Signal Flow Graph Report

19016354	محمد خالد محمد علي
19016113	عمرو عصام محمد الخطيب
19016013	علي حسن علي أحمد الشعراوي
19016582	محمود محمد فتح الله عبد الرازق

Problem Statement:

Giving signal flow graph representation of the system. Assuming that the total number of nodes and numeric branches gains are given, it's required to provide a GUI to draw the signal flow graph showing nodes, branches, and gains, listing all forward paths, individual loops, combinations of *n* non-touching loops, the values of Δ, Δ1, ..., Δm where m is number of forward paths, and overall system transfer function.

Main Features of the program:

- Graphical user interface with the ability to add nodes, branches between nodes, and specify branches' gains.
- Showing the result of:
    - Forward paths, individual loops, combinations of *n* non-touching loops, the values of Δ, Δ1, ..., Δm where m is number of forward paths, and overall system transfer function.

Data Structure:

- One-dimensional arrays to represent loops, nodes, and forward Paths.
- Two-dimensional array to represent branches' gain (row:start node) (column:end node)
- One-dimensional array to store the values needed to draw the nodes and set their gains

Main modules:

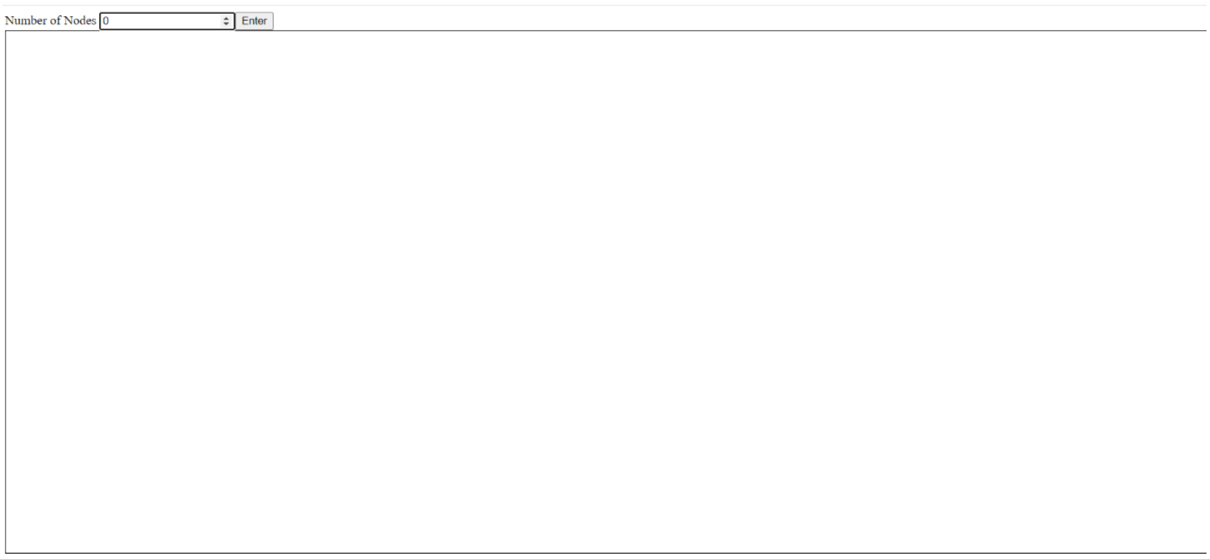
- Class "Node" to represent nodes, its keys, and array of other nodes to which a path goes starting from that node.
- Class "Node Drawer" to store the information about the drawed nodes and draw them.

Algorithms used:

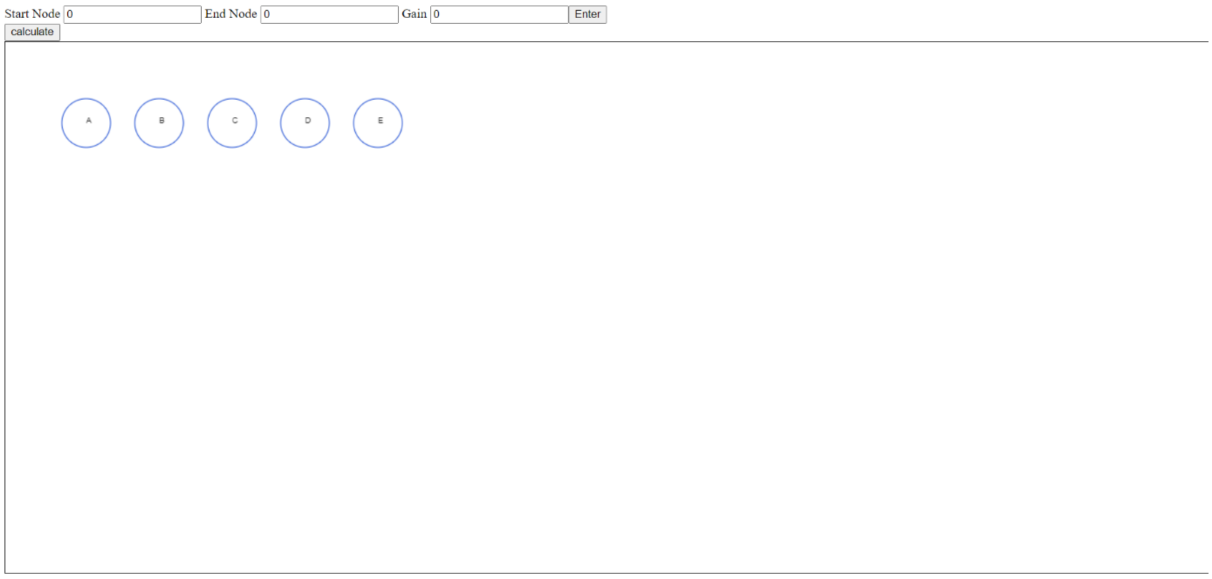
- “pathTraverse”: a recursive function to traverse a graph starting from a given node, trying all combinations of paths to obtain forward paths and loops from the graph, adding them to their corresponding arrays.
- “extractInfo”: function to call “pathTraverse” on each node, extracting the forward paths and filtering the duplicates from the extracted loops.
- We loop on each forward path and loop to calculate gain and push it to forward paths gain array and loop gain array.
- “Gain calculator”: we loop on edges of path or loop and multiple values in the adjacency matrix.
- We get all possible combinations of loops to check touching loops by “check touching validity”.
- “check touching validity”: we loop on each loop combination and check if a node is repeated or not to check validity a node can’t be duplicated.
- “Delta calculator”: we check the validity of each combination of loops so if it is valid we add its gain to the non touching len gain map according to its number of loops in combination.
- “Get delta loops”: get the loops after removing each path to calculate delta for this path.
- “Delta array calculator”: we loop on each path and get loops after removing this path and get all combinations of these loops to calculate delta and push it to the delta array.
- “Transfer function calculator”: from delta array and paths gain and delta we calculate transfer function value.

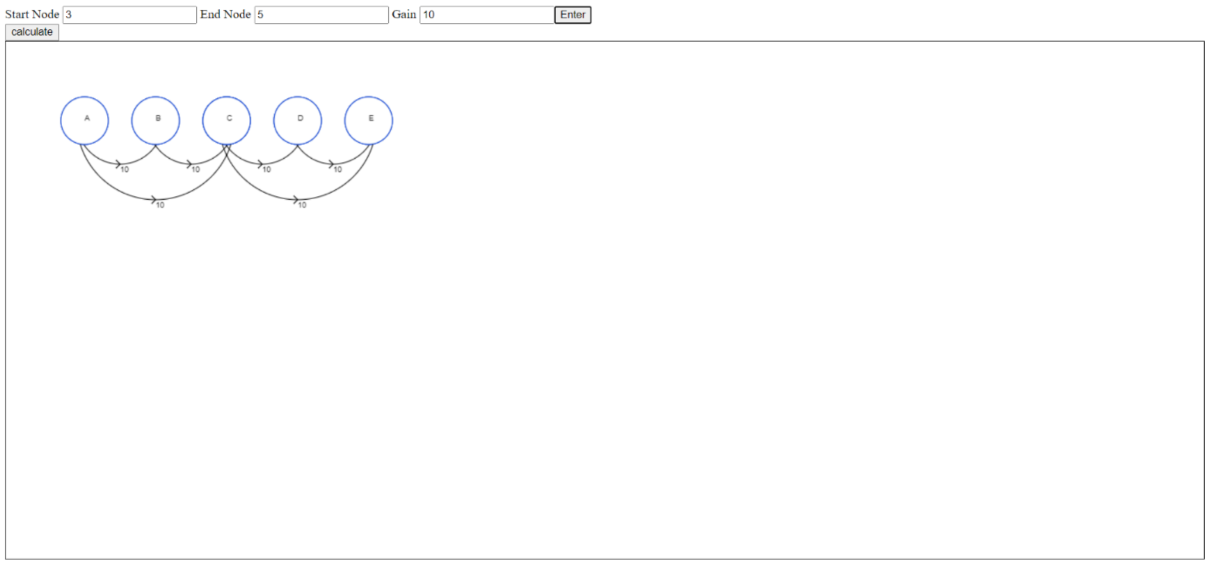
User guide:

At first you enter the total number of nodes in the signal flow graph.

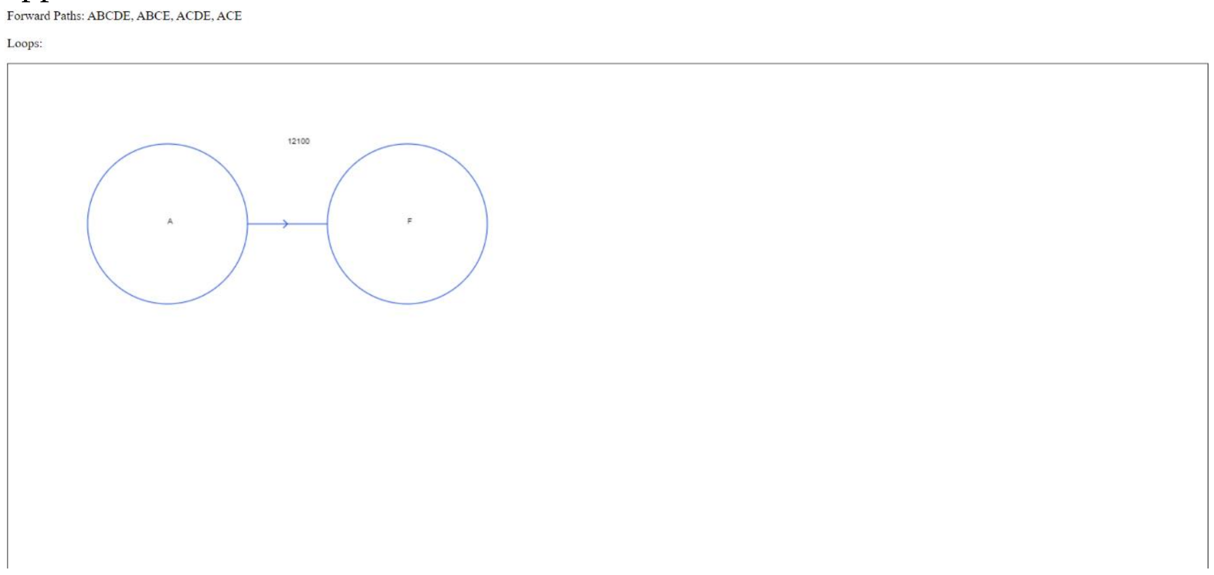


To set a gain between 2 nodes you enter the number of the first node and the number of the second node and the gain and press “Enter”





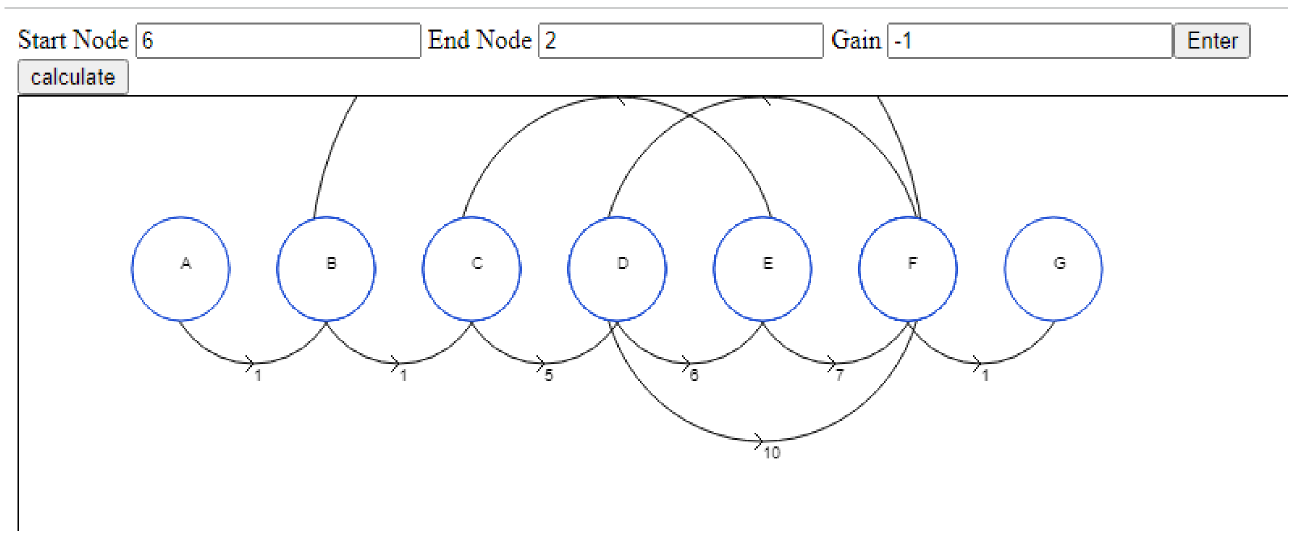
When you finish entering the gains press on “Calculate” and the result will appear.


Forward Paths: ABCDE, ABCE, ACDE, ACE

Loops:

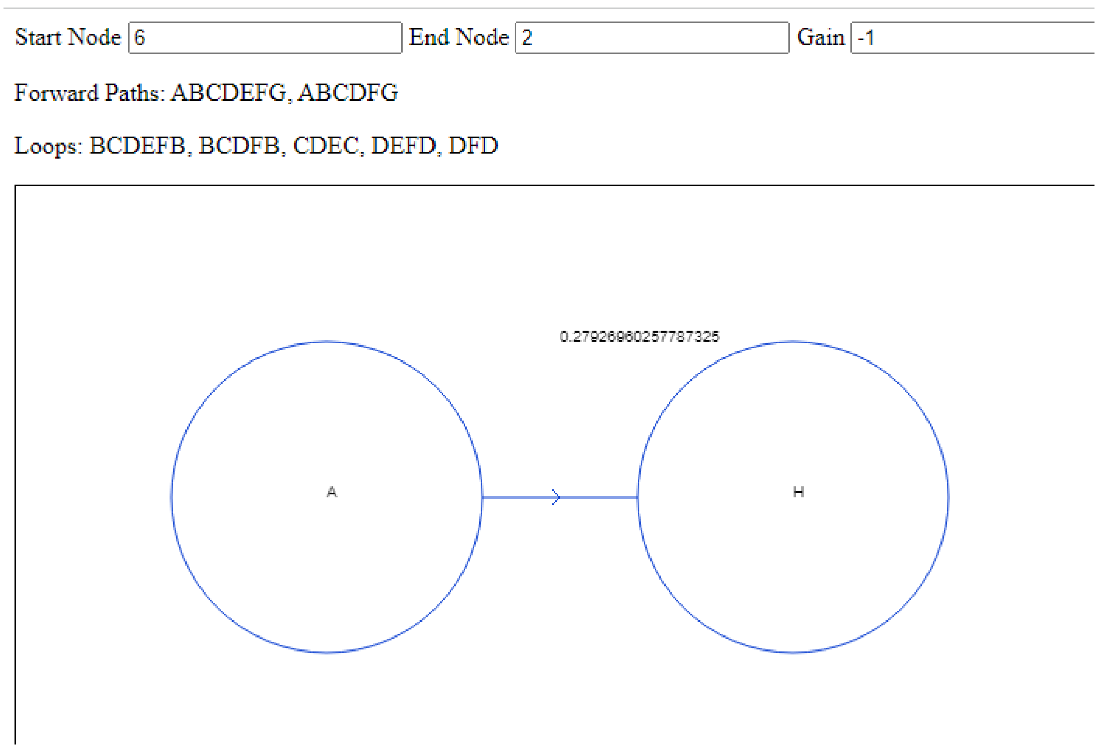
Sample runs:

First run: input


Start Node 6 End Node 2 Gain -1 Enter

calculate

Output

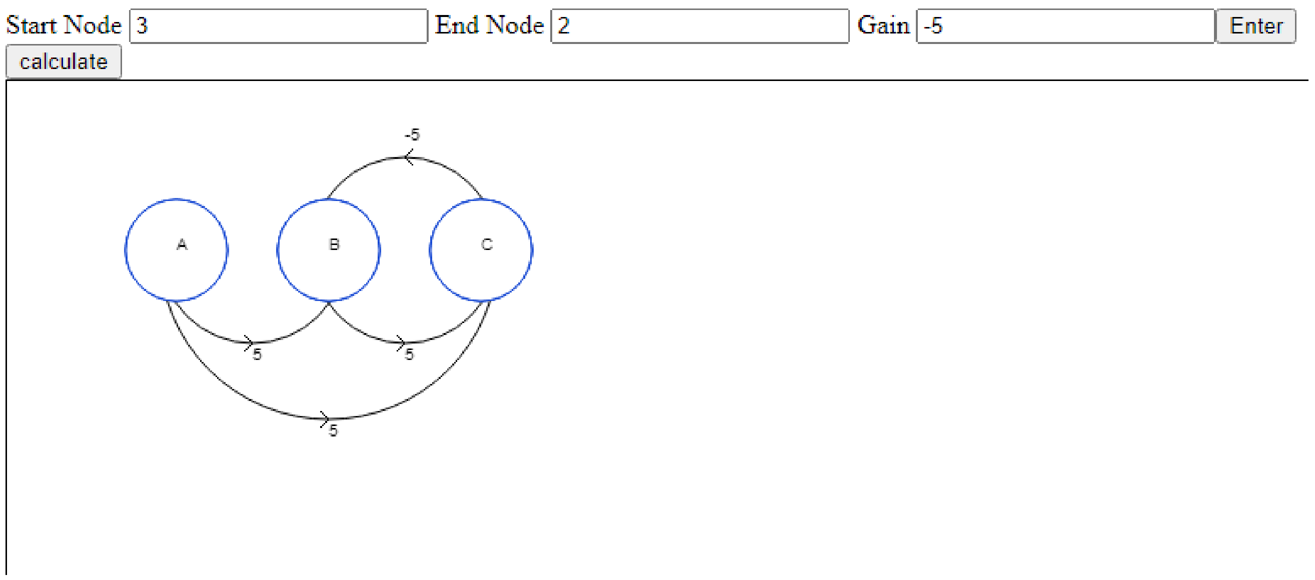


Start Node 6 End Node 2 Gain -1

Forward Paths: ABCDEFG, ABCDFG

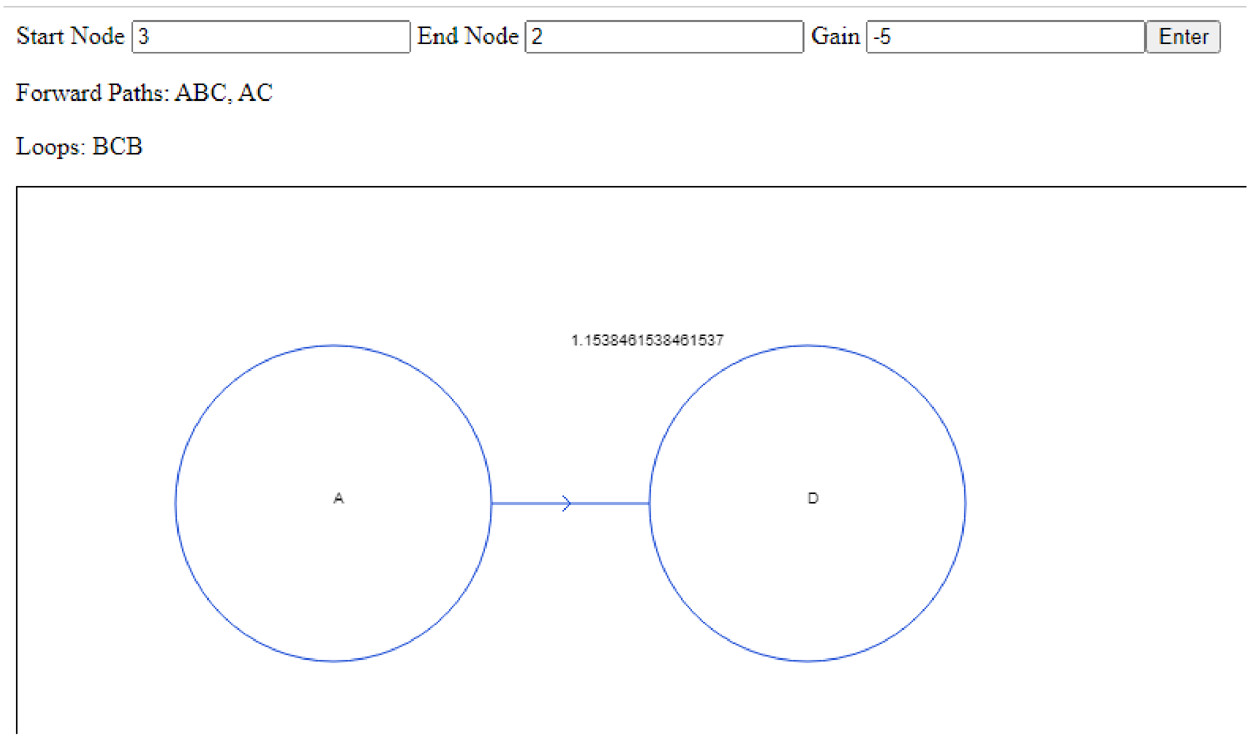
Loops: BCDEFB, BCDFB, CDEC, DEFD, DFD

Second run: input



Start Node 3 End Node 2 Gain -5

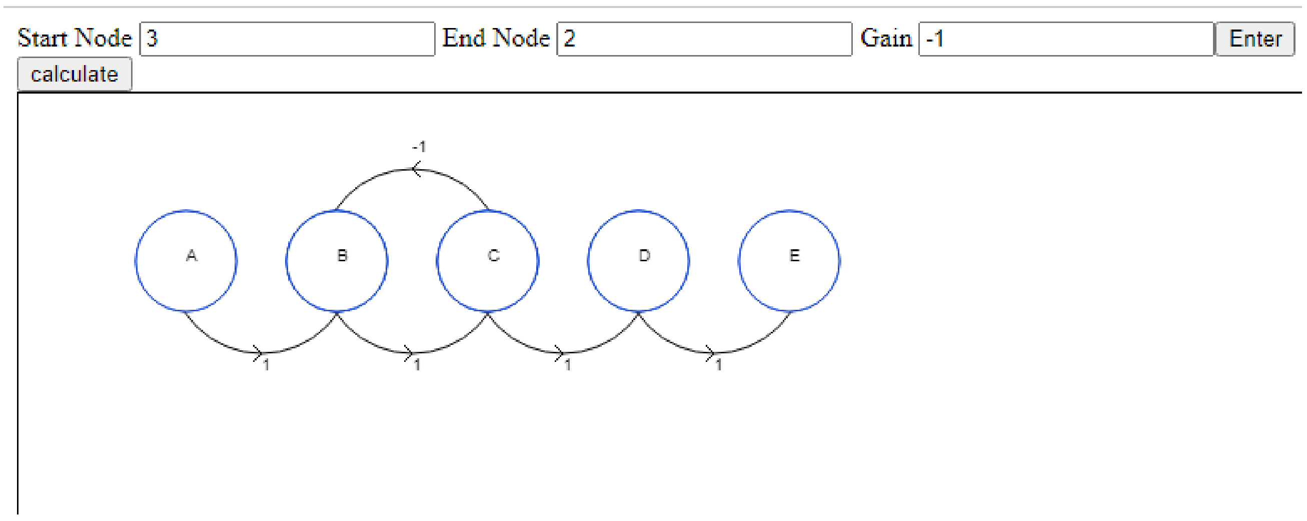
Output


Start Node 3 End Node 2 Gain -5 Enter

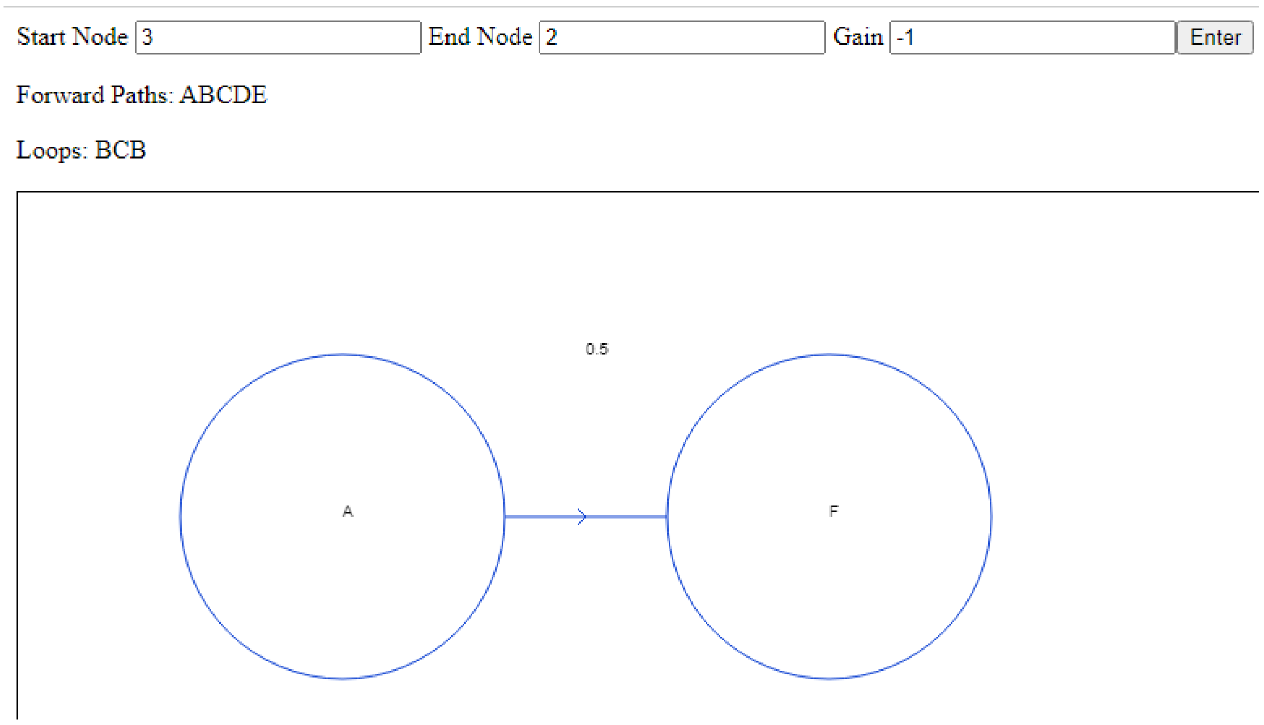
Forward Paths: ABC, AC

Loops: BCB

Third run: input


Start Node 3 End Node 2 Gain -1 Enter
calculate

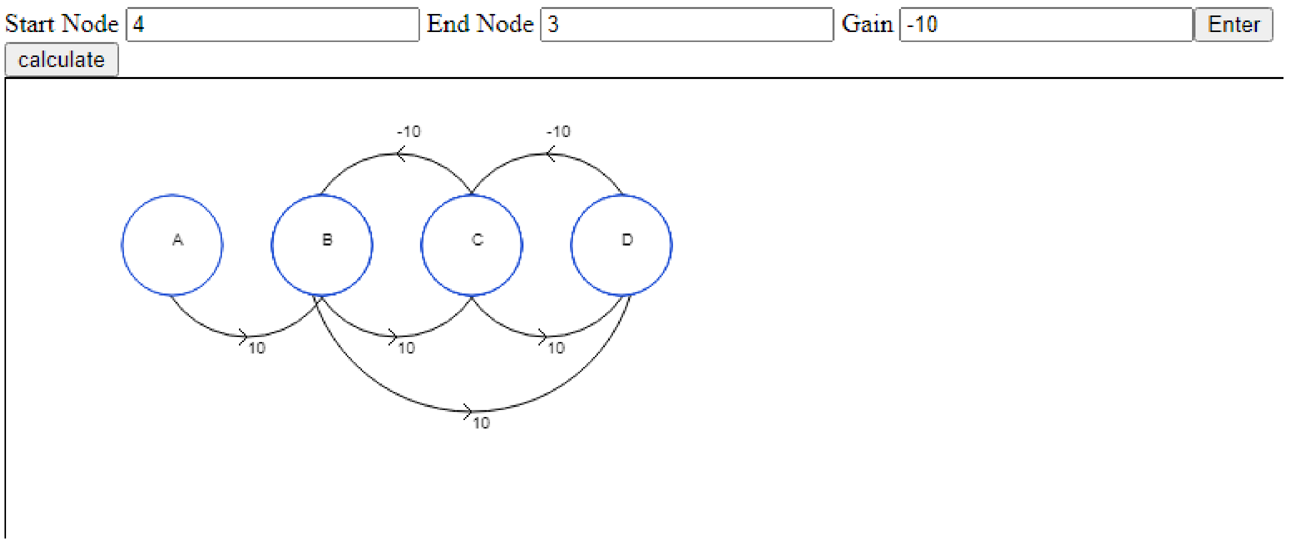
output


Start Node 3 End Node 2 Gain -1 Enter

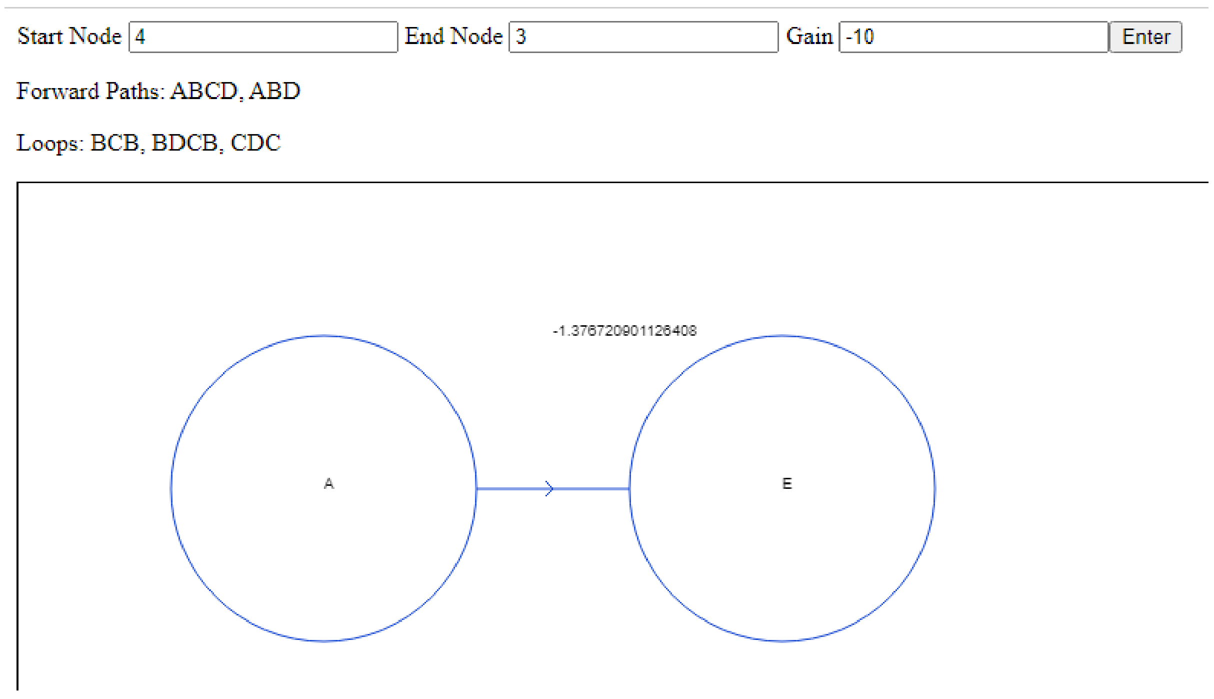
Forward Paths: ABCDE

Loops: BCB

Fourth run: input


Start Node 4 End Node 3 Gain -10 Enter
calculate

Output



Start Node 4 End Node 3 Gain -10 Enter

Forward Paths: ABCD, ABD

Loops: BCB, BDCB, CDC