Lab #1
Signal Flow Graphs

Students:

- Fareeda Mohamed Ali Abouzed – 19016154
- Esraa Hassan Mokhtar Aboshady – 19015407
- Ibrahim Tarek Ibrahim Abdelaal – 19015167
- Ziad Reda Ali Saad – 19015717

Problem Statement

Implementation of Signal flow graph representation of the system.

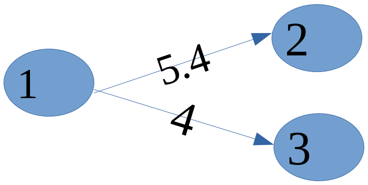
Main Features of the program

- Adding Nodes
- Adding Branches
- Numeric Gains for each Branch
- Solve the Signal Flow Graph
- Clear The Window

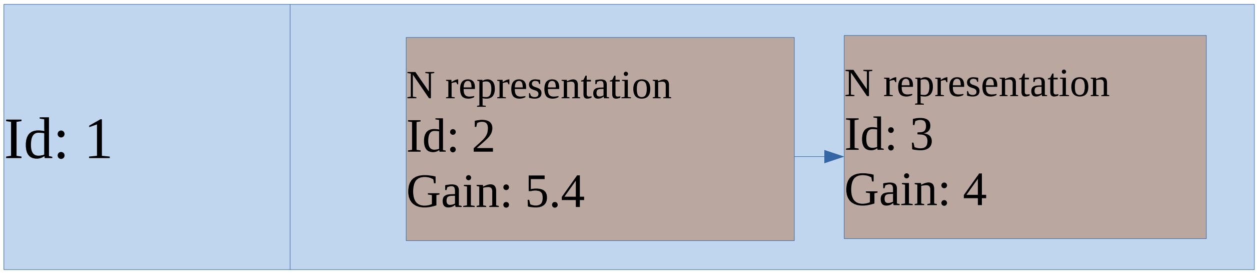
Data Structure

- Node

  each Node is represented as a List of N, and its id
  each N is an id, gain



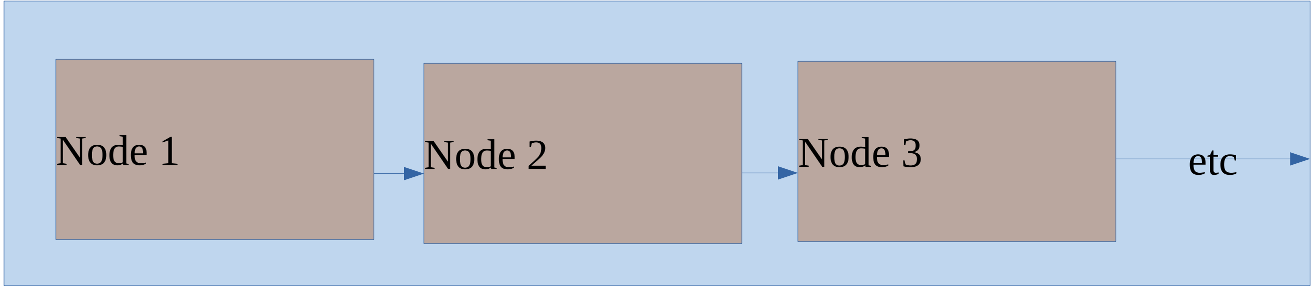
Node representation



- Graph

  Graph is simply a list of Nodes

  Graph representation



Built-in data structures in Java

- Map interface HashMap class
- List interface LinkedList class
- Arrays
- Set interface HashSet class

Main modules

we have used Java, and Javafx.
Main modules:

- controls
- fxml
- graphics
- media

Algorithms used

To get Forward Paths, and Feedback Loops, we have used DFS (Depth-First Search) traverse Algorithm:

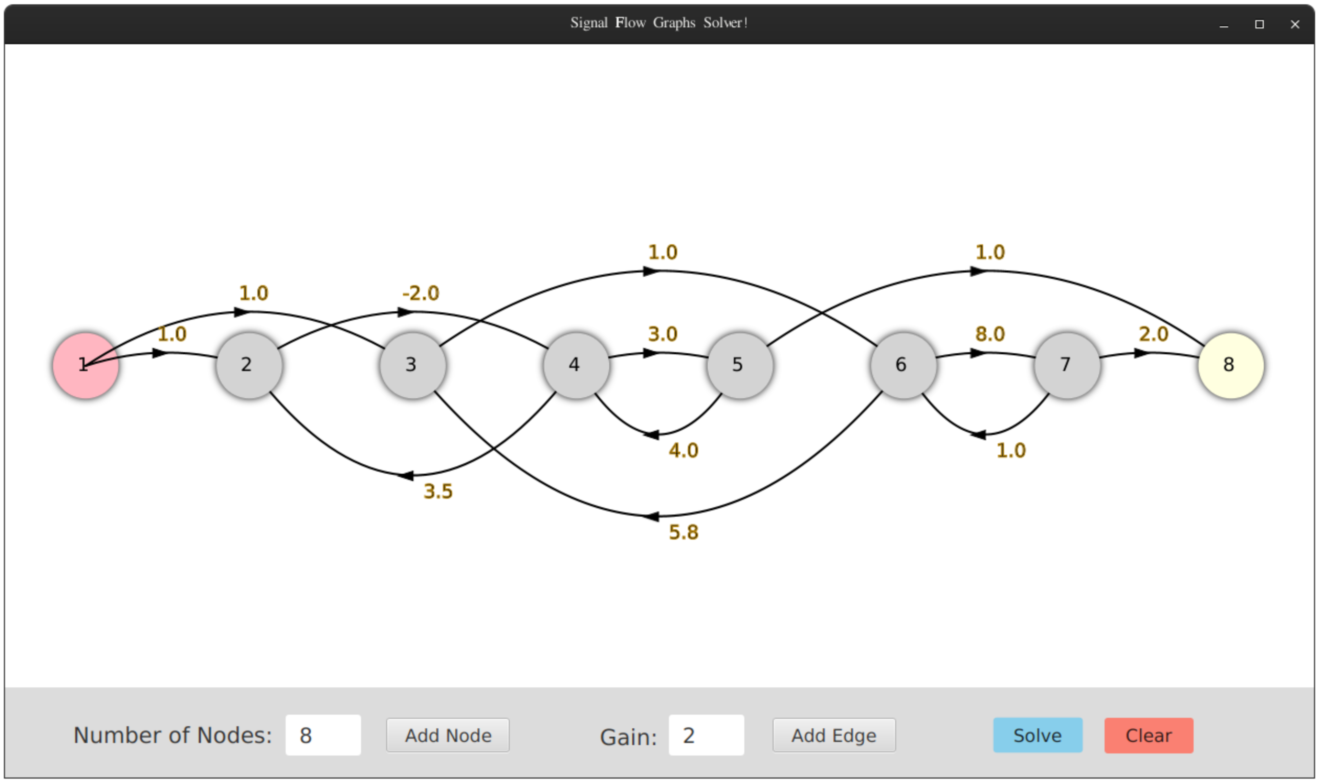
```
function dfs(src, des, graph, visited, temp, res):
    if src == des:
        temp.add(des)
        res.add(temp)
        return

    visited[src] = true
    temp.add(src)
    for every child → graph[src]:
        if !visited[child]:
            dfs(child, des, graph, visited, temp, res)
            temp.remove(temp_size -1) // last added
    visited[src] = false
```

To get Non Touching Loops, and Delta, we have used normal for Loops.

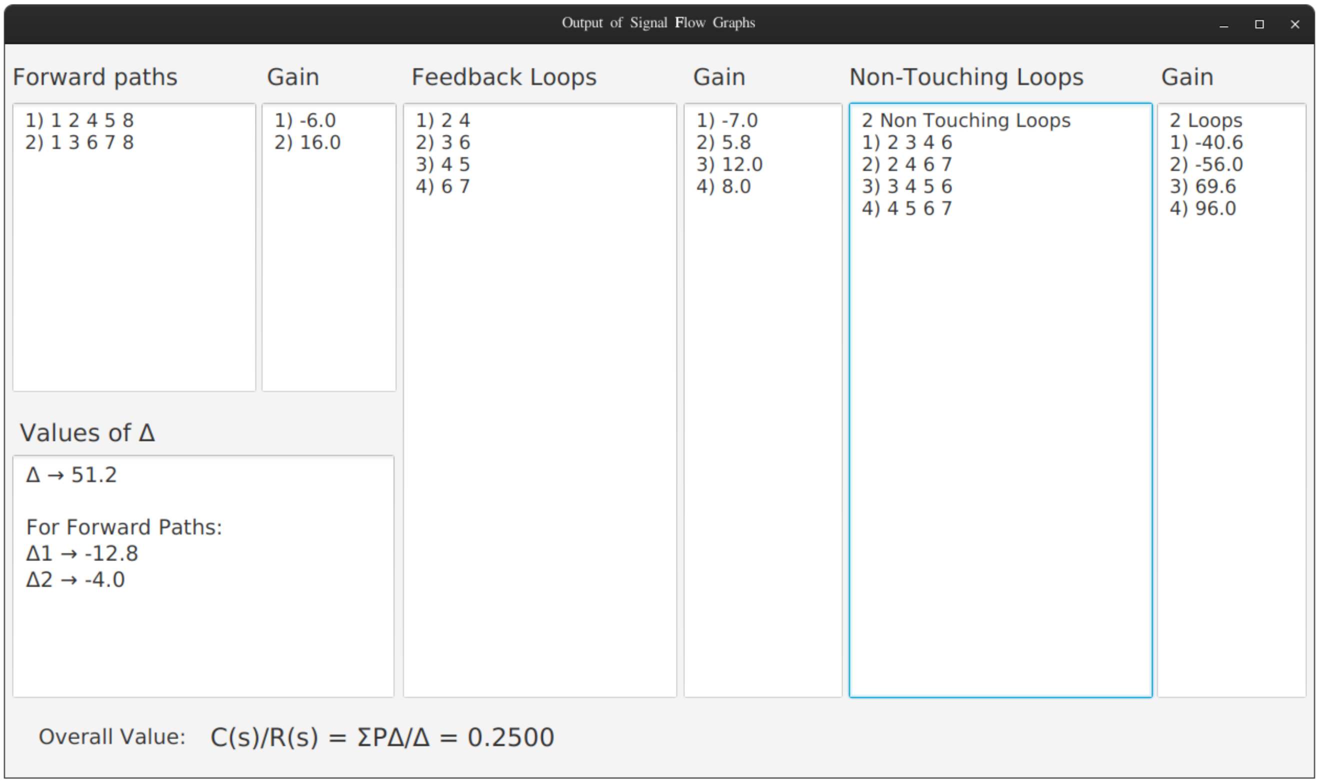
Sample Runs

Test 1:



Number of Nodes: 8

Gain: 2



Forward paths

1) 1 2 4 5 8
2) 1 3 6 7 8

Gain

1) -6.0
2) 16.0

Feedback Loops

1) 2 4
2) 3 6
3) 4 5
4) 6 7

Gain

1) -7.0
2) 5.8
3) 12.0
4) 8.0

Non-Touching Loops

2 Non Touching Loops
1) 2 3 4 6
2) 2 4 6 7
3) 3 4 5 6
4) 4 5 6 7

Gain

2 Loops
1) -40.6
2) -56.0
3) 69.6
4) 96.0

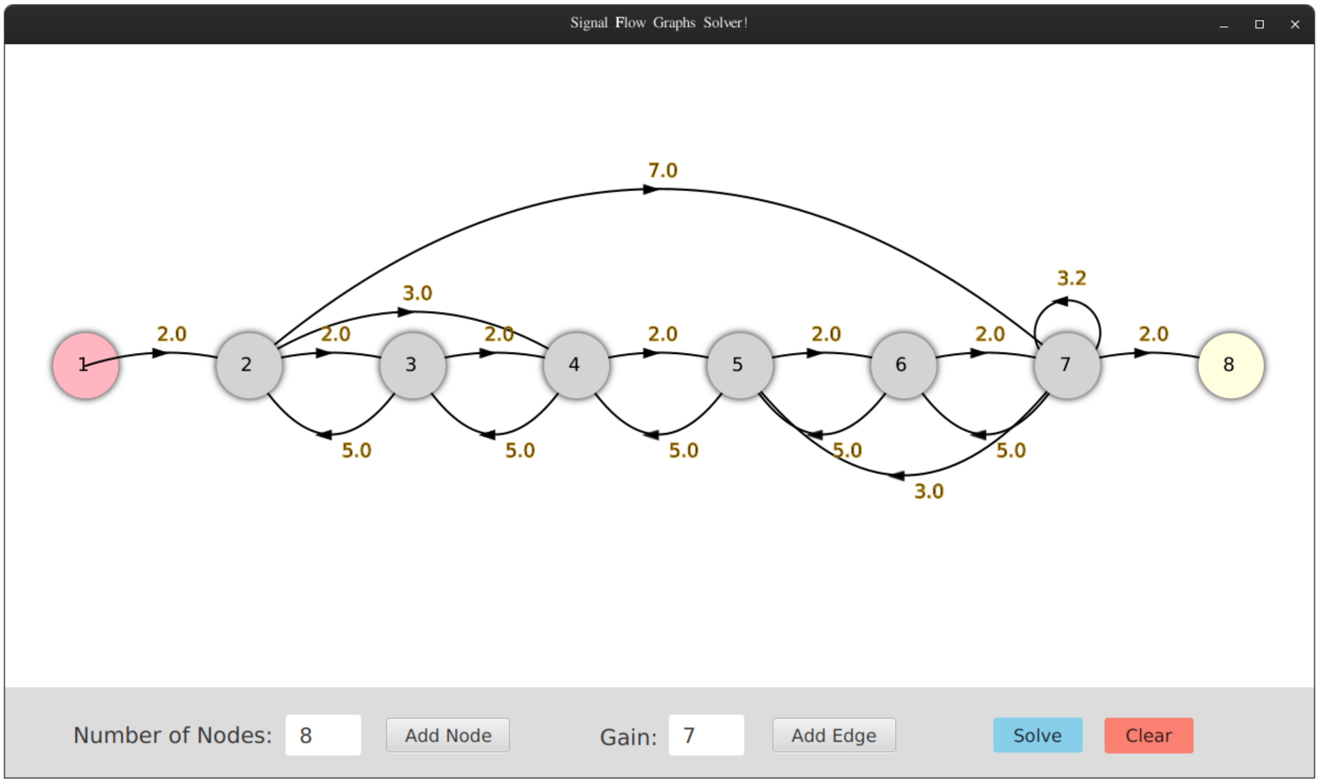
Values of Δ

Δ → 51.2

For Forward Paths:
Δ1 → -12.8
Δ2 → -4.0

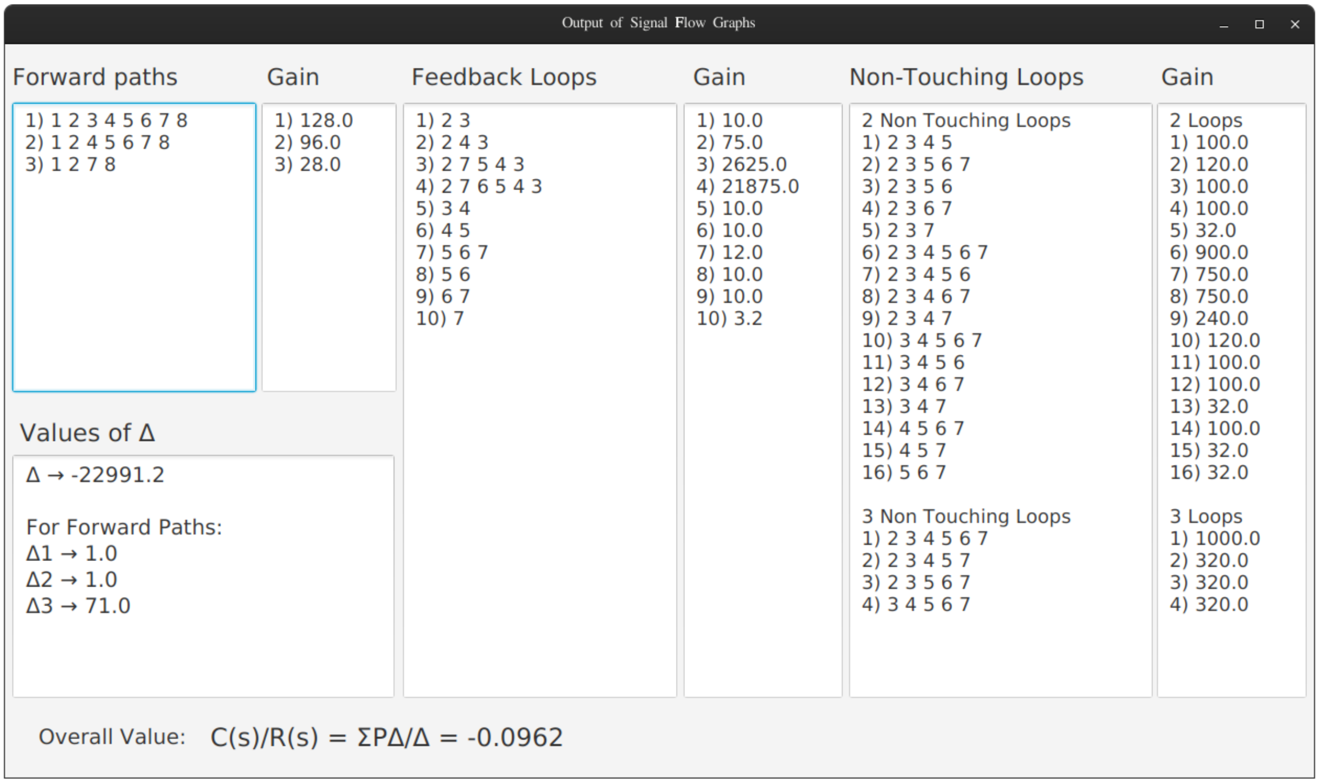
Overall Value: C(s)/R(s) = ΣPΔ/Δ = 0.2500

Test 2:



Number of Nodes: 8

Gain: 7



Forward paths	Gain
1) 1 2 3 4 5 6 7 8	1) 128.0
2) 1 2 4 5 6 7 8	2) 96.0
3) 1 2 7 8	3) 28.0

Feedback Loops	Gain
1) 2 3	1) 10.0
2) 2 4 3	2) 75.0
3) 2 7 5 4 3	3) 2625.0
4) 2 7 6 5 4 3	4) 21875.0
5) 3 4	5) 10.0
6) 4 5	6) 10.0
7) 5 6 7	7) 12.0
8) 5 6	8) 10.0
9) 6 7	9) 10.0
10) 7	10) 3.2

Non-Touching Loops	Gain
2 Non Touching Loops	2 Loops
1) 2 3 4 5	1) 100.0
2) 2 3 5 6 7	2) 120.0
3) 2 3 5 6	3) 100.0
4) 2 3 6 7	4) 100.0
5) 2 3 7	5) 32.0
6) 2 3 4 5 6 7	6) 900.0
7) 2 3 4 5 6	7) 750.0
8) 2 3 4 6 7	8) 750.0
9) 2 3 4 7	9) 240.0
10) 3 4 5 6 7	10) 120.0
11) 3 4 5 6	11) 100.0
12) 3 4 6 7	12) 100.0
13) 3 4 7	13) 32.0
14) 4 5 6 7	14) 100.0
15) 4 5 7	15) 32.0
16) 5 6 7	16) 32.0
3 Non Touching Loops	3 Loops
1) 2 3 4 5 6 7	1) 1000.0
2) 2 3 4 5 7	2) 320.0
3) 2 3 5 6 7	3) 320.0
4) 3 4 5 6 7	4) 320.0

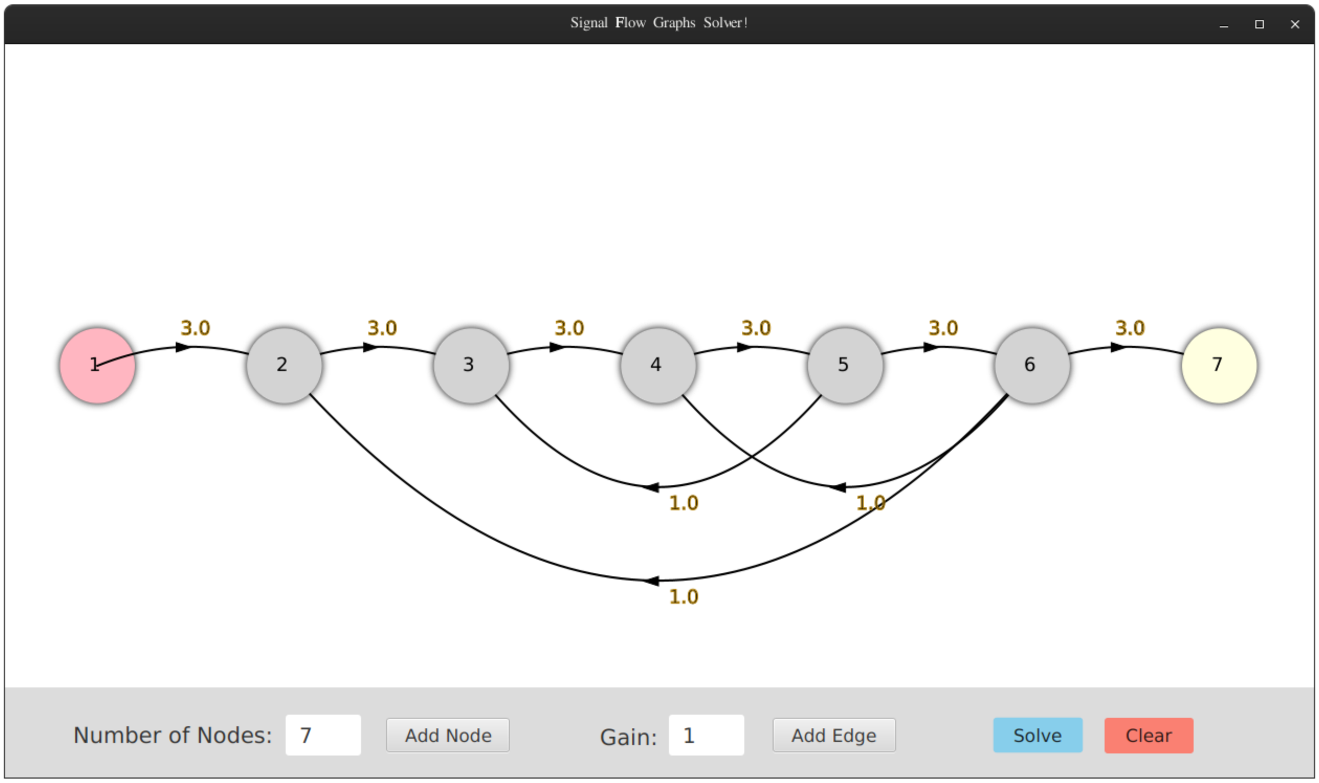
Values of Δ

Δ → -22991.2

For Forward Paths:
Δ1 → 1.0
Δ2 → 1.0
Δ3 → 71.0

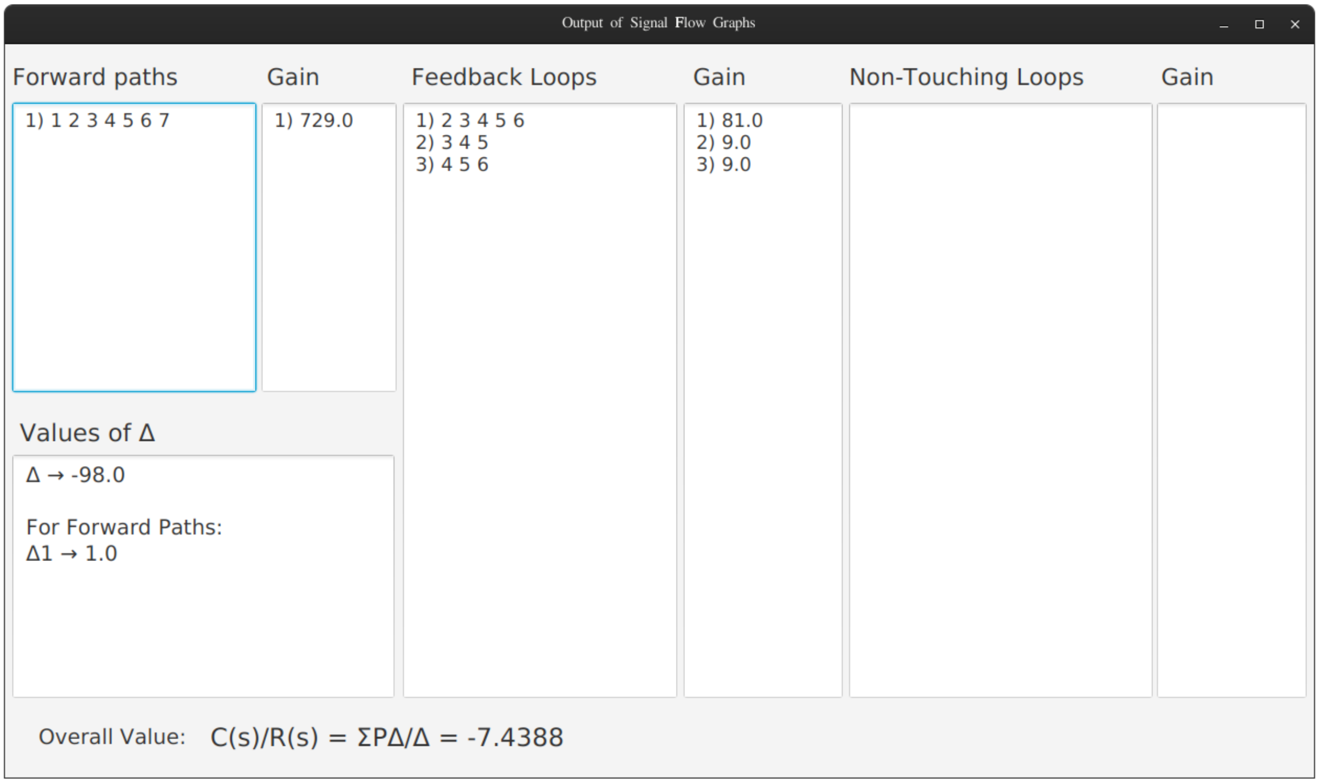
Overall Value: C(s)/R(s) = ΣPΔ/Δ = -0.0962

Test 3:



Number of Nodes: 7

Gain: 1



Forward paths

1) 1 2 3 4 5 6 7

Gain

1) 729.0

Feedback Loops

1) 2 3 4 5 6
2) 3 4 5
3) 4 5 6

Gain

1) 81.0
2) 9.0
3) 9.0

Non-Touching Loops

Gain

Values of Δ

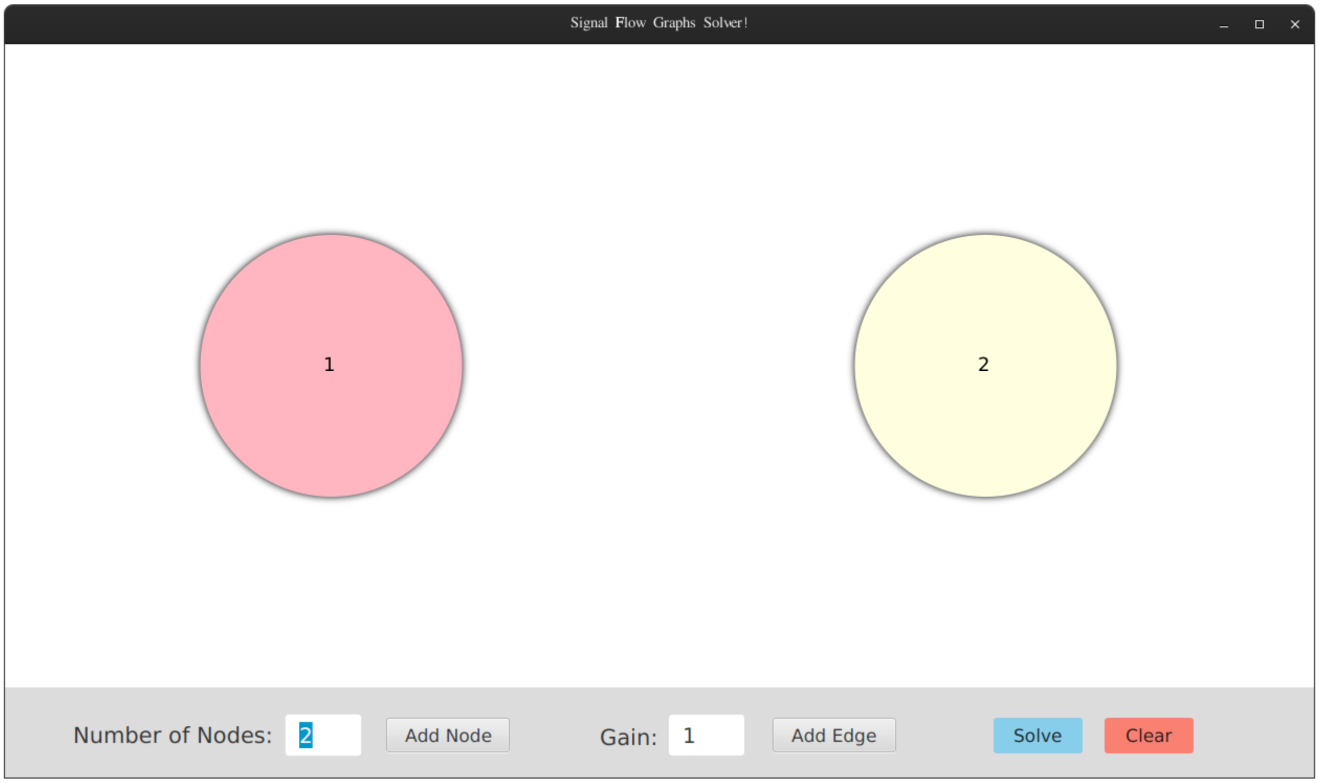
Δ → -98.0

For Forward Paths:
Δ1 → 1.0

Overall Value: C(s)/R(s) = ΣPΔ/Δ = -7.4388

Simple User Guide

Main Window when Open:

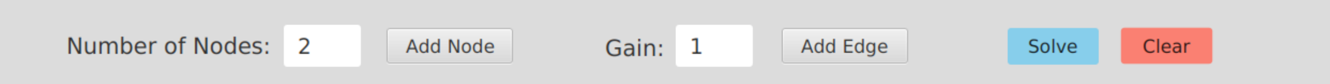


Assumption:

Node 1 → Source Node

Node 2 → Destination Node

Tools:



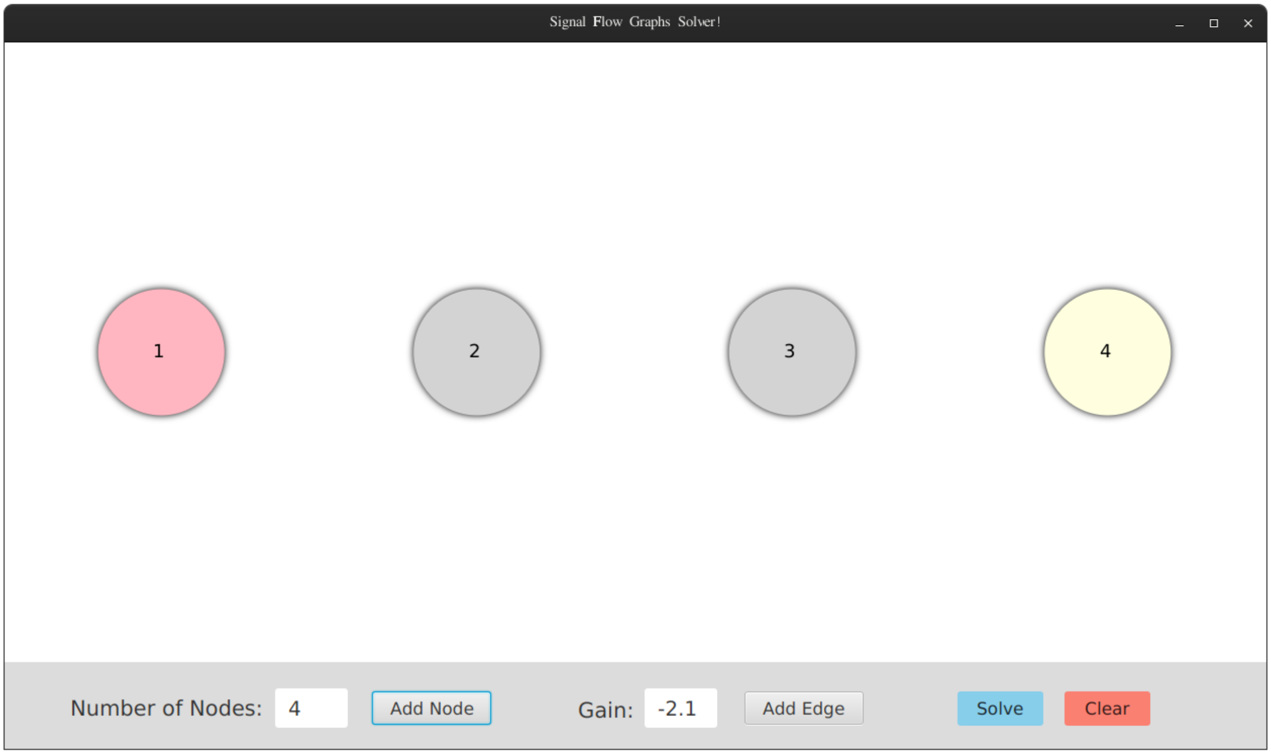
Number of Nodes: 2

Gain: 1

To add New Node:

- Right Click On Button (Add Node)

  The program will count “width / num_of_nodes” and return the new Nodes in their right place
- Number of Nodes Text Area will be increased by one
- Like:

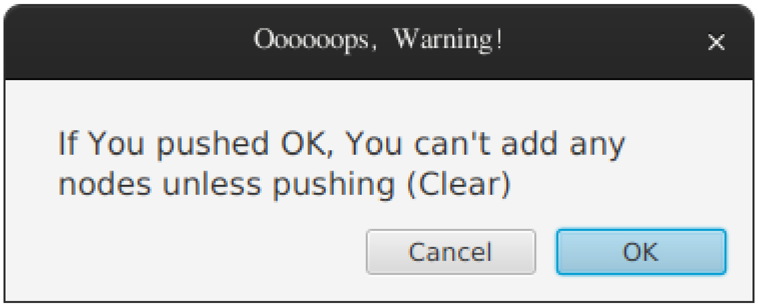


Number of Nodes: 4

Gain: -2.1

To add Edge:

- First Time you Press Right Click a Warning will pop up, saying if OK, you can't add any new Node unless you press Clear and Clear the scene



Oooooops, Warning!

If You pushed OK, You can't add any nodes unless pushing (Clear)

- If OK, you don't wanna any New Nodes but Branches
- if CANCEL, you can continue adding Nodes
- To draw a Branch:
- First Right Click on the first Node
- Second Right Click on the second Node
  (sound pop up as a message of passing, direction from first → second)
- Some constraints here:
  - No branch from a Specific Node to 1 (source)
  - No branch from last Node (destination) to any Node
  - No more than one Branch from a node to another

  (sound pop up a message of a constraint has been hacked)
- Default gain = 1, Gain text Area accepts only numbers (negative, or positive real numbers)

To Solve:

- Right Click on Solve Button, a new window will pop up as containing
  - Forward Paths
  - Feedback Loops
  - Non touching Loops
    - and their gain
  - Delta Values
  - Over all Value

To Clear The Scene:

- Right Click on Clear Button, congratulations you’re clean now :)

How To Run

- clone The repo (https://github.com/farida52369/Signal-Flow-Graph)
- download java (version 11)
- download javaFX (version 17) in your PC
- add javaFX library to program structure
- run HelloApplication.java
- don’t forget to enjoy the program ☺