Shortest paths

$\ell(u,v)$
$\ell(e)$

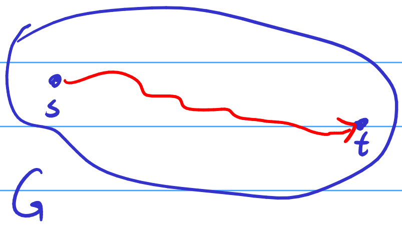
Graph $G=(V,E)$, $n=|V|$, $m=|E|$.

Length $\ell : E \to \mathbb{R}$ (ความยาว ≥ 0 โดยทั่วไป, ยกเว้นที่ความยาว <0)

ให้ path $P=(u_0,u_1,u_2,\dots,u_k)$

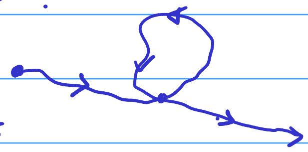
นิยาม $\ell(P) = \ell(u_0,u_1) + \ell(u_1,u_2) + \dots + \ell(u_{k-1},u_k)$

• ให้ s,t เพื่อโดยที่จะหา shortest path จาก s ไป t.



• ไม่มี shortest path — ① ไม่มี path
② มี path จาก s ไป t (st-path)

path จาก s ไป t = ?
• มีไม่จำกัด

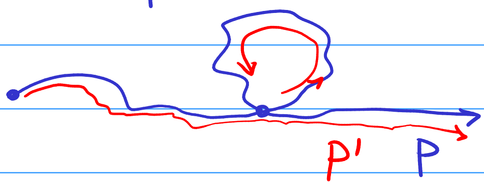


simple path จาก s ไป t

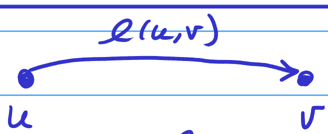
Def: negative length cycle

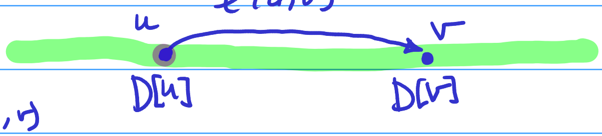
▶ จะมี shortest path จาก s ไป t ถ้าไม่มี negative-length cycle C ที่ s ไปถึง C และ C ไปถึง t.

Proof: • สำหรับ st-path P ใดๆ จะมี simple st-path P' ที่ไม่ยาวกว่า P



• ดังนั้นจะมี shortest path ที่เป็น simple st-path → มีจน. จำกัด
→ มี path ที่สั้นที่สุด





จะกล่าวว่า edge (u,v) relaxed เมื่อเทียบกับ D
ถ้า $D[v] \leq D[u] + \ell(u,v)$.

if $D[v] > D[u] + \ell(u,v)$
$D[v] \leftarrow D[u] + \ell(u,v)$

s - source node

~~edu~~ → D[

สมมุติ $D[s]=0$, $\boxed{\text{ถ้าทุก edge ในกราฟ relaxed}}$.

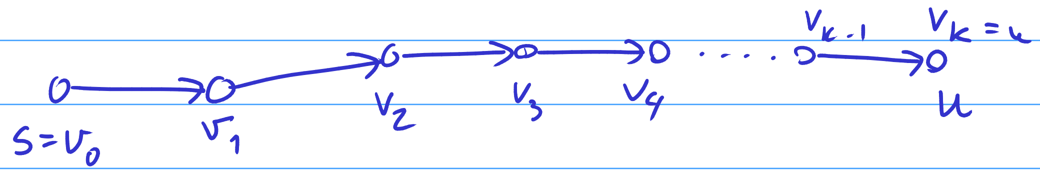
edge (u,v)

$D[v] \le D[u] + \ell(u,v)$

$\boxed{\ell(u,v) \ge D[v] - D[u]}$

▶ $D[u]$ จะมีค่าไม่เกินความยาวของ shortest path จาก s ไป u.

Proof: พิจารณา su-path P ใดๆ



$$\ell(P) = \ell(v_0,v_1) + \ell(v_1,v_2) + \ell(v_2,v_3) + \cdots + \ell(v_{k-1},v_k)$$

$$\ge (\cancel{D[v_1]} - D[v_0]) + (\cancel{D[v_2]} - \cancel{D[v_1]}) + (\cancel{D[v_3]} - \cancel{D[v_2]}) + \cdots + (D[v_k] - \cancel{D[v_{k-1}]})$$

$$\ge D[v_k] - D[v_0] = D[u] - D[s] = D[u]$$

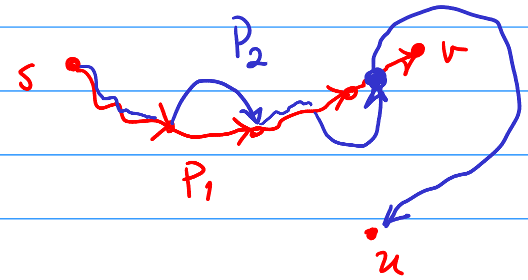
▶ ถ้าทุก edge ในกราฟ relaxed, $D[s]=0$ และมี path จาก s ไป u ที่ยาว $D[u]$, path นั้นจะเป็น shortest path.

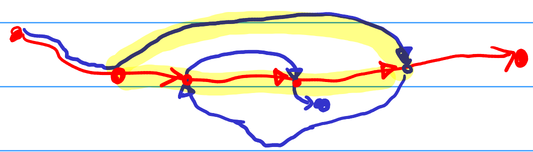
▶ **Single-source shortest path.**

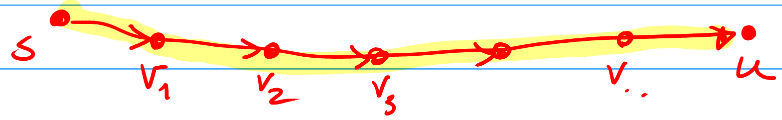
- label : $D[u]$ ....
- paths : shortest paths จาก s ไป node ต่างๆ — n paths, $O(n^2)$

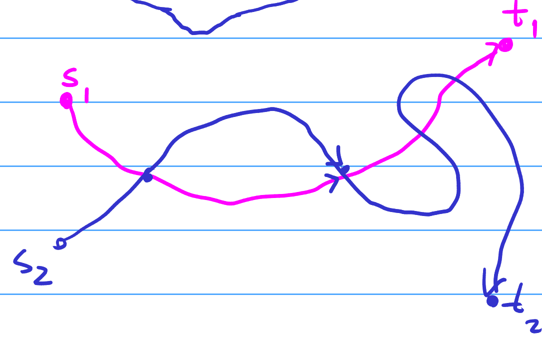
↳ เก็บเป็น tree (shortest path tree) แต่ละ [illegible]

$O(n)$



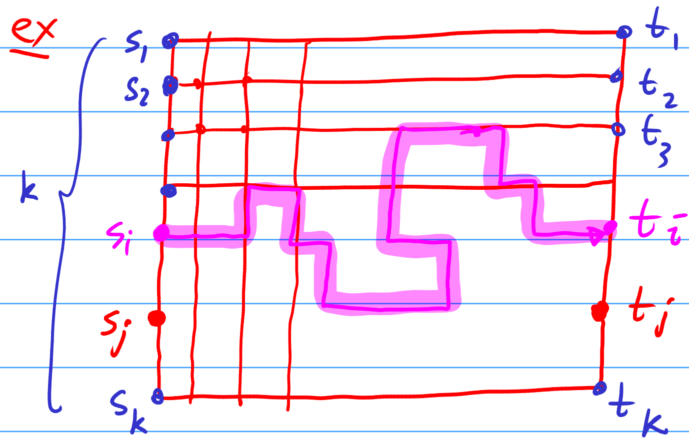




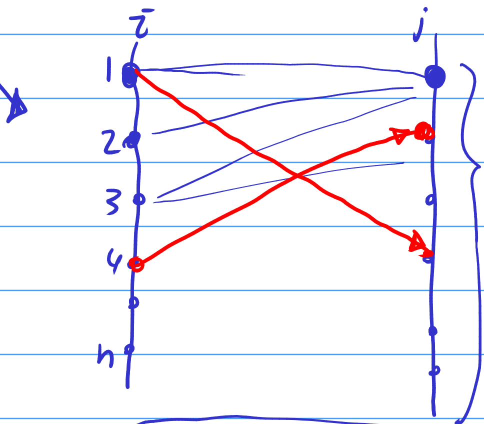


edge ยาวบวก ≥ 0

ex



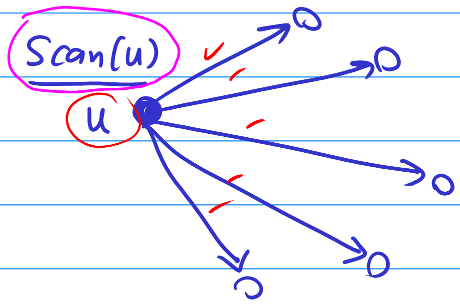
หา shortest path จากทุก s_i ไป t_i

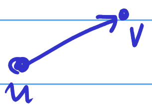


$\boxed{D[i] + d(i,j) + E[j]}$

ถ้ารู้ shortest paths จาก s ไปทุก node → มี shortest path tree T ที่ทุก path จาก s ไป node u ที่อยู่บน T จะเป็น shortest path.

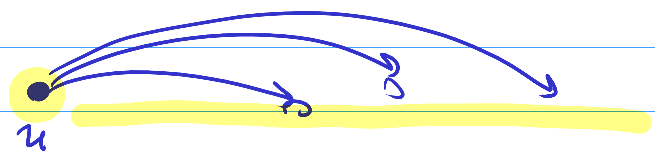
D[u] ← ∞ ∀u, D[s] ← 0
parent[s] ← nil
while มี edge (u,v) ที่ยังไม่ relaxed.
    เลือก edge (u,v) ที่ยังไม่ relaxed
        relax (u,v):
            D[v] ← D[u] + ℓ(u,v)
            parent[v] ← u.



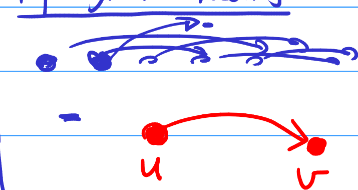


Efficient order ▶ DAG (directed acyclic graph)

- relax edge ตามลำดับ topological order ของจุดปลายด้านต้นของ edge.
-



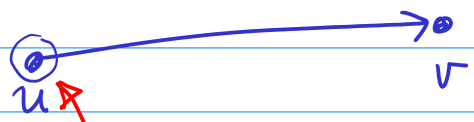
topological ordering



▶ **กราฟที่ไม่มี weight เป็นลบ.** (Dijkstra's algorithm).

$O(n^2)$

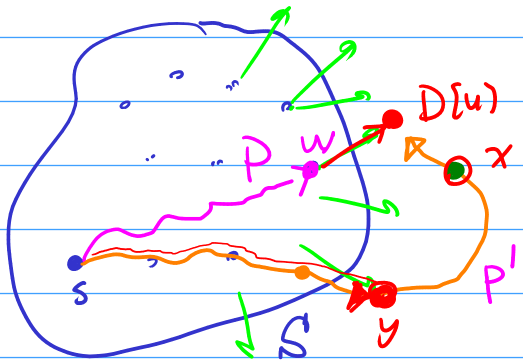
D[s] ← 0   Scanned[u] ← false ∀u.
D[u] ← ∞ ∀u
while มี node ที่ยังไม่ scan:
    u ← node ที่ scanned[u] = false และ D[u] ต่ำสุด
    Scan(u)
    scanned[u] ← true.



- (u,v) ไม่ relaxed
- ถ้า $scan(u)$
  edge (u,v) relaxed
- ถ้าเมื่อจบการทำงาน edge (u,v)
  $\boxed{\text{ต้อง update } D[u]}$

▶ เมื่อเริ่ม $scan(u)$, $D[u]$ เป็น shortest path distance จาก s ไป u

Proof: • $D[s] \overset{=0}{}$ เป็น shortest path distance จาก s ไป s



ให้ S เป็น set ของ node ที่ scanned แล้ว.

• induction hypothesis:
 · ทุกโหนด w ใน S, $D[w]$ เป็น shortest path distance จาก s ไป w

• ถ้า u เป็นโหนดที่จะไปถูก scaned แล้ว
ค่า $D[u]$ ต่ำสุด $\Rightarrow$ $D[u]$ จะเป็น shortest path dist จาก s ไป u.

• prove by contradiction.
 · assume $D[u]$ ไม่ใช่ shortest path distance จาก s ไป u
 · มี sub-path P' ที่ $\ell(P') < D[u]$.
 · claim: มี node $x \notin S$ ที่อยู่ใน P' ก่อนถึง u.
 · claim: มี node $y \notin S$ ที่เป็นโหนดแรกใน P' ที่ $\notin S$

y อาจจะเท่ากับ x ได้

 · สังเกตว่า $\ell(P') = dist(s,y) + dist(y,x) + \ell(x,u)$
  แต่ $\ell(P') < D[u]$
  ดังนั้น $dist(s,y) < D[u]$
  เพราะ: $dist(y,x) + \ell(x,u) \geq 0$

เมื่อ $dist(a,b)$ = shortest path distance จาก a ไป b

 · claim: $D[y] = dist(s,y)$
 $\Rightarrow D[y] < D[u]$ – ได้ contradiction ■

ครั้ง

Running time

$D[s] \leftarrow 0$ Scanned$[u] \leftarrow$ false $\forall u$.
$D[u] \leftarrow \infty$ $\forall u$

while มี node ที่ยังไม่ scan: | PQ

hn $O(n)$ $u \leftarrow$ node ที่ scanned$[u]$ = false และ $D[u]$ ต่ำสุด | $O(\log n)$

Scan(u):

for each edge $(u,v) \in E$ — $O(\deg(u))$, รวม $O(m)$ | รวม $O(m \log n)$

if $D[v] > D[u] + \ell(u,v)$

$D[v] \leftarrow D[u] + \ell(u,v)$

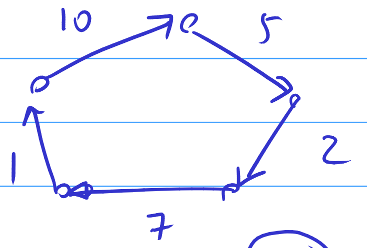
scanned$[u] \leftarrow$ true. | $O((n+m)\log n)$

↓ Fibonacci Heap $O(n\log n + m)$

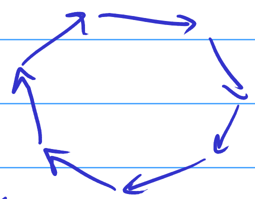
Special case $\ell(e)$ เป็น จน.เต็ม ที่มีค่า $\le \boxed{B}$

General weights

Minimum mean cycle. ในกราฟ G, หา cycle ที่มี ความยาว (length ℓ) เฉลี่ยต่ำสุด.



$\frac{25}{5} = 5$



Guess answer α: $G = (V,E)$, ℓ_α

ℓ_α นิยาม $\ell_\alpha(u,v) = \ell(u,v) - \alpha$

α สูงไป → มี negative cycle

α พอดี → มี cycle ความยาว 0

α ต่ำไป → ไม่มี negative cycle & ไม่มี cycle ยาว 0.

Binary search

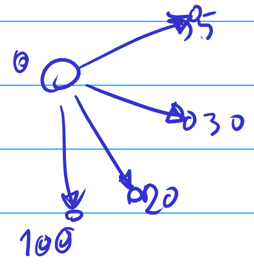
ex: $b(u,v)$ แทน อัตราส่วน ในการแลกเงิน $u \to v$

$Cost(u_1,u_2,u_3,\dots,u_k) = b(u_1,u_2)\cdot b(u_2,u_3)\cdot b(u_3,\dots)\dots$

Bellman-Ford-Moore

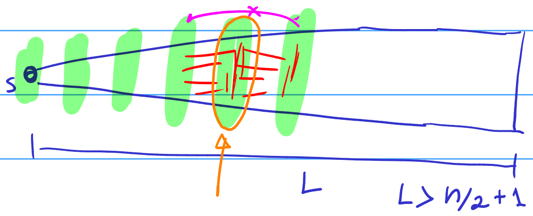
Assume กราฟไม่มี negative-length cycle

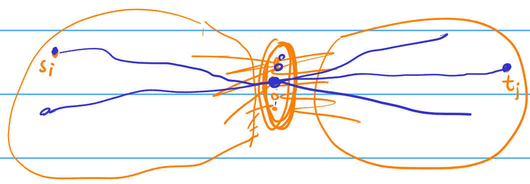
$\boxed{\begin{array}{l}\text{while ทำ } n-1 \text{ รอบ.} \\ \quad \text{relax all edges.}\end{array}}$



หลังจากทำงาน i รอบ
ทุกโหนดที่มี shortest path
จาก s ไปโหนดนั้นที่มี $\leq i$ edge
จะมี Distance label ถูกต้อง.

ให้กราฟ $G=(V,E)$, $n=|V|$, G is connected



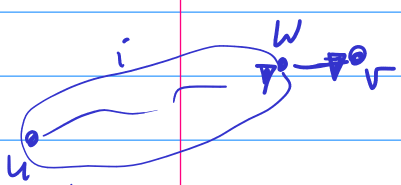


All-pair shortest paths (ทำงานได้แม้จะมี negative length)

▷ Dynamic programming

ให้ $D[i,u,v]$ แทนความยาว shortest path จาก u ไป v ที่มี $\leq i$ edge

- base case
- ถ้ามี $D[i,\cdot,\cdot]$ จะคำนวณ $D[i+1,\cdot,\cdot]$



$$\boxed{D[i+1,u,v] = \min \begin{cases} D[i,u,v] \\ \min\limits_{(w,v)\in E} D[i,u,w] + \ell(w,v) \end{cases}}$$

$$\boxed{D_k[i,j] \leftarrow \min \begin{cases} \underline{D_{k-1}[i,j]} \\ \underline{D_{k-1}[i,k] + D_{k-1}[k,j]} \end{cases}}$$

$\boxed{\text{Floyd-Warshall}}$

```
for k ← 1, 2, ..., n
    for i ← 1, 2, ..., n
        for j ← 1, 2, ..., n
            if D[i,j] > D[i,k] + D[k,j]
                D[ ← 
```

โดยที่ $D_k[i,j]$ = ระยะทางสั้นสุดจาก $i \to j$
โดยวิ่งผ่าน vertex $\{1,2,\ldots,k\}$

Graph ที่มี negative edge weights

คำนวณ all-pair shortest path.

① FW $O(n^3)$

② Run BellmanFord Moore n รอบ $O(mn \cdot n)$ แย่กว่า $O(n^3)$

③ Run Dijkstra $\underline{n \text{ รอบ}}$ $O(n \cdot (n+m) \log n)$

$= O(n^2 \log n + \underline{\underline{nm}} \log n)$

▶ $\boxed{\text{ปรับ edge weight}}$ ให้ ① ไม่มี weight เป็นลบ

② shortest path ในกราฟแสดงเหมือนเดิมทั้งหมด

สำหรับแต่ละ node u ให้ $\boxed{p(u)}$ แทน price function ที่ u

reduced cost

$$\underbrace{\ell_p(u,v)} = \underbrace{p(u)}_{\text{ซื้อ}} + \underbrace{\ell(u,v)}_{\text{ค่าขน}} - \underbrace{p(v)}_{\text{ราคาที่ขาย}}$$

พิจารณา path $P = (s = u_0, u_1, u_2, \ldots, u_k = t)$

$$\ell(P) = \ell(u_0,u_1) + \ell(u_1,u_2) + \ell(u_2,u_3) \cdots + \ell(u_{k-1},u_k)$$

reduced len ความยาวใหม่ $\ell_p(u_0,u_1) + \ell_p(u_1,u_2) + \cdots + \ell_p(u_{k-1},u_k)$

$$= \left(p(u_0) + \ell(u_0,u_1) - \cancel{p(u_1)}\right) + \left(\cancel{p(u_1)} + \ell(u_1,u_2) - \cancel{p(u_2)}\right) + \cdots + \left(\cancel{p(u_{k-1})} + \ell(u_{k-1},u_k) - p(u_k)\right)$$

$$= p(s) + \ell(P) - p(t)$$

หา price function p ที่ $\ell_p(u,v) \geq 0 \quad \forall (u,v) \in E$

$$0 \leq \ell_p(u,v) = p(u) + \ell(u,v) - p(v)$$

$$\Rightarrow \boxed{p(v) \leq p(u) + \ell(u,v)} \ \ldots$$

สังเกตว่า ค่า D เป็น shortest path distance จากของโหนด (s) ใดๆ
→ ทุกๆ edge จะ relaxed แล้ว

$$\forall (u,v) \qquad D[v] \le D[u] + \ell(u,v)$$

⇒ สามารถใช้ D เป็น price function ที่ทำให้

$$\ell_D(u,v) \ge 0 \qquad \forall (u,v) \in E .$$

Johnson's algorithm

- Run Bellman Ford Moore จาก s → ได้ distance fun D.
- ปรับ length $\ell \to \ell_D$: $\ell_D(u,v) = D[u] + \ell(u,v) - D[v]$
- for $u \in V$.

  รัน Dijkstra's alg จาก u

เวลา: $O(mn) + O(n \cdot (n+m) \log n)$

ให้ connected graph G ที่มี odd-degree node จน. k โหนด.

ลบ edge ให้เหลือแต่โหนด degree คู่ - ลบ edge ให้น้อยสุด.

hint: k = 2.