الأسم : عبدالعزيز السيد عبدالعزيز راشد الفرقة : الثانية

رقم الجلوس : 40 القسم : الحاسبات و النظم

Mason Formula Program

Project Line Here

Problem Statement

Given a general signal flow graph, write a program to calculate the over all transfer function using Mason Formula.

Formula

$$G = \frac{y_{in}}{y_{out}} = \frac{\sum_{k=1}^{N} G_k \Delta_k}{\Delta}$$

$$\Delta = 1 - \sum L_i + \sum L_i L_j - \sum L_i L_j L_k + \cdots + (-1)^m \sum \ldots + \cdots$$

where:

- Δ = the determinant of the graph.
- y_{in} = input-node variable
- y_{out} = output-node variable
- G = complete gain between y_{in} and y_{out}
- N = total number of forward paths between y_{in} and y_{out}
- G_k = path gain of the kth forward path between y_{in} and y_{out}
- L_i = loop gain of each closed loop in the system
- L_iL_j = product of the loop gains of any two non-touching loops (no common nodes)
- $L_iL_jL_k$ = product of the loop gains of any three pairwise nontouching loops
- Δ_k = the cofactor value of Δ for the k^{th} forward path, with the loops touching the k^{th} forward path removed. *

Definitions

- Path: a continuous set of branches traversed in the direction that they indicate.
- Forward path: A path from an input node to an output node in which no node is touched more than once.
- Loop: A path that originates and ends on the same node in which no node is touched more than once.
- Path gain: the product of the gains of all the branches in the path.
- Loop gain: the product of the gains of all the branches in the loop

Data Structures :

Vertex:

A class represent a vertex in a graph contains the vertex name and Edge list contains edges connected to this vertex

Edge:

A class represent the edge between two vertices contains the source vertex and the sink vertex and the gain between these two vertices

Path:

A class represent a path (forward or circular) contains the vertices in this path and the whole and the gain between these two vertices

Classes:

Control:

A control class for managing i/o in program window (Javafx)

Main:

A main class containing main method to run the program

Utility:

A utility class for getting all the possible combinations between a given loops with a target length

Algorithms and Functions :

Class SFG :

public ArrayList<Path> getForwardPaths()
Gets all the possible forward paths for node 0 to node n-1

public ArrayList<Path> getIndividualLoops()
Gets all the possible individual loops in the graph

public ArrayList<ArrayList<ArrayList<Path>>> getAllNonTouchingLoops()
Gets all the possible non touching loops in the graph

public Double getDelta()
returns Δ = the determinant of the graph

public ArrayList<Double> getDeltaK()
returns an array list of the Δ_k

public double getTotalTransferFunction()
returns the total transfer function value

private boolean isNonTouchingWithForwardPath(Path forwardPath, ArrayList<Path> nonTouchingCombination)
returns true if a combination of loops is not touching the forwardPath
returns false otherwise

private boolean isNonTouchingWithForwardPath(Path forwardPath, Path loop)

returns true if a loop is not touching the forward path
returns false otherwise

private boolean isNonTouchingLoops(ArrayList<Path> combination)
returns true if a combination of loops is not touching each other
returns false otherwise

private double getGain(List<String> cycle)
returns the whole gain of a cycle

Class Utility :

public static ArrayList<ArrayList<Path>> getAllCombinations(ArrayList<Path> loopm int length)

returns A array list of all possible combination between loops with length length of loops

private static void getAllCombinations(int index, ArrayList<Path> comb, int length)

A recursive function to get the combinations

User Guide :

-First enter number of nodes in nodes number box as shown and then press (Add nodes) then your nodes will be added to your graph and you can view them by pressing (Display graph) button

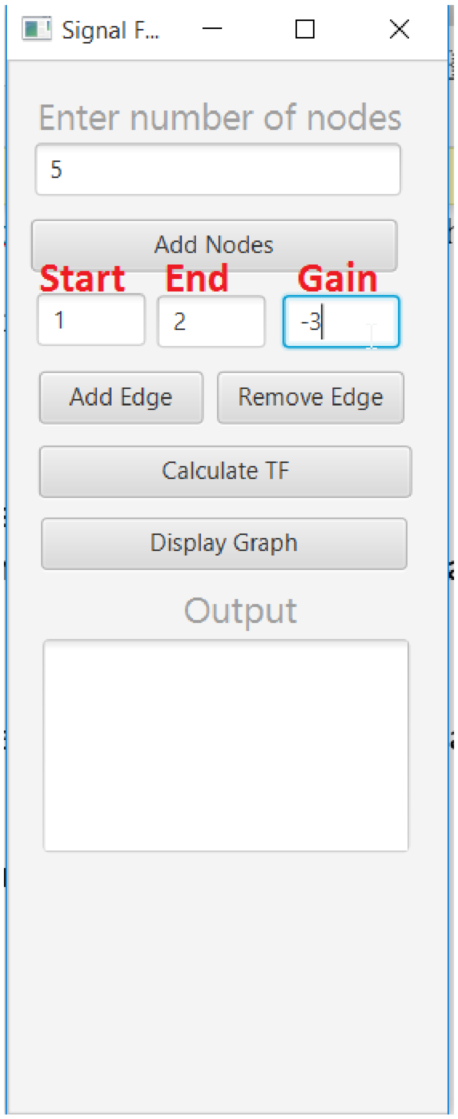
If you want to add more nodes you can by entering the number you want to add and pressing add nodes again

-According to program your source node should be number 1 and sink node should be number N where N is the total number of nodes in the graph

-you can add edges between nodes by typing in first box start node of edge and in second box end node of edge and in the third box the gain of edge (The program works only with numerical values no symbols allowed 😟 to be added isa)

And finally press add edge

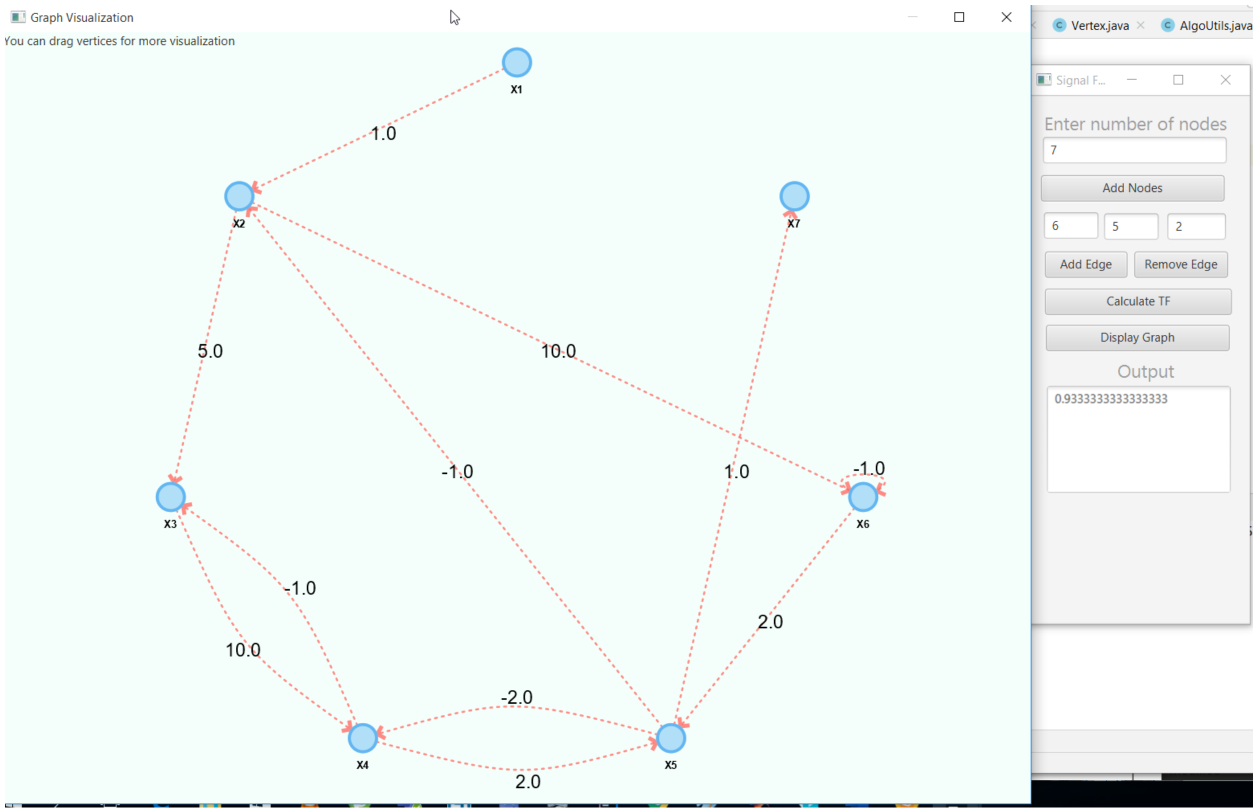
Ex picture



-When you finish entering your flow graph press (Calculate TF) to get the overall Transfer Function
-You can Display your graph at any stage

Sample runs

Sheet 3 Problem #3



```
0.9333333333333333
```

<u>**Note :**</u>

<u>**All needed external libraries will be found in “lib” folder**</u>