By Olena Chernilevska

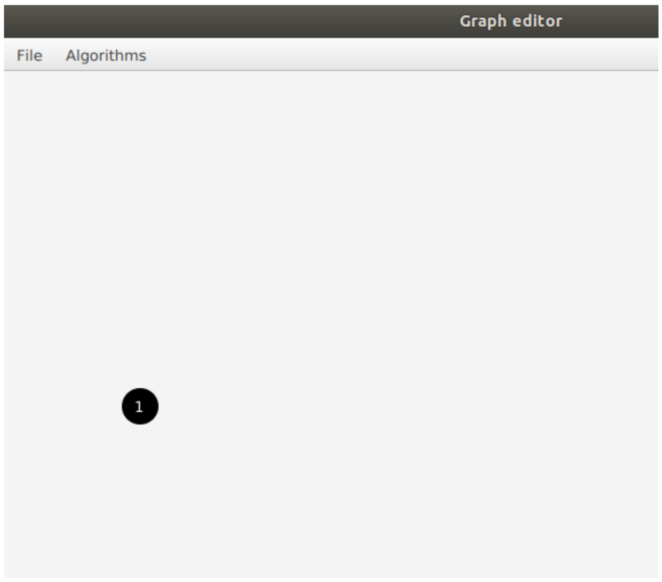
1) Instructions to run code

To run the program you need to type `mvn clean javafx:run` in the console. After that you can regenerate javadocs, if necessary. To do it, please, type `mvn javadoc:javadoc` in the console.

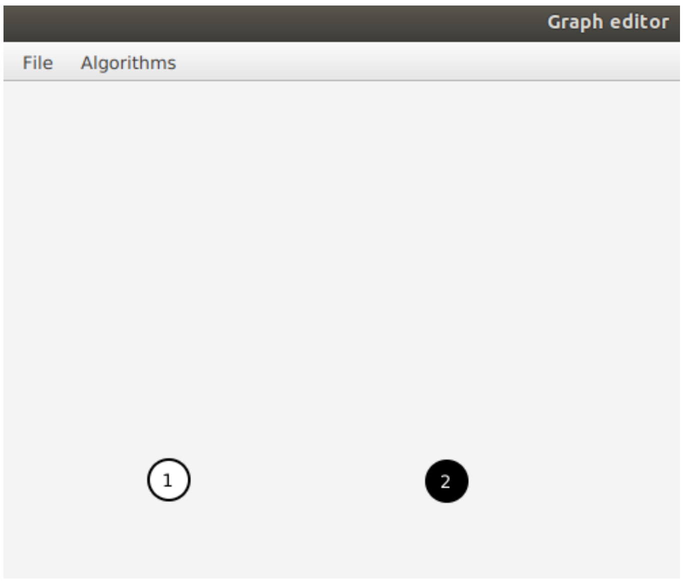
2) Short summary

The main use cases of the program:

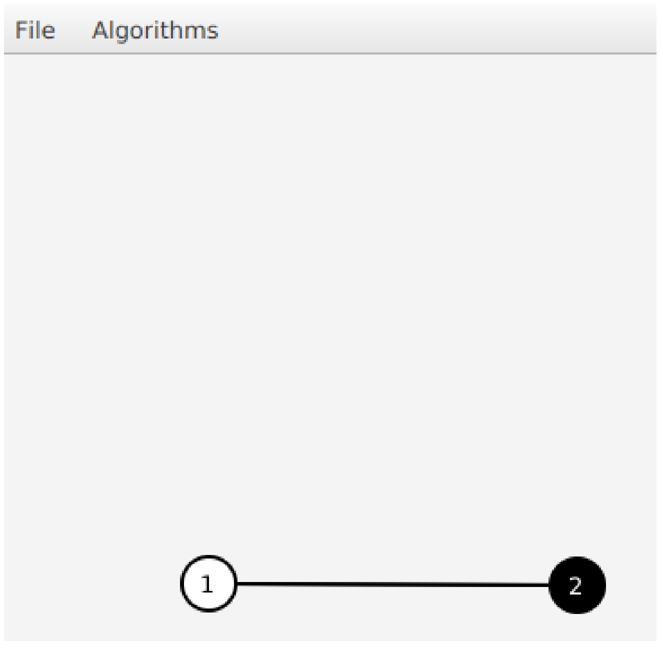
1) adding new vertex. The user can create a new vertex by holding down Shift and clicking in the window;



2) selecting vertex. The user can click any vertex to select it. Newly added vertex is auto selected. If the user clicks in the window outside any vertices, then the selection is cleared;



3) adding/deleting new edges. The user can hold down Control and click any vertex V to create an edge between the selected vertex and V. If there is already an edge between the selected vertex and V, then this action removes that edge;



4) dragging vertex. The user can click any vertex and drag it with the mouse to change its position;

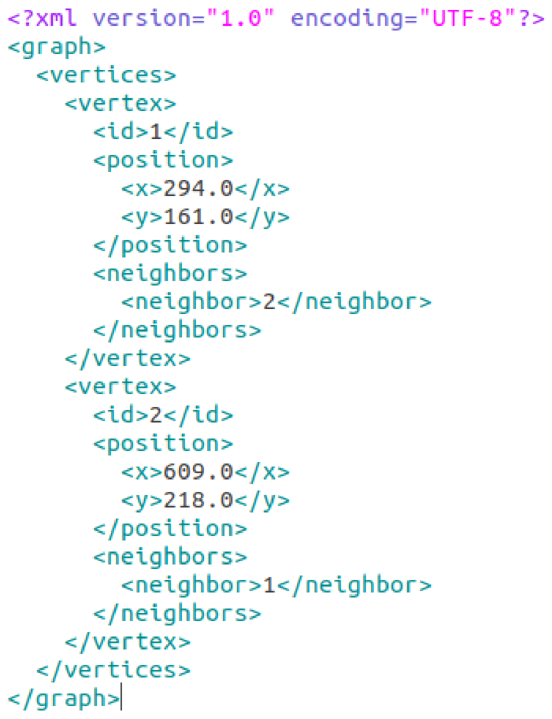
5) deleting vertex. The user can press the Delete key to delete the selected vertex with all edges that are attached to it.

Additionally, the program has a menu bar with two options: File and Algorithms.

The File menu contains the following menu items:

1) New (if clicked it clears all information about current graph and creates a new one);

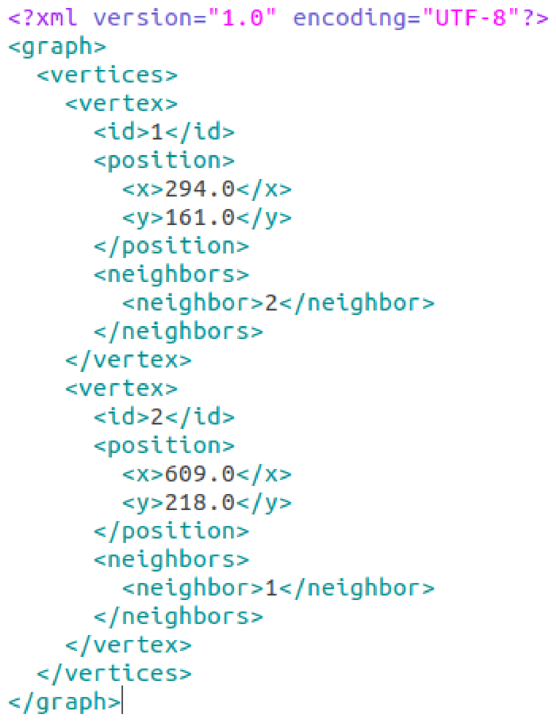
2) Open (if clicked prompts the user for a filename, then reads the graph from that file and displays it on the canvas). The file should have .xml format and display information about graph like:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<graph>
  <vertices>
    <vertex>
      <id>1</id>
      <position>
        <x>294.0</x>
        <y>161.0</y>
      </position>
      <neighbors>
        <neighbor>2</neighbor>
      </neighbors>
    </vertex>
    <vertex>
      <id>2</id>
      <position>
        <x>609.0</x>
        <y>218.0</y>
      </position>
      <neighbors>
        <neighbor>1</neighbor>
      </neighbors>
    </vertex>
  </vertices>
</graph>
```

where **id** indicates the unique identifier of vertex (always Integer), **x** and **y** - coordinates of vertex (Double), and **neighbors** - the ids of adjacent vertices;

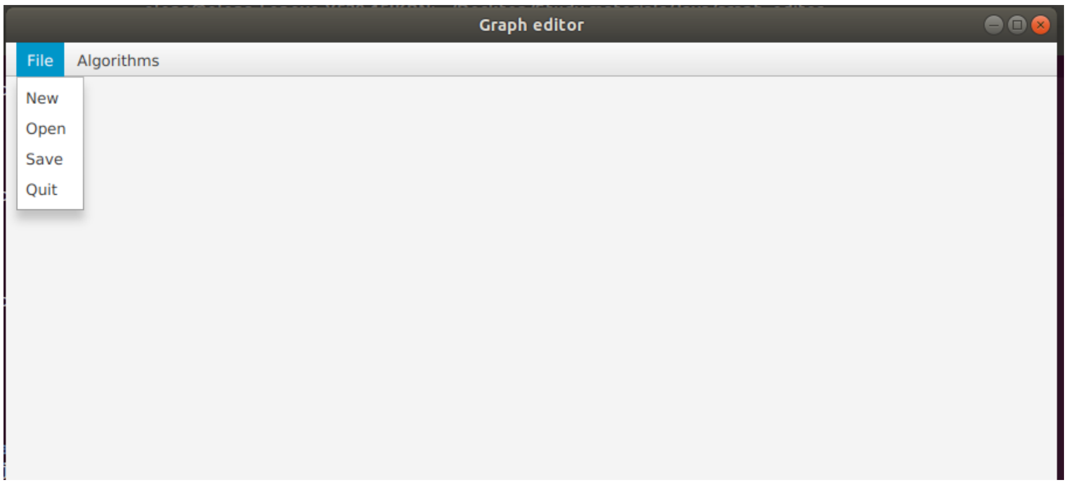
3) Save (if clicked prompts the user for a filename, then saves the current graph to that file). The file should have .xml format and it will display information about graph like:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<graph>
  <vertices>
    <vertex>
      <id>1</id>
      <position>
        <x>294.0</x>
        <y>161.0</y>
      </position>
      <neighbors>
        <neighbor>2</neighbor>
      </neighbors>
    </vertex>
    <vertex>
      <id>2</id>
      <position>
        <x>609.0</x>
        <y>218.0</y>
      </position>
      <neighbors>
        <neighbor>1</neighbor>
      </neighbors>
    </vertex>
  </vertices>
</graph>
```

where **id** indicates the unique identifier of vertex (always Integer), **x** and **y** - coordinates of vertex (Double), and **neighbors** - the ids of adjacent vertices;

4) Quit (if clicked exits the application).



The Algorithms menu the following menu items:

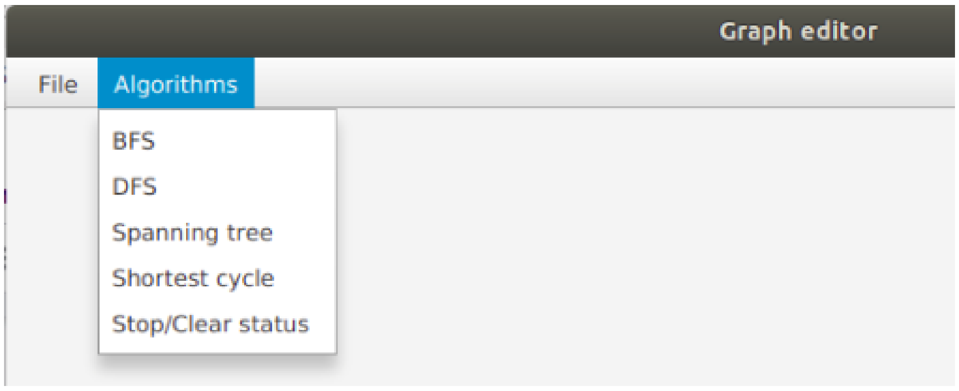
1) BFS (if clicked it runs BFS algorithm and displays the found result);

2) DFS (if clicked it runs DFS algorithm and displays the found result);

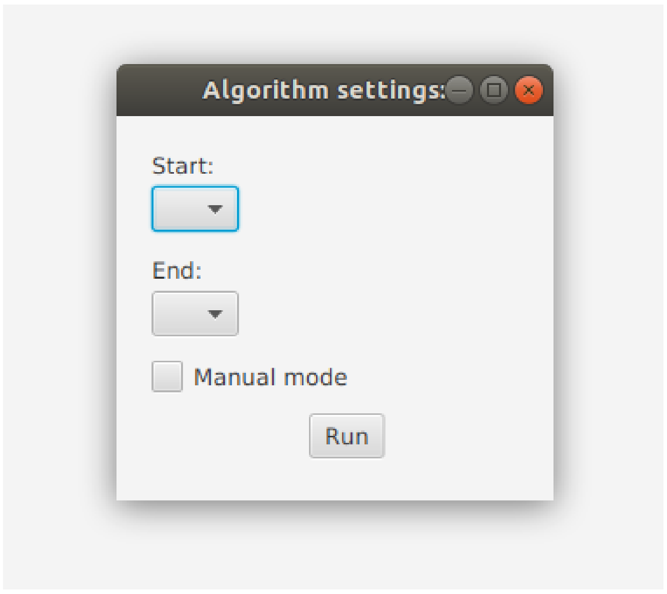
3) Spanning tree (if clicked it runs minimum spanning tree algorithm and displays the found result);

4) Shortest cycle (if clicked it runs shortest cycle algorithm and displays the found result);

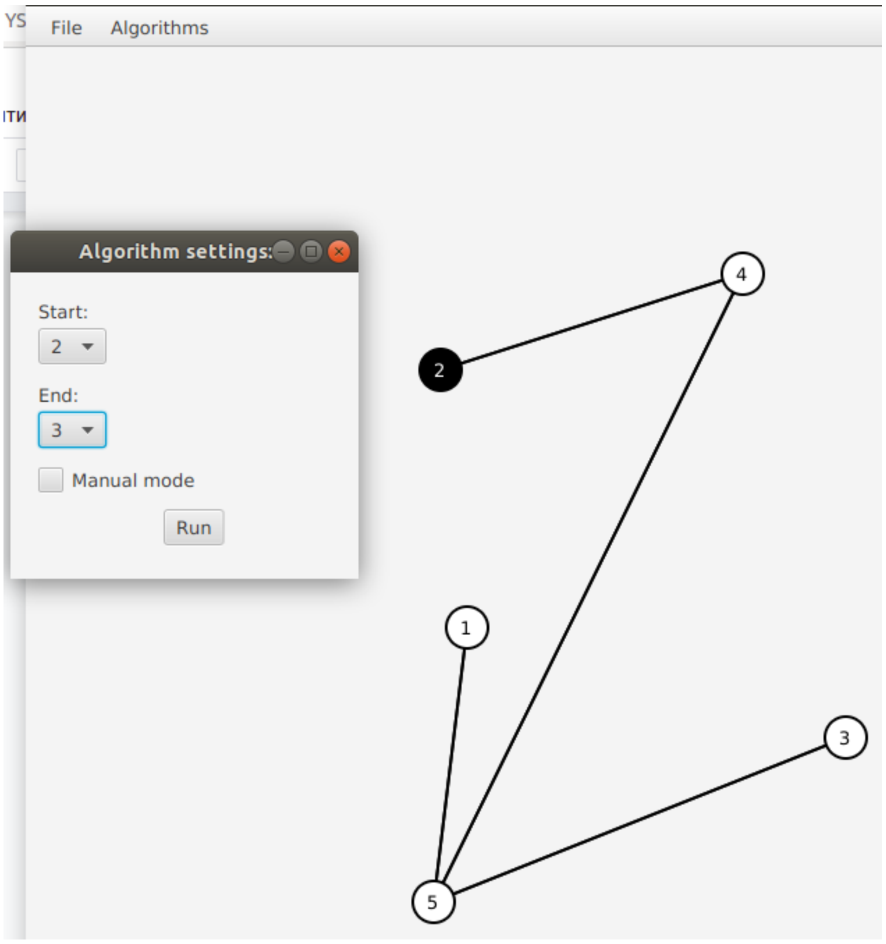
5) Stop/Clear status (if clicked it stops running algorithm & clears all algorithm related info).



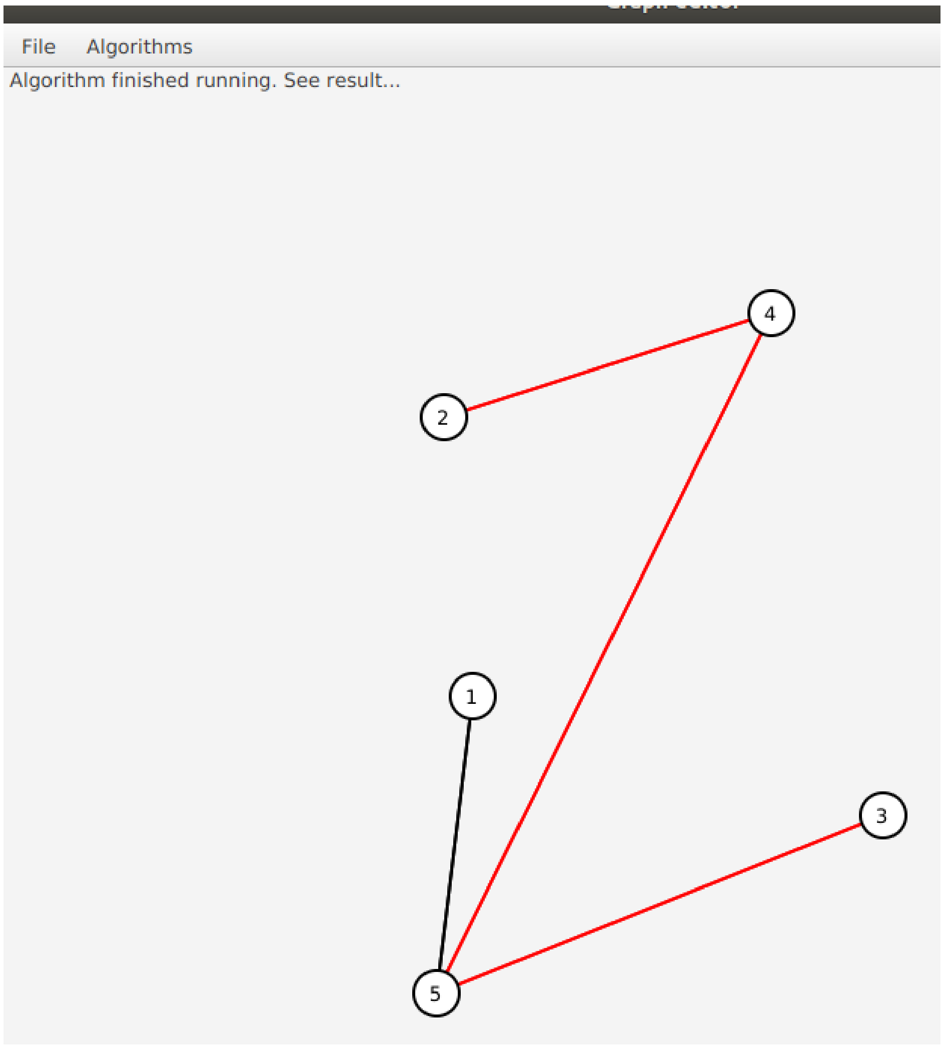
When the user selects the DFS/BFS algorithm the following form is displayed.



To proceed forward and run the algorithm it is necessary to fill in relevant data. The Start dropdown allows the user to select the start node of the path they are looking for, while the End dropdown gathers information about the end node. Both dropdowns display only ids of existing nodes, so they prevent the user from using incorrect data and breaking the program. Additionally, there is a checkbox that helps to indicate how the user prefers to run the algorithm: automatically or manually.



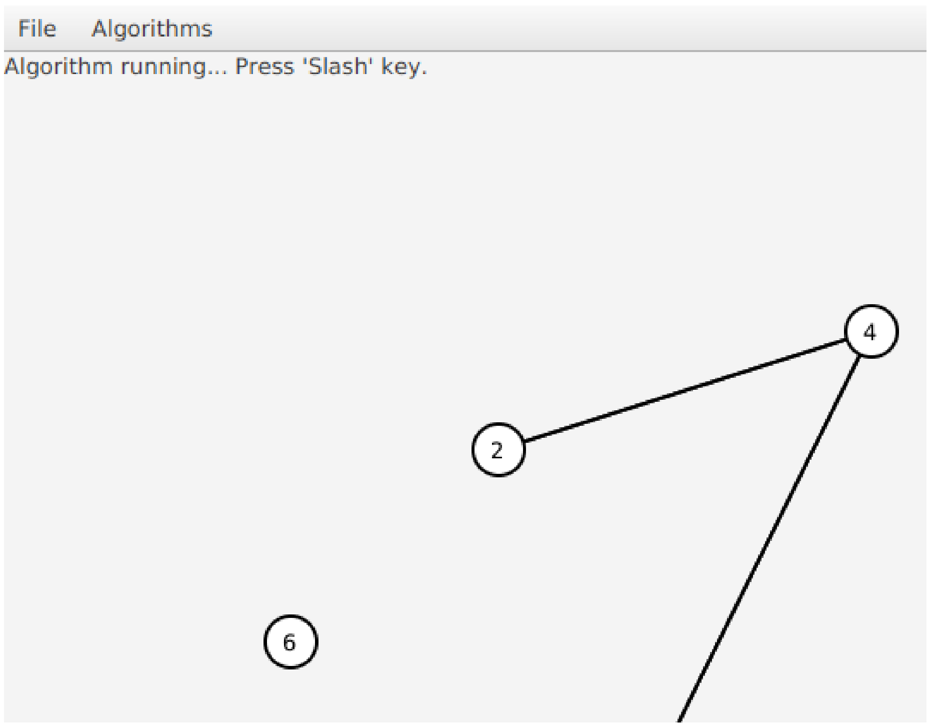
The auto mode is a default mode that runs in the background and shows the final result. It also displays info at the top of the program window to indicate that the algorithm finished running.



Algorithm finished running. See result...

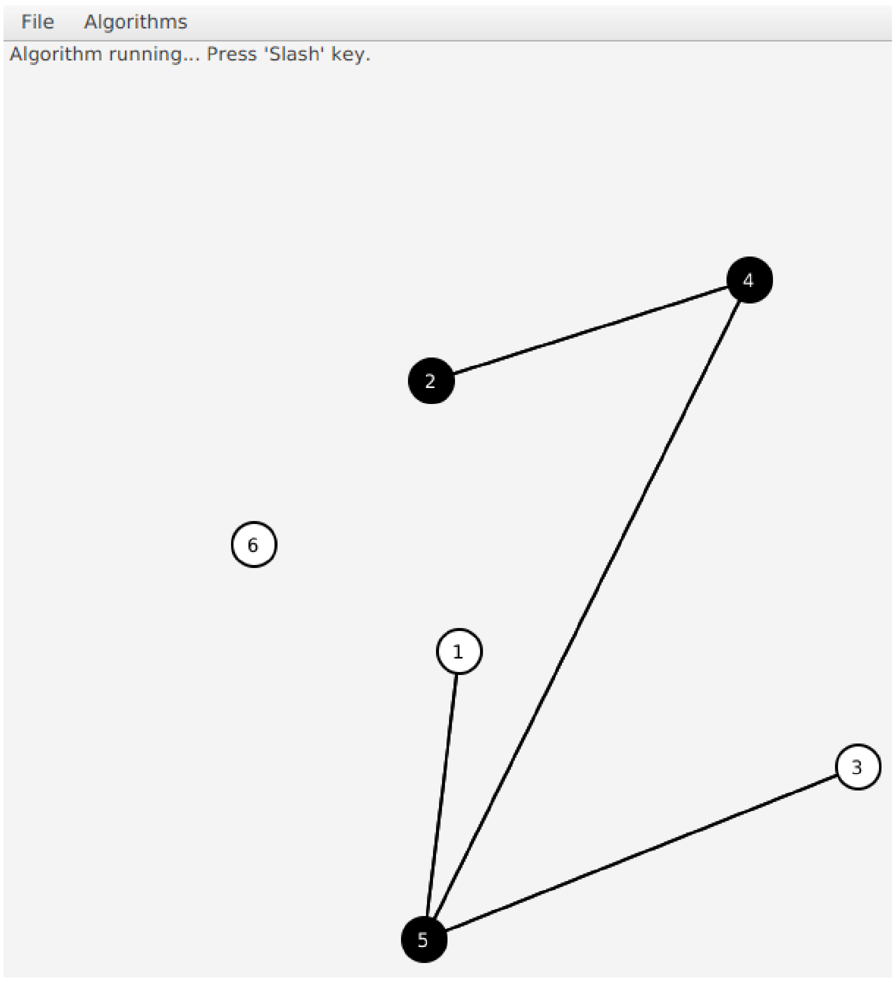
In case no result was found the program also informs about it, stating that "Algorithm finished running. No result found."

Meanwhile, the manual mode allows the user to follow algorithm steps one by one and see how nodes are being processed. Once the user checks "Manual mode" and presses "Run" the program displays the message at the top of the window that explains how to proceed forward.



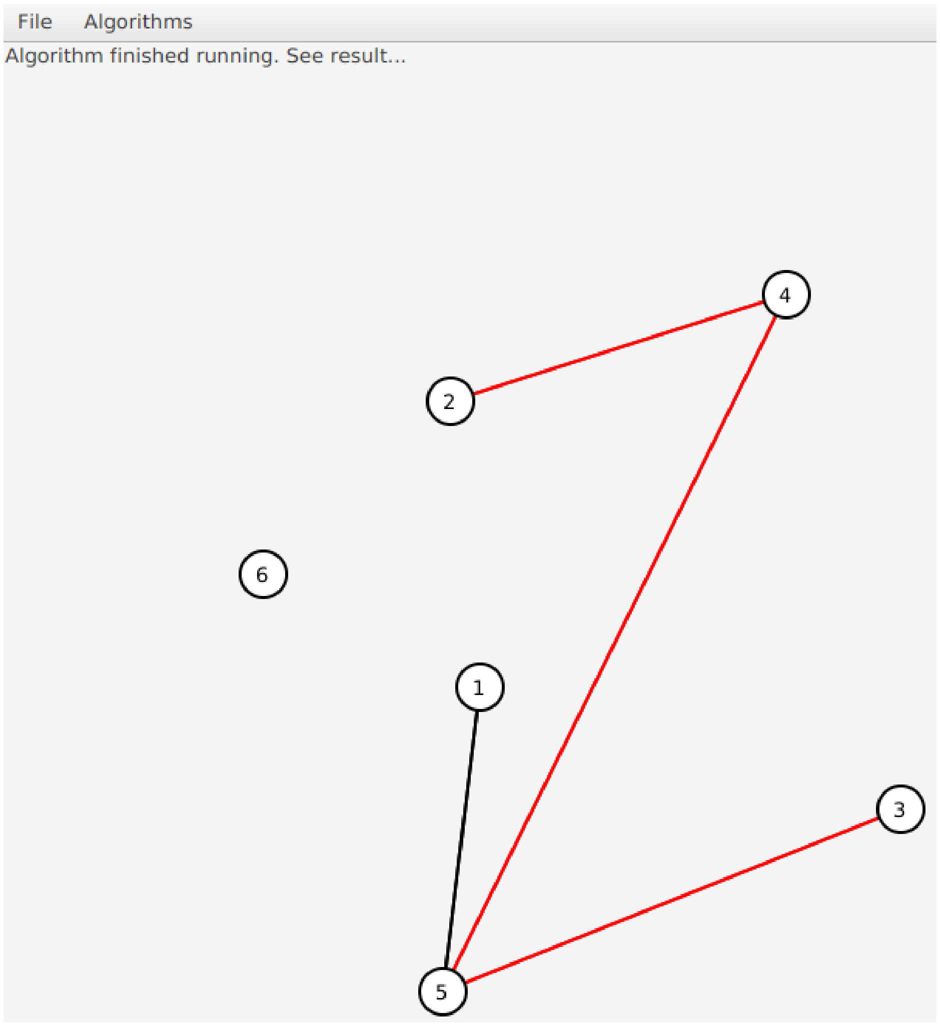
Algorithm running... Press 'Slash' key.

Every time the user presses Slash the new node is processed and displayed as such.



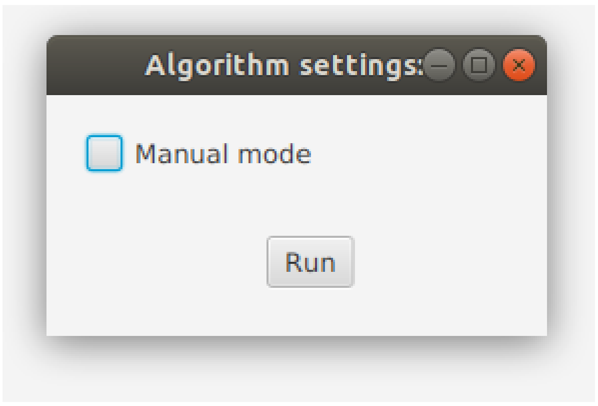
Algorithm running... Press 'Slash' key.

Once the result is found the user is again informed about it and can see it on the screen.



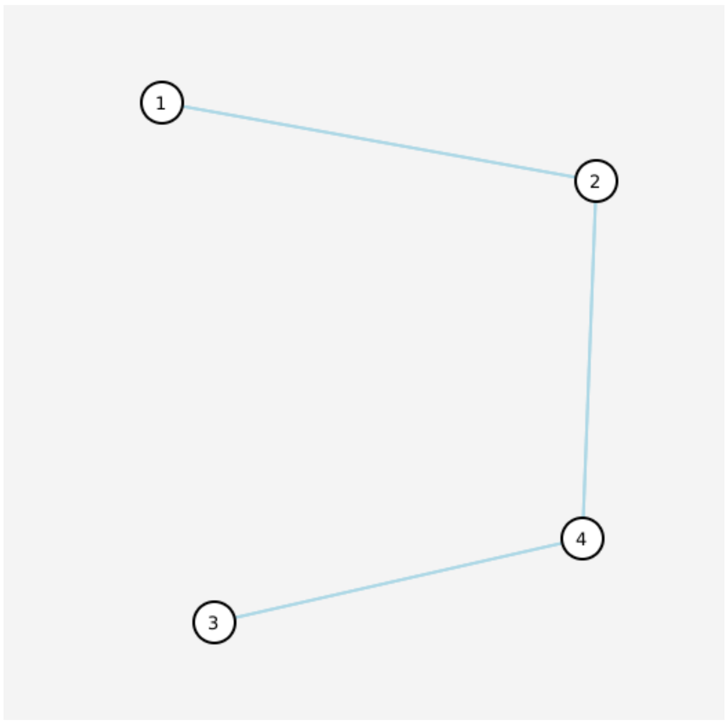
Algorithm finished running. See result...

In the case with two other algorithms - minimum spanning tree and shortest cycle - the logical flow is similar to DFS/BFS, yet has its own specifics. For example, when the user selects the spanning tree algorithm, the following form is displayed.

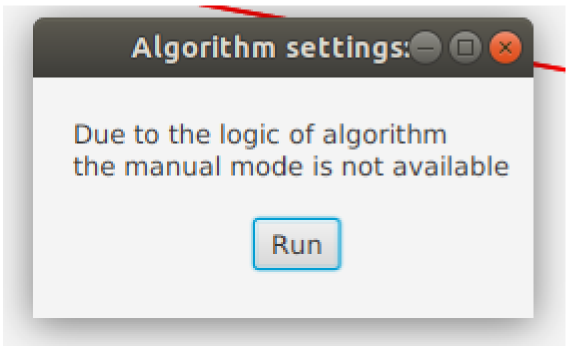


In comparison to DFS/BFS, the user cannot select nodes because the algorithm decides what vertices and edges are the part of spanning tree on its own. Also, during the manual search the explored edges are colored with blue, while the unexplored ones - with black.

This time, no vertices change their colors because the algorithm focuses on edges, not vertices.



Finally, the shortest cycle algorithm does not allow to split steps in such a way that it would be possible to display meaningful differences between every step. Therefore, manual mode is disabled for this algorithm and the user is informed about it.



Due to the logic of algorithm
the manual mode is not available