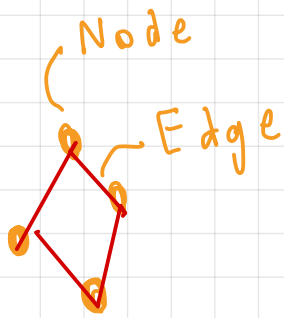




# Graph

→ มีองค์ประกอบ 2 อย่าง คือ

$$G = (V, E)$$

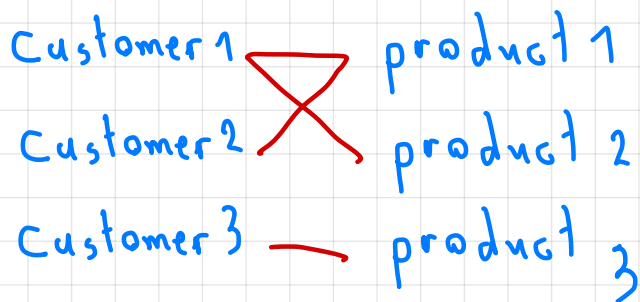


$V$  - set of Node (โหนด)

↳ ของที่พิจารณา เช่น คน, website

$E$  - set of Edge (เส้นเชื่อม)

↳ แสดงความสัมพันธ์ node



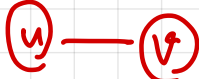
## ลักษณะพิเศษ

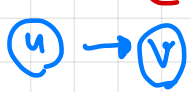
ของ Graph  $G = (V, E)$

1. กราฟมีทิศทาง ... หรือ ... เส้นเชื่อมมีทิศทาง

directed graph

directed edge

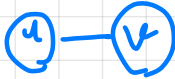
normal  $E = (u, v)$  เมื่อ  $u \in V, v \in V$  

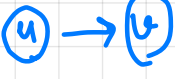

directed  $E = (u, v)$  เมื่อ  $u \in V, v \in V$  

↳  $(u, v) \neq (v, u)$




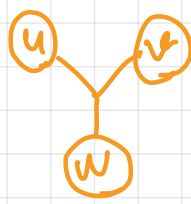
## 2. นิยาม Edge

→ normal edge 

→ directed edge   


→ multi edge 

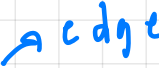
→ loop-back-edge 

→ hyper edge   
 $e = (u, v, w)$

จากข้อนี้ไป ข้อ 8 9

## 3. Path (จาก node หนึ่งไปอีก node หนึ่ง) (จากต้นไปสุดท้าย)

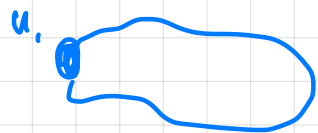


→  $p = e_1 e_2 e_3 e_4 e_5 \dots e_t$  

→  $p = (u_0, u_1)(u_1, u_2)(u_2, u_3) \dots (u_{t-1}, u_t)$

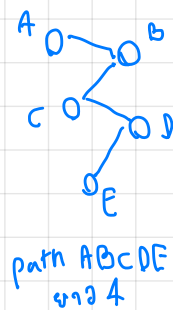
\* →  $p = u_0 u_1 u_2 u_3 u_4 \dots u_{t-1} u_t$

# 4. Cycle

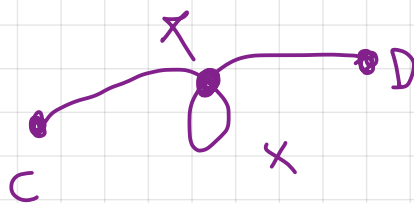
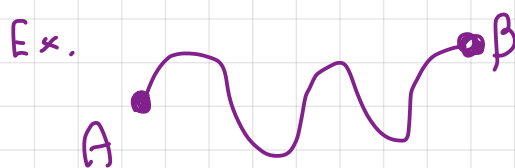


↳ path ที่มี จุดเริ่ม = จุดสิ้นสุด,  $M_0 = M_1$

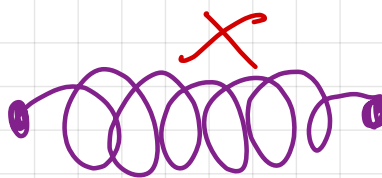
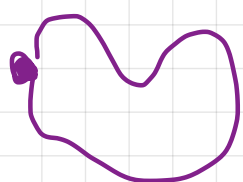
E.g. ABCDEA



Simple Path คือ path ที่ไม่มี cycle เป็นองค์ประกอบ



Simple Cycle คือ cycle ที่ไม่มี cycle อันเป็นองค์ประกอบ



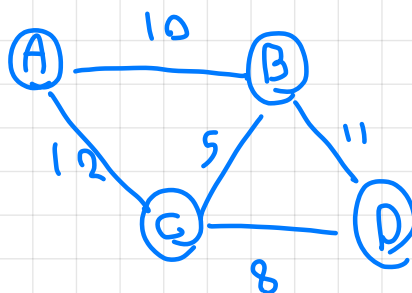
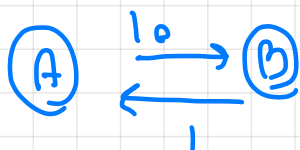
Edge

มีน้ำหนักได้  $\Rightarrow$  weighted Edge

(ราคา/ค่า)

(weighted graph  $G=(V,E,w)$ )

$\forall e \in E, w_e \in W$



# Special Graph

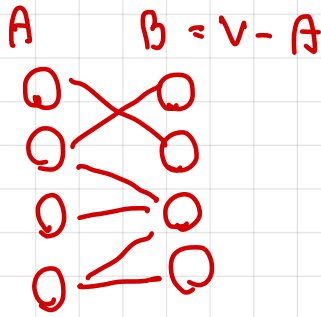
## 1. Bipartite graph

$$G = (V, E)$$

$$A \subseteq V, B = V - A$$

$$\forall e \in E \quad e = (u, v)$$

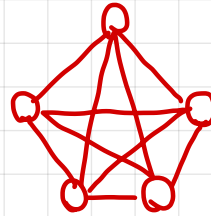
ถ้า  $u \in A$  แล้ว  $v \in B$   
 หรือ  $u \in B$  แล้ว  $v \in A$



## 2. Complete Graph

$$G = (V, E)$$

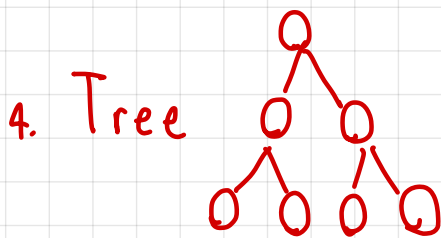
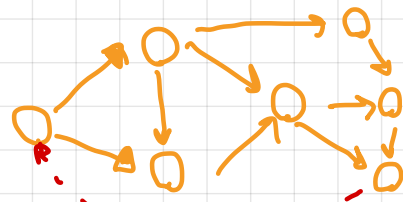
$$\forall u, v \in V \quad (u, v) \in E$$



## 3. Directed Acyclic Graph (DAG)

- directed of graph

ที่ไม่มี directed cycle



## 4. Tree

กราฟที่มี n โหนด

1. เป็น connected graph
2. มีเส้นเชื่อม  $n-1$  เส้น
3. ไม่มี cycle

เมื่อมี 2 คุณสมบัติใด ๆ ก็สามารถสรุปคุณสมบัติที่ 3 ได้

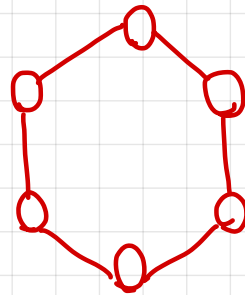
Graph  $G(V, E)$  จะเป็น connected graph

ก็ต่อเมื่อ  $\forall u, v \in V$  จะมี path  $(u, v)$  เสมอ  
 [ทุกตัวของโหนดมี path เชื่อมถึงกัน]



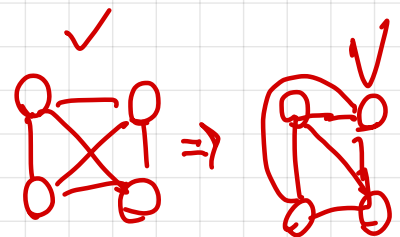
5. Ring

→ Connected Graph  
 → ทุกโหนดมี degree = 2

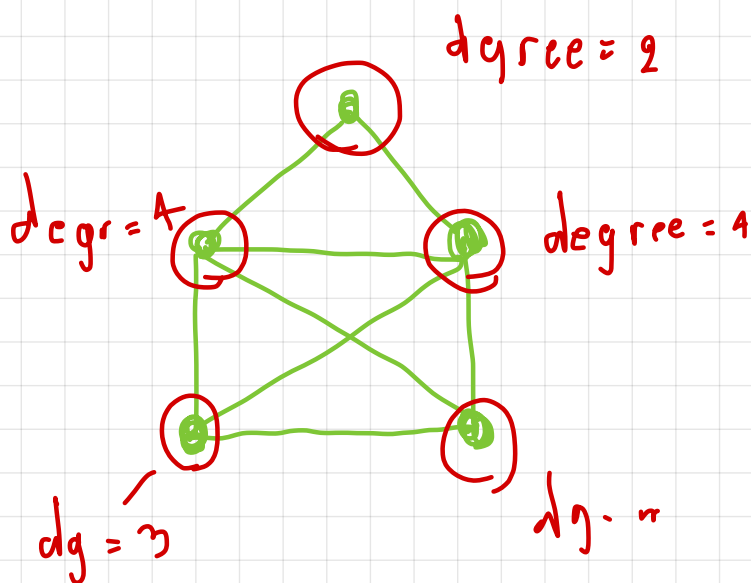


6. Planner Graph

→ กราฟที่เขียนโดยไม่มีเส้นตัดกัน



วงจรที่ ดีกรี ต่ำ น้อยกว่า 2 อัน



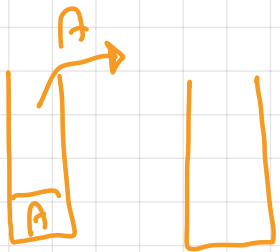
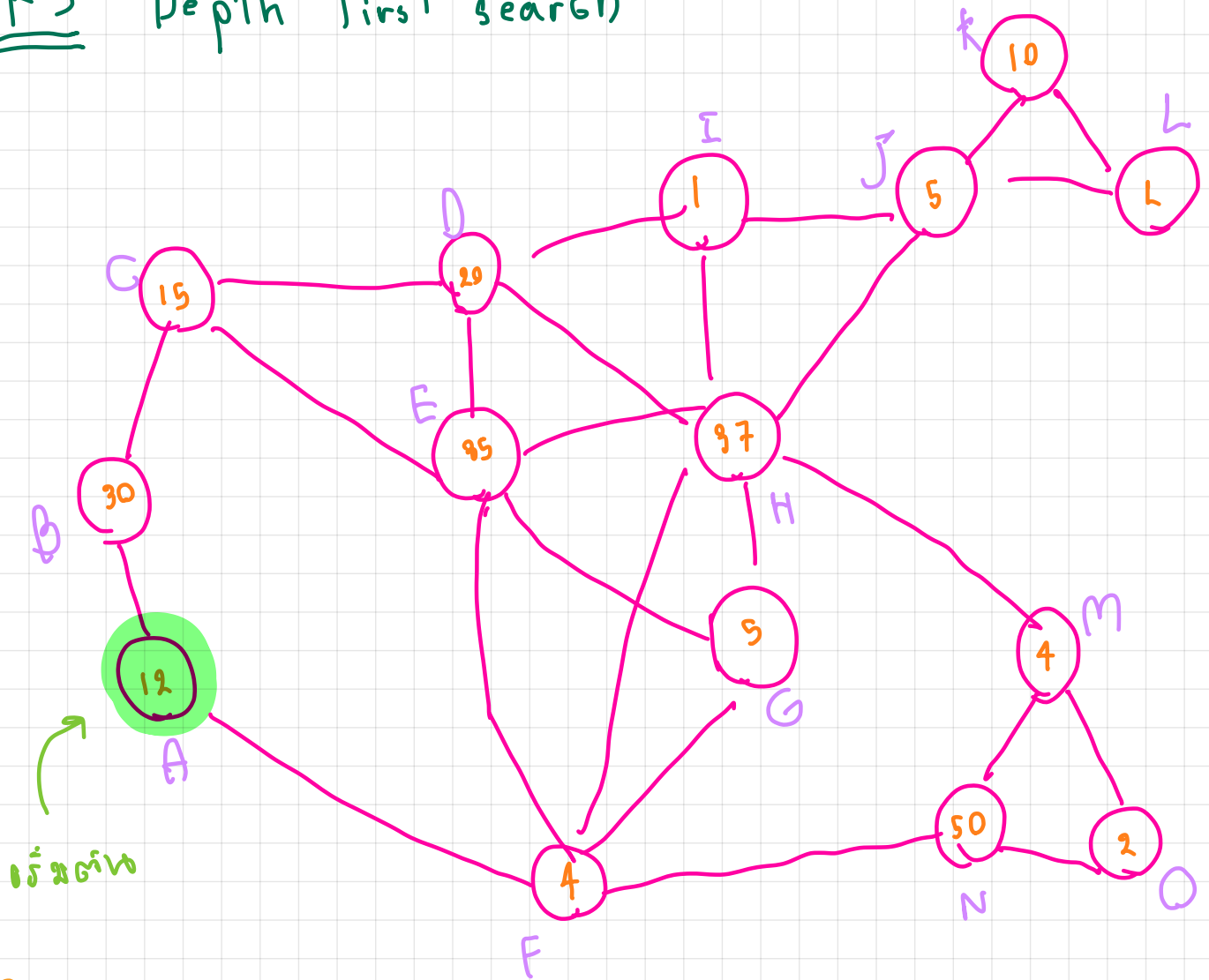
∴ สามารถได้เมื่อ node ที่มี deg  
 เป็นคู่ ไม่เกิน 2 โหนด

$$\sum_{u \in V} \deg u = 2 |E|$$

ดีกรี เป็น 2 เท่าของเส้นเชื่อม

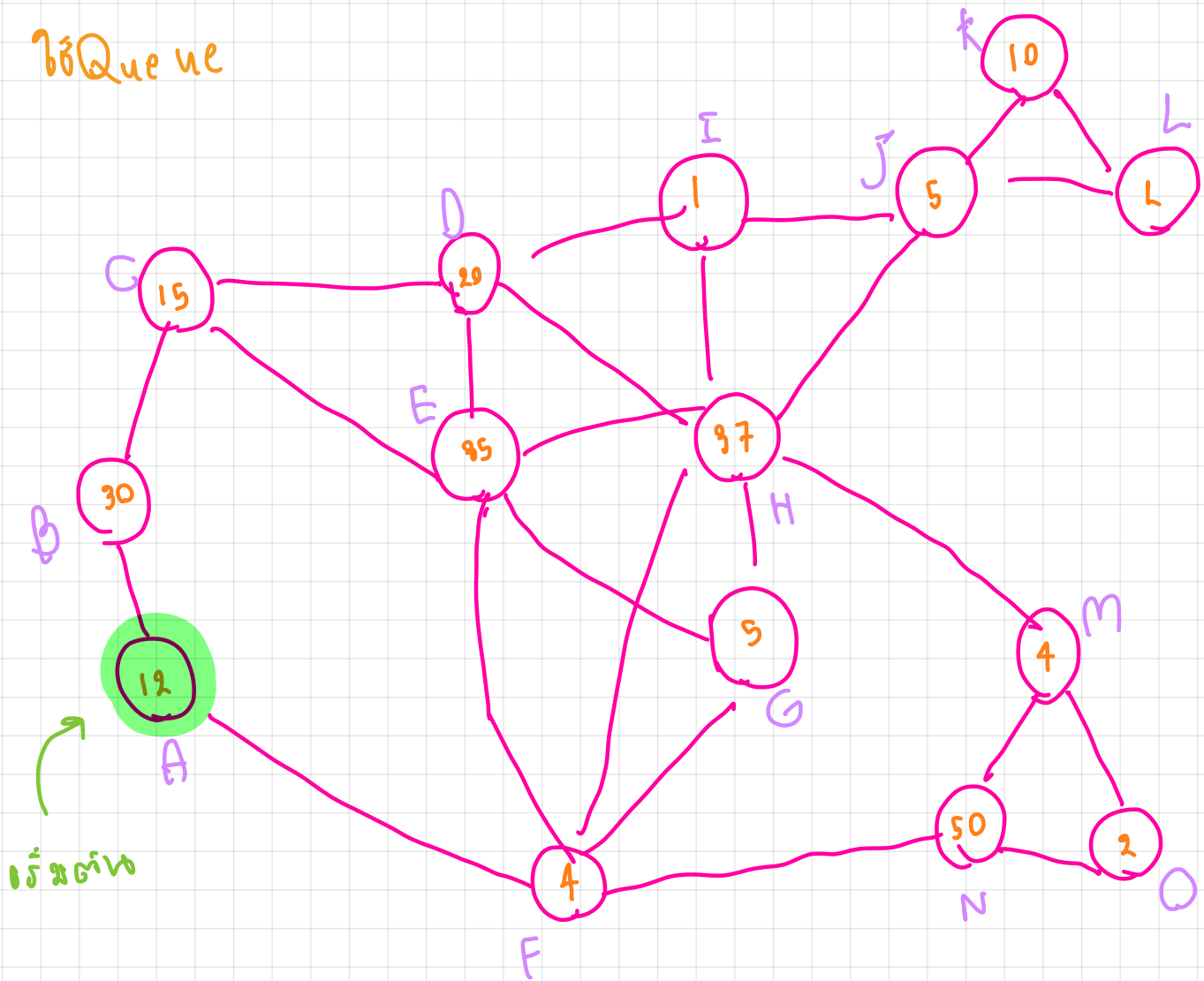
ให้  $G = (V, E)$  หา(จัดเรียง) โหนดที่ต้องการในกราฟนี้  
ไปทุกโหนดในกราฟนี้

DFS Depth first search



# Breadth First Search

ใช้ Queue



→ A →

เพื่อน A → F, B →  
B, F

เพื่อน B → C, F → F  
C

เพื่อน F



# BFS/DFS

DFS (u) โหนดที่กำหนด / ส่วนที่ไว้

- เริ่มต้นที่ u

- stack Q = {}

- Q.push(u)

while (Q ≠ empty)

{ u ← Q.pop()

visit(u) = 1

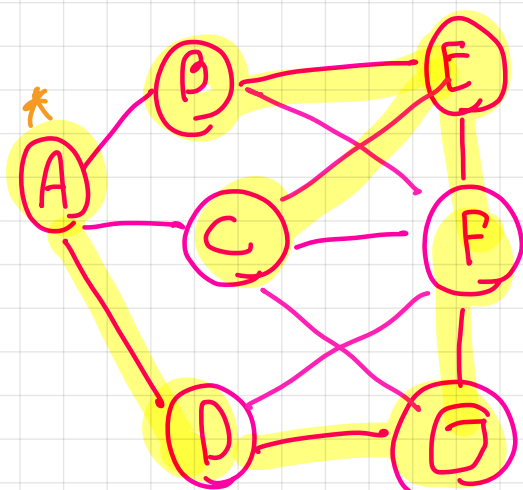
for all edges (u,v) // edge ที่เชื่อมติดกับ u

if visit(v) ≠ 1

Q.push(v)

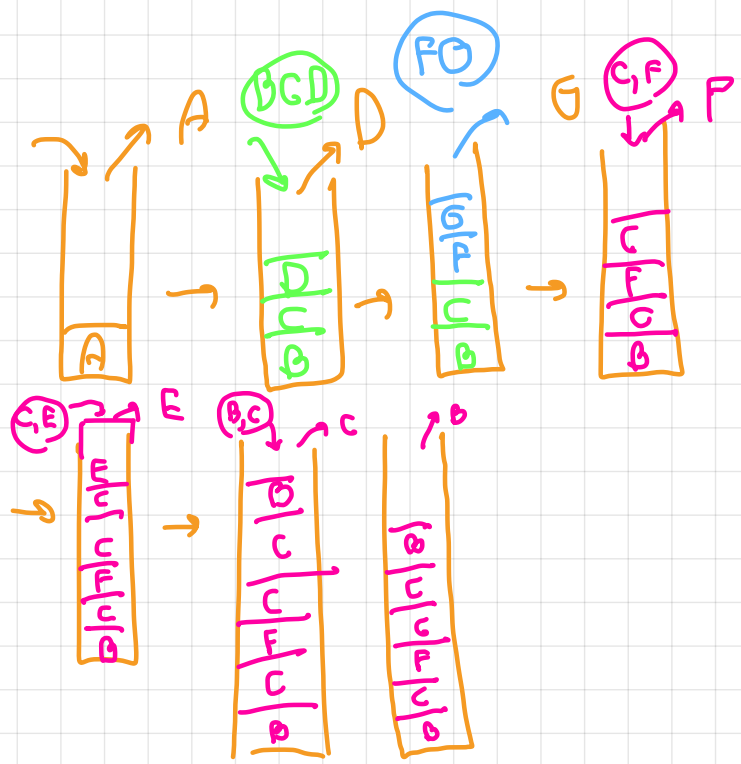
end for

end while



order: A, D, G, F, E, C, B

A → D → G → F → E → C → B



DFS (u) ← โหนดที่กำหนด / สุ่มมาก็ได้

- เริ่มต้นที่ u

- stack  $Q = \{ \}$

- $Q.push(u)$

while ( $Q \neq \text{empty}$ )

{  $u \leftarrow Q.pop()$

$visit(u) = 1$

for all edges  $(u, v)$  // edge ที่ต่อติดกับ u

if  $visit(v) \neq 1$

$Q.push(v)$

end for

end while

A → ~~A~~ → A

DCB → ~~DC~~ → B

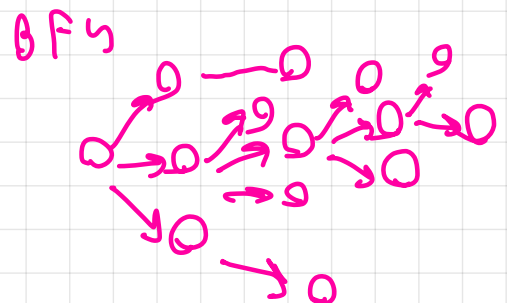
FE → ~~FEP~~ → C

GFE → ~~GFEFE~~ → D

GF → ~~GFGEFE~~ → E

DEGFGFE(F) → F

DFS : 0 → 0 → 0 → 0 → 0 → 0  
 ↳ 0 → 0 → 0



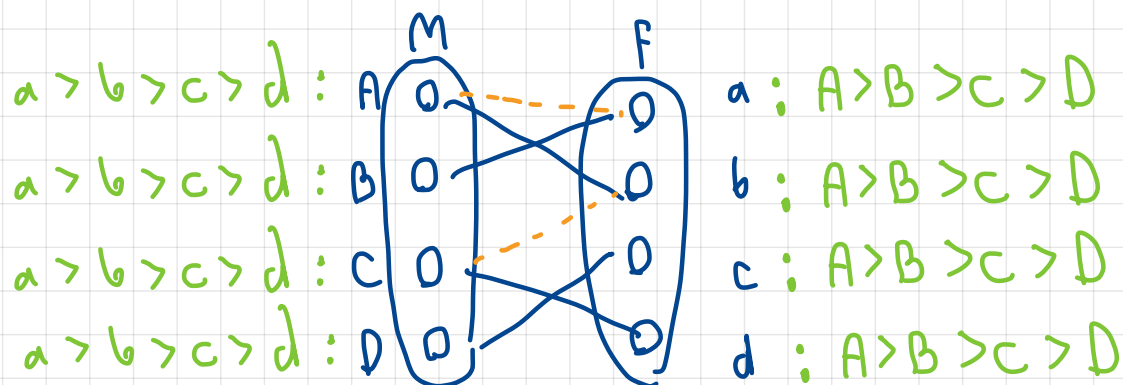
# Stable Marriage Matching

มีผู้ชาย  $n$  คน หญิง  $n$  คน

ผู้ชายแต่ละคน มีรายการความชอบผู้หญิงทุกคน

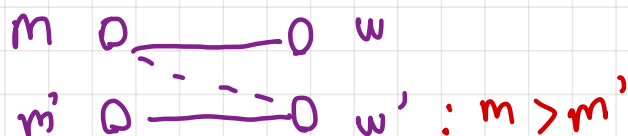
ผู้หญิง ,, ~~.....~~ ,, ผู้ชายทุกคน

เป้าหมาย : ให้นิยามการจับคู่คนทั้งหมดให้ stable

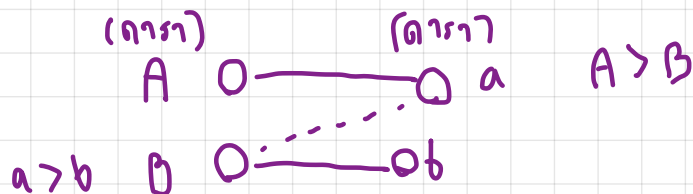


การจับคู่ที่ไม่ stable

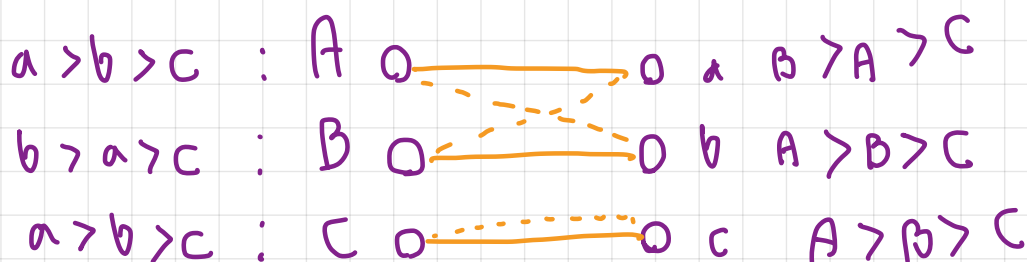
สอบ  $w'$  มา ทกว่า  
 $w' > w$



stable



ขย.



# Gale - Shapley's Algorithm

while (มีผู้ชายบางคนที่ยังไม่มีคู่)

• ในขั้นที่  $m$

•  $m$  propose  $w$  (คนแรกสุดที่ยังไม่เคยปฏิเสธ)

if  $w$  ว่าง

รับ  $m$

$(m, w) \in A_m$

else // ตอหน้า  $w$  คู่กับ  $x$  อยู่,  $(x, w) \in A_w$

$w$  เลือกว่าชอบ  $m$  มากกว่า  $x$  กับ  $x$   
แล้ว  $w$  เลือกคนที่ชอบมากกว่า และทิ้งอีกคน

1. Gale - Shapley's Algo ให้คำตอบเป็น stable matching

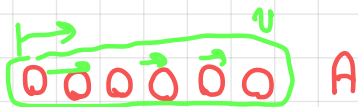
2. ผู้ชายได้ คู่ที่ดีที่สุด

3. ผู้หญิงได้ คู่ที่ชอบที่สุด

4. จำนวนการขอ (prop) ที่เกิดขึ้น

Q. GS Algo ทำงานจบ (ไม่จำเป็นต้องวิ่งได้คำตอบ stable)

ตอบ จบการทำงาน



B

0000000 C



b 000000

c 000000

สังเกต  $\rightarrow$  เมื่อผู้หญิง มีแฟนจะไม่ได้ตอบอีกเลย

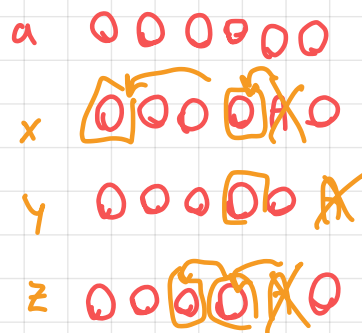
$\rightarrow$  เมื่อ ณ ทุกๆขั้นตอนของ GS Algo จบ

(เมื่อ ณ ตอน  $x$  จะตอบด้วย)

$\rightarrow$  Algo จะจบการทำงานแน่นอน เมื่อมี  $x$  บางคน ขอไปจนถึงคนที่ชอบที่สุดท้าย

$\therefore$  Algo จบในเวลา  $\leq n^2 = O(n^2)$

# 1. GS Algo ได้ stable matching



๑. ขั้วไล่จากตัวสุดท้ายไปเรื่อยๆ

จาก ตช.  $(A, w) \in M^{\text{matching}}$  คำตอบ

$\therefore x, y, z$  ทั้ง 3 มาแล้ว ไปหาโพรพอสิตที่ต่ำกว่า a

$\therefore$  คู่ของ  $x, y, z$  จะดีกว่า a

$\therefore$  A ไม่สามารถ ทำให้เกิด unstable pair ได้

GS มีจำนวนการขอ (ท.จับ/proposal) ทั้งสิ้นที่คงที่ (โดยประมาณ/เลข)

GS Algo (while) ทำซ้ำ ๑ รอบ (คือ ๓)

๓ จับ พ. ดีสุด (คือ พ)

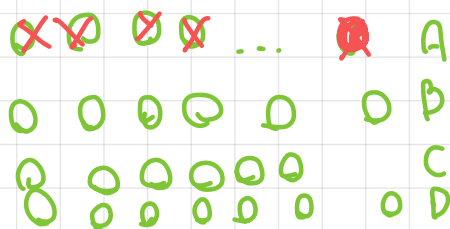
$\rightarrow$  พ. = เลือกว่าคนที่ดีที่ สุด เก็บไว้  
(แล้วทิ้งไว้ถ้ามีคน)

## Observation

• พ. ถูกจับแล้ว จะไม่กลับไปโหวตอีกแล้ว

• เมื่อ พ. ทุกคนโหวตแล้ว  $\Rightarrow$  ๑. ทุกคนมีคู่ตรง

เวลาทำงานไม่เกิน  $n^2$  (๑ มีทุกคน แต่ละคนจากหน้าไปหลัง)



ถึง ตัวสุดท้าย

โปรแกรมละขั้นตอนทำงาน

ทศิข จำนวน ๑

|               | ↓ | ↑ |
|---------------|---|---|
| ● a b c d e f | A | a |
| ● a b c d e f | B | b |
| ● a b c d e f | C | c |
| ● a b c d e f | D | d |
| ● a b c d e f | E | e |
| ● a b c d e f | F | f |

สังเกตว่า a, b, c, d, e, f

$$\therefore \text{จำนวนวิธี} = 6 + 5 + 4 + 3 + 2 + 1 = 21 \text{ วิธี}$$

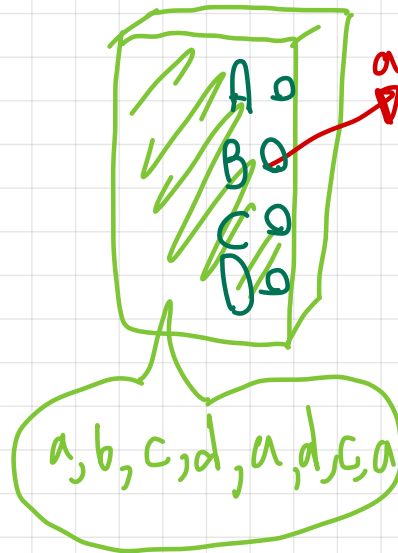
$$= \frac{n(n+1)}{2} = \frac{6(7)}{2} = 21 \text{ ฟ.}$$

# Coupon Collection

• คุปอง  $n$  ประเภท ต้องสุ่ม คุปองกี่ใบ จึงจะได้ครบ  $n$  ประเภท

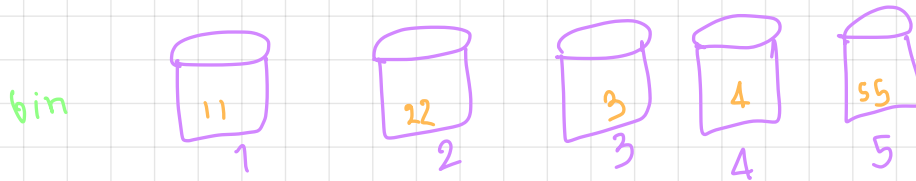
• มองที่ดู

• ระบบ  $\{A, B, C, D\}$  จะสุ่มจนกว่าจะได้  
ชื่อ ทุกประเภท



- a  $A > B > C > D$
- b  $B > C > D > A$
- c
- d

ball 1 3 5 2 1 3 1 2 5 4 2 4 4 3 4 2 1



∴ ไม่สิ้น จำนวนเงินถึง พอแก่ใจถึง มีของครบตาม ถึง

Random Variable

$X$  = จำนวนลูกบอลทั้งหมดที่ใช้เพื่อให้ถึงทุกถึงไม่ว่า

Expected value

$$E[X] = \text{ค่าเฉลี่ย} \sum_{x=1}^n x P(X=x)$$

Sum of Expectation = Expectation of Sum

$$* E[X] + E[Y] = E[X+Y]$$

$$\sum E[X_i] = E[\sum X_i]$$

# Random Variable



$$X_2^{2 \rightarrow 1} = 1$$

$$X_1^{1 \rightarrow 0} = 2 \text{ ครั้งโดยเฉลี่ย}$$

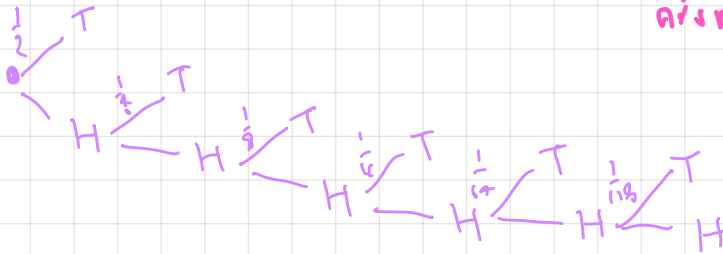
$$E[\text{# วงกบโหนด T}] = \sum x P(x)$$

ดังนี้

$$5 = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + \dots$$

$$2.5 = 1 + 2 \cdot \frac{1}{2} + \dots$$

$$\therefore 5 = 2$$



$X$  = จำนวนลูกบอลทั้งหมดที่ใช้เพื่อให้ถึงทุกถังไม่ว่า

จะว่างดกล

$$X = X_n + X_{n-1} + X_{n-2} + X_{n-3} + X_{n-4} + \dots + X_3 + X_2 + X_1 + X_0$$

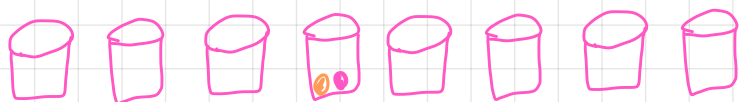
โดย  $X_i$  แทนจำนวนบอลที่ใช้มากกว่า บอลจะตกลงถังไหน

Ex.  $n=8$  (ถัง 8 ใช้บอลที่ลูกโดยเฉลี่ย)



⋮

⋮



ถัง 8  $\rightarrow$  8  $\frac{100\% = 1}{\text{โอกาสลงถังไหน}}$   $E[X_8] = 1$  # บอล

7  $\rightarrow$  7  $\frac{7}{8}$   $E[X_7] = \frac{8}{7}$

6  $\rightarrow$  6  $\frac{6}{8}$   $E[X_6] = \frac{8}{6}$

1  $\rightarrow$  1  $\frac{1}{8}$   $E[X_1] = 8$

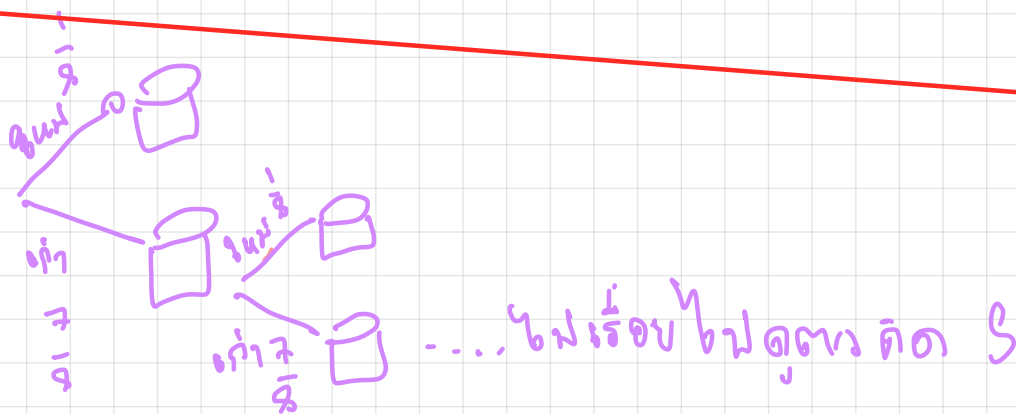


$$E[x] = E[X_8 + X_7 + X_6 + \dots + X_1]$$

$$= E[X_8] + E[X_7] + E[X_6] + \dots + E[X_1]$$

$$= 1 + \frac{8}{7} + \frac{8}{6} + \frac{8}{5} + \dots + \frac{8}{2} + \frac{8}{1} = 8 \ln 8$$

$$= 8 \left( 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} \right) \leftarrow \text{harmonic sum}$$



$$E[x] = n \ln n$$

Linear Expectation

∴ \* ครั้งการจูน (proposal) คือ  $n \log n$

∴ โดยเฉลี่ย ๘ จะได้ ๗ ดีสุด อีกตัว  $\log n$

∴ โดยเฉลี่ย ๗ จะทิ้ง ๘ แก่  $\log n - 1$  คน

ลำดับ  
... คนดีสุด  $\approx \frac{1}{\log n}$

