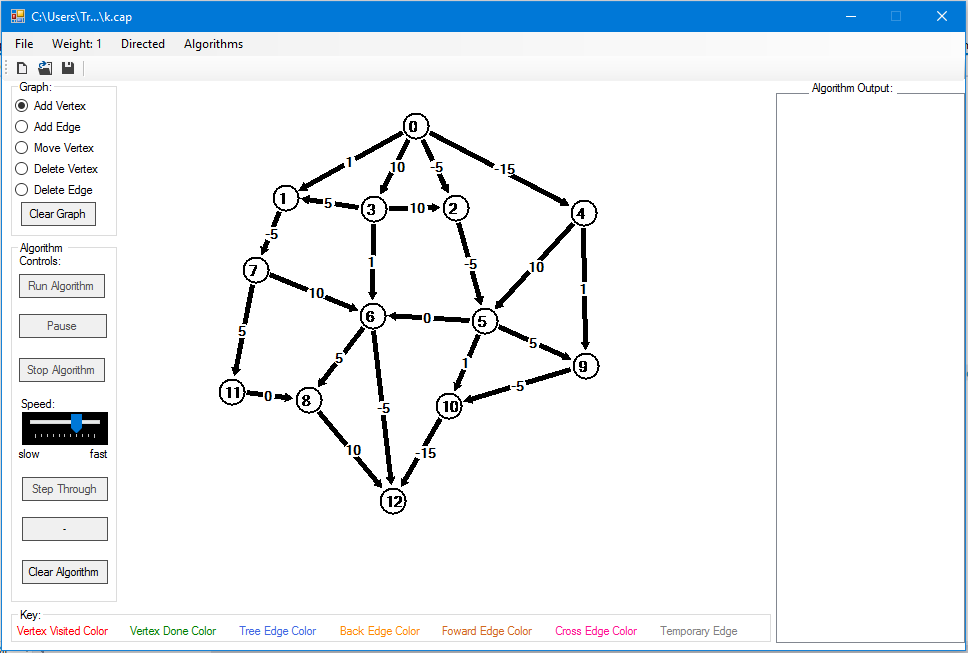
Visualizing Graph Algorithms

User Manual



1. How to get started:

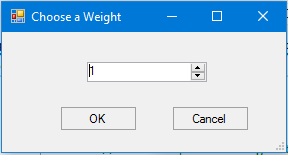
* Download the .exe file labeled “Run Project” on website
* Double click on download and the project should open
* Ignore warning about MRU.txt missing – it will still work and create that file on its own.

1. How to open a saved graph:

* Click on one of the following:
  + Open icon in the menu strip
    - Normal open dialog box will open
    - Choose a .cap file
  + Open under File
    - Normal open dialog box will open
    - Choose a .cap file
  + Open Recent
    - Shows files (graphs) that have been recently used – or none if none have been saved
    - Click on the one you want to open

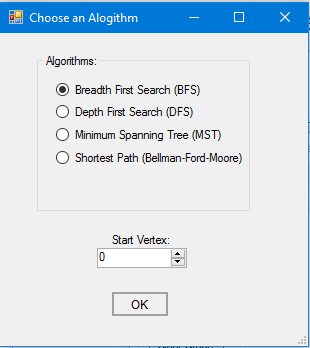


1. How to draw a vertex:
   * Make sure the “Add Vertex” radio button is checked
   * Click on the screen where you want the vertex to be drawn
     + Note: You are not allowed to draw near the edges of the window or on top of an existing vertex
2. How to draw an edge:
   * Make sure the “Add Edge” radio button is checked
   * The graph is defaulted to directed edges. By clicking on directed in the menu strip, you can change the edge to non-directed.
     + By changing the edge type, all the edges that exist will change. If directed edges were previously drawn, they will now go in both directions.
     + Clicking on it when it says non-directed will result in the edges becoming directed.
   * The default edge weight is set to 1. To change the edge weight:
     + Click on “Weight” in the menu
     + A dialog box will pop up where you can change the weight to any number between -99 and 99
     + Press OK and the edge weight will be whatever number was selected



* + To actually draw the edge, click down on the starting vertex and drag the mouse to the vertex you want the edge going to.
    - The edge will appear in gray as you are drawing it.
    - Mouse down and mouse up have to happen near a vertex (and different vertices) for a line to be drawn.

1. How to move a vertex:
   * Make sure the “Move Vertex” radio button is checked
   * Click on the vertex you want to move and drag it with the mouse
     + Note: Vertex cannot be dragged on top of another vertex or too close to the edges of the window
2. How to delete a vertex:
   * Make sure the “Delete Vertex” radio button is checked
   * Click on the vertex you wish to delete
3. How to delete an edge:
   * Make sure the “Delete Edge” radio button is checked
   * Click on the edge you would like to delete
     + Note: If you click near the intersection of two edges, it will delete the edge it finds first when going through the two-dimensional array.
4. How to save a graph:
   * Click on save in File menu or in menu strip.
     + If graph was not previously saved, Save As box will pop up and save like normal but as a .cap file.
     + Otherwise the graph will just save under its previous name.
   * Click on save as in File menu
     + Save As box will pop up and save like normal but as a .cap file
5. How to start a new graph:
   * If you just started the program, this is pointless.
   * Click on New in either the menu strip or the File menu
     + A box may pop up asking if you wish to save your current graph. Choose the option you feel is best.
     + Everything will be completely cleared and ready for you to draw a new graph.
6. How to select an algorithm:
   * Click on Algorithms in the menu strip
   * Choose the algorithm you wish to see – Breadth First Search is the default.
   * Select a starting vertex – 0 is the default.
   * Press OK



1. How to run an algorithm:
   * This only works if an algorithm is selected first. (See “How to select an algorithm” above) The button will become enabled.
   * Click on the “Run Algorithm” button
   * Note that changes to the graph cannot be made while an algorithm is running.
   * The Speed track bar controls the speed at which the algorithm runs.
     + Move left for slower and right for faster.
   * If the “Pause” button is clicked, the algorithm will pause where it is.
   * If “Continue” is clicked, the algorithm will continue where it left off.
   * If the “Stop Algorithm” button is clicked, the algorithm is completely stopped and will have to be restarted. However, this does enable changes to the graph.
2. How to step through an algorithm:
   * This only works if an algorithm is selected first. (See “How to select an algorithm” above) The button will become enabled.
   * Start by clicking on the “Step Through” button. The text of this button should then change to say “Next Step”.
   * To go forward, click the button that says “Next Step”.
   * To go backwards, click the button that says “Step Back”.
     + Note: If stepping back is not possible, the text of this button is “-”.
   * At the end of the algorithm, the forward button will say “End – Clear”. Clicking on it will clear the algorithm.
3. How to clear an algorithm:
   * Click on the button that says “Clear Algorithm”
   * It is also clears when a new algorithm is selected, an algorithm starts over, or a change to the graph was made.
4. How to exit:
   * Click on either Exit in the File menu or the “x” in the upper right hand corner.
   * If changes were made to a graph, a box will pop up asking if you wish to save. Choose the option you feel is best.

Advanced – Need entire program and knowledge of C#

* Change minimum and maximum of edge weights:
  + Go into the frmChooseWeight.cs [Design]
  + Click on the Numeric Up-Down box
  + Go under Properties
  + Find the minimum and maximum and change them to desired values
* Add an algorithm:
  + Add new algorithm choice to frmChooseAlgorithm.cs [Design] so it can be selected.
  + In main, add it in the “if else”s of onRunAlgorithmClick and onNextStepClick as the enumerator to run.
  + Add the algorithm itself to the bottom of MyDocument.cs.
    - All functions are IEnumerable<StepResult>
    - Data structure is an adjacency matrix
    - Variables may need to be added and declared depending on what the algorithm needs
    - For each step you want shown, return int parent (parent vertex), int vertex (current vertex), VertexState state (current state of vertex), VertexState prevState (state of vertex before step occurred), string edgeType (type of edge between parent and current vertex), string prevEdgeType(type of edge between parent and current vertex before step occurred), string output (string to be displayed in algorithm output text box), List<OtherEdge> otherEdges = null (list of edges – very helpful with the temporary edges in MST) – Some of these are to make stepping backwards easier.