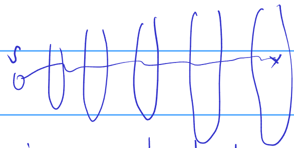
Graph algorithms

Recap - Graph representation

- DFS / BFS

↓ → levels



Levels สอดคล้องกับ shortest path length จาก s

Input graph → Gmm *

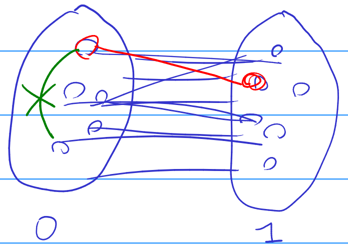
Input ↗

Two coloring : · ให้กราฟ $G=(V,E)$ ไม่มีทิศทาง. $n=|V|$, $m=|E|$.

· หา assignment ของ สี $c: V \to \{0,1\}$ ที่ไม่มี จุดยอด ที่ติดกัน มี สี เดียวกัน (proper coloring)

adjacent

▶ ถ้าทำได้



สามารถแบ่ง V เป็น A, B ($\subseteq V$, $\subseteq V$)

$A \cup B = V$

$A \cap B = \emptyset$ disjoint.

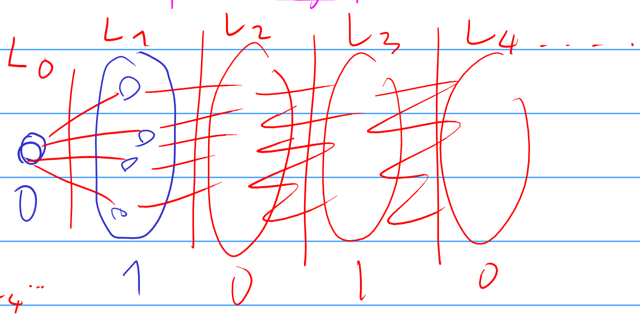
ที่ ทุก edge $(u,v) \in E$

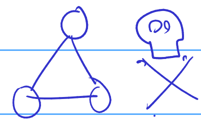
(1) $u \in A$, $v \in B$

(2) $v \in A$, $u \in B$.

ถ้า สามารถ แบ่งได้

จะเรียก G ว่าเป็น bipartite graph





Candidate:

รัน BFS เริ่มที่ไหน s

1 ครั้ง พอไหม

→ assign ให้

ทั้งหมดใน $L_0 \cup L_2 \cup L_4 \ldots$

→ ให้ $c(u)=0$ $\forall u \in L_0 \cup L_2 \cup \ldots$

→ ให้ $c(u)=1$ $\forall u \in L_1 \cup L_3 \cup L_5 \cup \ldots$

odd-length cycle

C_i ที่ เป็นคี่

→ check ว่า c ok?

→ ถ้า ok — ตอบ ✓

→ ถ้า ไม่ ok — NO!

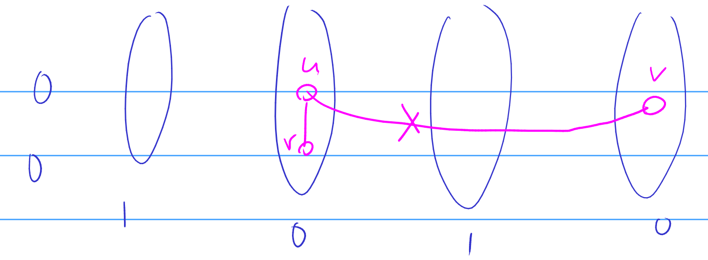
Lemma ถ้า algo ตอบ NO จะมี odd-length cycle. ⇒ กราฟแบ่ง 2 สี ไม่ได้

$P \Rightarrow Q$

Proof:

assume มี edge (u,v) ที่ $c(u) = c(v)$ ⇒ u และ v อยู่ level เดียวกัน

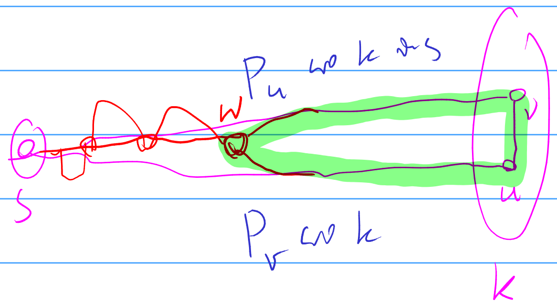
ใน level k



$P_u \cdot (u,v) \cdot P_v$ → cycle ยาว $2k+1$

bipartite test

two teams



P_u ยาว k ตั้ง s

P_v ยาว k

▷ ถ้า algo ตอบ NO → ไม่มีการแบ่ง (มี odd len)

▷ ถ้าแบ่งได้ 2 สี → algo จะตอบ YES / ตอบ N → ไม่ได้

▷ มี odd-len cycle → ไม่สามารถแบ่ง 2 สีได้

+

▷ ไม่มี odd-length cycle → แบ่ง 2 สีได้.

มี odd len cycle ⟺ แบ่ง 2 สีไม่ได้.

obstacle

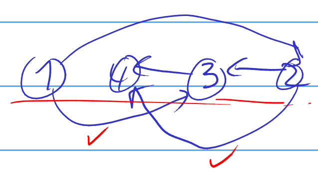
Topological ordering

Input: directed graph $G=(V,E)$ $\qquad n=|V|,\ m=|E|$

Output: ลำดับของจุดยอด $v_1, v_2, \ldots, v_n$ ที่ไม่มี edge $(v_i, v_j) \in E$ ที่ $i > j$

degree

 ✓

 ไม่มี topo order

in-deg

out

G เป็น DAG ⟺ G มี topo order.

▷ ถ้า G มี cycle ⇒ ไม่มี topological ordering.

?? G มี cycle ⇐?✓ ไม่มี topo order ≡ ถ้า G ไม่มี cycle ⇒?? G มี topo ordering

obstacle สำหรับ topo ordering.

▷ out ← [] (เก็บ indeg)

▷ while $|V| > 0$:

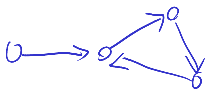
  หา u ที่ indegree = 0 $O(n)$

  out ≪ u

  ลบ u จาก G. (check indegree 0 ไหม)

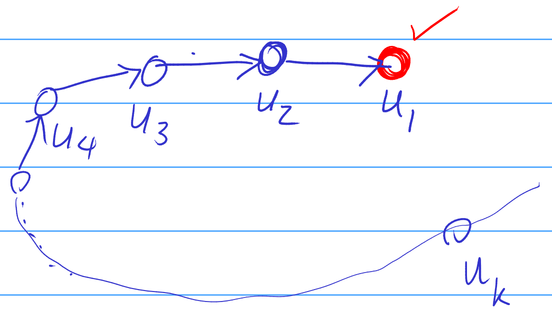
$O(n^2+m)$

ถ้า G ไม่มี cycle จะ output topological ordering.

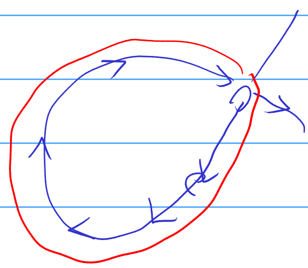


Lemma: ถ้า directed graph G ไม่มีโหนด u ที่ indegree(u) = 0 G จะมี cycle.

Prf:



$u_k = u_\ell$



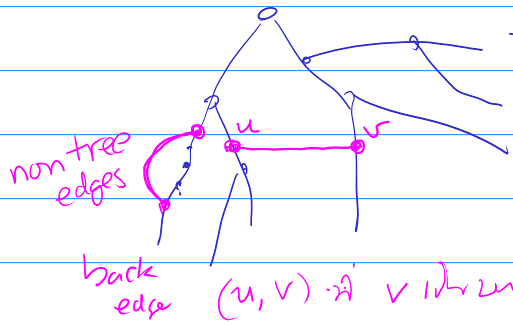
G จะมี cycle ถ้า proof นี้ถูกต้องไหม

▶ ถ้ากราฟไม่มี cycle จะมี node ที่ indegree = 0

▶ $O(n+m)$ Implementation:

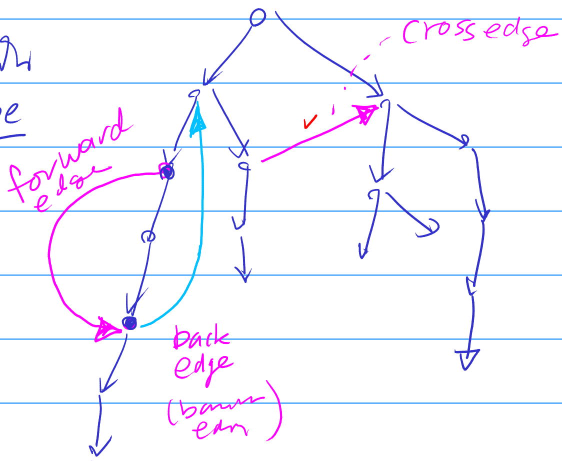
DFS ▷ undirected

▷ directed



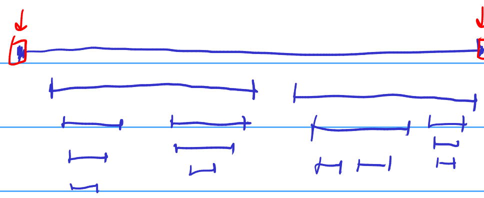
(u,v) ที่ v เป็นบรรพบุรุษของ u

ได้ T เป็น DFS tree



(u,v) เป็น

- forward edge ถ้า v เป็นลูกหลานของ u บน T
- backward edge ถ้า u เป็นลูกหลานของ v บน T
- cross edge ไม่ใช่อื่น



dfs(u)

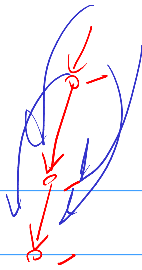
tin ← timer; timer++

tout ← timer; timer++

out.push_front(u)

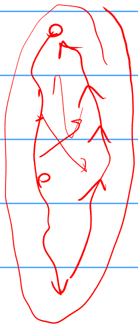
HW

▶ ถ้า G มี cycle จะมี backward edge เสมอ



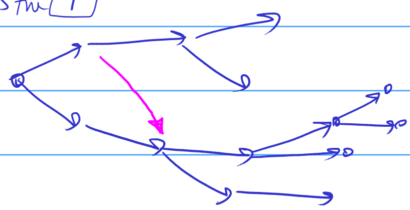
Topological ordering

ถ้ามี topological order ⟺ $\boxed{G \text{ จะไม่มี cycle}}$



$\boxed{G \text{ ไม่มี}}$ backward edges

dfs tree $\boxed{T}$



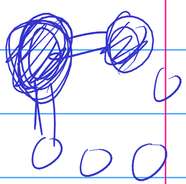
strongly connected component

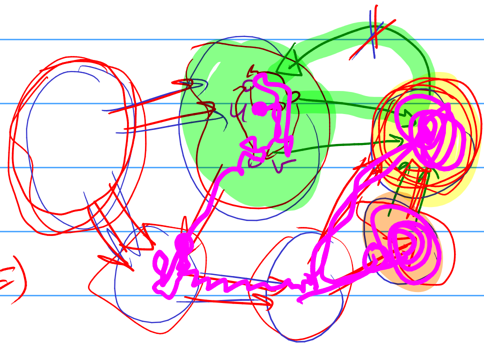
Strongly connected component (SCC)

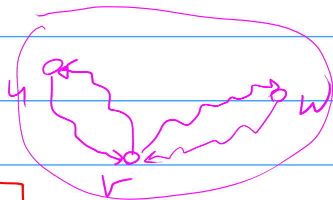
Definition: สำหรับ graph $G=(V,E)$

u และ v อยู่ใน SCC เดียวกัน ถ้า มี path $u \rightsquigarrow v$

$v \rightsquigarrow u$.







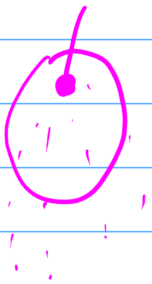
$G=(V,E)$

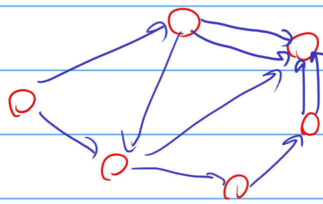
edge ข้าม

$\boxed{H=(V_s, E_s)}$

V_s เซตของ SCC

E_s edge ที่เชื่อมระหว่างโหนด





H เป็น DAG — directed acyclic graph

SCC

DFS-ALL(G) //เก็บ ลำดับที่ return A

DFS-ALL(G^R) visit ตามลำดับ reverse ของ A