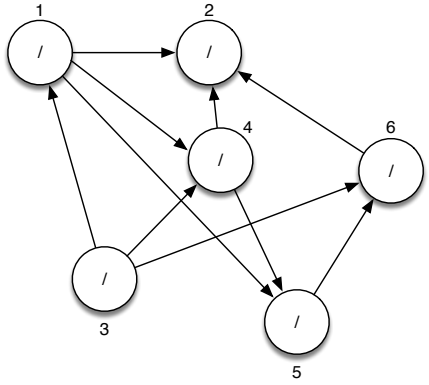
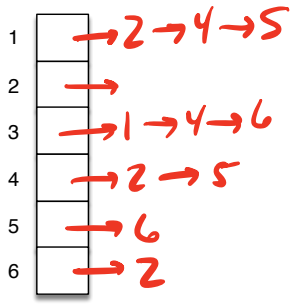
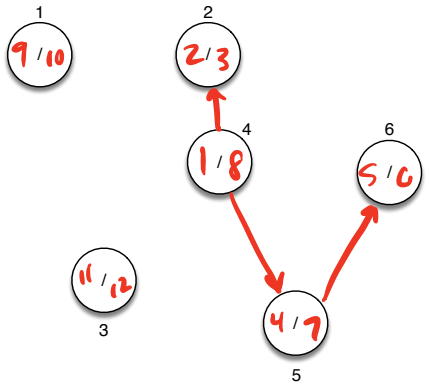
Topological Sort. For the following graph,

(a) Give an *adjacency list* representation for the graph. Order the lists by increasing vertex indices.



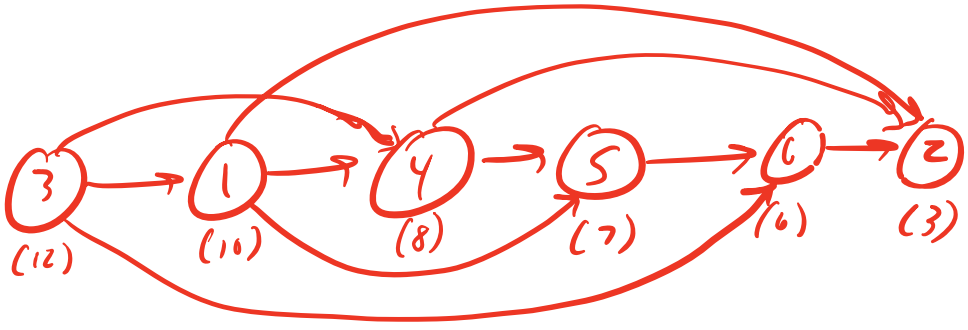


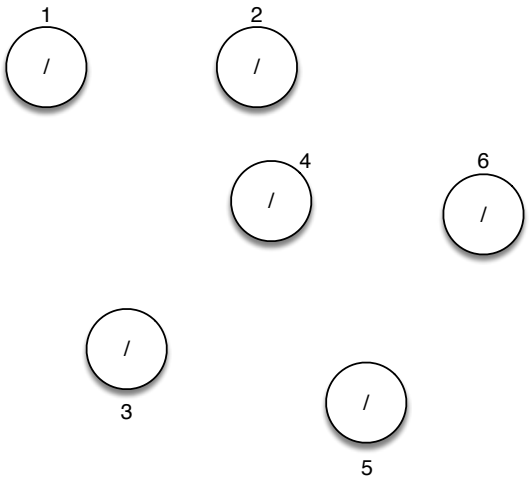
(b) Perform a DFS using vertex 4 as the source. (For any new trees, start at the smallest available vertex index).



	1	2	3	4	5	6
d	9	2	11	1	4	5
f	10	3	12	8	7	6
π	/	4	1	/	4	5

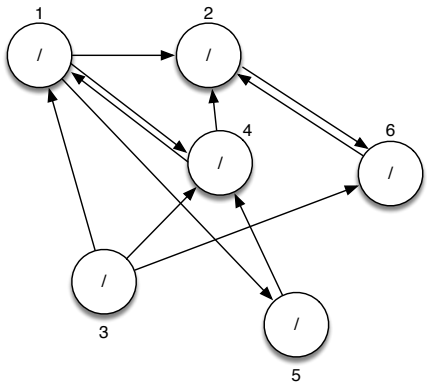
(c) Perform a topological sort on the resulting depth first forest.

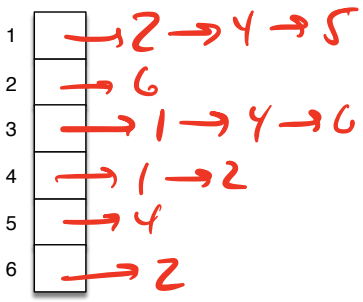



1
/
2
/
4
/
6
/
/
3
/
5

Strongly Connected Component Decomposition. For the following graph,

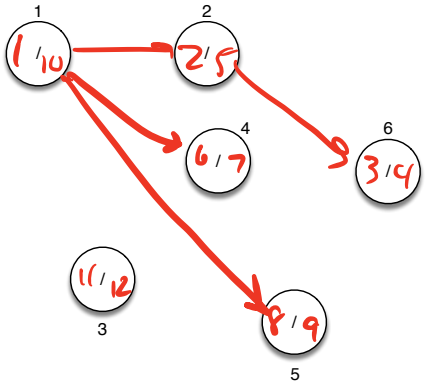
(a) Give an *adjacency list* representation for the graph. Order the lists by increasing vertex indices.





(b) Perform SCCD.

i. Run DFS staring at vertex 1. (For any new trees, start at the smallest available vertex index).

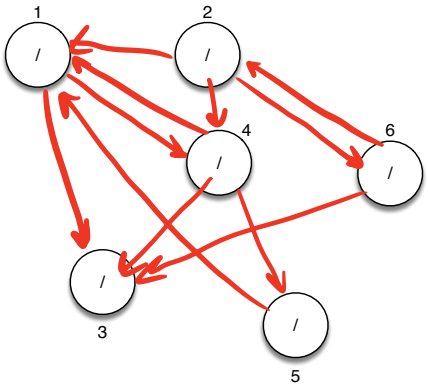


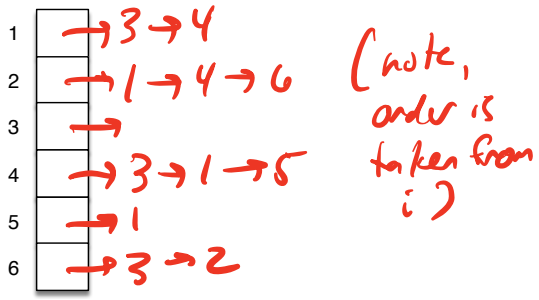
	1	2	3	4	5	6
d	1	2	11	6	8	3
f	10	5	12	7	9	4
π	1	1	1	1	1	2

List the vertices in order of *decreasing* finishing times.

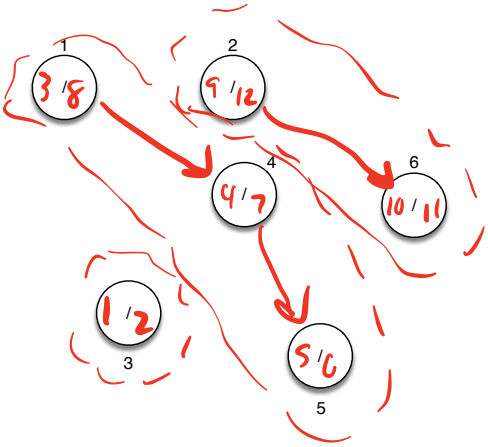
⟨3, 1, 5, 4, 2, 6⟩

ii. Construct G^T





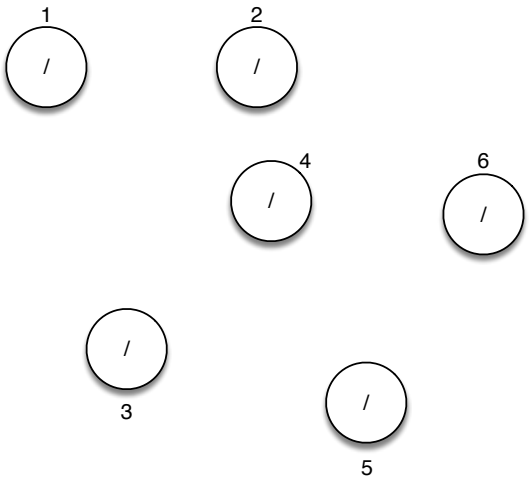
iii. Run DFS on G^T taking vertices in order from step ii

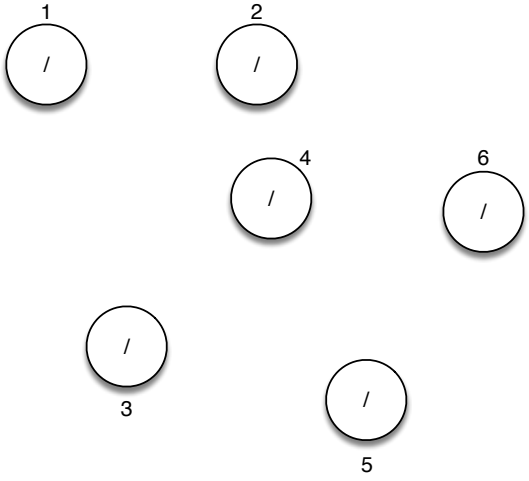


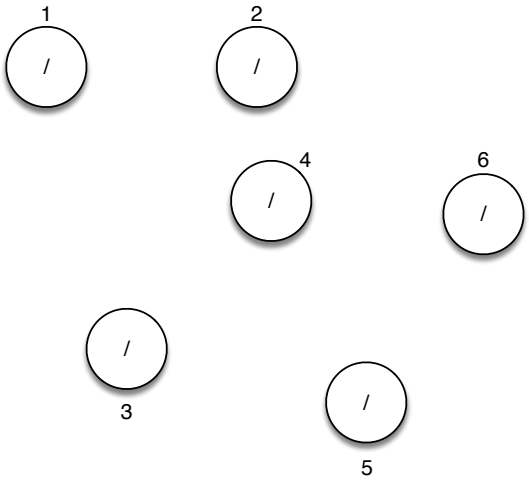
	1	2	3	4	5	6
d	3	9	1	4	5	10
f	8	12	2	7	6	11
π	1	1	1	1	4	2

(c) List the strongly connected components.

<3>, <1, 4, 5>, <2, 6>


1
/
2
/
4
/
6
/
/
3
/
5


1
/
2
/
4
/
6
/
/
3
/
5


1
/
2
/
4
/
6
/
/
3
/
5