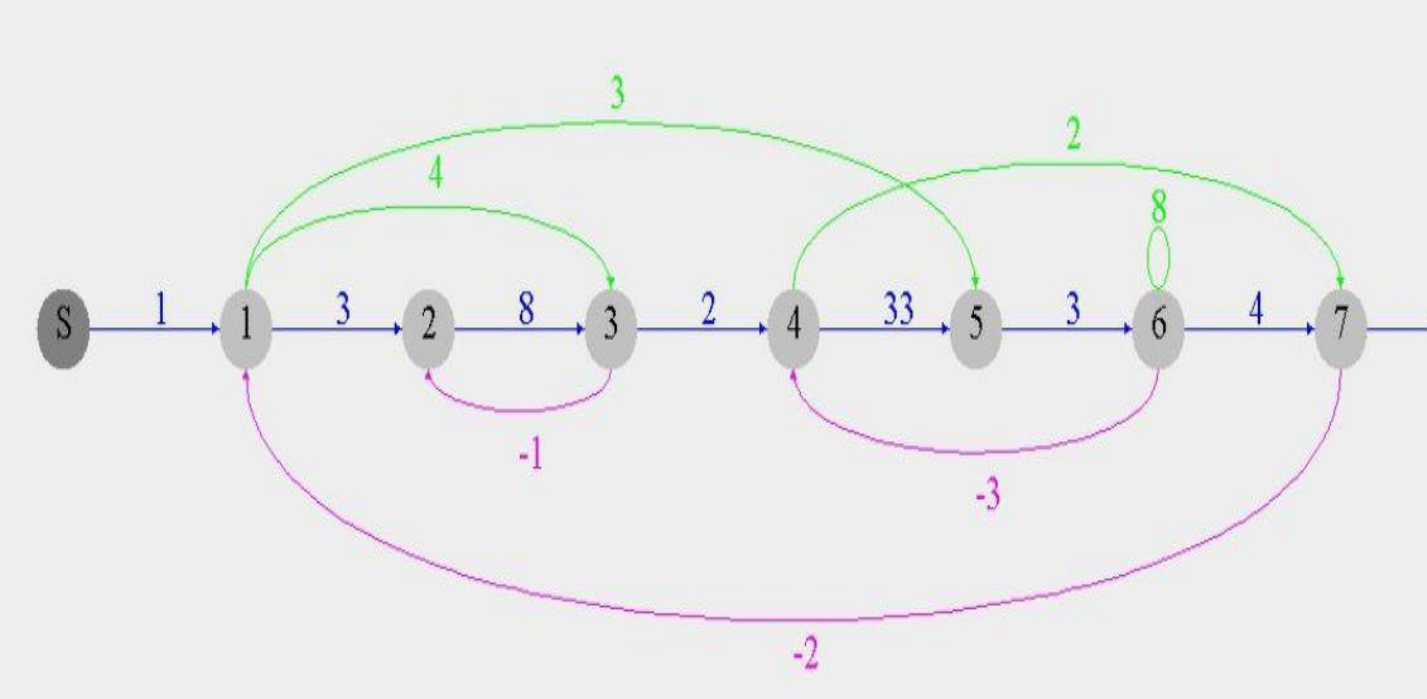
Signal Flow Graph



by

[Mohamed Murad Shafiq]
[AlexU, CSED19, 69]

TABLE OF CONTENTS

1 PROBLEM STATMENT

Signal flow graph representation of the system. Assume that total number of nodes and numeric branches gains are given.

> *"A signal-flow graph or signal-flow graph (SFG), invented by Claude Shannon, but often called a Mason graph after Samuel Jefferson Mason who coined the term, is a specialized flow graph, a directed graph in which nodes represent system variables, and branches (edges, arcs, or arrows) represent functional connections between pairs of nodes. Thus, signal-flow graph theory builds on that of directed graphs (also called digraphs), which includes as well that of oriented graphs. This mathematical theory of digraphs exists, of course, quite apart from its applications.*
>
> *SFG's are most commonly used to represent signal flow in a physical system and its controller(s), forming a cyber-physical system. Among their other uses are the representation of signal flow in various electronic networks and amplifiers, digital filters, state variable filters and some other types of analog filters. In nearly all literature, a signal-flow graph is associated with a set of linear equations." Wikipedia*

2 STRUCTURES AND FETURES

This software is implemented using **Java** programming language.
GUI is fully programmed in **Swing**

2.1 MAIN FETURES

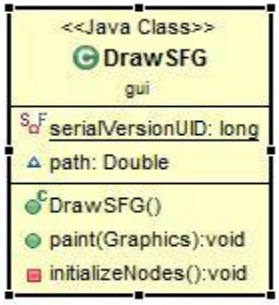
You can use a SFG desktop app with high user friendly GUI.

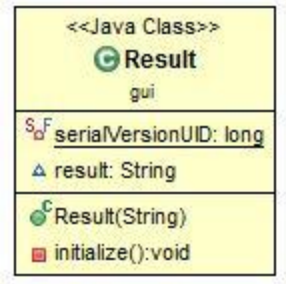
2.2 DATA STRUCTURES

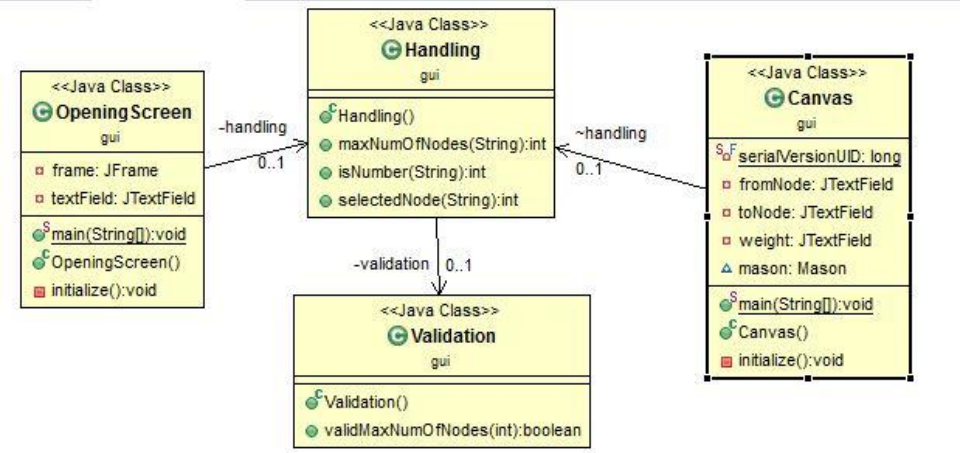
There are many data structures used in the Mason implementation such as:

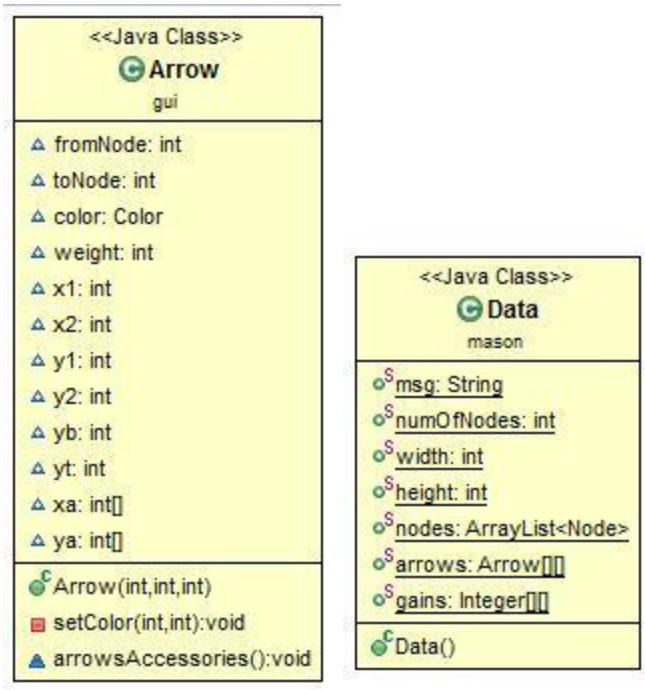
1- `Array List`: used to store forward paths, loops, untouched loops and their gains, the implementation used Array List of Array List of integers, where integers are the node's numbers, so I can easily manipulate the nodes operations.
2- `Queue`: used in two positions:
a- in finding loops, implementation save the suspected paths which is probably a loop – before checking it's similar to taken one or not – after checking if is valid loop It's saved, else it's removed from the queue, as long as my path splits I save new split path in the same queue, when queue is empty method is finished.
b- the same thing is done on finding the untouched loops of rank 3 or more [three untouched loops or more], there are two instances of queues one get the new untouched of rank k, the other has the previous untouched loops of the previous ranks, after checking all k-1 untouched loops with original loops, the queue containing the new untouched loops is added to the original untouched, used it as k and so on.
3- `String Builder`: because of high text accumulated to display, it's necessary to use efficient way to build the result, so String builder in **Java** is the best choice.

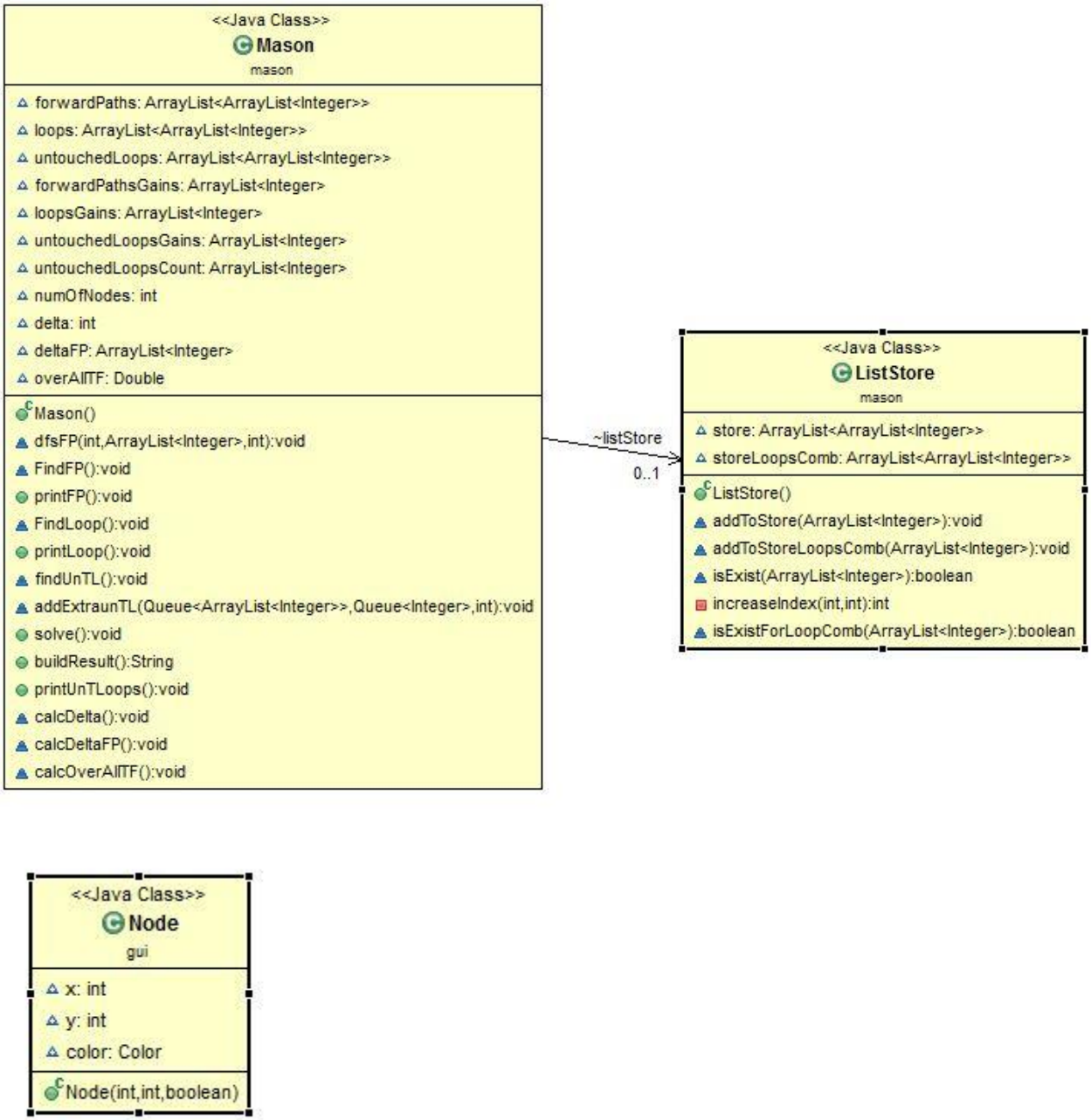
2.3 MAIN MODULES






<<Java Class>>
OpeningScreen
gui
frame: JFrame
textField: JTextField
main(String[]):void
OpeningScreen()
initialize():void
-handling
0..1
<<Java Class>>
Handling
gui
Handling()
maxNumOfNodes(String):int
isNumber(String):int
selectedNode(String):int
~handling
0..1
<<Java Class>>
Canvas
gui
serialVersionUID: long
fromNode: JTextField
toNode: JTextField
weight: JTextField
mason: Mason
main(String[]):void
Canvas()
initialize():void
-validation
0..1
<<Java Class>>
Validation
gui
Validation()
validMaxNumOfNodes(int):boolean


<<Java Class>>
Arrow
gui
fromNode: int
toNode: int
color: Color
weight: int
x1: int
x2: int
y1: int
y2: int
yb: int
yt: int
xa: int[]
ya: int[]
Arrow(int,int,int)
setColor(int,int):void
arrowsAccessories():void
<<Java Class>>
Data
mason
msg: String
numOfNodes: int
width: int
height: int
nodes: ArrayList<Node>
arrows: Arrow[][]
gains: Integer[][]
Data()



2.4 ALGORITHMS USED

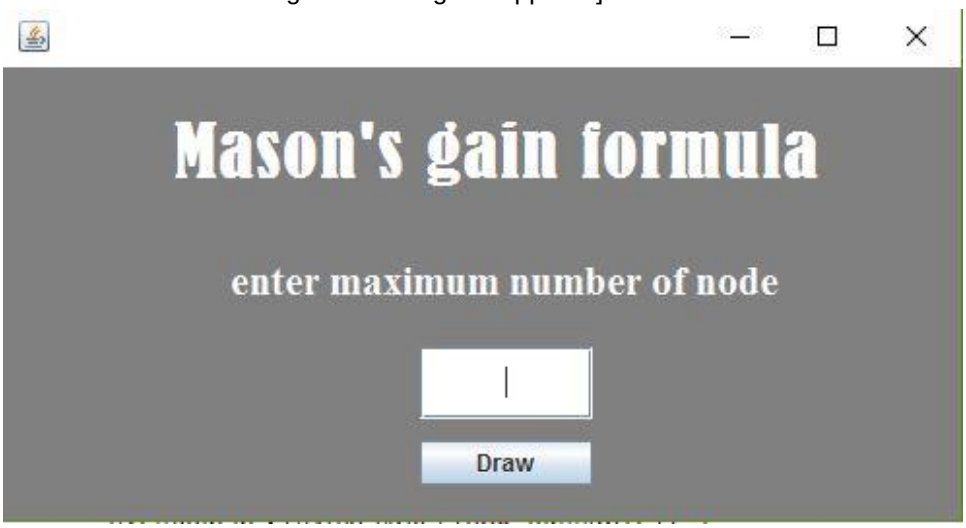
DFS algorithm takes the graph as a matrix representation, and used to get the forward paths, to find loops it's easily to use the DFS method with some enhancement, but there are another implementation used as a replacement of DFS to find loops, Techniques also used as the two pointer to set ranks of untouched loops during searching them, and save all untouched loops with various ranks in one place using another list parallel to it, which is set by those pointers.

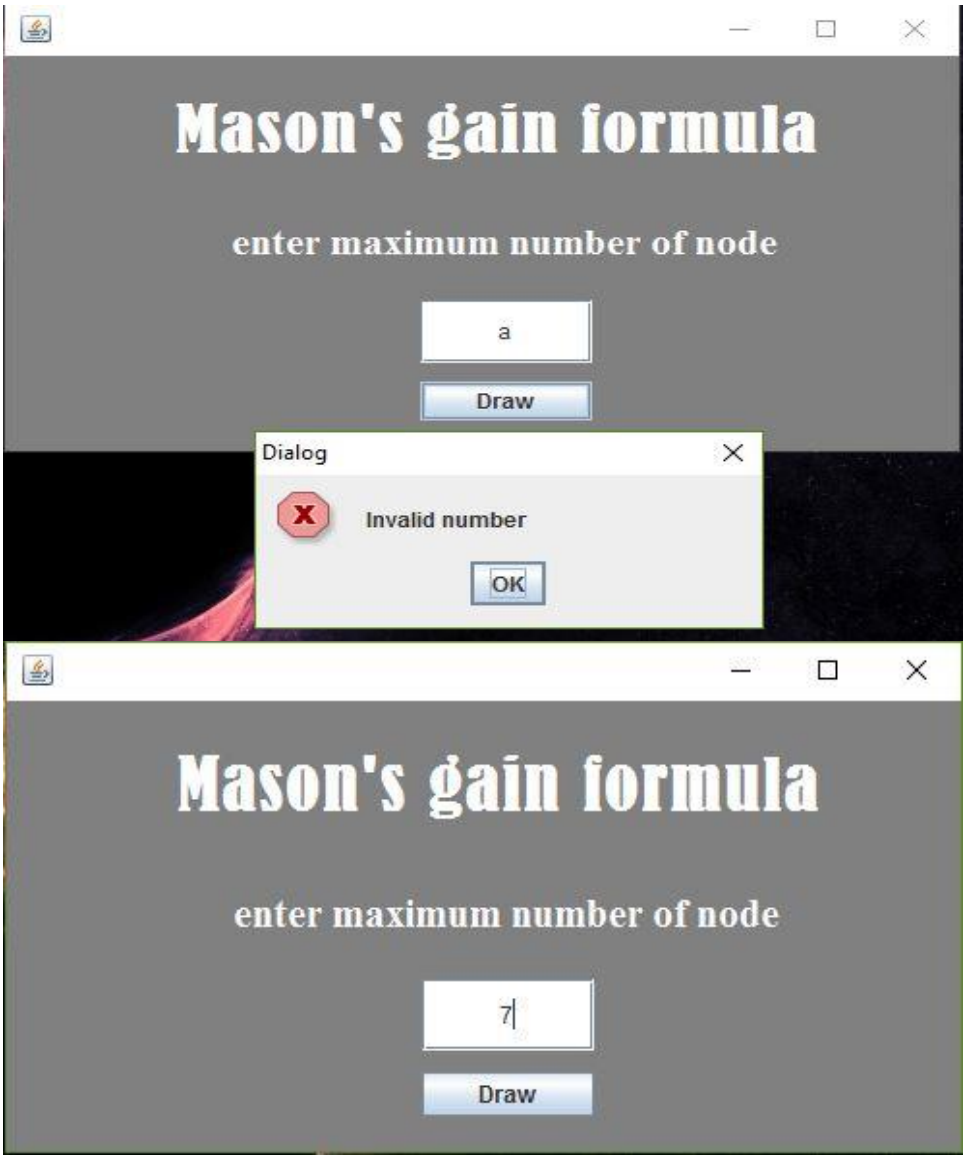
3 USER GUIDE

As we mentioned, the most feature in this app is that it's user friendly, let's take a look of how to use it step by step;

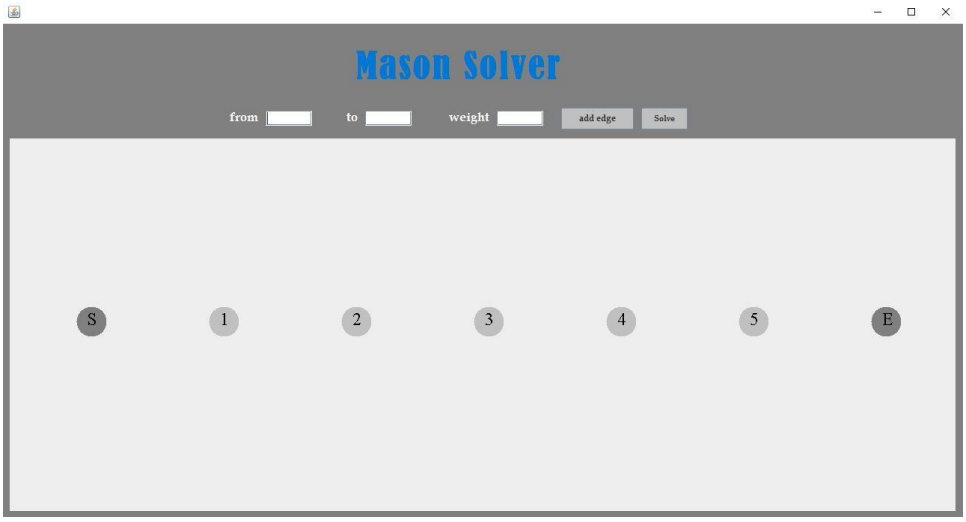
3.1 USER GUIDE

1- open the app, then select number of nodes [in case of invalid number or overflow number massage or warning will appears]

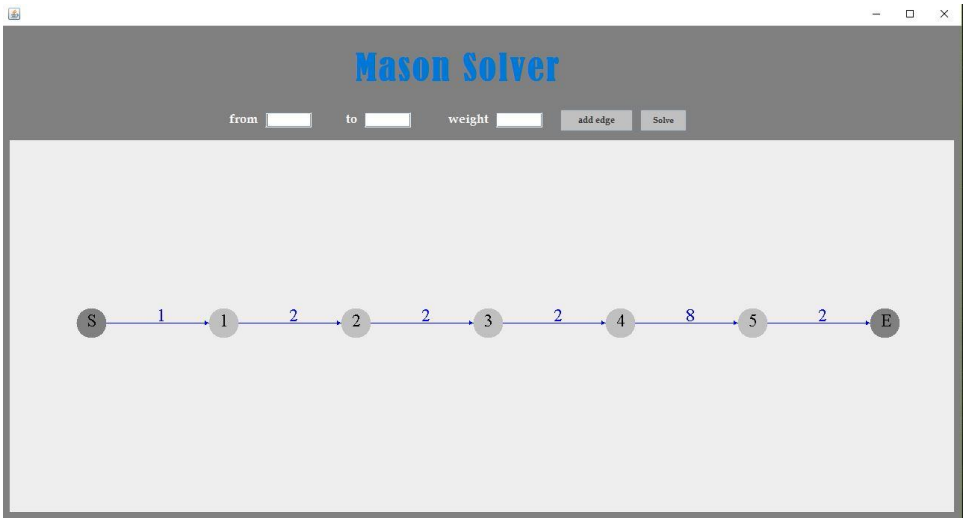



Mason's gain formula
enter maximum number of node
a
Draw
Dialog
Invalid number
OK
Mason's gain formula
enter maximum number of node
7
Draw

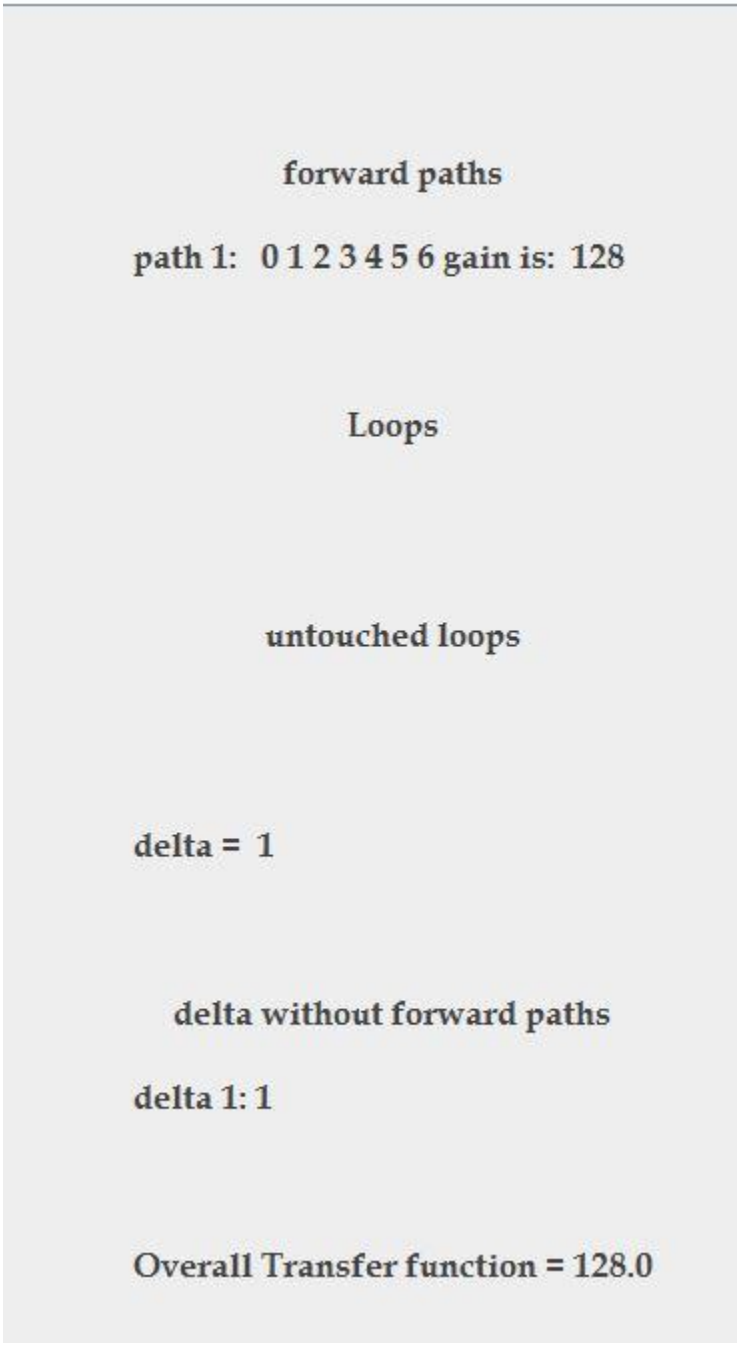
2- enter the edge in displayed fields as shown.



3- after entering the edges
[s>1 : 1] [1>2 : 2] [2>3 :2] [3>4 : 2] [4>5 : 8] [5>e : 2]

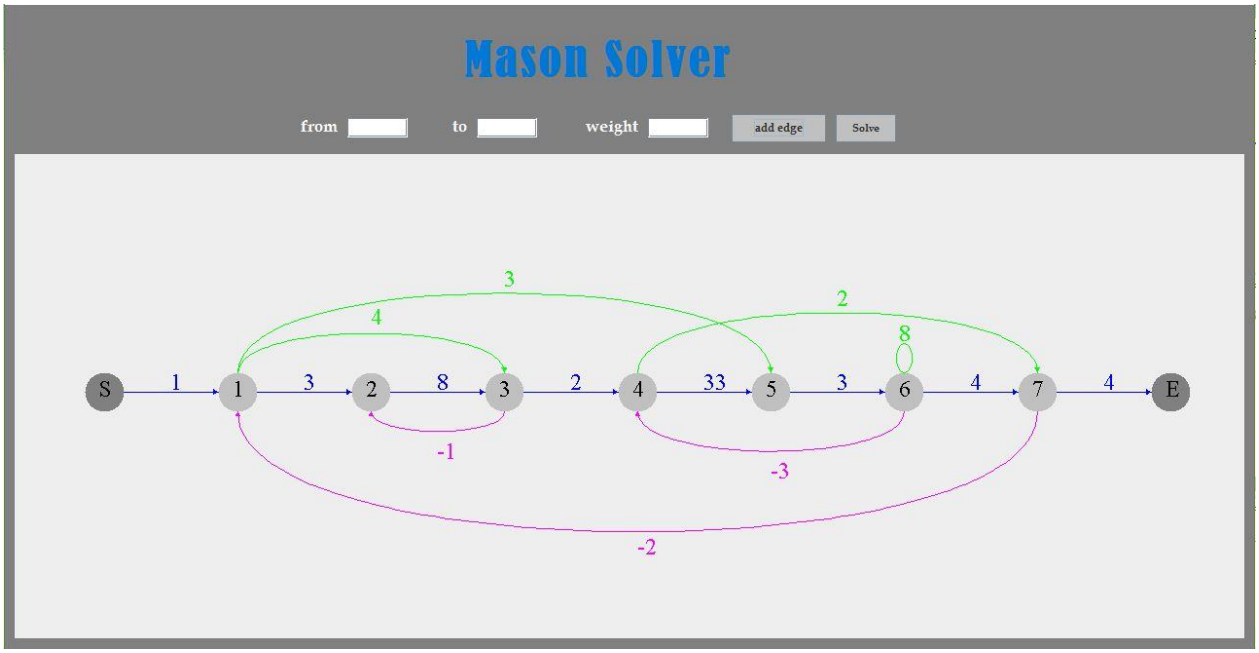


4- Result will be displayed in another window as following,



let's show other example runs,

3.2 SAMPLE RUNS



sample1

forward paths

path 1: 0 1 2 3 4 5 6 7 8 gain is: 76032
path 2: 0 1 2 3 4 7 8 gain is: 384
path 3: 0 1 3 4 5 6 7 8 gain is: 12672
path 4: 0 1 3 4 7 8 gain is: 64
path 5: 0 1 5 6 7 8 gain is: 144

Loops

loop 1: 1 3 4 7 1 gain is: -32
loop 2: 1 5 6 7 1 gain is: -72
loop 3: 1 2 3 4 7 1 gain is: -192
loop 4: 1 5 6 4 7 1 gain is: 108
loop 5: 1 3 4 5 6 7 1 gain is: -6336
loop 6: 1 2 3 4 5 6 7 1 gain is: -38016
loop 7: 2 3 2 gain is: -8
loop 8: 4 5 6 4 gain is: -297
loop 9: 6 6 gain is: 8

untouched loops

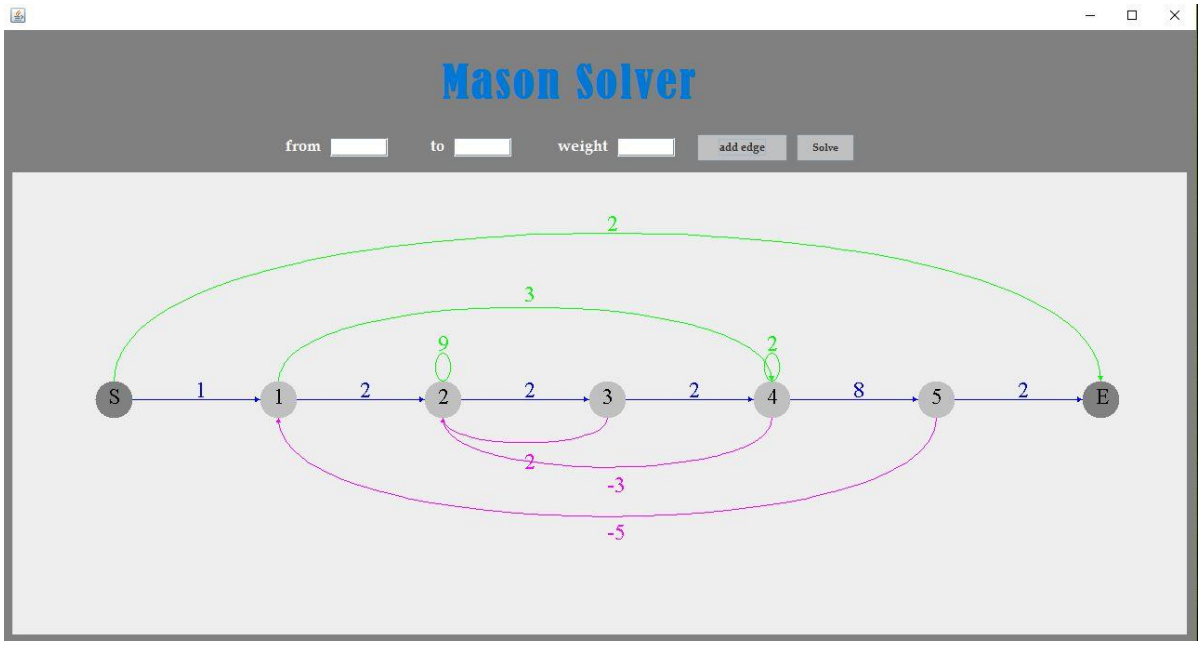
untouched loop: 1 of rank: 2 gain is:-256
untouched loop: 2 of rank: 2 gain is:576
untouched loop: 3 of rank: 2 gain is:-1536
untouched loop: 4 of rank: 2 gain is:-864
untouched loop: 5 of rank: 2 gain is:2376
untouched loop: 6 of rank: 2 gain is:-64

delta = 45070

delta without forward paths

delta 1: 1
delta 2: -7
delta 3: 1
delta 4: -7
delta 5: 9

Overall Transfer function = 1.9273130685600177


Mason Solver
from
to
weight
add edge
Solve
2
3
9
2
S
1
1
2
2
2
3
2
4
8
5
2
E
2
-3
-5

sample2

forward paths

path 1: 0 1 2 3 4 5 6 gain is: 128
path 2: 0 1 4 5 6 gain is: 48
path 3: 0 6 gain is: 2

Loops

loop 1: 1 4 5 1 gain is: -120
loop 2: 1 2 3 4 5 1 gain is: -320
loop 3: 2 2 gain is: 9
loop 4: 2 3 2 gain is: 4
loop 5: 2 3 4 2 gain is: -12
loop 6: 4 4 gain is: 2

untouched loops

untouched loop: 1 of rank: 2 gain is:-1080
untouched loop: 2 of rank: 2 gain is:-480
untouched loop: 3 of rank: 2 gain is:18
untouched loop: 4 of rank: 2 gain is:8

delta = -1096

delta without forward paths

delta 1: 1
delta 2: 1
delta 3: -12
delta 4: -1096

Overall Transfer function = -0.1386861313868613