# Week 11 - DFS app., MST

Algorithm Analysis

School of CSEE

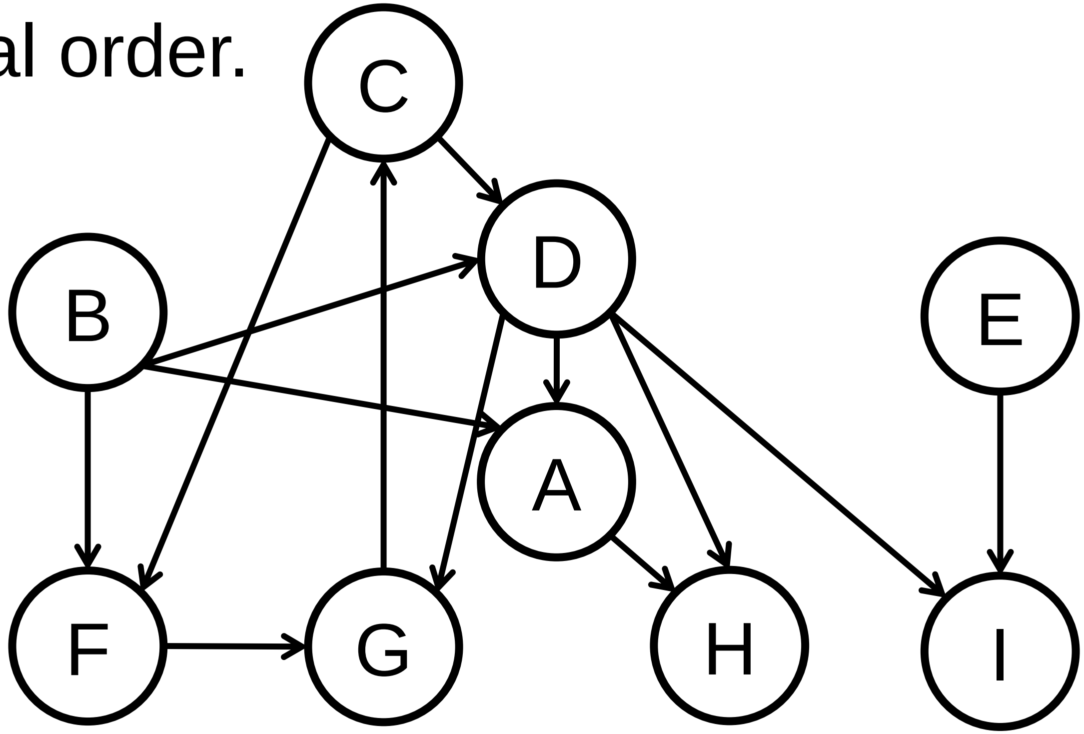
# HW5

- Due by 5/24(Wed.)
- Computes transpose of graph using adjacency array/list to save time.

# DFS application

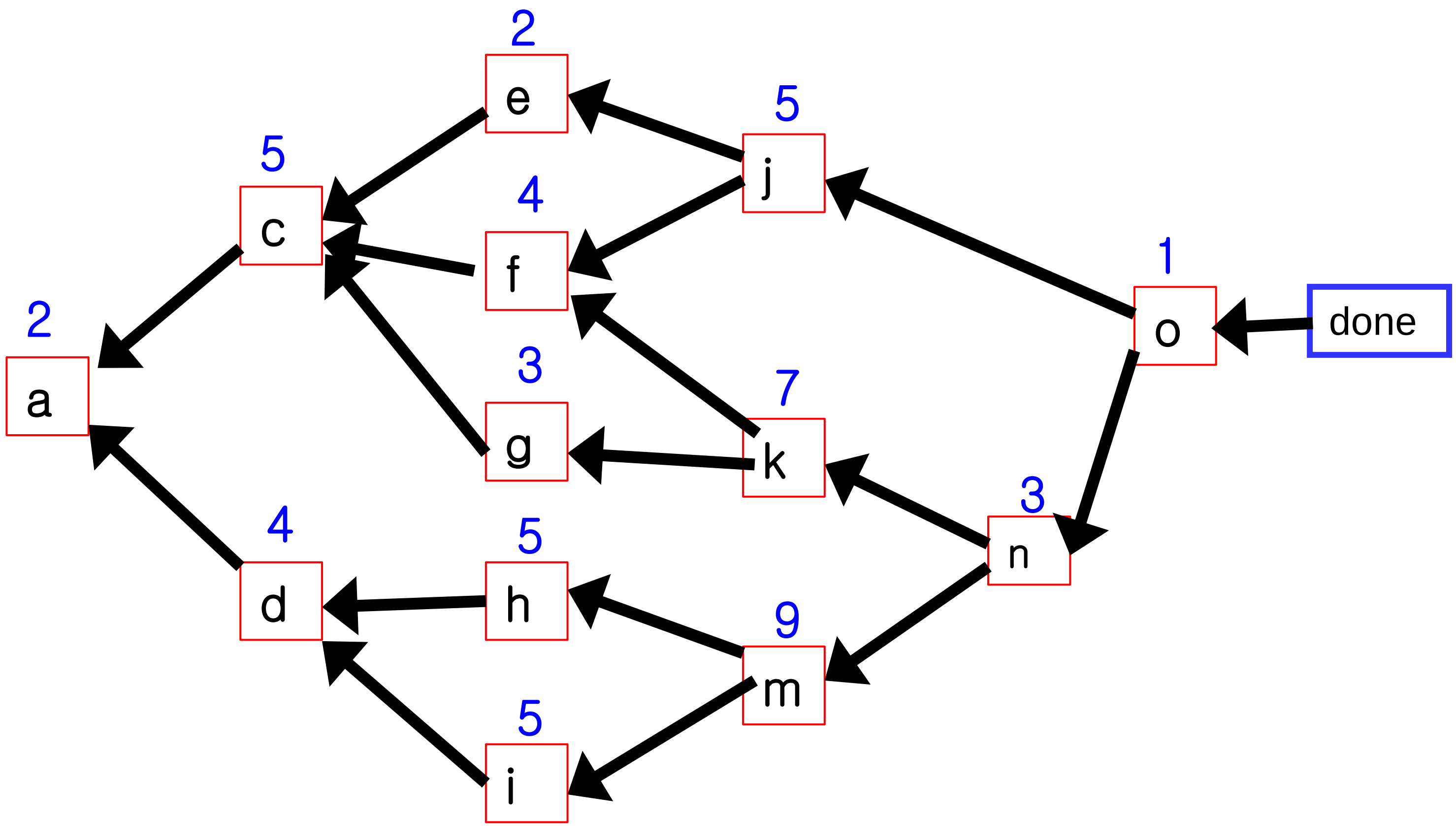
# Exercise 2

From the following directed graph, determine the strongly connected component. Assume that vertices and each adjacency list are in alphabetical order.

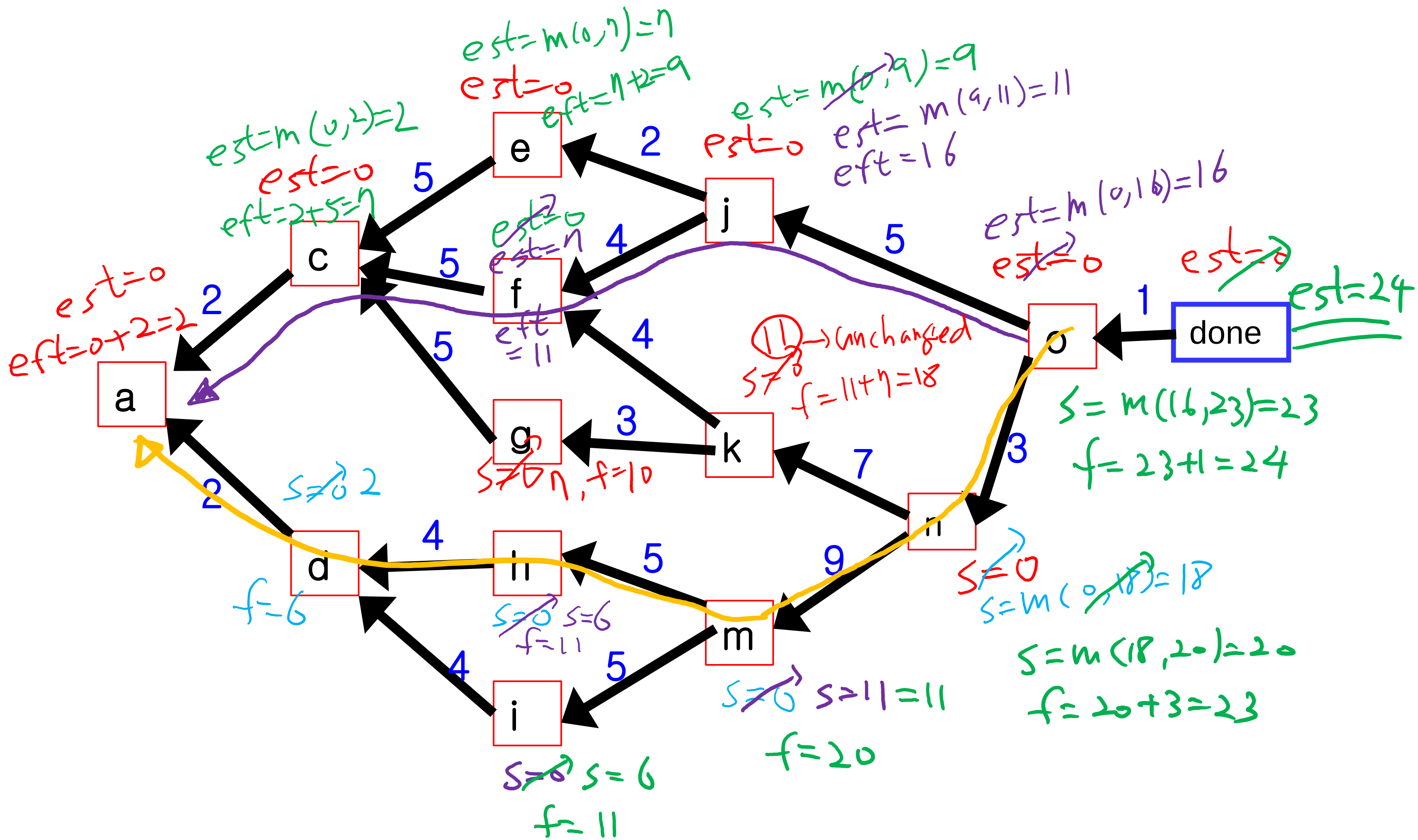


# Exercise 3

Determine ‘est’ and ‘eft’ of each node.



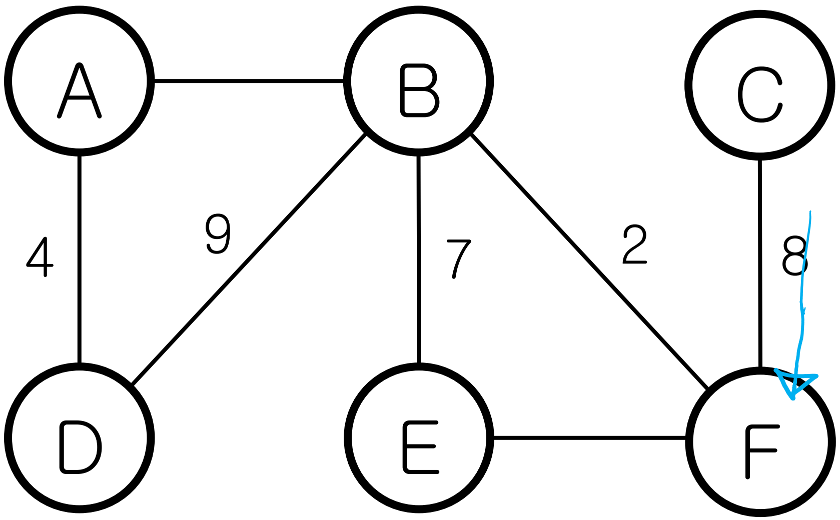
# Exercise 3


est=m(0,7)=7
est=0
eft=7+2=9
est=m(0,2)=2
est=0
eft=2+5=7
est=m(0,9)=9
est=0
est=m(9,11)=11
eft=16
est=m(0,16)=16
est=0
est=0
est=24
est=0
eft=0+2=2
est=7
eft=11
11 → unchanged
s=8
f=11+7=18
s=7 7, f=10
s=0 2
f=6
s=0 s=6
f=11
s=6 s=11=11
f=20
s=0 s=6
f=11
s=0
s=m(0,18)=18
s=m(18,20)=20
f=20+3=23
s=m(16,23)=23
f=23+1=24
a
c
e
f
g
j
k
d
h
i
m
n
o
done
5
2
5
4
5
4
3
5
7
1
2
3
2
4
5
9
4
5

# Exercise 4

Determine the value of $d[E]$ and $back_F$ when the node B is determined as articulation point. Assume adjacency array and adjacency list is in alphabetical order.

(Discovery time of vertex A is 1.)



# Articulation Point

For all $v < y$, check back edge $vw$.

a. *back* is initialized to $d[v]$.

b. When encounteded with back edge $vw$

$back_v = \min(back_v, d(w))$

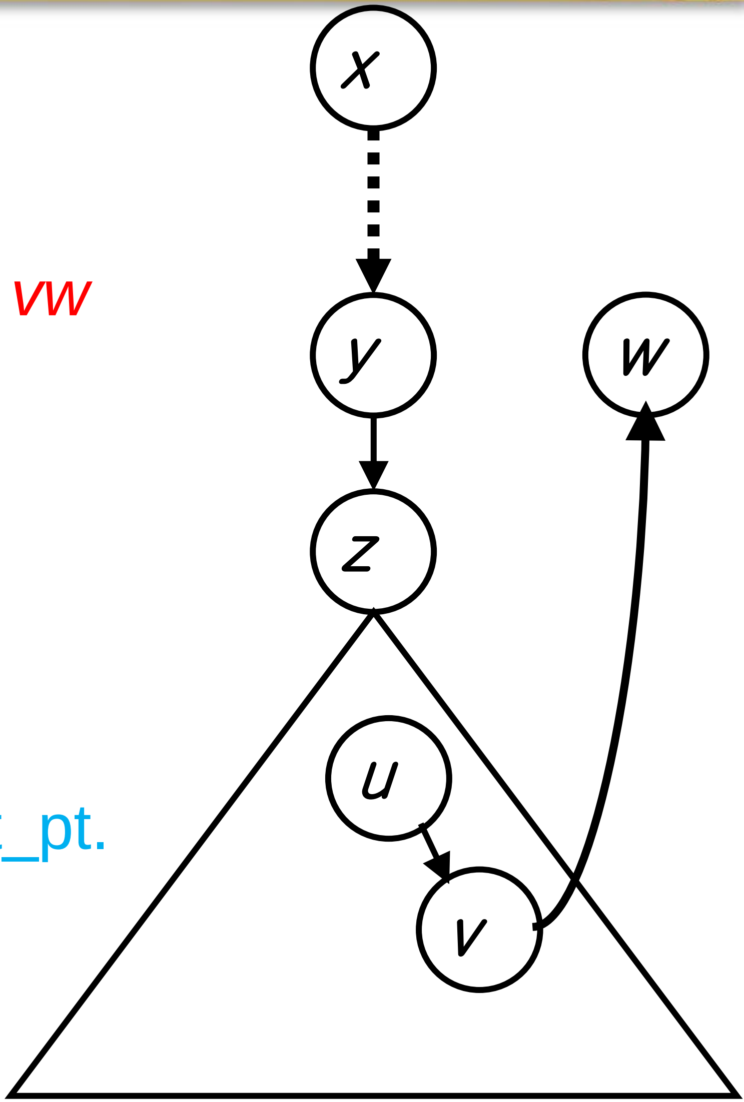
c. When backtracking from $v$ to $u$,

$back_u = \min(back_u, back_v)$

When backing up form $z$ to $y$,

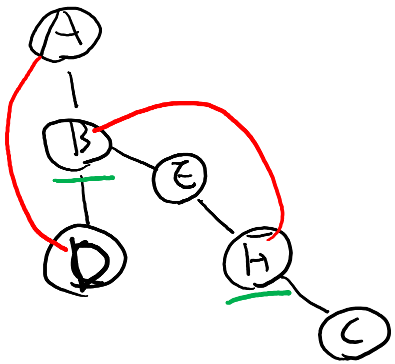
if all vertices $\le z$, $back_z \ge d(y)$. : $y$ is art_pt.

else $back_z < d(y)$. : $y$ isn't art_pt.



When backing up form $z$ to $y$,
if all vertices $\leq z$, $back_z \geq d(y)$. ∴ $y$ is art_pt.
else $back_z < d(y)$. ∴ $y$ isn't art_pt.

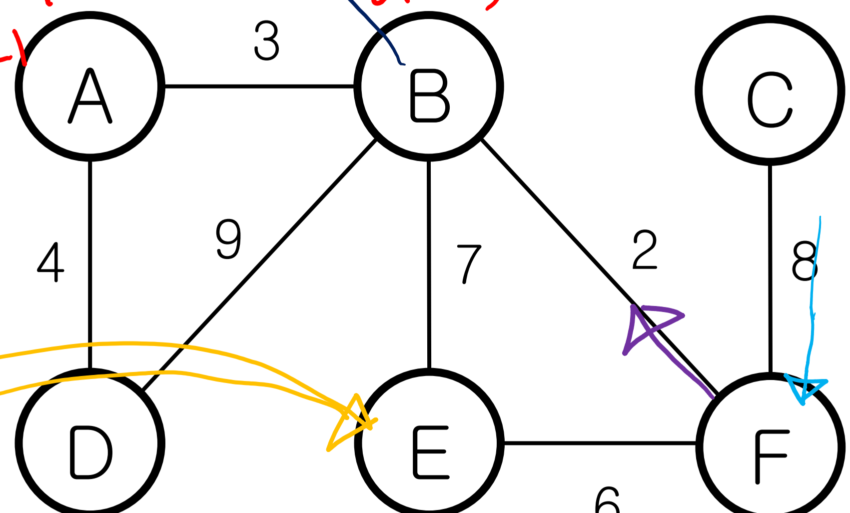
$(back_E = 2) \geq d_B = 2$ ∴ B is A.P.




$(back_D = 1) < (d_B = 2)$ ∴ B isn't A.P.

d=2, b=2 ⇒ b=1

d=1
b=1




d=7
b=7

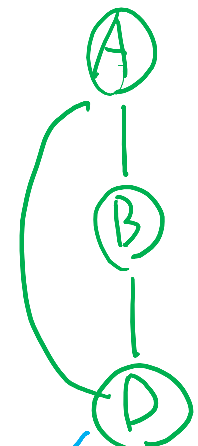
$(back_H = 2)$
$< (d_E = 5)$
∴ E isn't A.P.

d=3, b=3

b = min(3, 1) = 1

d=5
b=~~5~~ 2
b=2

d=6
b=~~6~~ 2




B isn't A.P.

$(back_C = 7) \geq (d_F = 6)$ ∴ F is A.P.

# MST

# Exercise 1

What is data structure used for simple implementation of Prim's algorithm?