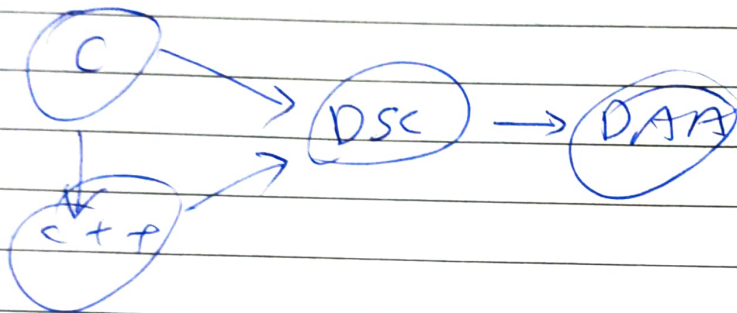


## 2b) Topological Sort

↓  
not a sorting algorithm

→ Topological ordering of a directed acyclic graph (DAG) is the linear ordering of all the nodes in the graph, such that if there is an edge from node  $x$  to node  $y$ ,  $x$  should be placed before placing  $y$ .

Eg:-



Topological order: C, C++, DSC, DAA

2 methods to obtain topological ordering

1) DFS

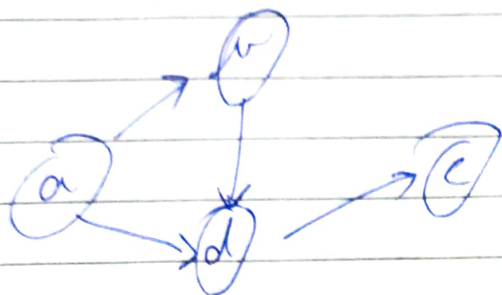
2) vertex deletion method (source removal method)

DFS method

→ First traverse the graph using DFS method

→ Note down the order in which the nodes become the dead end (i.e., note down the popping order of the nodes)

→ Reverse the popping order, we will get Topological order



a, 4

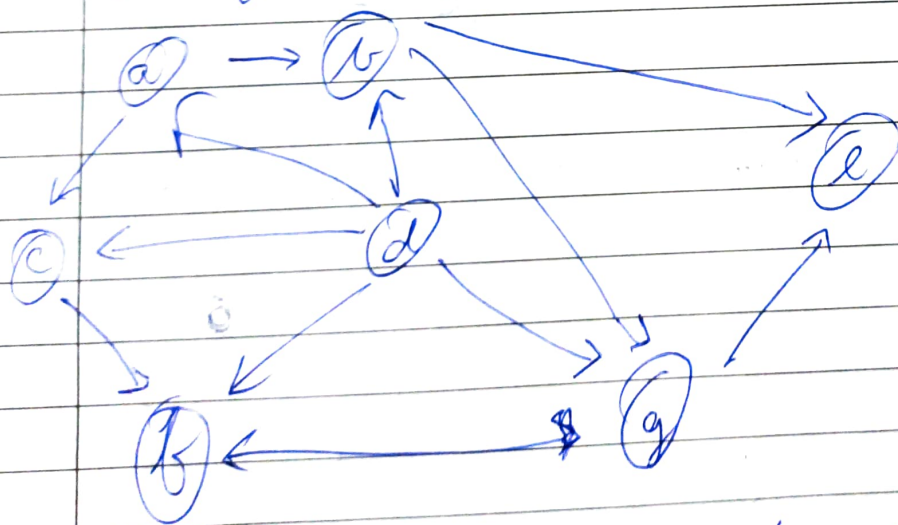
b, 3

d, 2

c, 1

Popping order :- c d b a

Topological order :- a b d c



a, 6

b, 4

c, 1

e, 3

Popping order :- e, f, g, b, c, a, d

f, 2

Topological order :- d, a, c, b, g, f, e

b, 5

d, 7

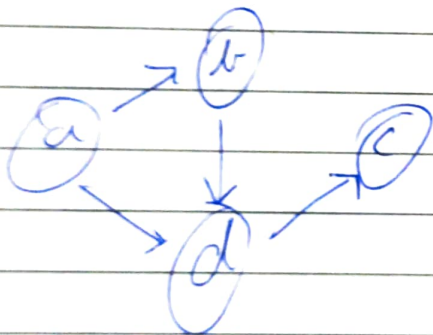


## Vertex deletion method (source removal method)

The procedure for this is

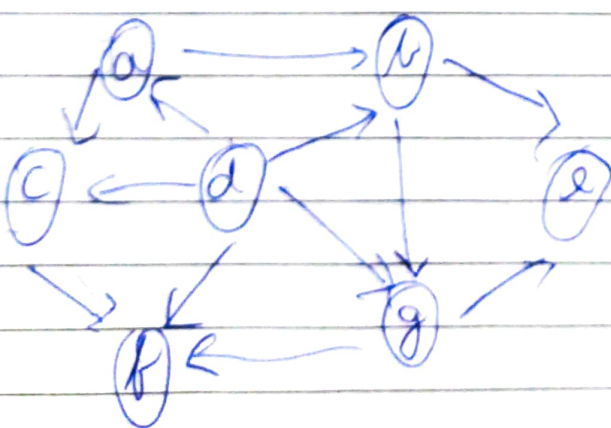
- ① Note down the in degree of all the nodes
- ② Place the node/nodes which has in degree of zero
- ③ Decrement the in degree of those nodes where there was an incoming edge from the node placed in step 2 (deleting the vertex)
- ④ Repeat step 2 & step 3 until all nodes are placed

Eg -



$in(a) = 0$				
$in(b) = 1$	0			
$in(c) = 1$	1	0		
$in(d) = 2$	1	1	0	

Topological order  $\rightarrow a \ b \ d \ c$



$in(a) = 1$	0					
$in(b) = 2$	1	0				
$in(c) = 2$	1	0	0			
$in(d) = 0$						
$in(e) = 2$	2	2	1	1		
$in(f) = 3$	2	2	2	1		
$in(g) = 2$	1	1	0	0		

Topological order  $\rightarrow$   
 $d \ a \ b \ c \ g \ f \ e$

