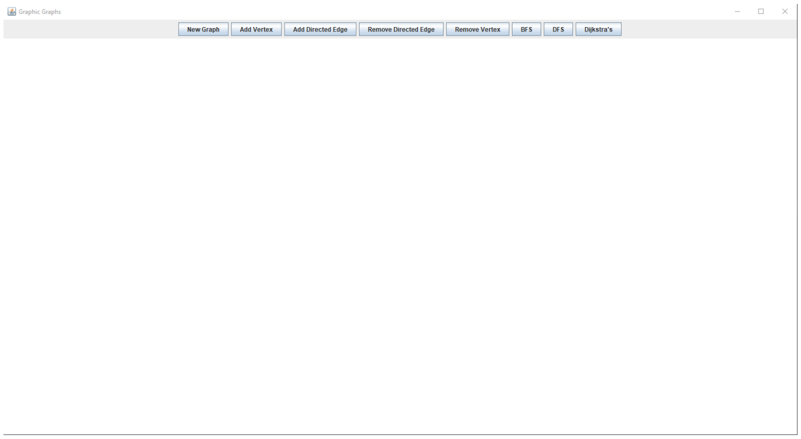
Graphic Graphs

Getting Started

To run the program, unzip the files and open up the src folder within the Graphic graphs folder. Then, right click on GraphicGraphs and run its main method.

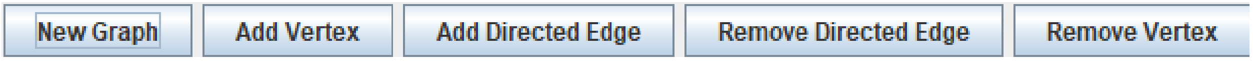


The following screen should pop up:



Building Your Own Graph

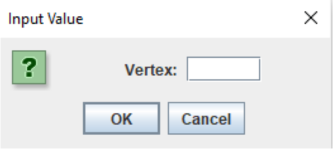
To construct your very own graph, the following 5 buttons will be your best friends:



New Graph:

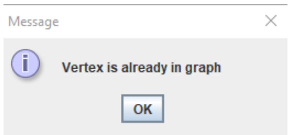
This button will clear your current graph. Nothing too fancy here.

Add Vertex:
This button will prompt you to give a label for the vertex you want to add.



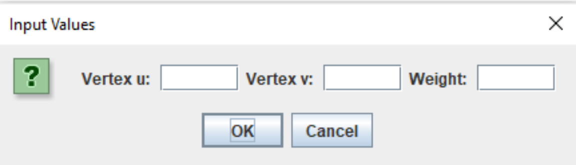
If there is no vertex in the graph with that label, then you have successfully added a vertex. For example, adding a vertex with the label “a” will look like the following:

If there is already a vertex in the graph with that label, then the following window will appear:

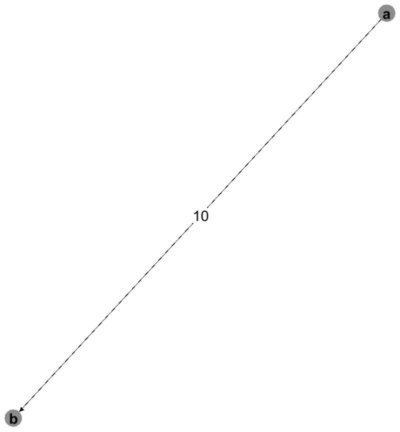


Vertex is already in graph

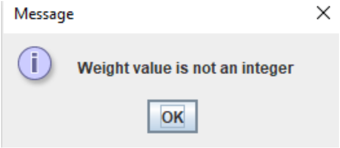
Add Directed Edge
This button will prompt you to give inputs via the following pop up:



Given proper inputs, the program will add an edge from vertex “u” to vertex “v” with the weight given, adding the vertices if they aren’t already in the graph. For example, inputting “a”, “b”, and 10 will add the following edge:

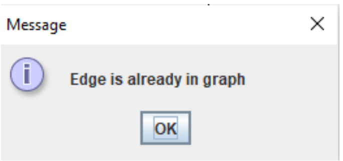


If the weight of the edge is not an integer, then the following window will show up:



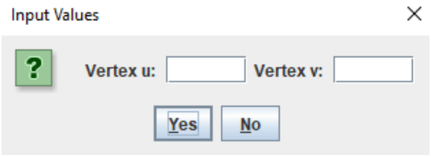
Weight value is not an integer

If there is already an edge between u and v, no matter what weight is inputted (given that it is an integer), then the following window will show up:



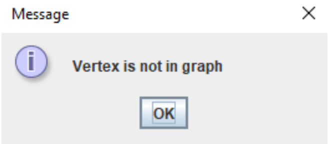
Edge is already in graph

Remove Directed Edge
This button will prompt you to give inputs via the following pop up:



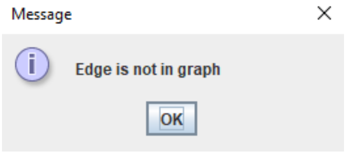
If the directed edge exists, the edge is removed, but not the vertices. For instance, removing the “a” to “b” edge in the above example would just leave “a” and “b” in the graph.

If an inputted vertex does not exist in the graph, then the following window pops up:



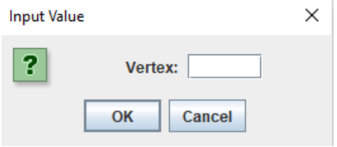
Vertex is not in graph

If the vertices exist, but the directed edge does not, then the following window pops up:

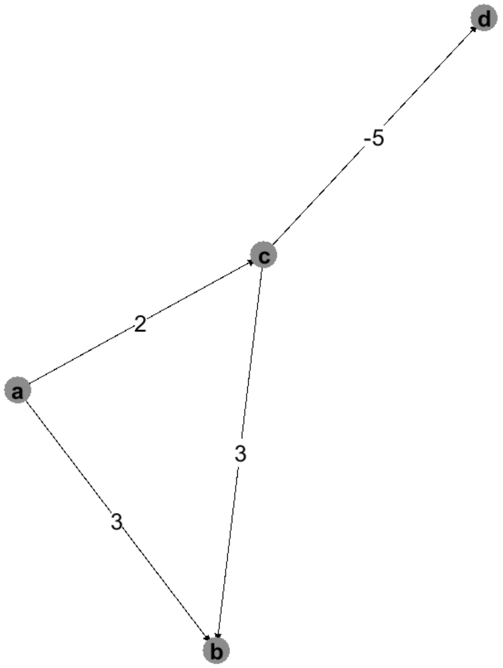


Edge is not in graph

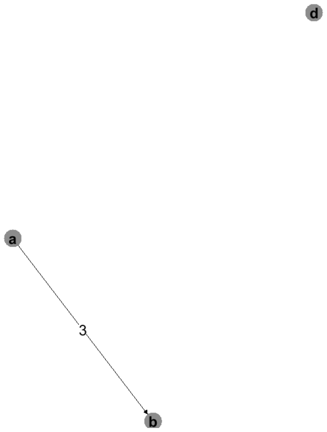
Remove Vertex
This button will prompt you to give a label for the vertex you want to remove.



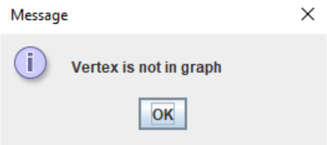
If the vertex exists, then it will be removed along with its incoming and outgoing edges. For example, consider the following graph:



Removing “c” will result in this new graph:



If the vertex does not exist, then the following window pops up

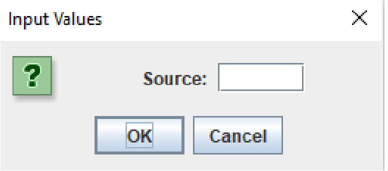


Vertex is not in graph

Graph Algorithms

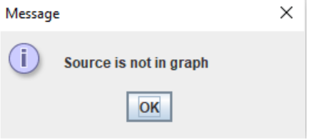
Now that you know how to build your own graph, you can call Breadth-First Search (BFS), Depth-First Search (DFS), or Dijkstra’s algorithm.

Clicking any of these three buttons will bring up the following window for the vertex you want to start the algorithm at.



Source:

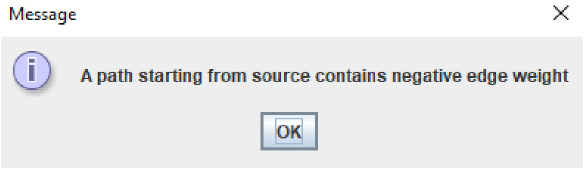
If the source vertex is not in the graph, the following window will pop up:



Source is not in graph

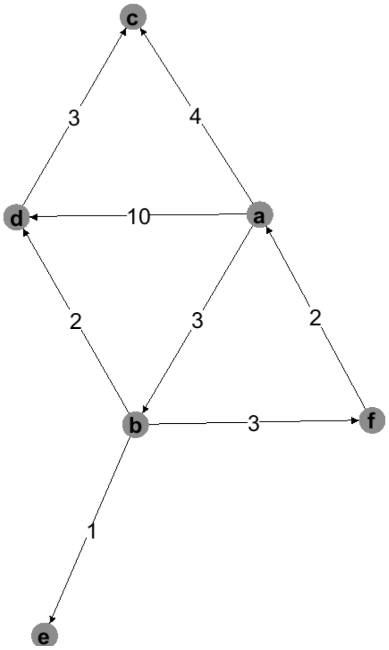
Clicking the button will run the appropriate algorithm on the current graph and display its corresponding tree/forest. If the algorithm is DFS, then the result will show the start and finish times. If the algorithm is Dijkstra's, then the result will show the edge weights.

Note that if Dijkstra's finds a negative edge weight during the algorithm, the following message will be displayed since Dijkstra's only works for positive edge weights:

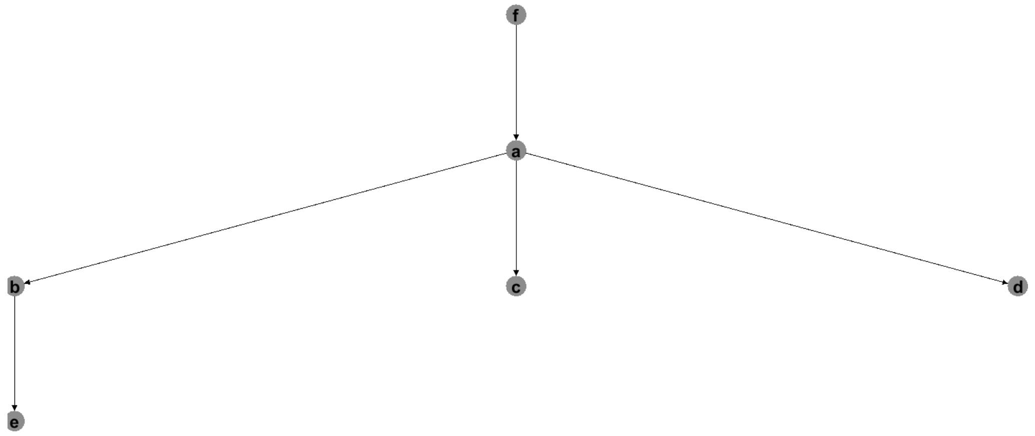


A path starting from source contains negative edge weight

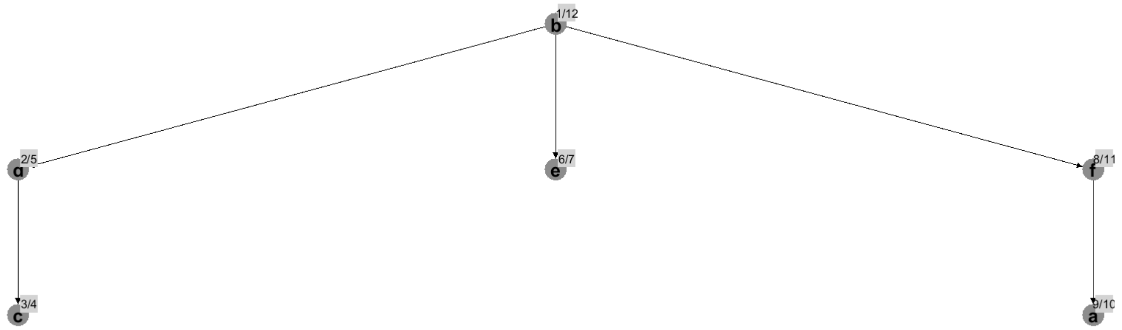
For example, consider the following graph.



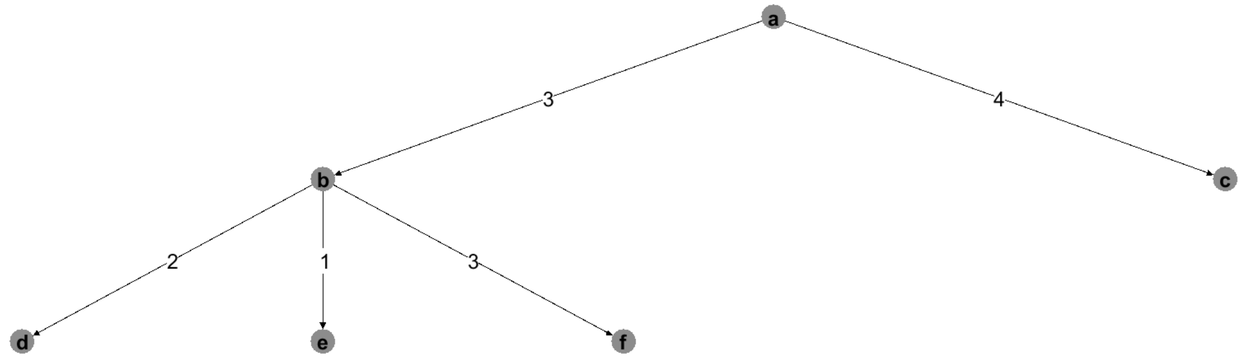
Running BFS starting at “f” will result in the following graph:



Running DFS starting at “b” will result in the following graph:



Running Dijkstra’s starting at “a” will result in the following graph:



Final Note

If the nodes in the graph overlap, you can drag them using your mouse to make it more visible. Thank you for reading and have fun with Graphic Graphs!