

Signal flow Assignment Report

Mohamed el-maghraby Mohamed 55

Abdelrahman Ahmed Mohamed Abdelfttah Omran 37

# Problem statement

**Given**

Signal flow graph representation of the system. Assume that total number of nodes and numeric

branches gains are given.

**Required**

1- Graphical interface.

2- Draw the signal flow graph showing nodes, branches, gains, …

3- Listing all forward paths, individual loops, all combination of n non-touching loops.

4- The values of , 1 , …, m where m is number of forward paths.

5- Overall system transfer function.

# Main features

* User friendly gui made with javafx .
* Drawing without any restrictions on the position of node .
* Adding the node’s name and the arrow gain immediately on the draw .
* The the connections between the nodes are designed to be either straight line or curve to avoid intersections (most of the time).
* Showing the solution with the graph to track forward paths.
* The user can switch between forward paths ,loops ,untouched loops and the total transfer function with single click .
* If any error happened the user will be notified that his graph has an error in the solution pane.

# Bonus

The user can find the Transfer function between any two nodes even there weren’t a sink or source nodes.

# Data structures

* Class for nodes, Arrows, loops and forward paths.
* Arraylists to keep nodes, Arrows, forward paths and loops.
* Arraylist of Arraylists to keep untouched loops.
* Hashmap to store the icons with their names as keys
* Linked Lists for the gui nodes and arrows .

# Main modules

* Control module called SFG that handle all operations .
* View module to handle GUI .
* Structure Module that keep all data (Nodes, Arrows, Forward paths and loops).

# Algorithms used

* DFS to find the forward paths.
* DFS to find loopst.
* To find untouched loops, I found all ….. And removed touched and repeated ones.

# 

# 

# 

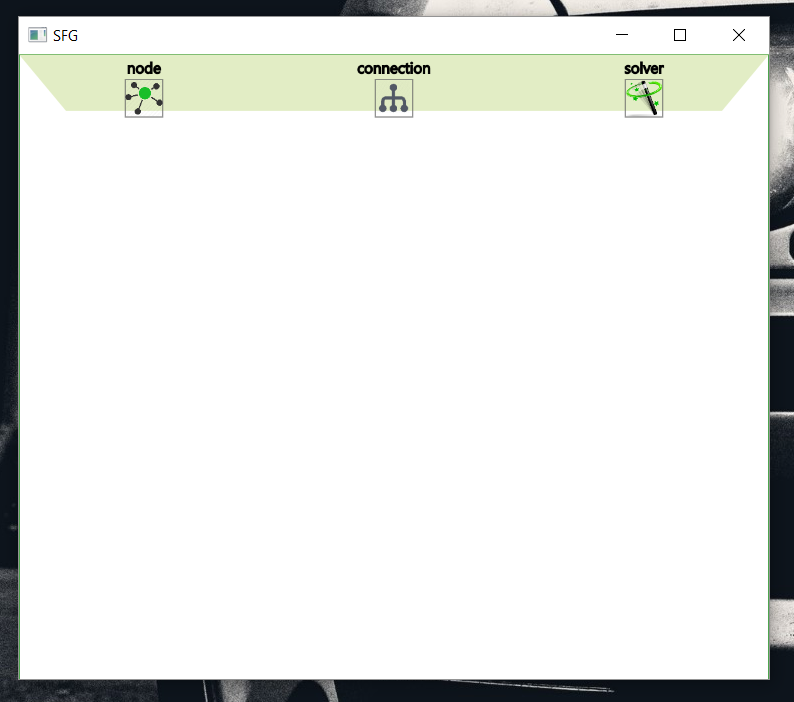
# 

# 

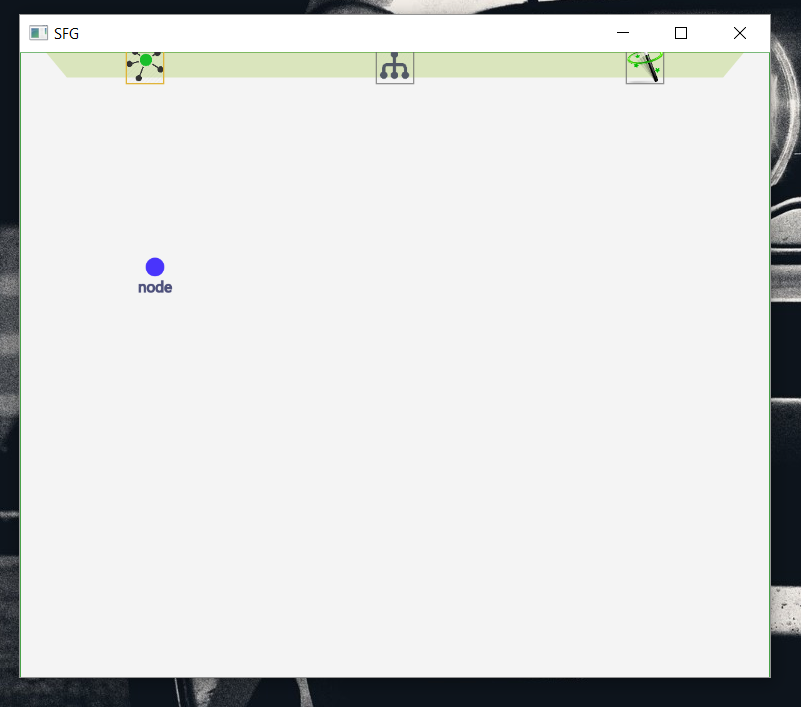
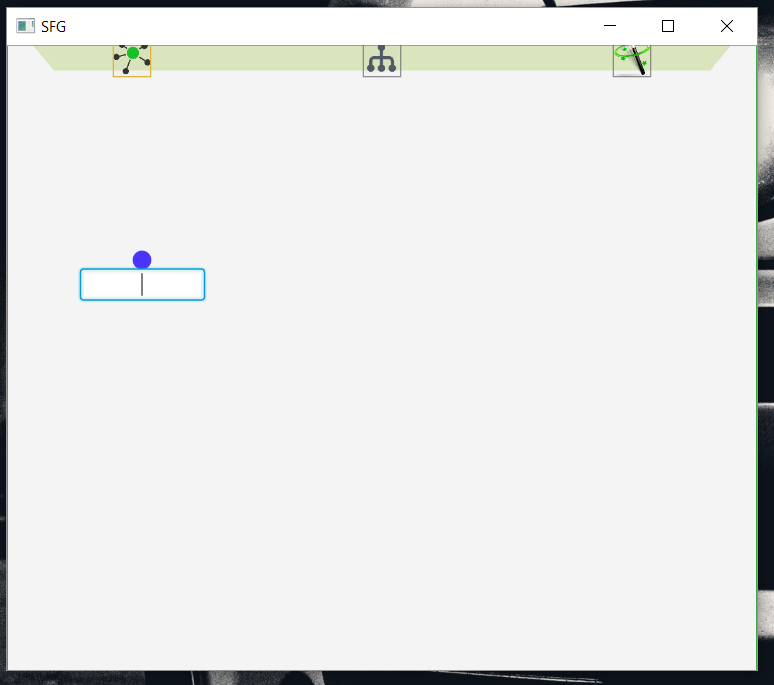
# 

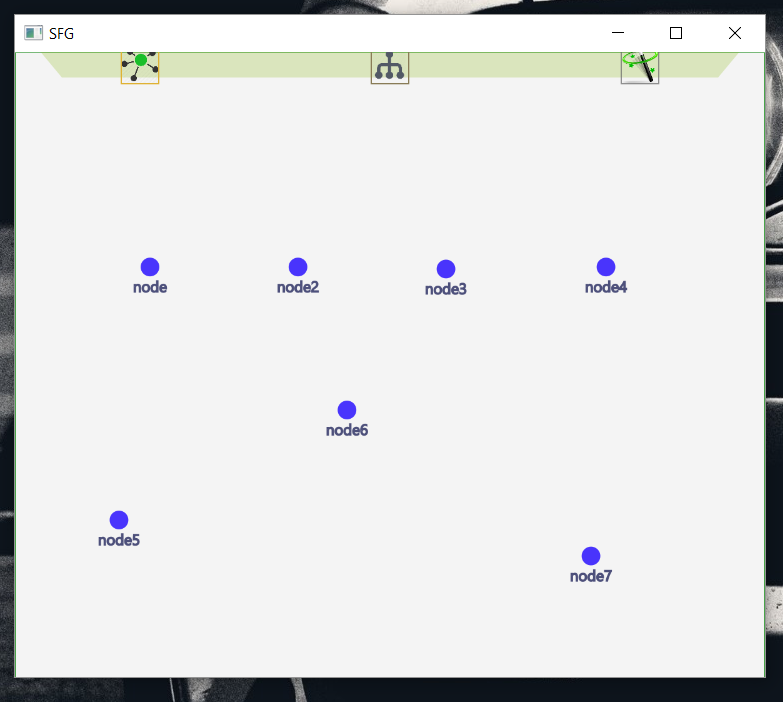
# Sample runs

1. Main layout : Three main actions : draw node , draw connection and solve the graph

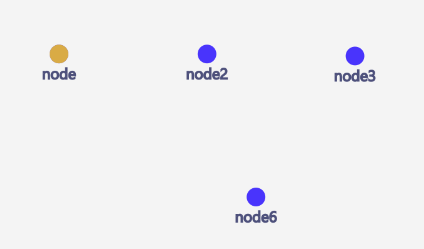
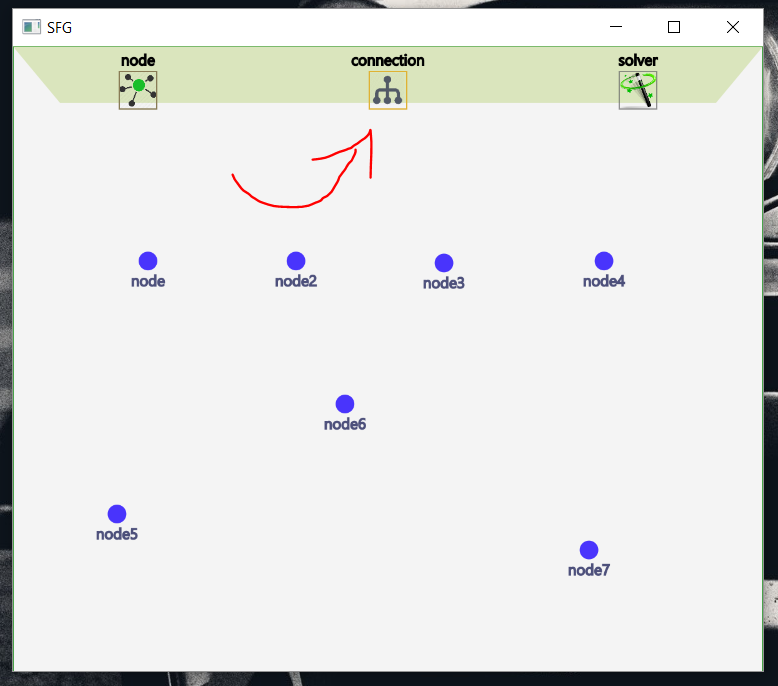


2. Constructing node and write its name , the user can put the node wherever with no restrictions on node positions or name .

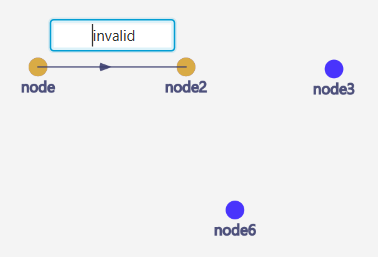
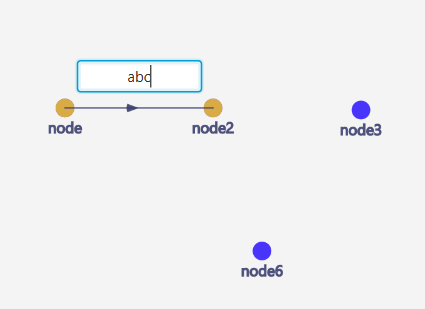




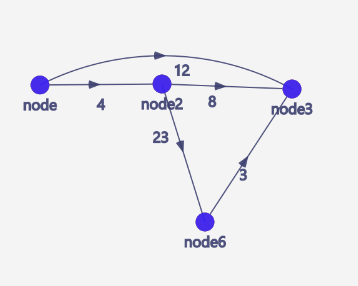
3. The user clicks on the connection tab , and select two nodes to draw a connection from the first selected to the second .



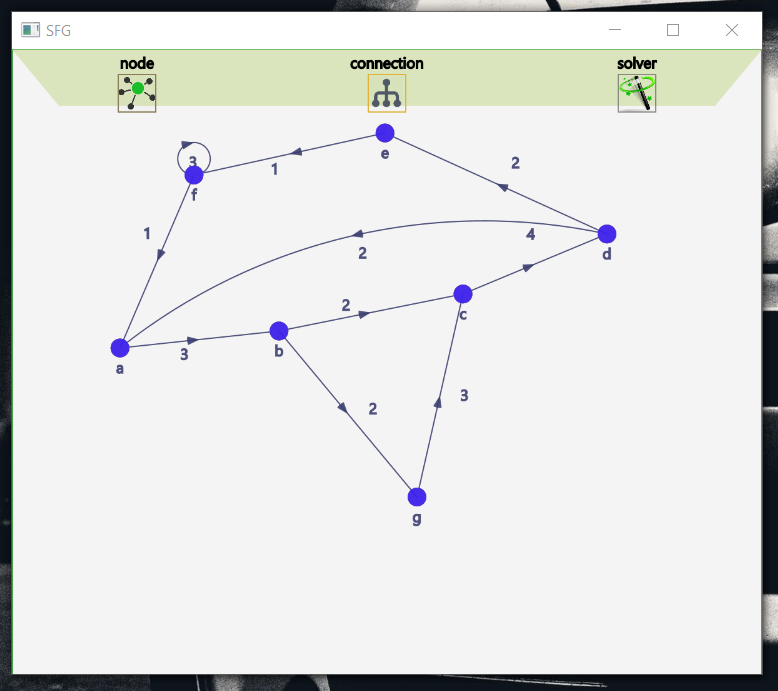
* Note that if the user entered non integer gain , the gui tells him that it is invalid .



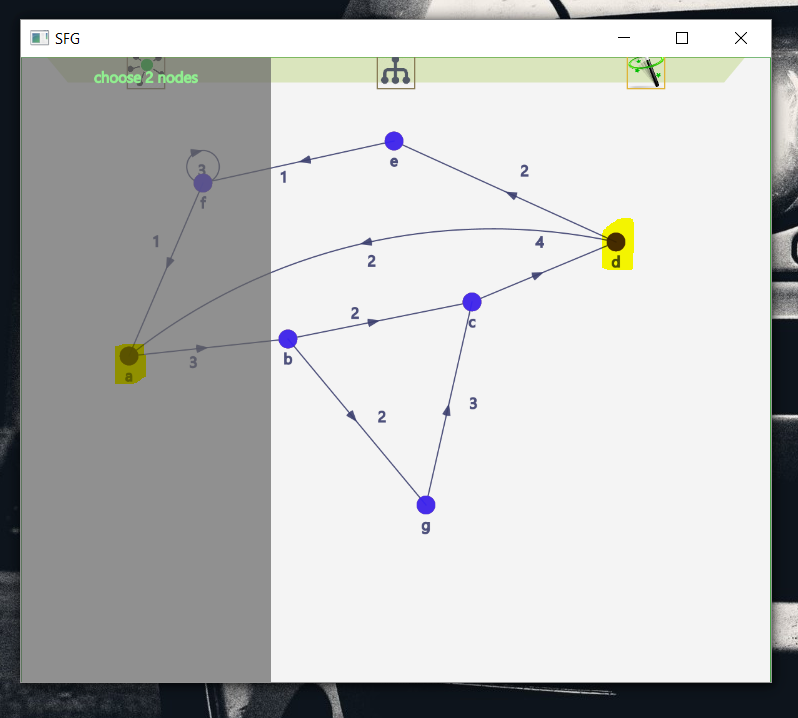
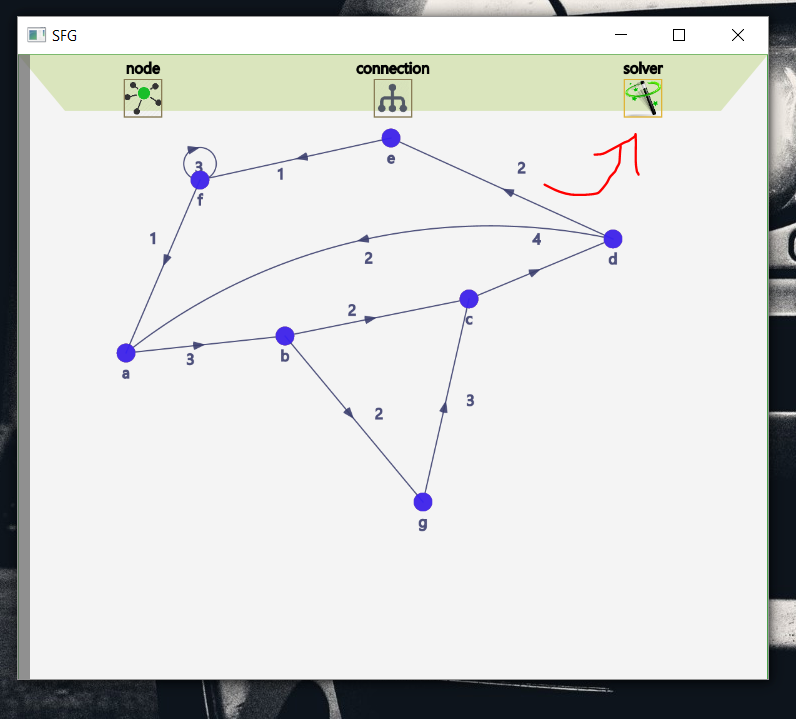
* Here is what connections will look like :



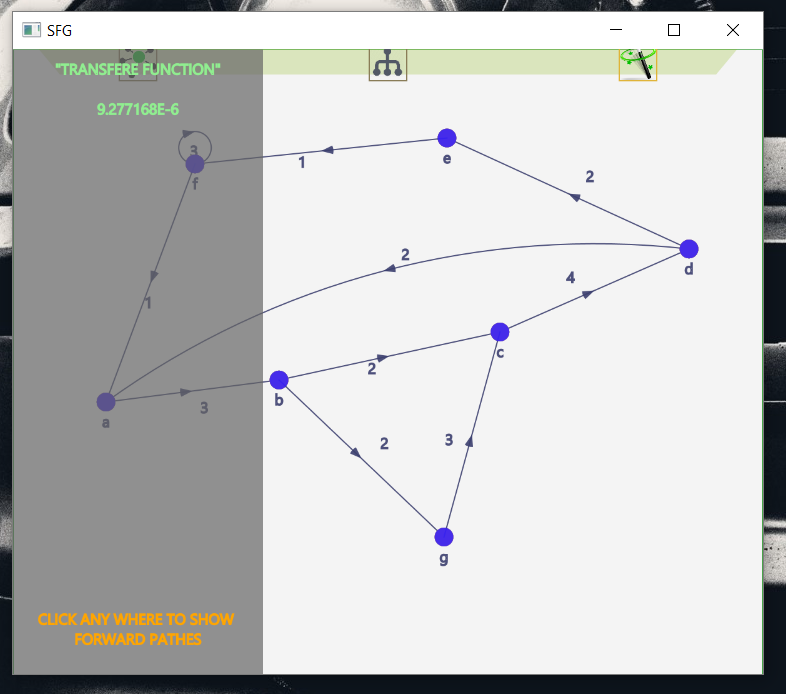
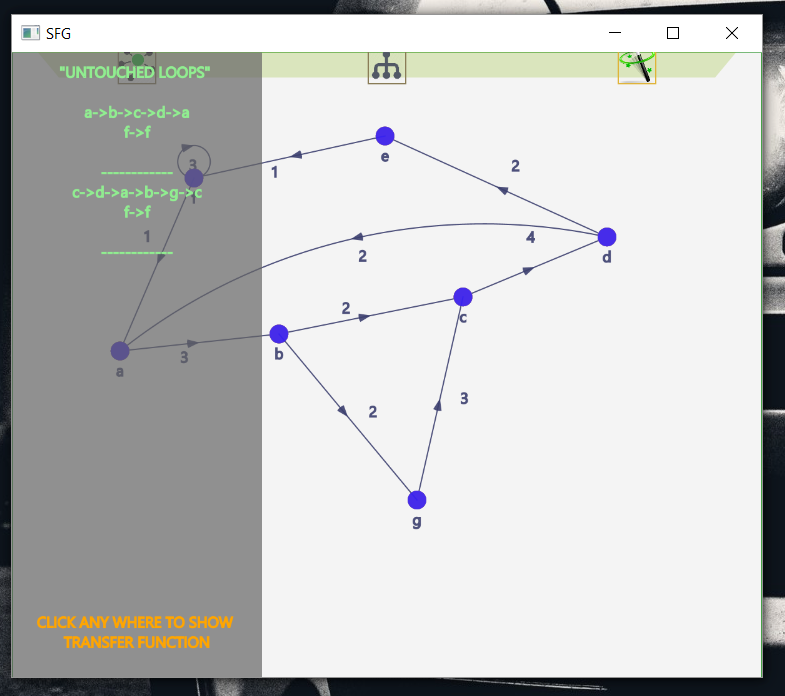
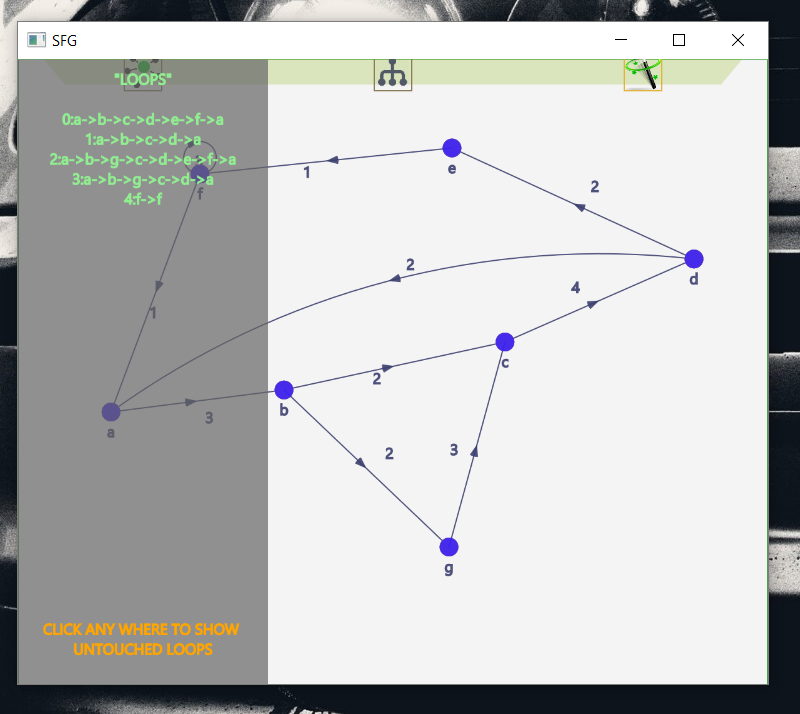
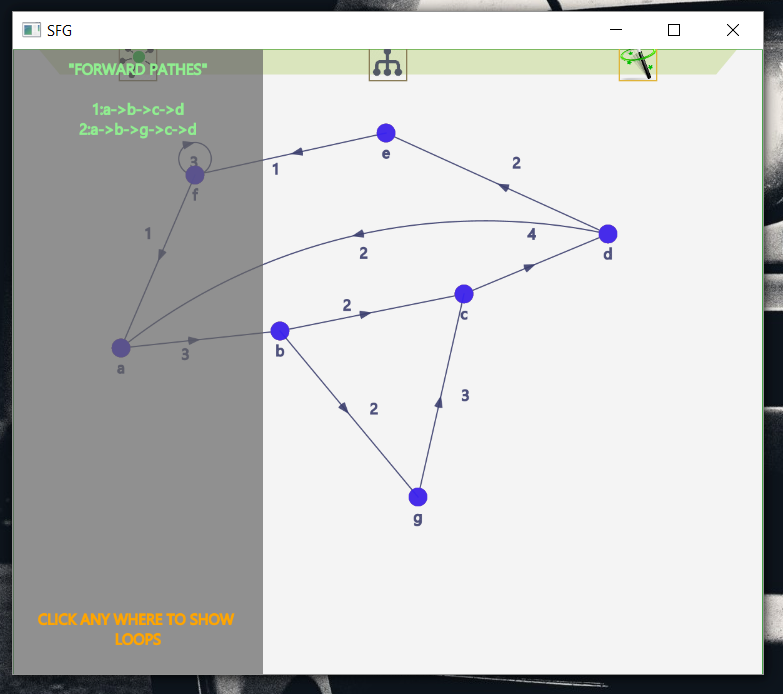
3. Another Sample Run with big sfg to show the results :



5. On clicking on the third tab , the gui asks the user to choose two nodes as source node and sink node .



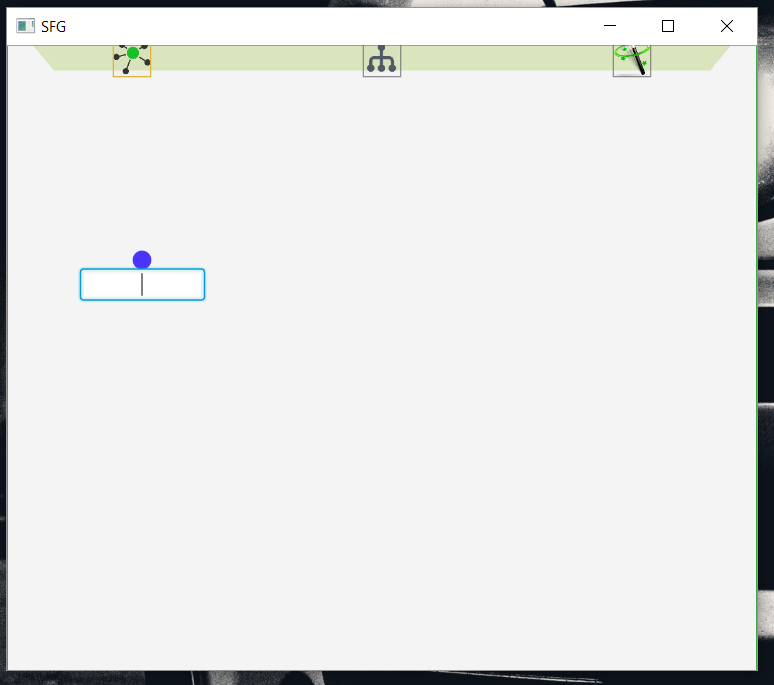
6. After choosing node a then b (marked on the previous image) , the gui views the result .



# 

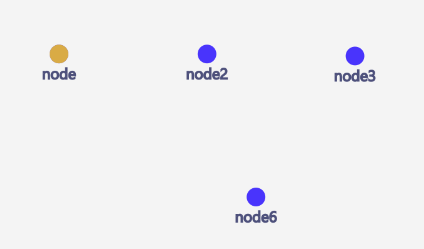
# Simple user guide

* To draw node :

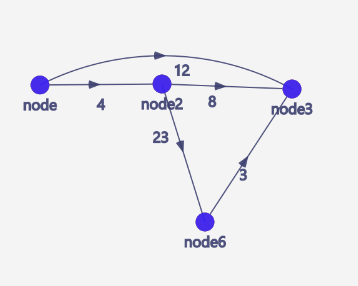
1. Move the mouse towards the upper tab .
2. The tab will move down .
3. Click on “Node”
4. Choose where to place the node on the white pane by clicking .
5. Then the gui will ask you to name the node .
6. The node’s name must be unique .

* To draw a connection :

1. Move the mouse towards the upper tab.
2. Click on “connection”
3. Choose the node to draw the connection from .
4. The node will turn orange .

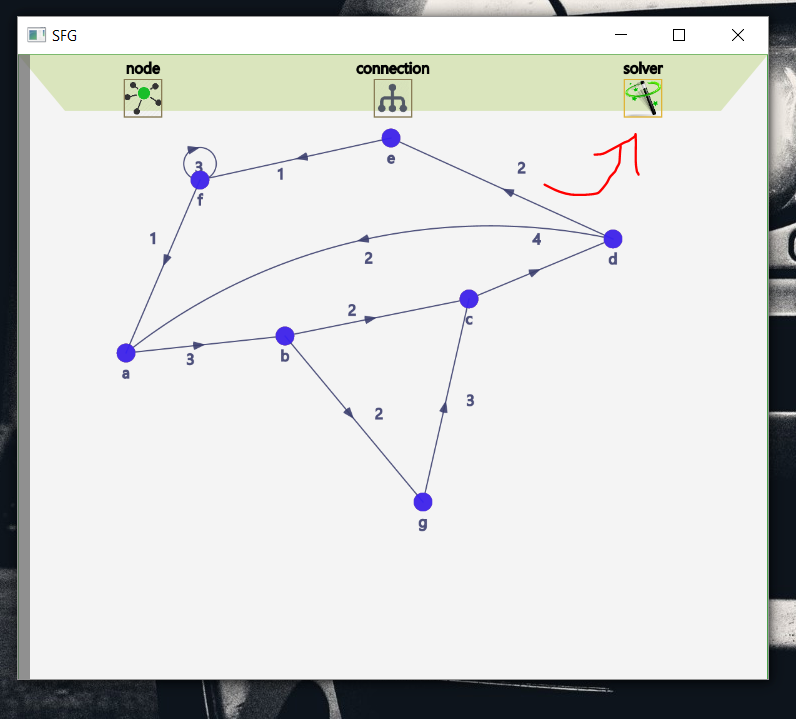


1. Choose the second node .
2. The gui will draw the connection and ask you to enter the gain .



1. Then the gui will put that gain on the canvas

* To show result :

1. 
2. Move the mouse towards the upper tab.
3. Click on “solver”.
4. A small gray pane called the solution pase will appear on the left side
5. Moving the mouse on it , it will expand .
6. The solution pane will ask you to choose 2 nodes the source and the sink .
7. Then the forward paths will appear there.
8. By clicking on anywhere on the solution pane , it will show loops then untouched loops then the transfer function .

