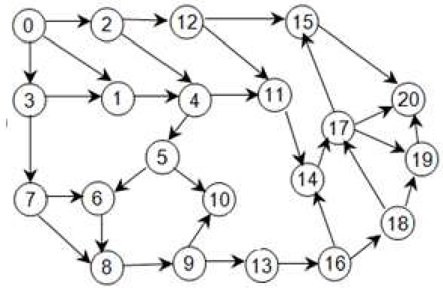
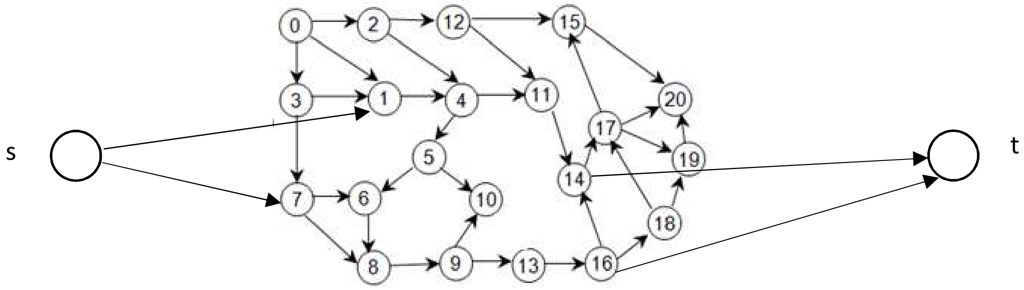
Illustration

Directed Graph



Consider the graph shown above where the set of source nodes is (1, 7) and set of terminal nodes is (14, 18). Before we apply graph reduction, we add two more nodes, artificial source ‘s’ and artificial terminal ‘t’.



REDUCE-DIRECTED

- LsOC 1 to 8: Scans the graph for the vertices which don’t have any incoming nodes or outgoing nodes, leaving s and t. Nodes not having incoming nodes are stored into To_delete_in and nodes having no outgoing nodes into To_delete_out. For the above example node 0 will be pushed into To_delete_in and node 10,20 will be pushed into To_delete_out.
- LsOC 9 to 18: This loop treats the nodes of the stack To_delete_in:
    - > A node is taken out of the stack; it is noted by x
    - > Any y in OUT_N(x) whose card(IN_N(y)) =1 is pushed into the sack To_delete_in
    - > Delete node x from the graph
- LsOC 19 to 28: This loop treats the nodes of the stack To_delete_out:
    - > A node is taken out of the stack; it is noted by x
    - > Any y in IN_N(x) whose card(OUT_N(y)) =1 is pushed into the sack To_delete_out
    - > Delete node x from the graph

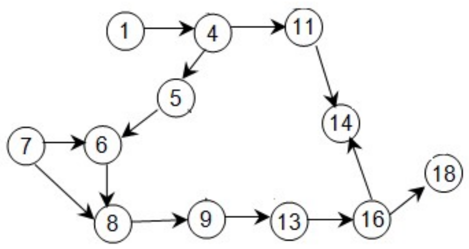
REDUCE-DIRECTED: Lines of code between 9 and 18

To_delete_in	The current node to be deleted	Nodes to be pushed in To_delete_in	Deleted nodes	Deleted edges
{0}	0	2 ,3	0	(0, 1), (0, 2), (0, 3)
{3,2}	3	-	0,3	(0, 1), (0, 2), (0, 3), (3, 1), (3, 7)
{2}	2	12	0,2,3	(0, 1), (0, 2), (0, 3), (3, 1), (3, 7), (2, 4), (2, 12)
{12}	12	-	0,2,3,12	(0, 1), (0, 2), (0, 3), (3, 1), (3, 7), (2, 4), (2, 12), (12, 11), (12, 15)

REDUCE-DIRECTED: Lines of code between 19 and 28

To_delete_out	The current node to be deleted	Nodes to be pushed in To_delete_out	Deleted nodes	Deleted edges
{10,20}	10	-	10	(5,10), (9, 10)
{20}	20	15, 19	20	(19, 20), (15, 20)
{19, 15}	19	-	19,20	(17, 19), (18, 19)
{15}	15	17	15,19,20	(17, 15)
{17}	17	18	15,17,19,20	(14, 17), (18, 17)
{18}	18	-	15,17,18,19,20	(16, 18)

After reduction:



FIND-PATH-DIRECTED

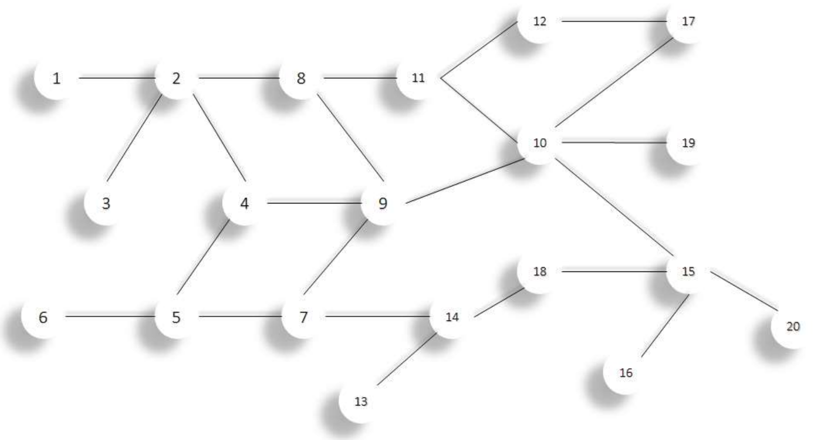
FIND-PATH-DIRECTED takes the reduced graph as the input and find all the paths.

FIND-PATH-DIRECTED: Lines of code from 1 to 20

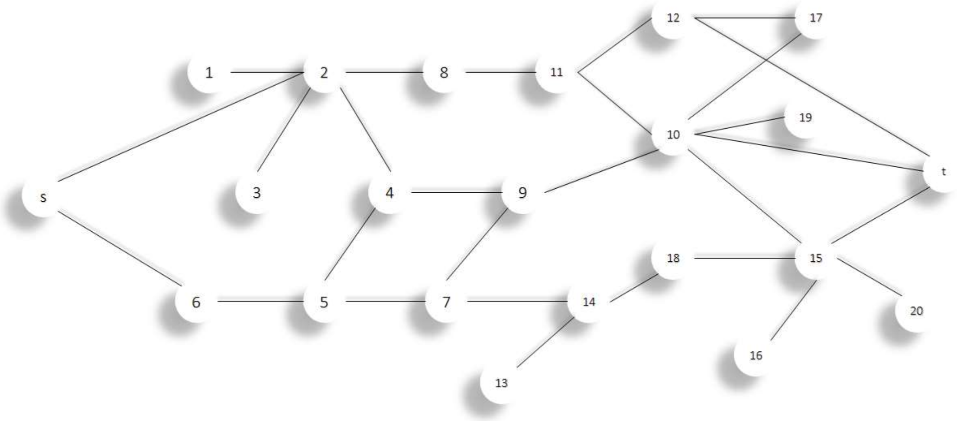
Current Source node	Current node	Next node	L	Path set	Node to be removed

1	1	4	{1,4}	-	-
1	4	5	{1,4,5}	-	-
1	5	6	{1,4,5,6}	-	-
1	6	8	{1,4,5,6,8}	-	-
1	8	9	{1,4,5,6,8,9}	-	-
1	9	13	{1,4,5,6,8,9,13}	-	-
1	13	16	{1,4,5,6,8,9,10,13,16}	-	-
1	16	14	{1,4,5,6,8,9,10,13,16,14}	{1,4,5,6,8,9,13,16,14}	14
1	16	18	{1,4,5,6,8,9,10,13,16,18}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	18
1	16	-	{1,4,5,6,8,9,10,13,16}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	16
1	13	-	{1,4,5,6,8,9,10,13}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	13
1	9	-	{1,4,5,6,8,9}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	9
1	8	-	{1,4,5,6,8}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	8
1	6	-	{1,4,5,6}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	6
1	5	-	{1,4,5}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	5
1	4	11	{1,4,11}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}	-
1	14	-	{1,4,11,14}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	14
1	11	-	{1,4,11}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	11
1	4	-	{1,4}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	4
1	1	-	{1}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	1
7	7	6	{7,6}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	-
7	6	8	{7,6,8}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	-
7	8	9	{7,6,8,9}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	-
7	9	13	{7,6,8,9,13}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	-
7	13	16	{7,6,8,9,13,16}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}	-
7	16	14	{7,6,8,9,13,16,14}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}, {7,6,8,9,13,16,14}	14
7	16	18	{7,6,8,9,13,16,18}	{1,4,5,6,8,9,13,16,14}, {1,4,5,6,8,9,13,16,18}, {1,4,11,14}, {7,6,8,9,13,16,14}, {7,6,8,9,13,16,18}	-

UNDIRECTED-GRAPH



Consider the graph shown above where the set of source nodes is (2, 6) and set of terminal nodes is (10, 12, 15). Before we apply graph reduction, we add two more nodes, artificial source 's' and artificial terminal 't'.



REDUCE-DIRECTED

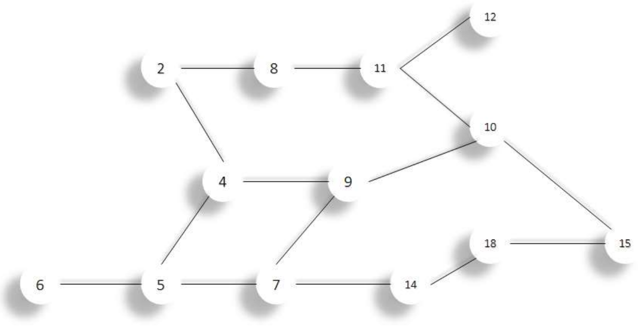
- LsOC 1 to 5: Scans the graph for nodes connected to only one node. All such type of nodes is pushed into To_delete
- LsOC 6 to 15: This loop treats the nodes of the stack To_delete:
  - > A node is taken out of the stack; it is noted by x
  - > Any y in L[x] whose card(L[y]) =1 is pushed into the sack To_delete
  - > Delete node x from the graph

REDUCE-UNDIRECTED: Lines of code between 6 and 15

To_delete	The current node to be deleted	Nodes to be pushed in To_delete	Deleted nodes	Deleted edges

{1,3,13,16,17,19,20}	1	-	1	(1,2)
{3,13,16,17,19,20}	3	-	1,3	(1,2), (3,2)
{13,16,17,19,20}	13	-	1,3,13	(1,2), (3,2), (13, 14)
{16,17,19,20}	16	-	1,3,13,16	(1,2), (3,2), (13, 14), (16, 15)
{17,19,20}	17	-	1,3,13,16,17	(1,2), (3,2), (13, 14), (16,15), (12,17), (10,17)
{19,20}	19	-	1,3,13,16,17,19	(1,2), (3,2), (13, 14), (16,15), (12,17), (10,17), (10,19)
{20}	20	-	1,3,13,16,17,19,20	(1,2), (3,2), (13, 14), (16,15), (12,17), (10,17), (10,19), (15,20)

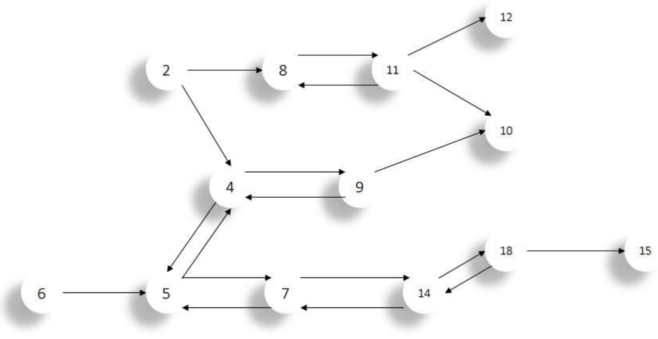
Reduced graph



FIND-PATH-UNDIRECTED

After the graph is reduced it is converted to directed graph and then FIND-PATH-UNDIRECTED is called.

Directed form of the reduced graph



FIND-PATH-UNDIRECTED: Lines of code from 1 to 26

Current Source node	Current node	Next node	l	Visited_nodes	Path set	Node to be removed
2	2	8	{2,8}	{2,8}	-	-

2	8	11	{2,8,11}	{2,8,11}	-	-
2	11	12	{2,8,11,12}	{2,8,11,12}	{2,8,11,12}	12
2	11	10	{2,8,11,10}	{2,8,11,10}	{2,8,11,12}, {2,8,11,10}	10
2	11	-	{2,8,11}	{2,8,11}	{2,8,11,12}, {2,8,11,12}	11
2	8	-	{2,8}	{2,8}	{2,8,11,12}, {2,8,11,12}	8
2	2	4	{2,4}	{2,4}	{2,8,11,12}, {2,8,11,12}	-
2	4	5	{2,4,5}	{2,4,5}	{2,8,11,12}, {2,8,11,12}	-
2	5	7	{2,4,5,7}	{2,4,5,7}	{2,8,11,12}, {2,8,11,12}	-
2	7	14	{2,4,5,7,14}	{2,4,5,7,14}	{2,8,11,12}, {2,8,11,12}	-
2	14	18	{2,4,5,7,14,18}	{2,4,5,7,14,18}	{2,8,11,12}, {2,8,11,12}	-
2	18	15	{2,4,5,7,14,18,15}	{2,4,5,7,14,18,15}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}	15
2	18	-	{2,4,5,7,14,18}	{2,4,5,7,14,18}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}	18
2	14	-	{2,4,5,7,14}	{2,4,5,7,14}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}	14
2	7	-	{2,4,5,7}	{2,4,5,7}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}	7
2	5	-	{2,4,5}	{2,4,5}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}	5
2	4	9	{2,4,9}	{2,4,9}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}	-
2	9	10	{2,4,9,10}	{2,4,9,10}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	10
2	9	-	{2,4,9}	{2,4,9}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	9
2	4	-	{2,4}	{2,4}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	4
2	2	-	{2}	{2}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	2
6	6	5	{6,5}	{6,5}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	-

6	5	4	{6,5,4}	{6,5,4}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	-
6	4	9	{6,5,4,9}	{6,5,4,9}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}	-
6	9	10	{6,5,4,9,10}	{6,5,4,9,10}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}	10
6	9	-	{6,5,4,9}	{6,5,4,9}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}	9
6	4	-	{6,5,4}	{6,5,4}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}	4
6	5	7	{6,5,7}	{6,5,7}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}	-
6	7	14	{6,5,7,14}	{6,5,7,14}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}	-
6	14	18	{6,5,7,14,18}	{6,5,7,14,18}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}	-
6	18	15	{6,5,7,14,18,15}	{6,5,7,14,18,15}	{2,8,11,12}, {2,8,11,12}, {2,4,5,7,14,18,15}, {2,4,9,10}, {6,5,4,9,10}, {6,5,7,14,18,15}	-