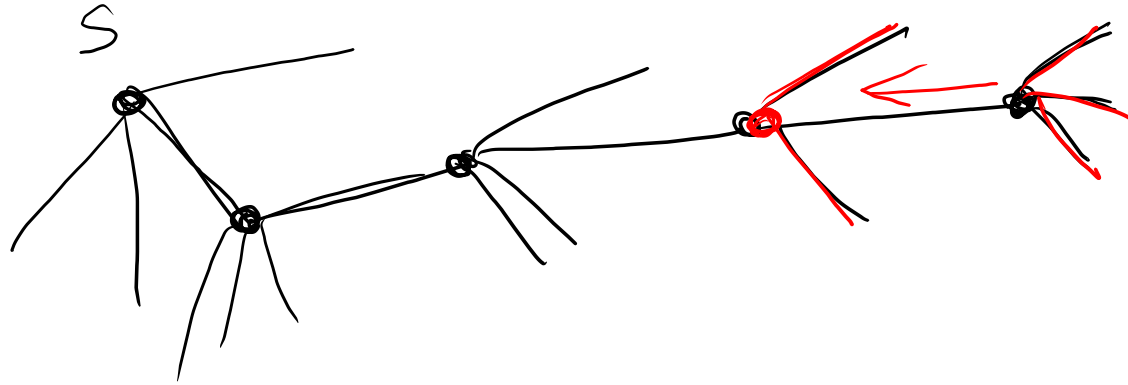






# Depth First Search (DFS)



DFS( $G$ )

for each vertex  $u \in G[V]$

$u.color = white$

$u.pred = NIL$

time = 0

for each vertex  $u \in G[V]$

if  $u.color = white$

DFS-visit( $u$ )

DFS-visit( $u$ )

$u.color = gray$

time = time + 1

$u.starttime = time$

for  $v \in Adj[u]$

if  $v.color = white$

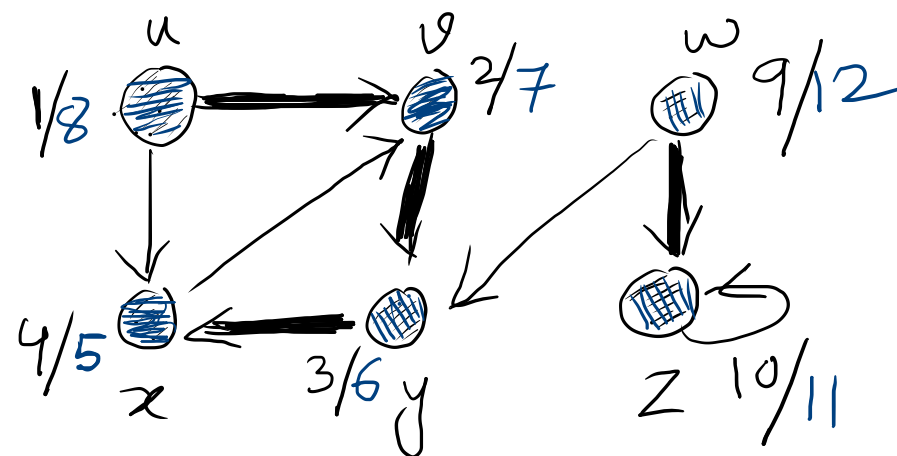
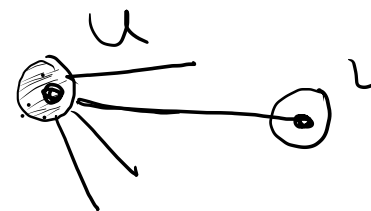
$v.pred = u$

DFS-visit( $v$ )

$u.color = black$

time = time + 1

$u.endtime = time$



Total time:  $O(|V| + |E|)$

# Properties

## Parenthesis theorem

( ( ) ) X

( ) ( ) ( ( ) )

In previous example

u

w

v

z

y

x

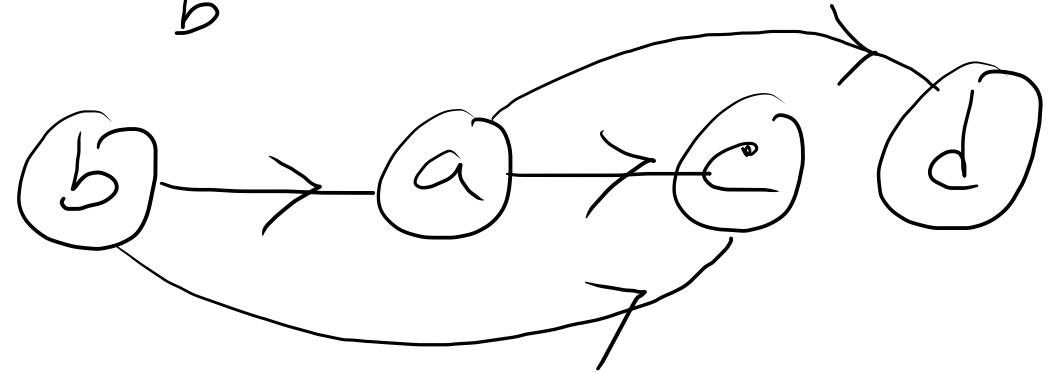
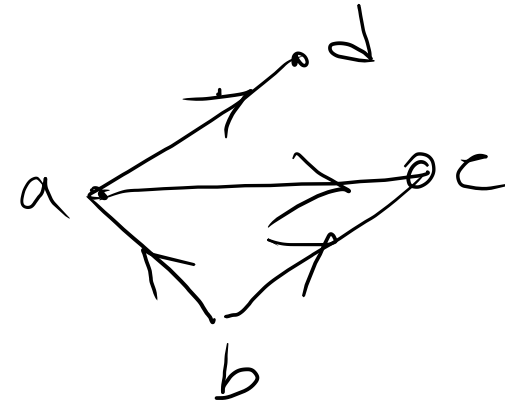
1 2 3 4 5 6 7 8 9 10 11 12  
(u (v (y (x x) y) v) u) (w (z z) w)

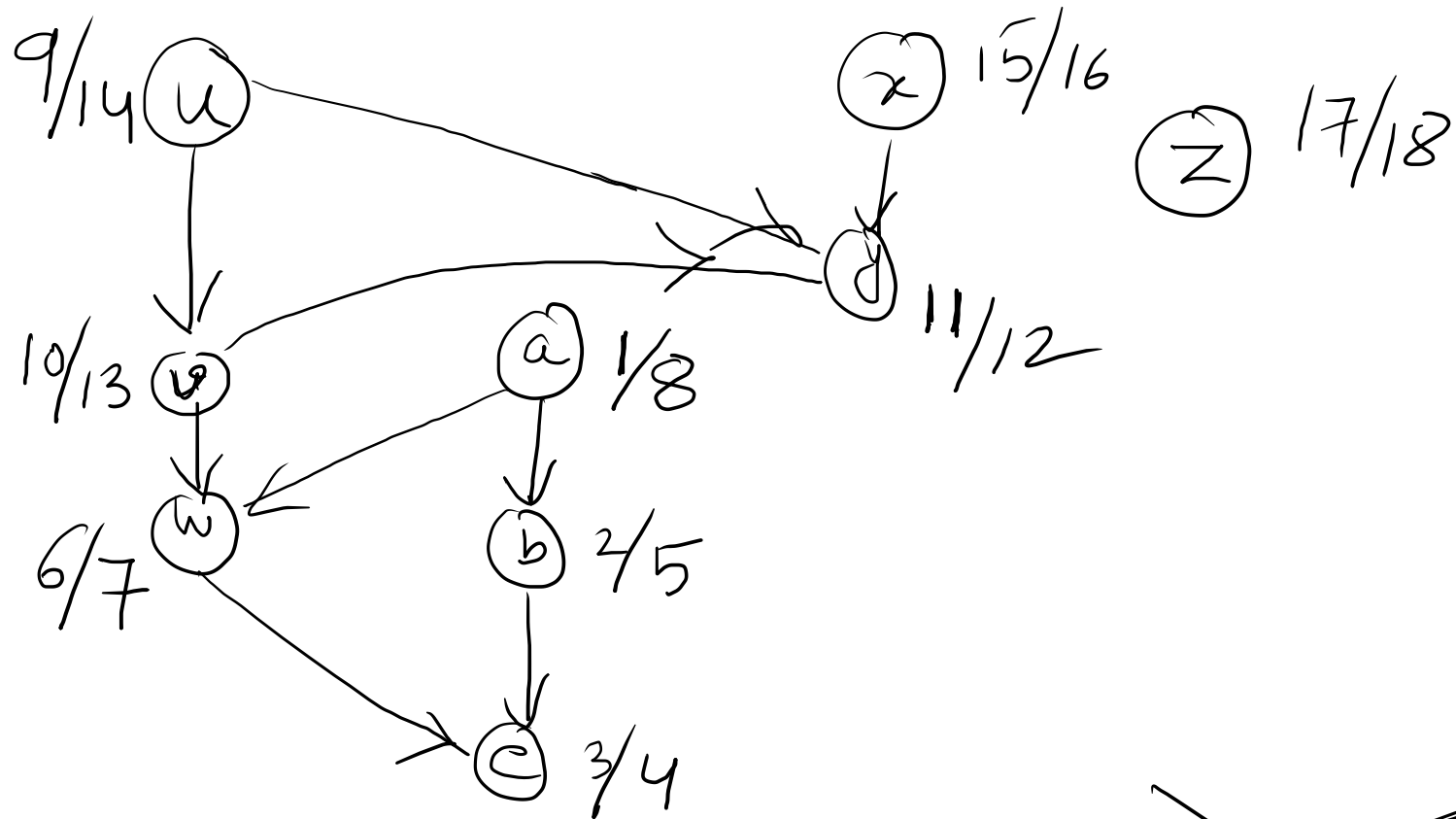
# Topological Sorting for DAG Directed Acyclic Graph.

A topological sorting of a DAG  $G(V, E)$  is a linear order of all its vertices such that if  $(u, v)$  be an edge in  $G$  then  $u$  appears before  $v$  in the order.

Topological-sort( $G$ )

- Run DFS( $G$ )
- As each vertex finish, insert it in a front of a queue
- return the queue.





(z)

(x)

(u)

(v)

(y)

(a)

(w)

(b)

(c)