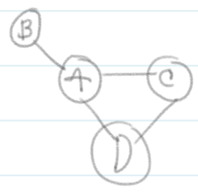
Graph ใช้ยังไง~~~

- Graph in everyday life → เครือข่ายโครงสร้าง, แผนที่ต่างๆ
- ประกอบด้วย vertex และ edges
- อนุญาตให้เกิด loop ได้

Graph Representation (Edge list)
Adjacency Matrix, list ออกสอบ!!

Edges list

Edges : (A,B), (A,C), (A,D), (C,D)



List :

Adjacency Matrix ออกสอบ!!

เป็น Matrix

	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	1	0	0	1
D	1	0	1	0

hash ใช้ได้ integer ก่อน

0: ไม่มี edge
1: มี edge

สร้าง Arr 2 มิติ ของ int ออกสอบ

เริ่มได้ไงอ่ะ → boolean[][] arr;
-true = มีเส้นทาง, false = ไม่มีเส้นทาง

directed graph - ไปทางเดียว

Adjacency List
→ เป็น arr ของ list เก่งตัว

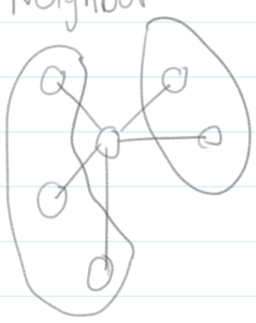
0	A
1	B
2	C
3	D

A → B → C → D
B → A
C → A → D
D → A → C

push back ได้
ถ้าจะ find ก็วนหาเอาได้

Map index to object

Neighbor



เพื่อนบ้าน

Runtime
- ขึ้นอยู่กับทั้ง Vertex และ Edge
เช่น O(|V|+|E|)
ความหนาแน่น graph E เทียบ V

โครงเหล็ก

Graph Depth First Search

path - ลำดับของ Vertex

Reachability - เข้าถึงได้ย้อนหลัง
เพิ่ม boolean visited
if v.visited ไปมาแล้ว
→ ปั๊มทางไว้จะได้ไม่ซ้ำ

Exploration
visited(v) ← true
for (v,w) ∈ E
if not visited(w);
Explore(w)

BFS (visited, cc num)
- Length = edge
- Distance = Length ที่สั้นที่สุด, ไม่มีระยะทางหลบ ∞, s.v = 0 → ①
BFSC(G,S) dist
v = เพื่อนบ้าน, u = เรา

Shortest-path tree
constructing (prev)
Reconst ใช้ stack

O(|E|+|V|)

ฝากไว้เผื่อสอบ

push a เข้า q
pop a set dist = 0
push b เข้า q
push C เข้า q
pop b set dist = 1
push D เข้า q
push E
push ไปเรื่อยๆ จนหมด

Theorem — ทุกๆ vertex ที่ เชื่อมต่อกัน จะถูก Mark ที่ visit

DFS (G)

Runtime
- total O(|V|+|E|)

Connect component → มีกี่กลุ่ม ใช้จัดกลุ่มข้อมูล

*ก่อนตอบ เวลาหา Runtime ใช้ |V|,|E| แทน n เสมอ~