Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
CS241: Automatic Control

Signal Flow Graph

Reported By

Amr Khaled #28

Mohamed Alaa #42

Problem statement

Design software that calculates the overall gain of a system represented as signal flow graph

Main Features

Easy to use and friendly user interface

Data Structure

- Adjacency list to store the edges
- Adjacency matrix to store edges' gain

Main modules

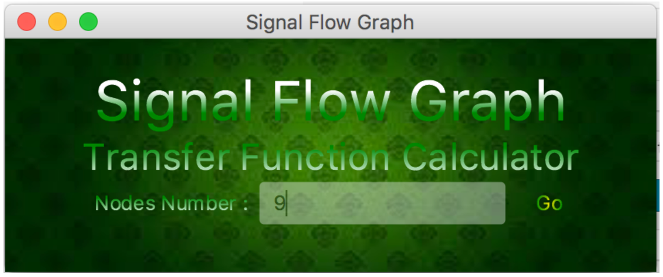
- Drawer: draws the Signal Flow Graph, take the input from the user and show the results
- Solver: calculates the overall gain and store important information about the graph

Algorithms used

- Depth First Search (DFS): to find the forward paths and loops
- Inclusion-Exclusion principle: to get all the combinations of non-touching loops and calculate Δ & Δi

Simple user guide

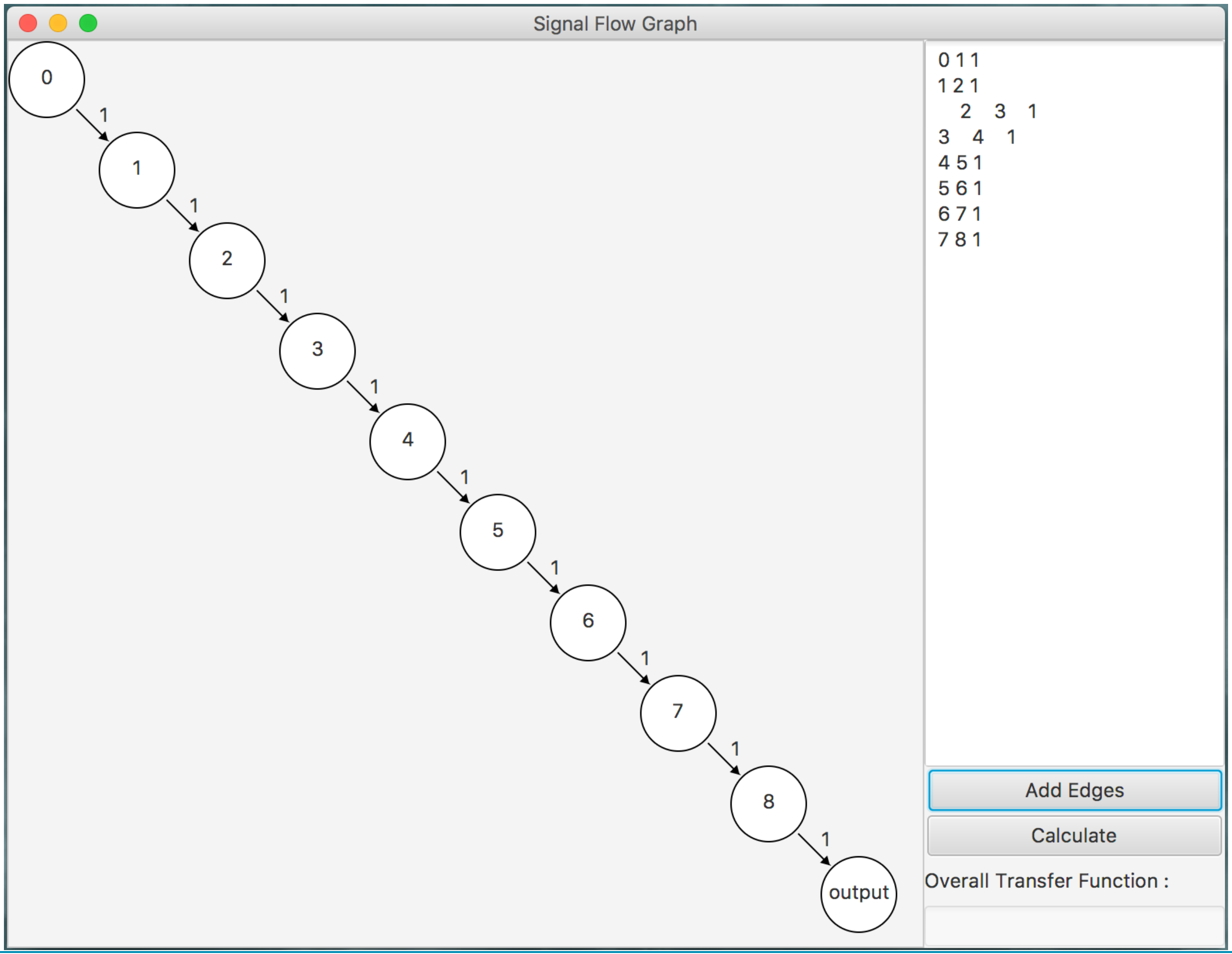
STEP 1: enter number of nodes



Nodes Number : 9

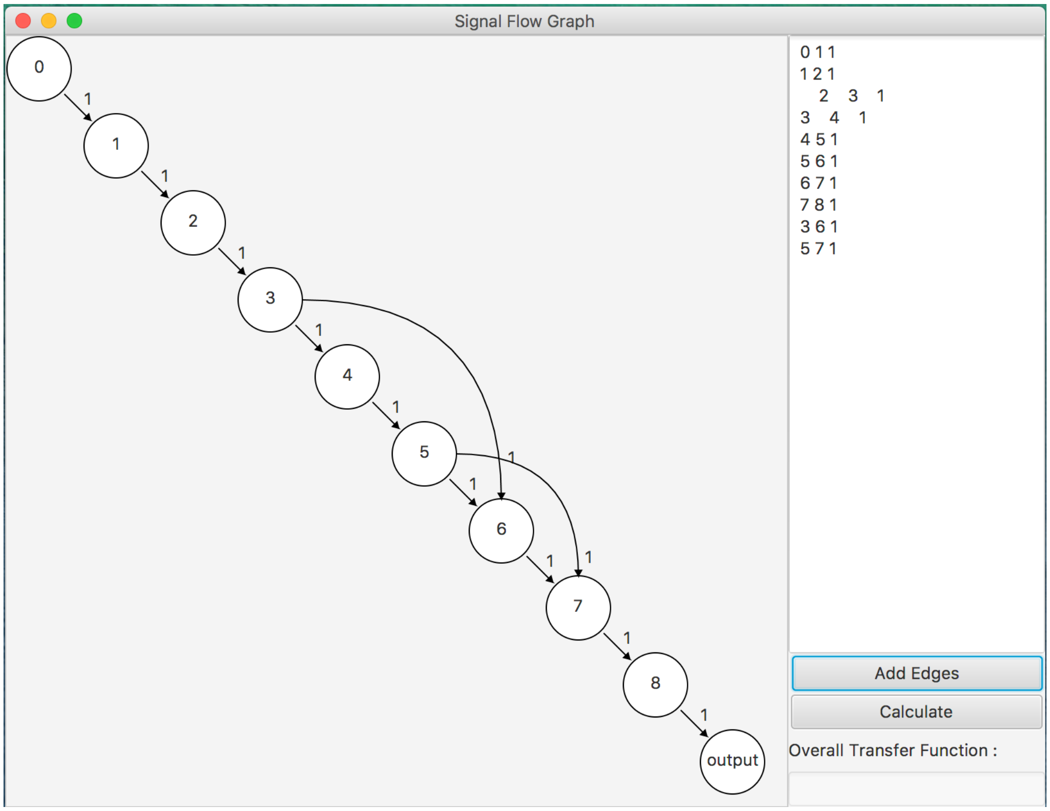
STEP 2: add edges

- *for each edge add its start, end and weight values respectively separated by spaces*
- *each line represents an edge*

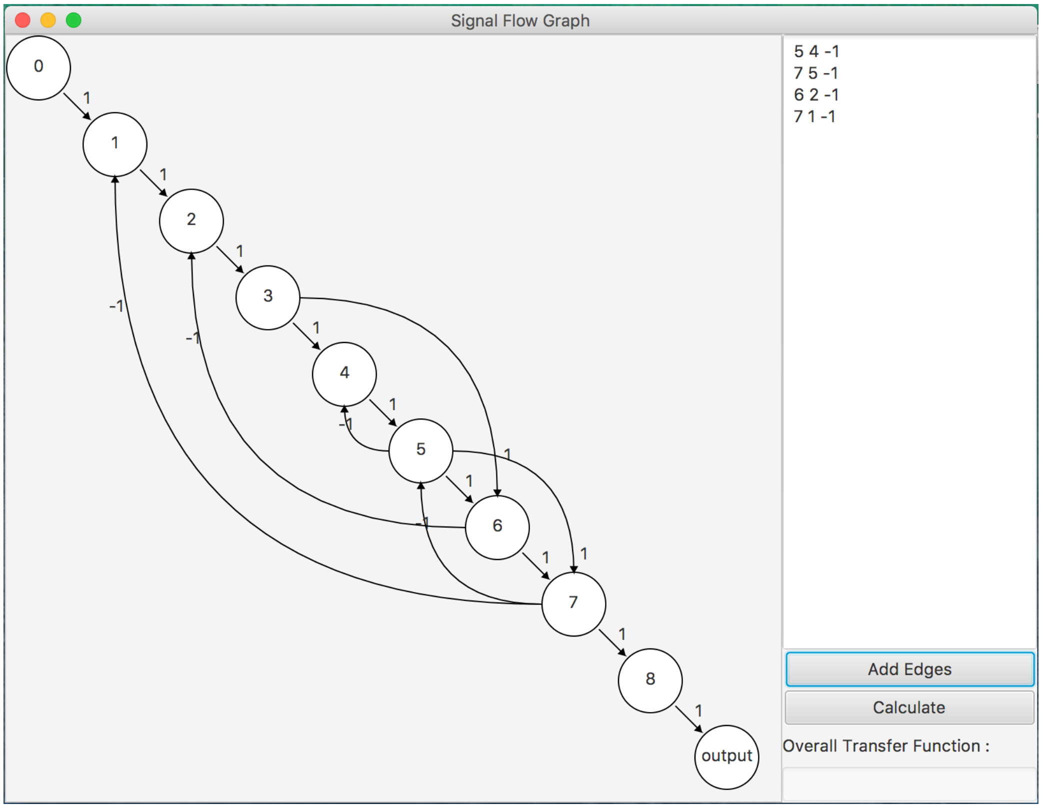


```
0 1 1
1 2 1
  2  3  1
3  4  1
4 5 1
5 6 1
6 7 1
7 8 1
```

User can always add more edge

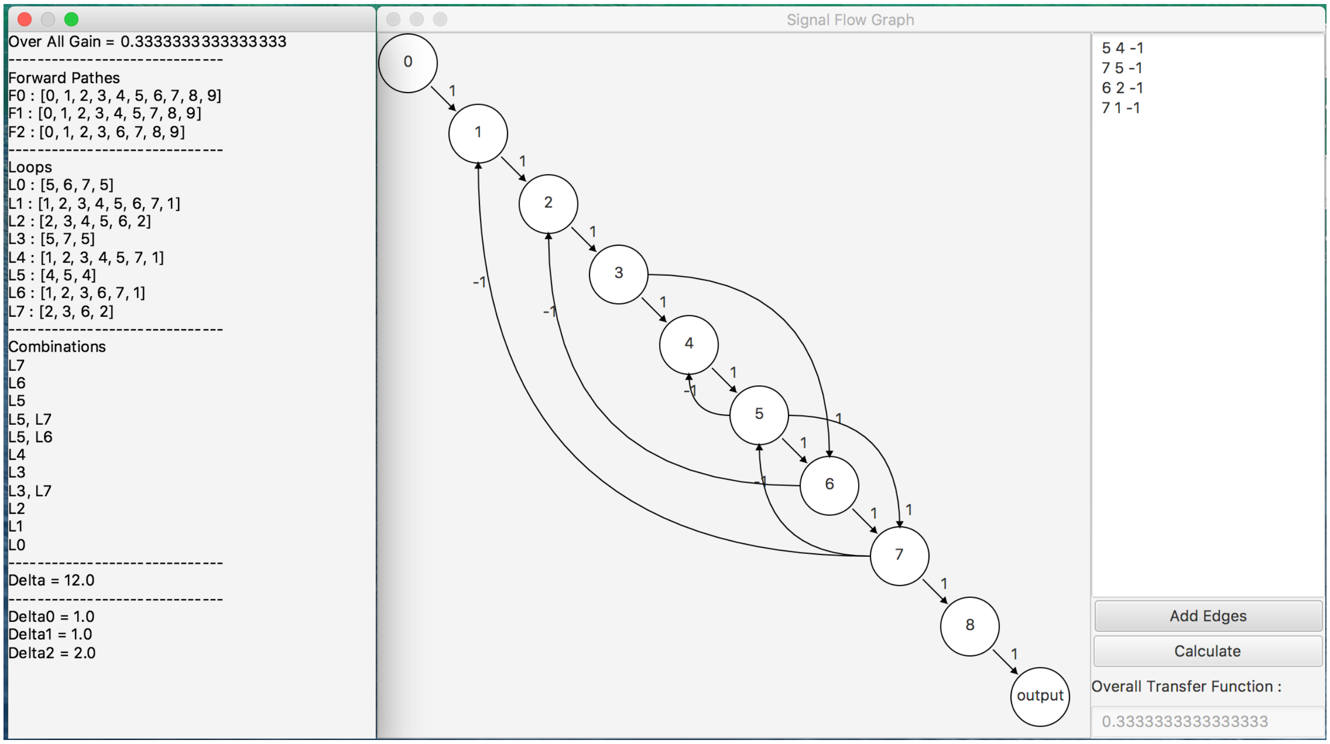


```
0 1 1
1 2 1
   2   3   1
3   4   1
4 5 1
5 6 1
6 7 1
7 8 1
3 6 1
5 7 1
```



```
5 4 -1
7 5 -1
6 2 -1
7 1 -1
```

STEP 3: Calculate



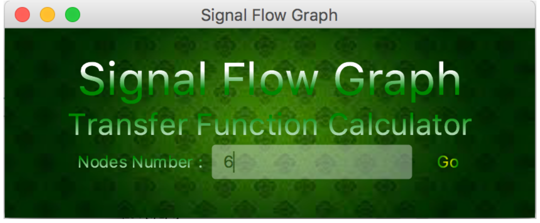
```
Over All Gain = 0.3333333333333333
------------------------------
Forward Pathes
F0 : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
F1 : [0, 1, 2, 3, 4, 5, 7, 8, 9]
F2 : [0, 1, 2, 3, 6, 7, 8, 9]
------------------------------
Loops
L0 : [5, 6, 7, 5]
L1 : [1, 2, 3, 4, 5, 6, 7, 1]
L2 : [2, 3, 4, 5, 6, 2]
L3 : [5, 7, 5]
L4 : [1, 2, 3, 4, 5, 7, 1]
L5 : [4, 5, 4]
L6 : [1, 2, 3, 6, 7, 1]
L7 : [2, 3, 6, 2]
------------------------------
Combinations
L7
L6
L5
L5, L7
L5, L6
L4
L3
L3, L7
L2
L1
L0
------------------------------
Delta = 12.0
------------------------------
Delta0 = 1.0
Delta1 = 1.0
Delta2 = 2.0
```

```
5 4 -1
7 5 -1
6 2 -1
7 1 -1
```

Overall Transfer Function :

0.3333333333333333

Sample runs

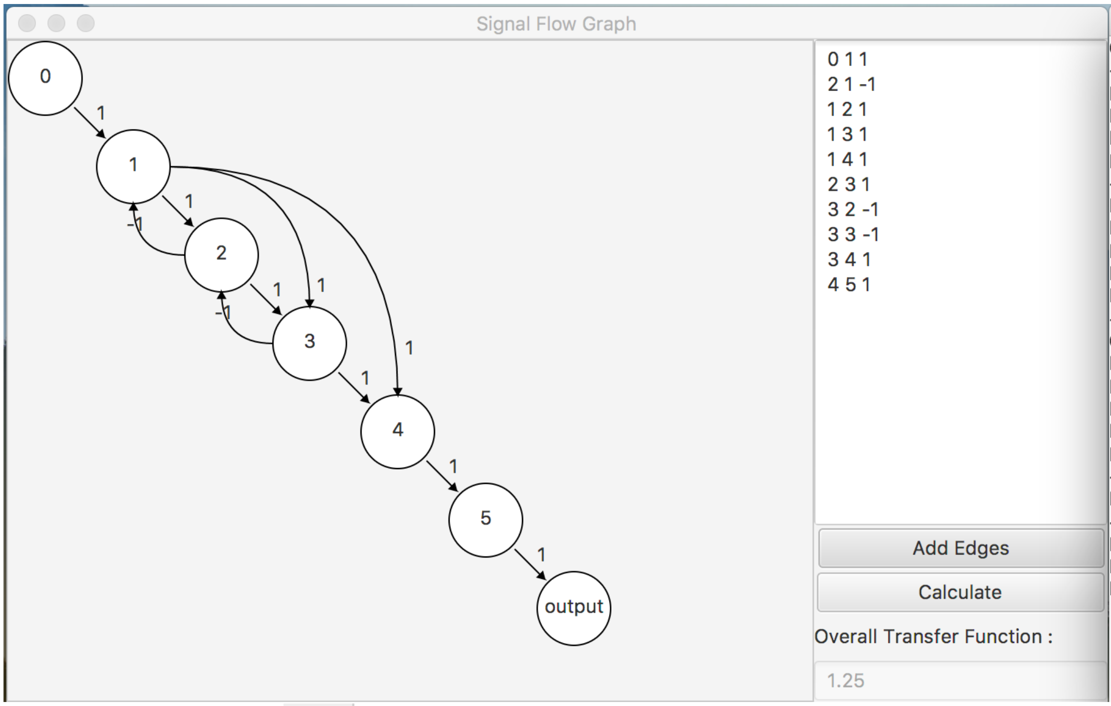


Signal Flow Graph

Signal Flow Graph

Transfer Function Calculator

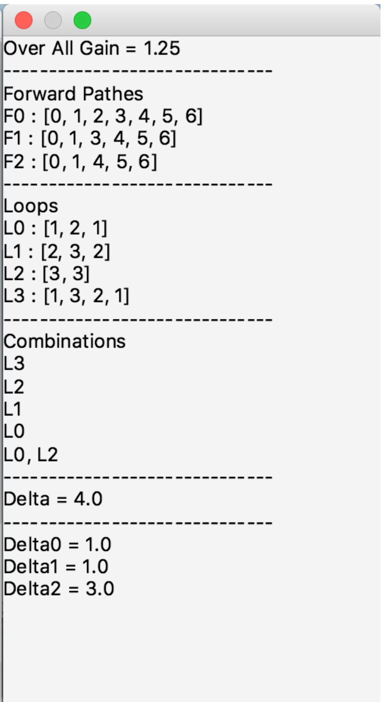
Nodes Number : 6



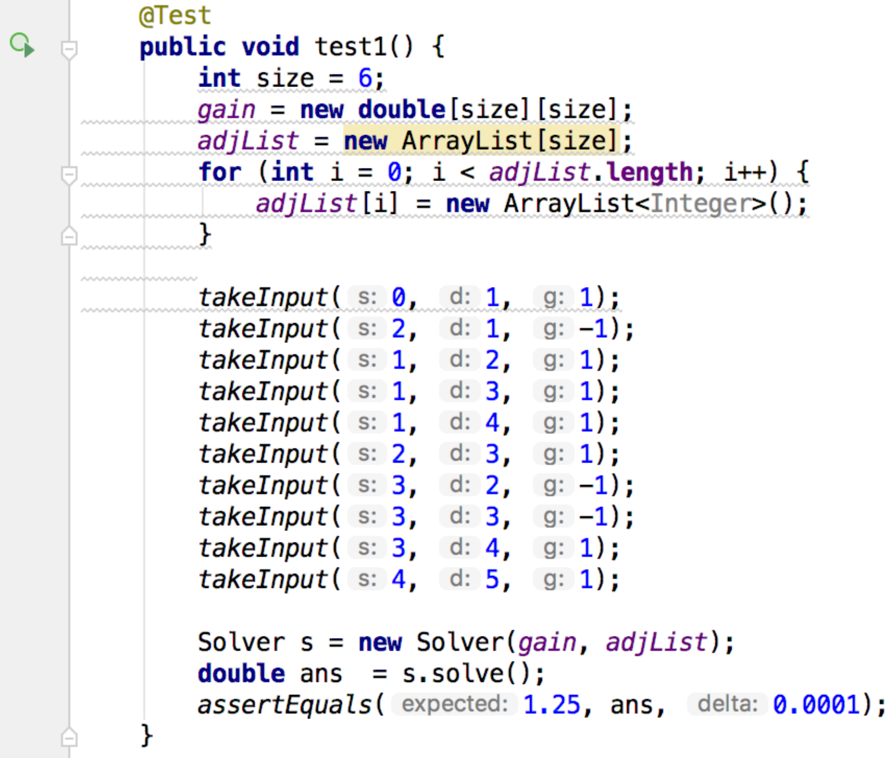
```
0 1 1
2 1 -1
1 2 1
1 3 1
1 4 1
2 3 1
3 2 -1
3 3 -1
3 4 1
4 5 1
```

Overall Transfer Function :

1.25



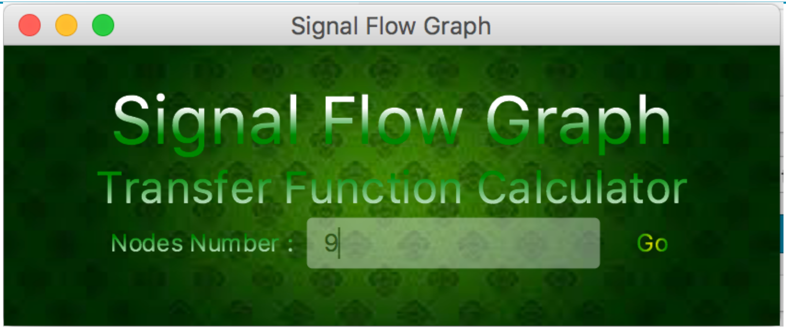
```
Over All Gain = 1.25
------------------------------
Forward Pathes
F0 : [0, 1, 2, 3, 4, 5, 6]
F1 : [0, 1, 3, 4, 5, 6]
F2 : [0, 1, 4, 5, 6]
------------------------------
Loops
L0 : [1, 2, 1]
L1 : [2, 3, 2]
L2 : [3, 3]
L3 : [1, 3, 2, 1]
------------------------------
Combinations
L3
L2
L1
L0
L0, L2
------------------------------
Delta = 4.0
------------------------------
Delta0 = 1.0
Delta1 = 1.0
Delta2 = 3.0
```

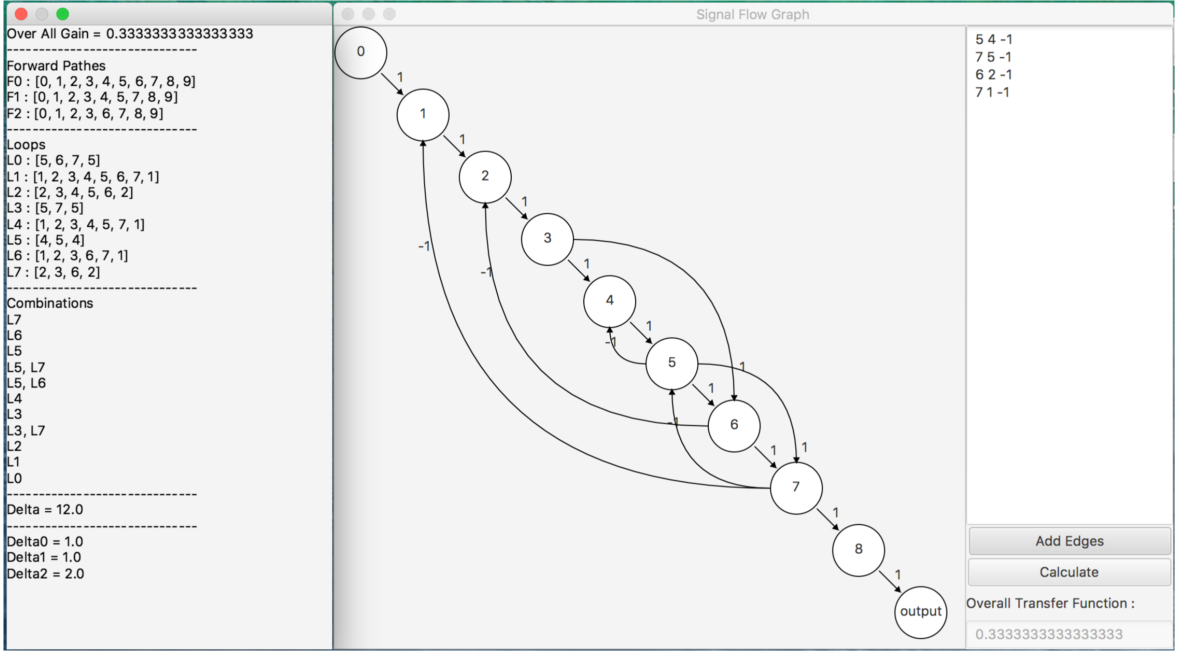


```
@Test
public void test1() {
    int size = 6;
    gain = new double[size][size];
    adjList = new ArrayList[size];
    for (int i = 0; i < adjList.length; i++) {
        adjList[i] = new ArrayList<Integer>();
    }

    takeInput( s: 0,  d: 1,  g: 1);
    takeInput( s: 2,  d: 1,  g: -1);
    takeInput( s: 1,  d: 2,  g: 1);
    takeInput( s: 1,  d: 3,  g: 1);
    takeInput( s: 1,  d: 4,  g: 1);
    takeInput( s: 2,  d: 3,  g: 1);
    takeInput( s: 3,  d: 2,  g: -1);
    takeInput( s: 3,  d: 3,  g: -1);
    takeInput( s: 3,  d: 4,  g: 1);
    takeInput( s: 4,  d: 5,  g: 1);

    Solver s = new Solver(gain, adjList);
    double ans  = s.solve();
    assertEquals( expected: 1.25, ans,  delta: 0.0001);
}
```



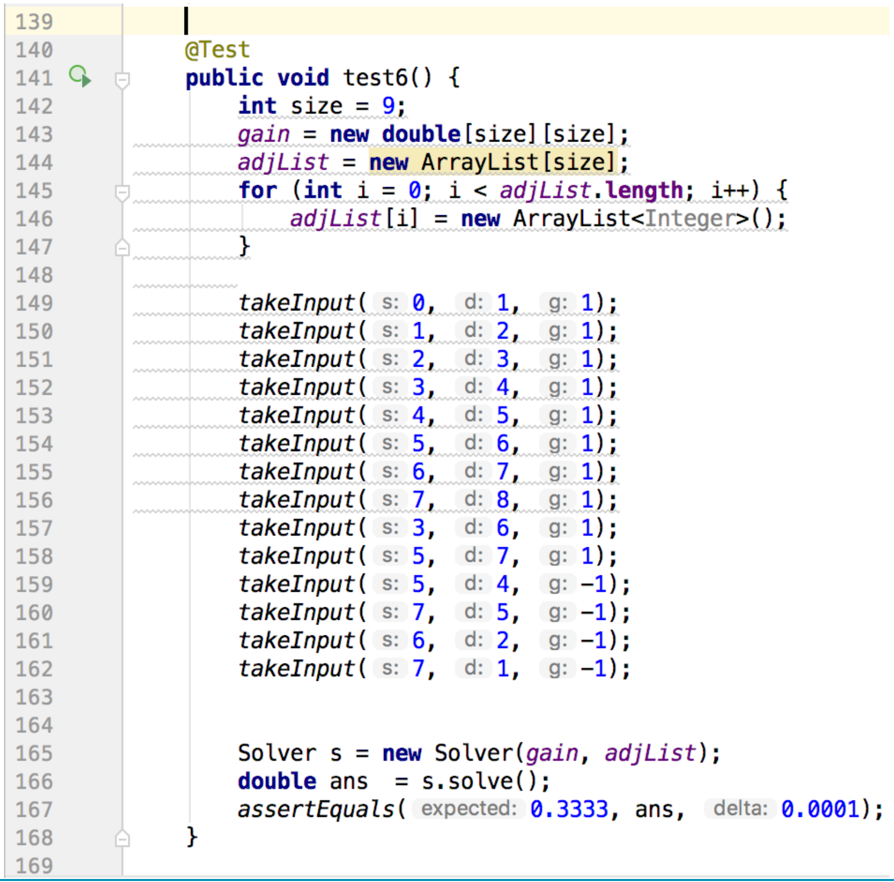


```
Over All Gain = 0.3333333333333333
--------------------------------
Forward Pathes
F0 : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
F1 : [0, 1, 2, 3, 4, 5, 7, 8, 9]
F2 : [0, 1, 2, 3, 6, 7, 8, 9]
--------------------------------
Loops
L0 : [5, 6, 7, 5]
L1 : [1, 2, 3, 4, 5, 6, 7, 1]
L2 : [2, 3, 4, 5, 6, 2]
L3 : [5, 7, 5]
L4 : [1, 2, 3, 4, 5, 7, 1]
L5 : [4, 5, 4]
L6 : [1, 2, 3, 6, 7, 1]
L7 : [2, 3, 6, 2]
--------------------------------
Combinations
L7
L6
L5
L5, L7
L5, L6
L4
L3
L3, L7
L2
L1
L0
--------------------------------
Delta = 12.0
--------------------------------
Delta0 = 1.0
Delta1 = 1.0
Delta2 = 2.0
```

```
5 4 -1
7 5 -1
6 2 -1
7 1 -1
```

Overall Transfer Function :

0.3333333333333333



```java

@Test
public void test6() {
    int size = 9;
    gain = new double[size][size];
    adjList = new ArrayList[size];
    for (int i = 0; i < adjList.length; i++) {
        adjList[i] = new ArrayList<Integer>();
    }

    takeInput( s: 0,  d: 1,  g: 1);
    takeInput( s: 1,  d: 2,  g: 1);
    takeInput( s: 2,  d: 3,  g: 1);
    takeInput( s: 3,  d: 4,  g: 1);
    takeInput( s: 4,  d: 5,  g: 1);
    takeInput( s: 5,  d: 6,  g: 1);
    takeInput( s: 6,  d: 7,  g: 1);
    takeInput( s: 7,  d: 8,  g: 1);
    takeInput( s: 3,  d: 6,  g: 1);
    takeInput( s: 5,  d: 7,  g: 1);
    takeInput( s: 5,  d: 4,  g: -1);
    takeInput( s: 7,  d: 5,  g: -1);
    takeInput( s: 6,  d: 2,  g: -1);
    takeInput( s: 7,  d: 1,  g: -1);


    Solver s = new Solver(gain, adjList);
    double ans  = s.solve();
    assertEquals( expected: 0.3333, ans,  delta: 0.0001);
}

```