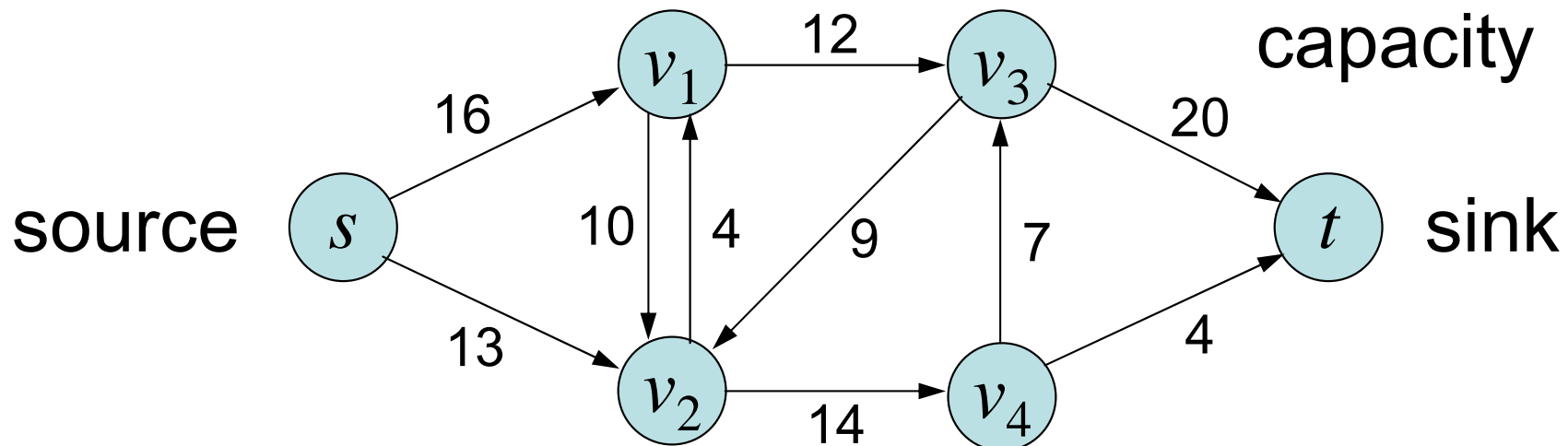


Max Flows

สมชาย ประสิทธิ์จตุระกุล
2550

การไหล (flow)

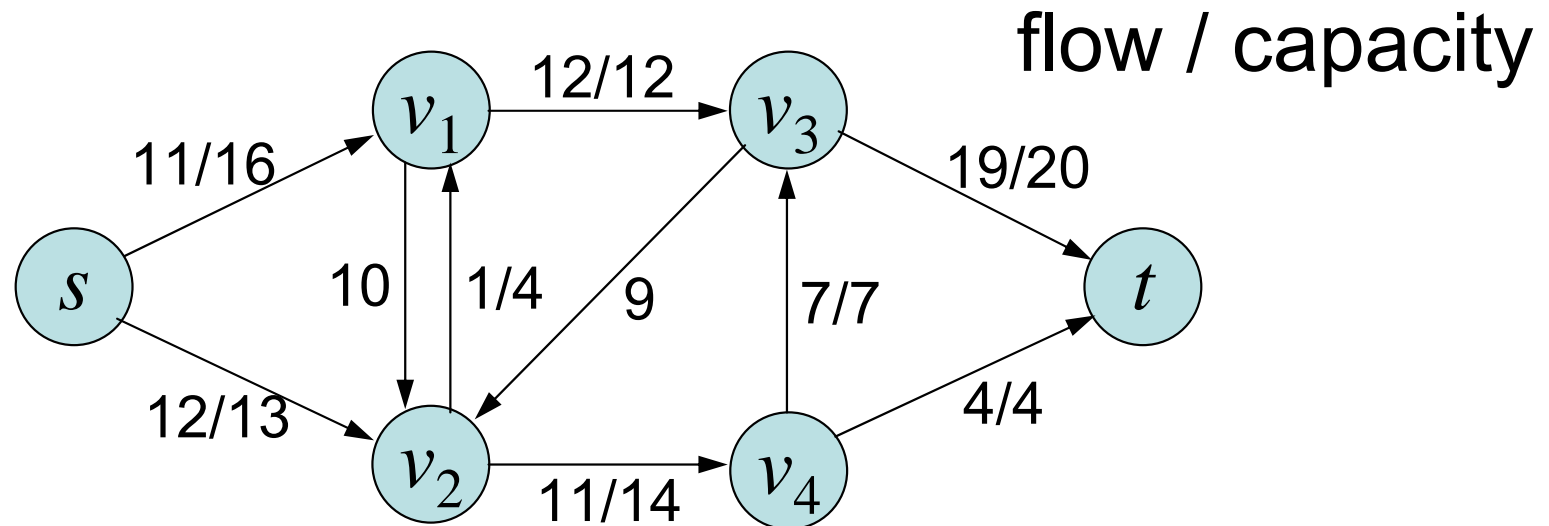
- เส้นเชื่อมแต่ละเส้นมีความจุเป็นจำนวนไม่ติดลบ
- ปมพิเศษ : source s และ sink t
- กำหนดให้ปมทุกปมอยู่บนวิถีใดวิถีหนึ่งจาก s ถึง t



หาปริมาณการไหลจาก s ถึง t ที่มากที่สุดในข่ายงาน

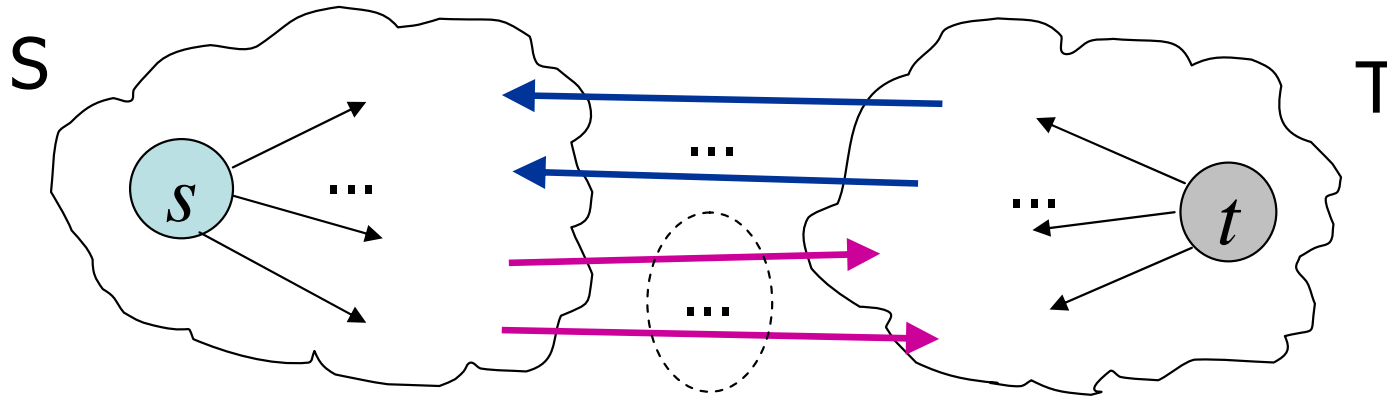
การไหล (flow)

- ให้ $f(u,v)$ คือปริมาณการไหลในเส้นเชื่อม (u,v)
 - ปริมาณการไหลในเส้นเชื่อมต้องไม่เกินความจุ
 - ปริมาณการไหลเข้าปมหนึ่ง ต้องเท่ากับปริมาณการไหลออกจากปมนั้น (ยกเว้นปม s กับ t)



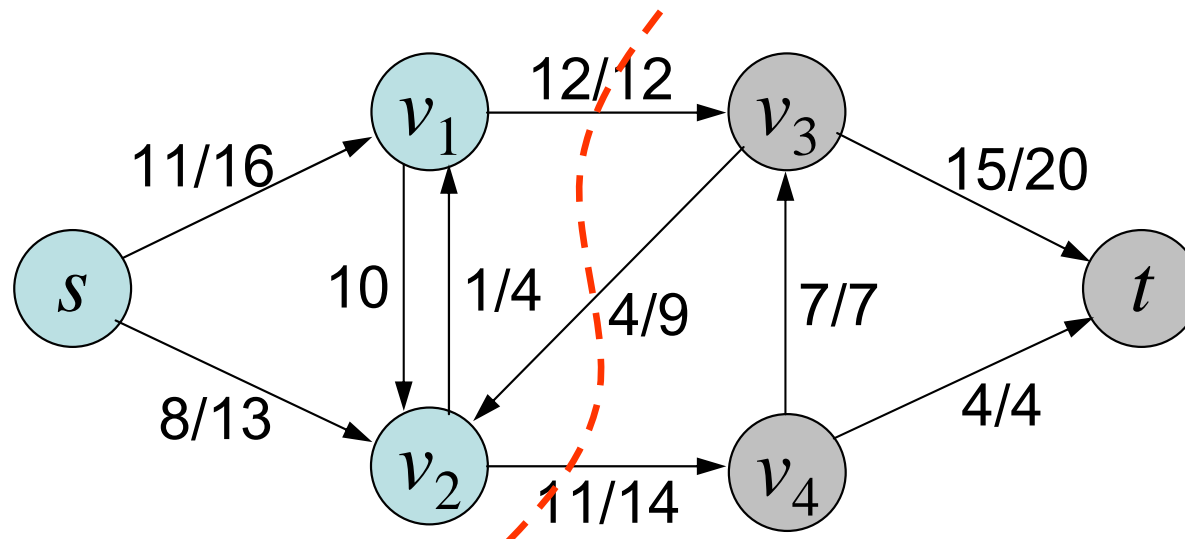
$$|f| = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t)$$

ส่วนตัด (cut)



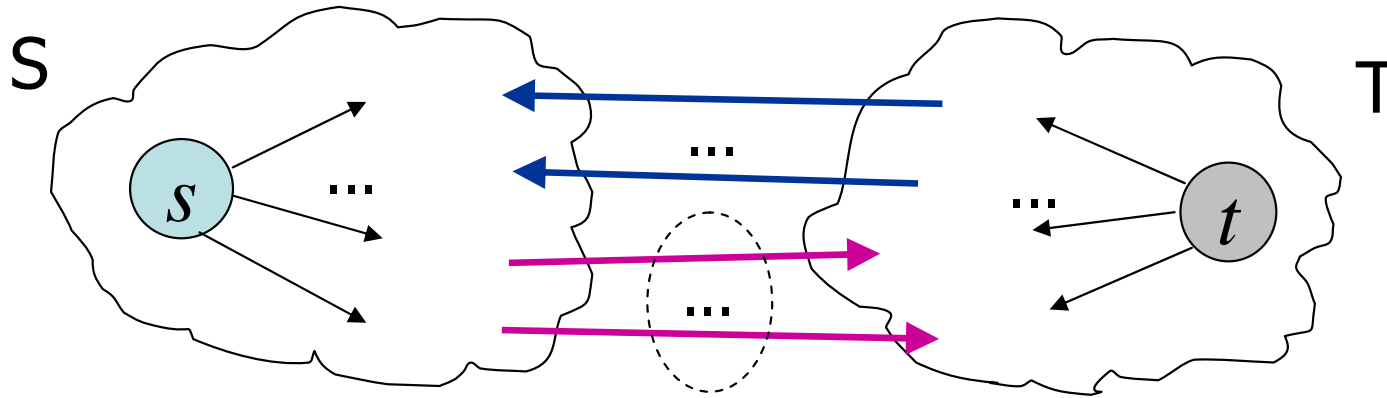
ความจุของส่วนตัด (S,T) = ผลรวมความจุเส้นเชื่อมจาก S ไป T

$$f(S,T) \leq c(S,T)$$

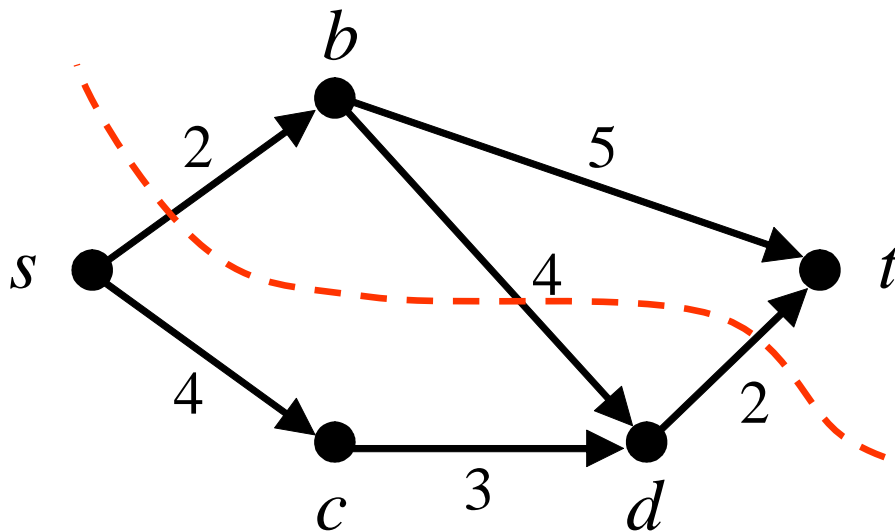


$$12 - 4 + 11 \leq 12 + 14$$

Max-Flow Min-Cut Theorem



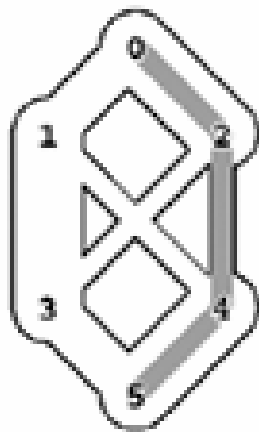
$$f_{\max}(S, T) = c_{\min}(S, T)$$



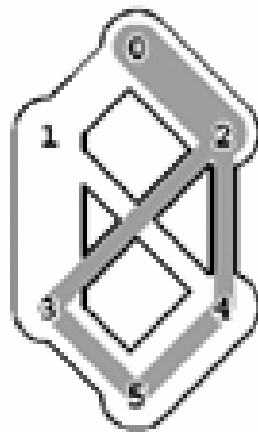
S	$c(S, T)$
{ s }	6
{ s, b }	13
{ s, c }	5
{ s, d }	8
{ s, b, c }	12
{ s, b, d }	11
{ s, c, d }	4
{ s, b, c, d }	7

วิธีหา max-flow

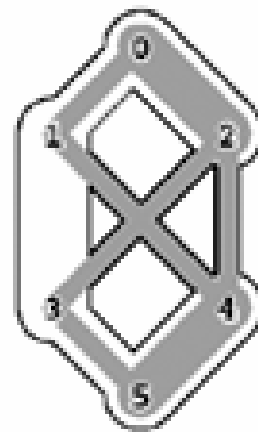
- Augmenting-path method



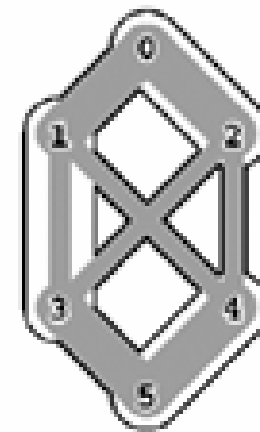
0-2-4-5



0-2-3-5

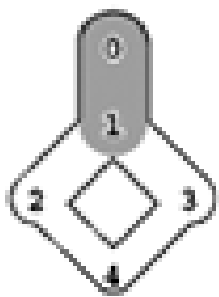


0-1-4-5

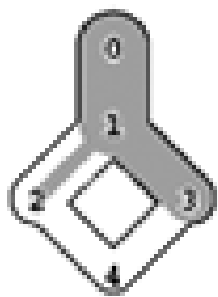


0-1-3-5

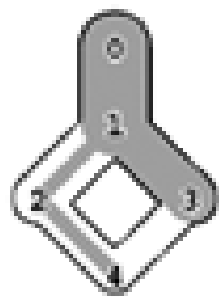
- Preflow-push method



1



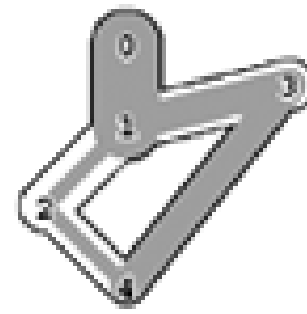
2 3



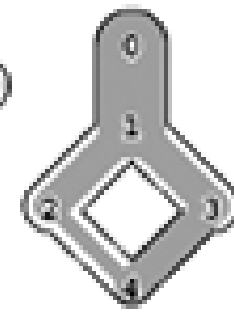
3



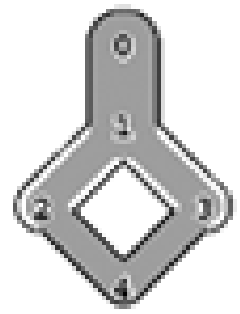
3



1

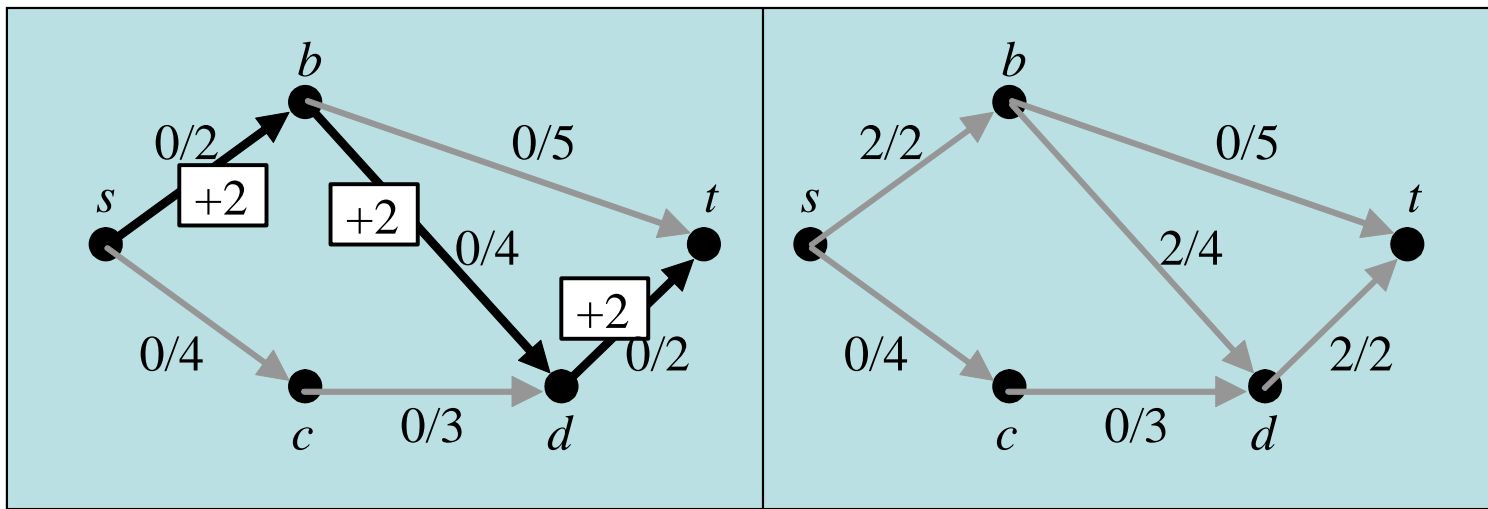
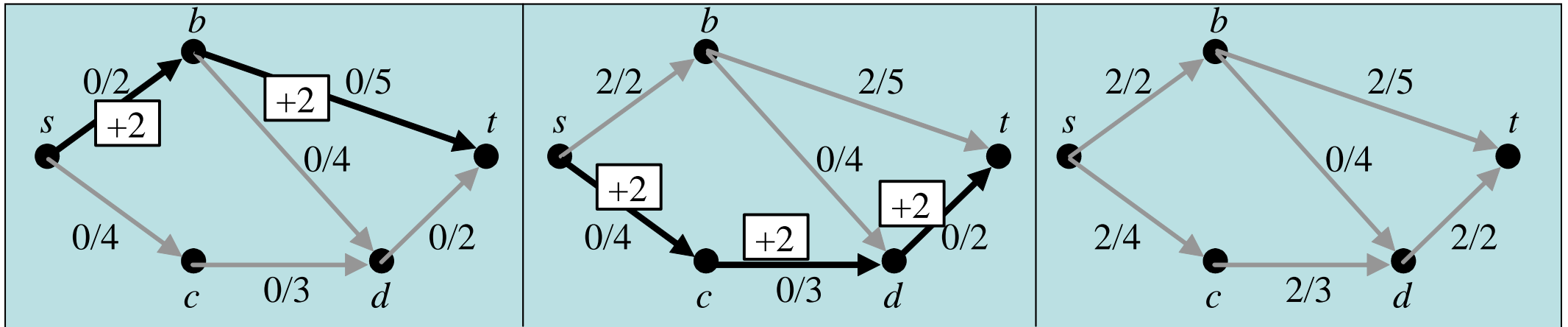
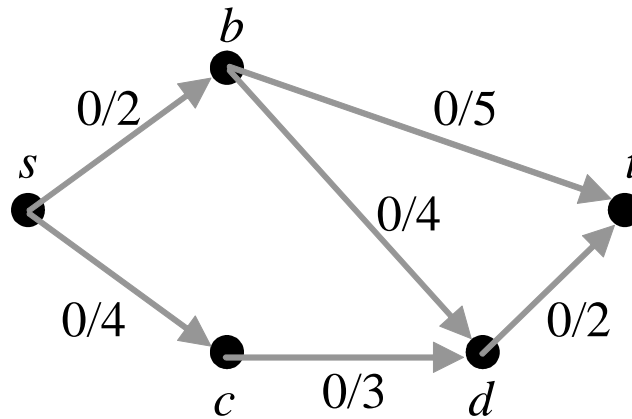


2

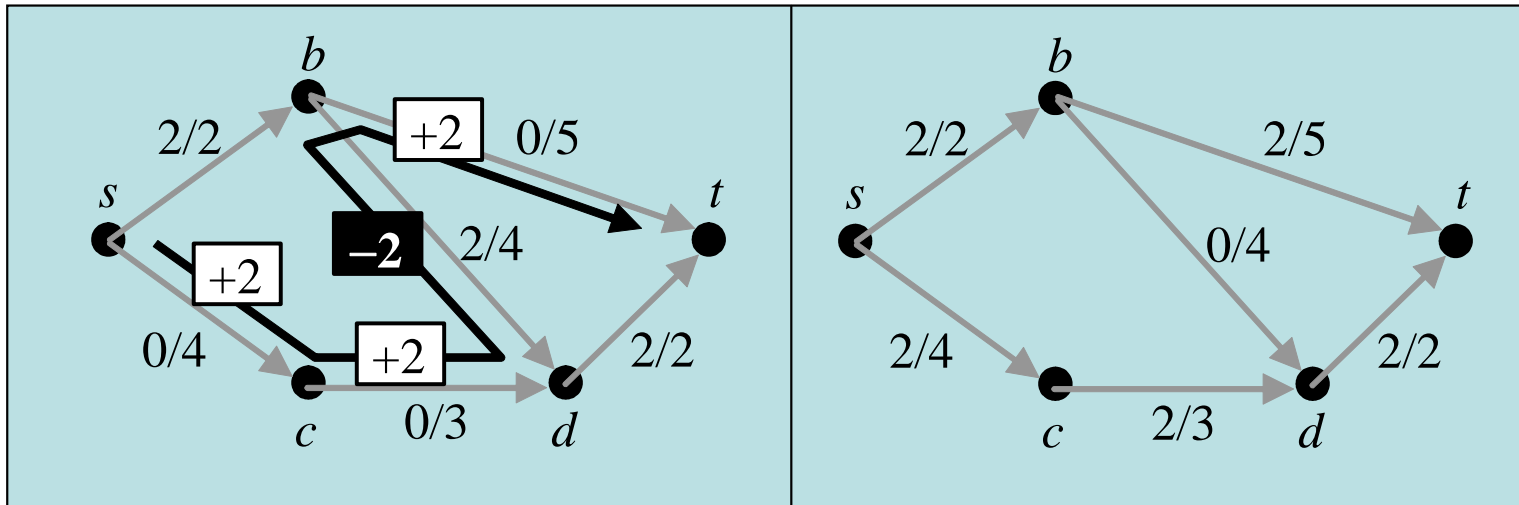
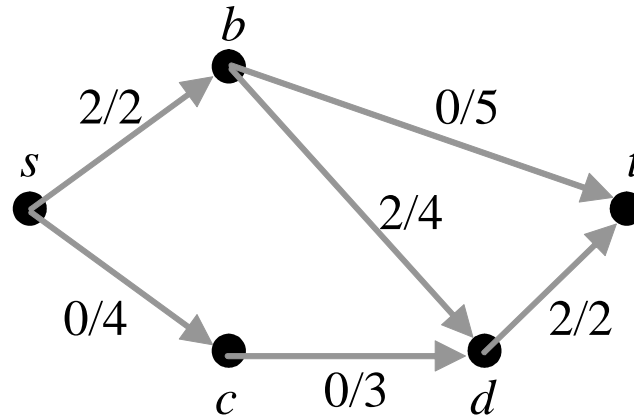


3

หาวิธีเพื่อเพิ่มการไหล



วิธีการไหลแบบส่วนที่ใส่ได้



Residual Network

- ให้ความจุคงเหลือ (residual capacity) ของเส้นเชื่อม (u,v) ในข่ายงาน G ที่มีการไหล f คือ

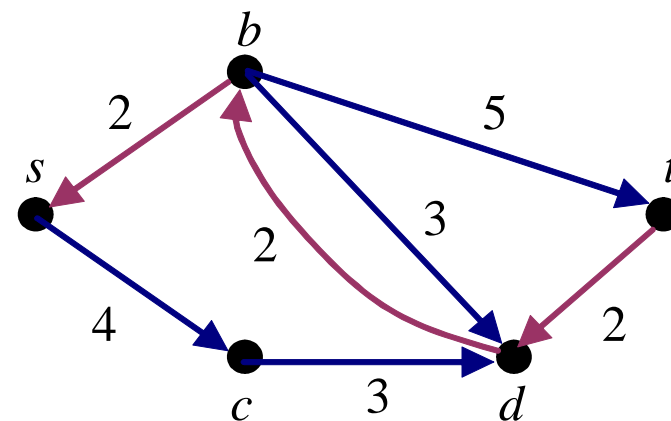
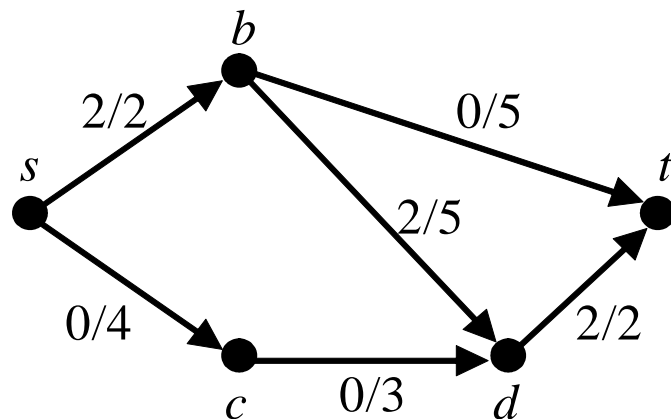
$$c_f(u,v) = c(u,v) - f(u,v)$$

- ให้ข่ายงานคงเหลือ (residual network) ของ G ที่มีการไหล f คือ

– ข่ายงานที่มีปมเหมือน G

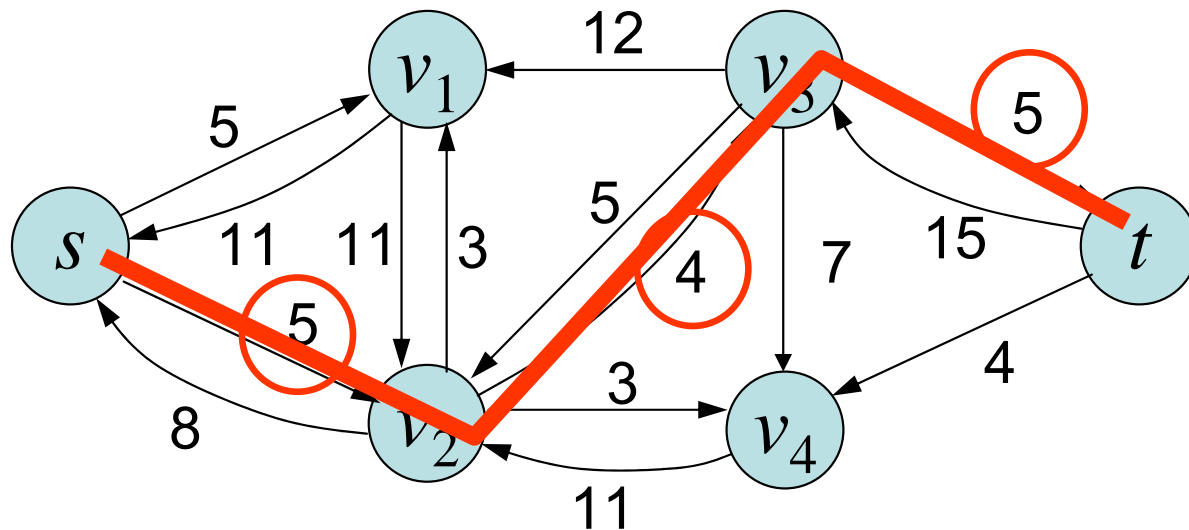
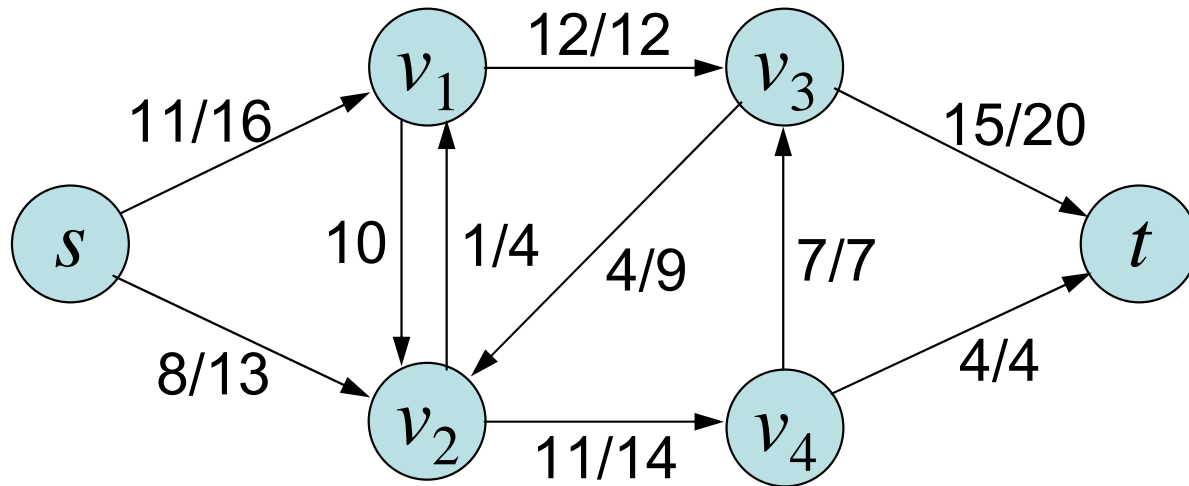
– มีเส้นเชื่อม (u,v) ที่มีความจุ $c_f(u,v)$ ถ้า $c_f(u,v) > 0$

– มีเส้นเชื่อม (v,u) ที่มีความจุ $f(u,v)$ ถ้า $f(u,v) > 0$



Augmenting Path Method

- หาวิถีแต่งเติม (augmenting path) ในข่ายงานคงเหลือ เพื่อเพิ่มการไหลในข่ายงาน

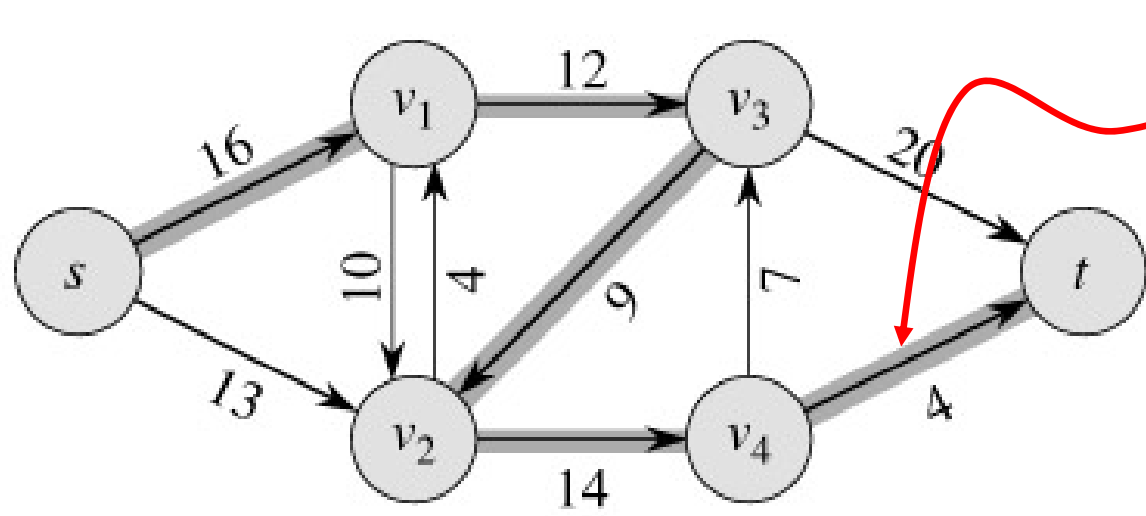


Ford & Fulkerson, 1962

FORD-FULKERSON(G, s, t)

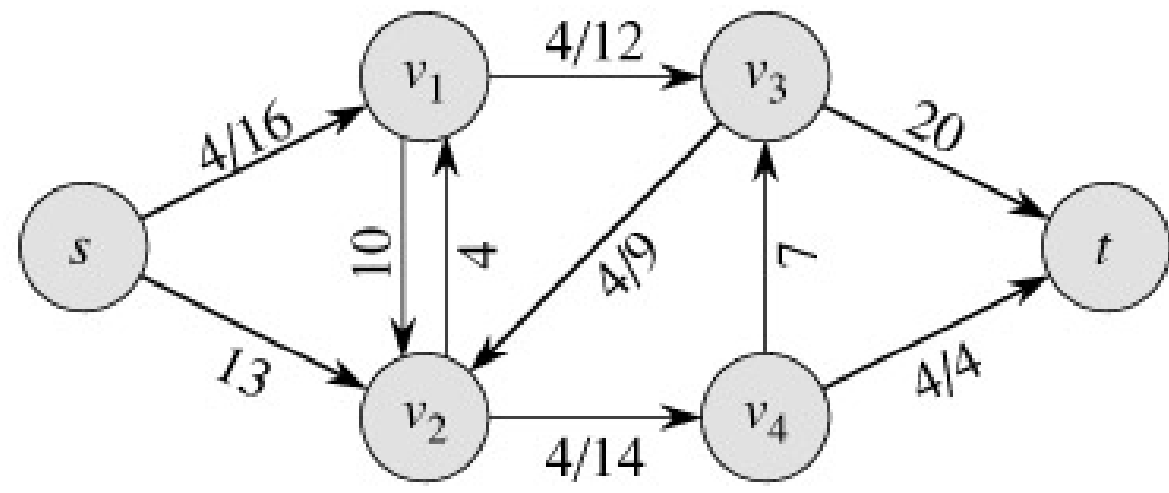
```
1  for each edge  $(u, v) \in E[G]$ 
2      do  $f[u, v] \leftarrow 0$ 
3      do  $f[v, u] \leftarrow 0$ 
4  while there exists a path  $p$  from  $s$  to  $t$  in the residual network  $G_f$ 
5      do  $c_f(p) \leftarrow \min \{c_f(u, v) : (u, v) \text{ is in } p\}$ 
6      for each edge  $(u, v)$  in  $p$ 
7          do  $f[u, v] \leftarrow f[u, v] + c_f(p)$ 
8          do  $f[v, u] \leftarrow -f[u, v]$ 
```

ตัวอย่าง



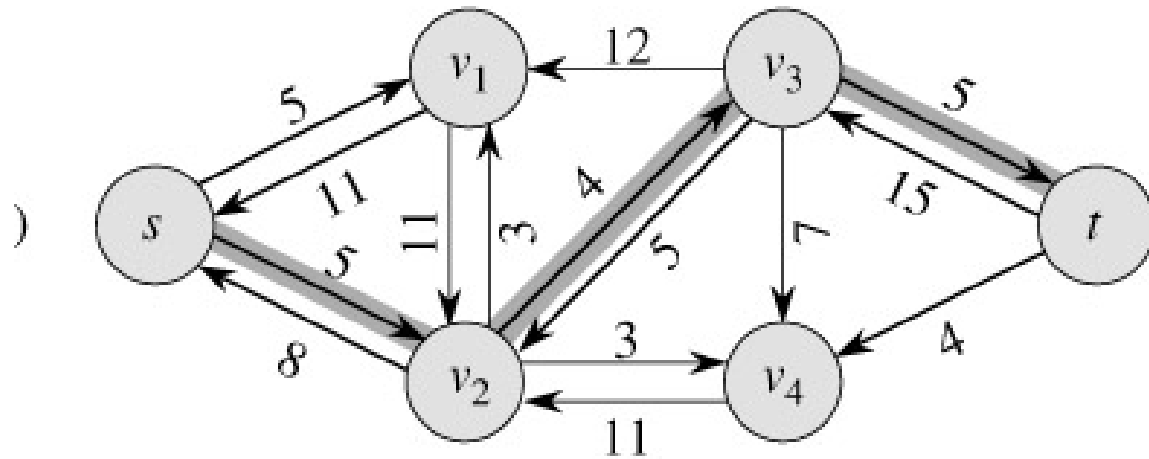
augmenting path

Original Network

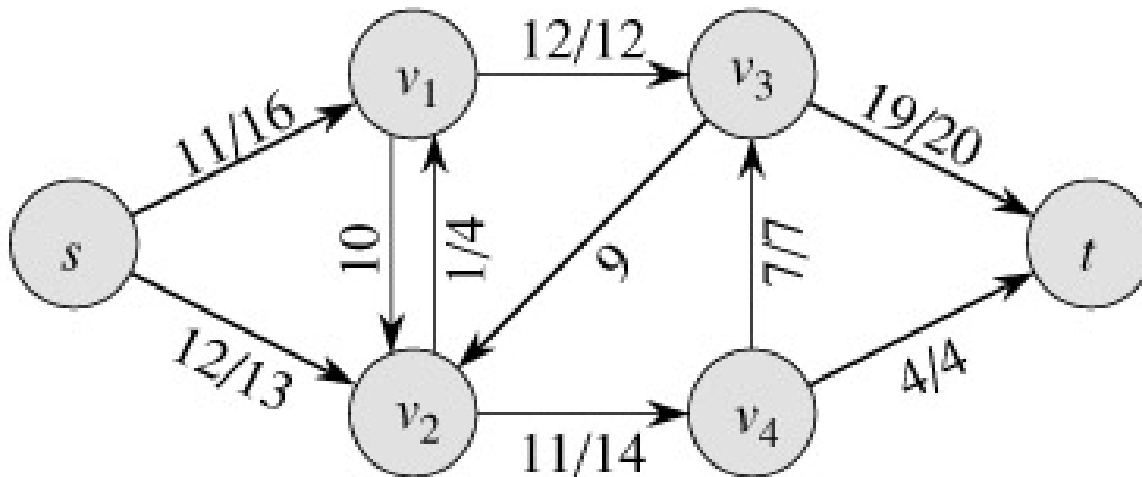


Resulting Flow = 4

ตัวอย่าง (ต่อ)



Residual Network



Resulting
Flow = 23

วิเคราะห์

- ถ้าความจุทุกเส้นเป็นจำนวนเต็ม วิธีแต่งเติมหนึ่งวิธี เพิ่มการไหลได้อย่างน้อยหนึ่งหน่วย
- ถ้า max flow คือ f^* , ใช้เวลารวม $O(E f^*)$

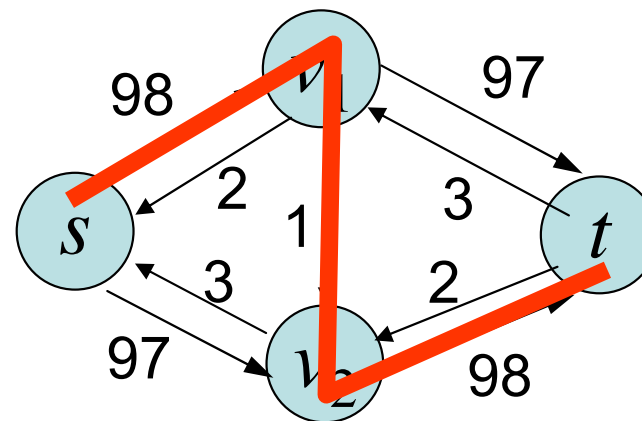
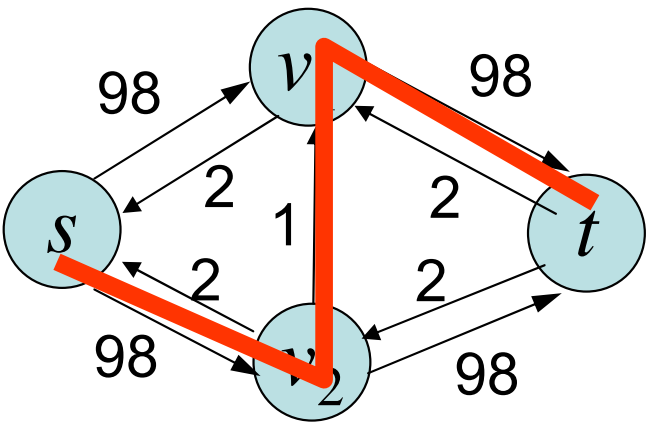
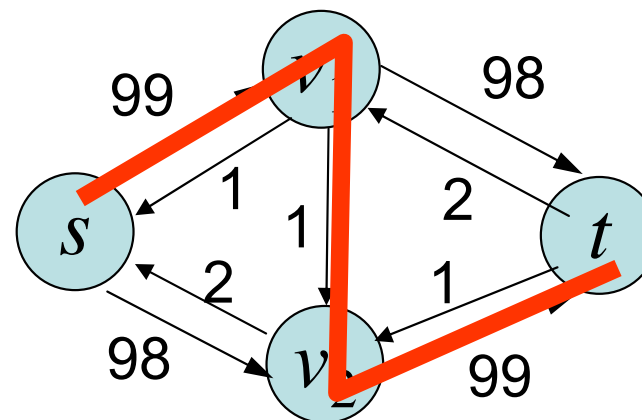
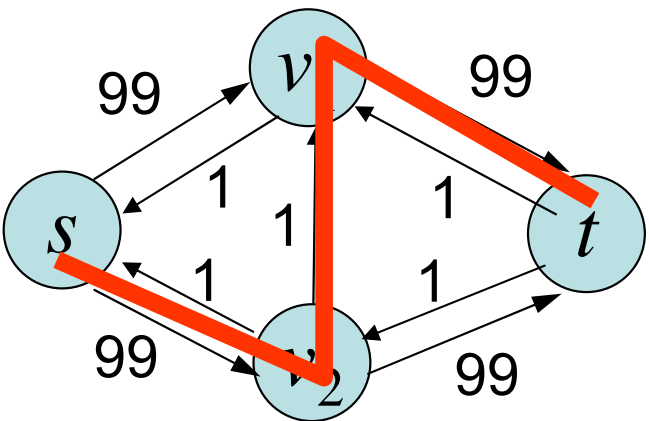
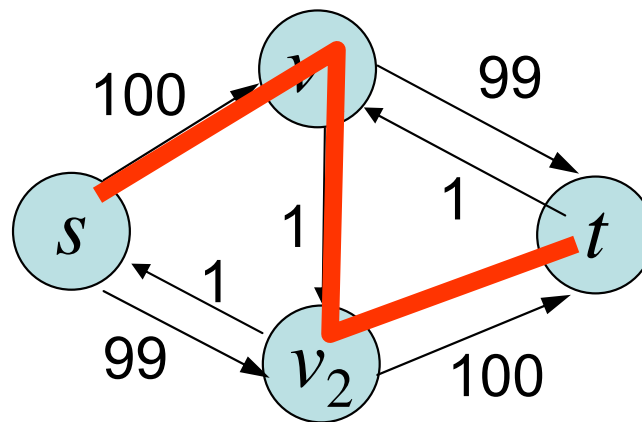
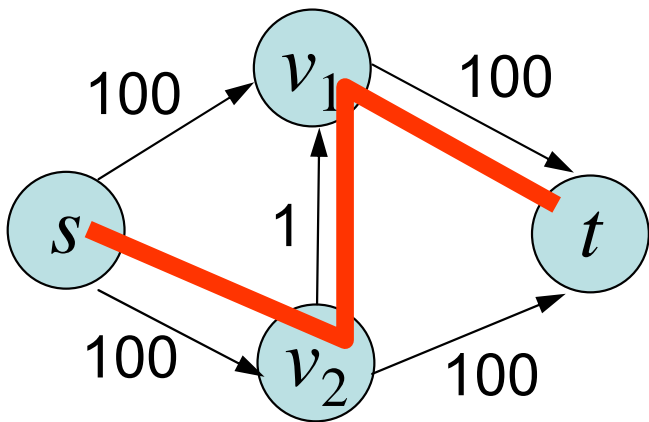
FORD-FULKERSON(G, s, t)

```
1  for each edge  $(u, v) \in E[G]$ 
2      do  $f[u, v] \leftarrow 0$ 
3      do  $f[v, u] \leftarrow 0$ 
4  while there exists a path  $p$  from  $s$  to  $t$  in the residual network  $G_f$ 
5      do  $c_f(p) \leftarrow \min \{c_f(u, v) : (u, v) \text{ is in } p\}$ 
6      for each edge  $(u, v)$  in  $p$ 
7          do  $f[u, v] \leftarrow f[u, v] + c_f(p)$ 
8          do  $f[v, u] \leftarrow -f[u, v]$ 
```

$O(E)$

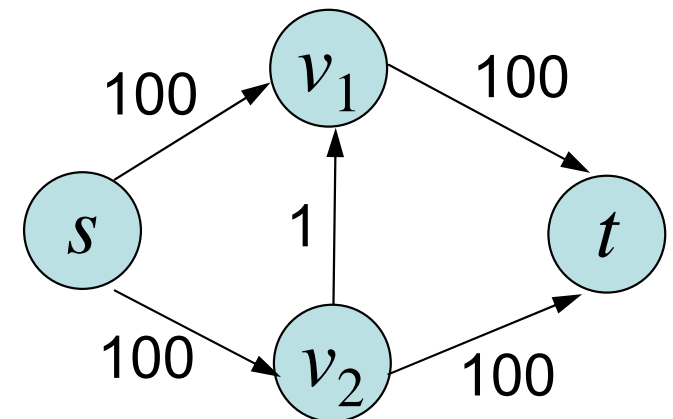
$O(E)$

กรณีแย ๆ



Augmenting Path : ปรับปรุง

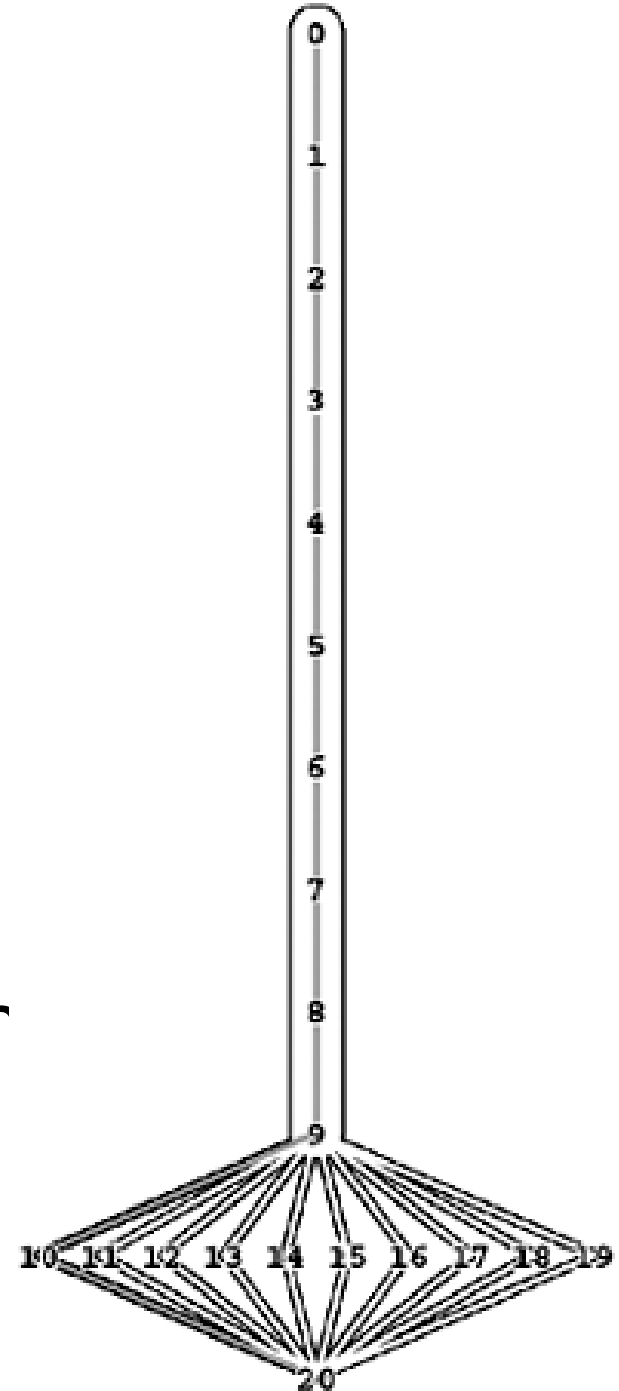
- Capacity scaling $O(E^2 \log U)$
 - Δ -residual network คือ residual network ที่เส้นเชื่อมมีความจุไม่น้อยกว่า Δ
 - เริ่ม $\Delta = 2^{\lg M}$ แล้วค่อย ๆ ลดลงทีละครึ่งจนเป็น 1
- Edmonds และ Karp เส้นอ
 - เลือกวิถีสั้นสุด (จำนวนเส้นน้อยสุด) $O(VE^2)$
 - เลือกเส้นที่มีความจุมากที่สุด $O(V^2 \lg U \lg V)$



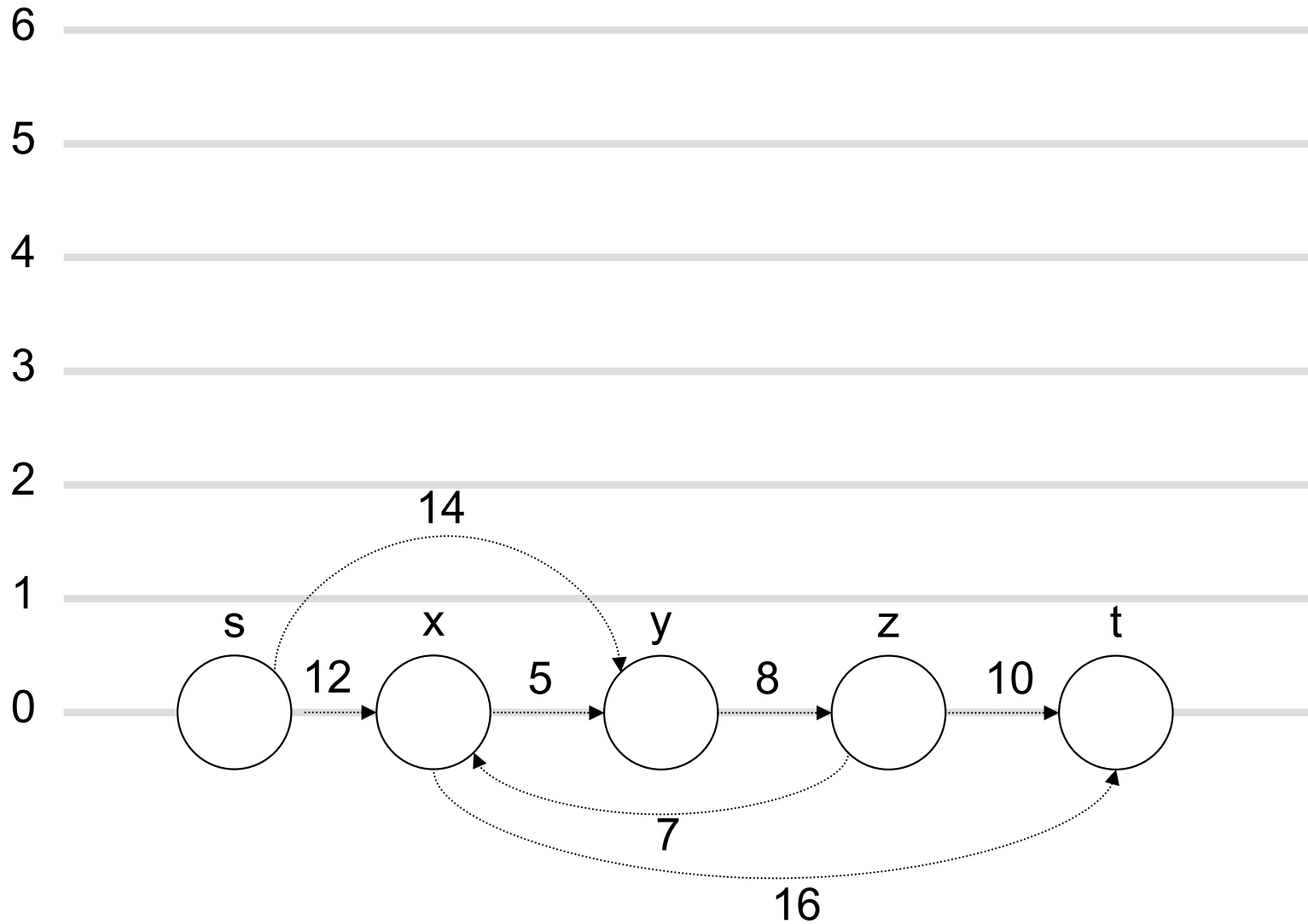
(U คือความจุมากที่สุดของเส้นเชื่อม)

Preflow-Push Method

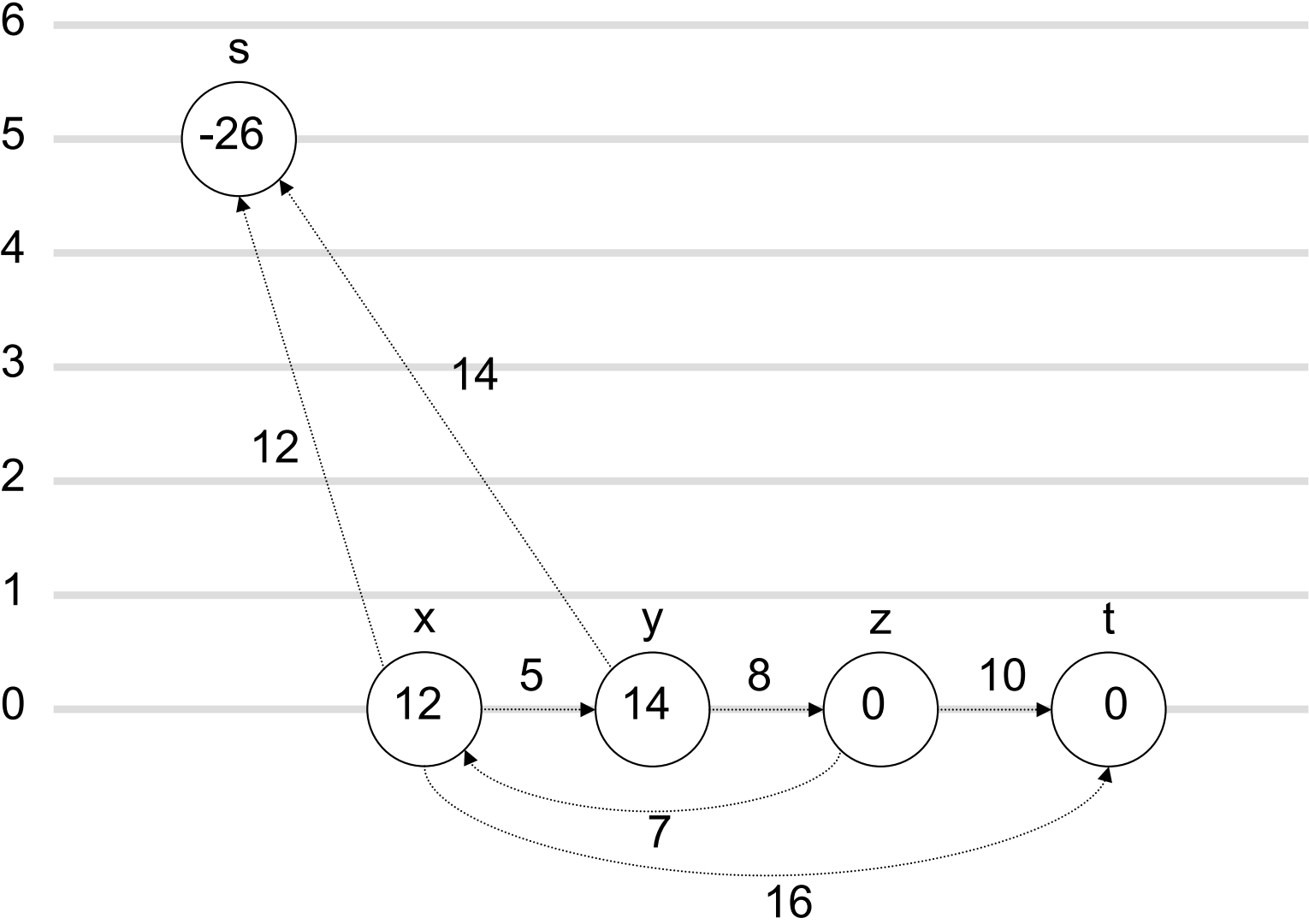
- augmenting path method หาวิถีเพื่อเพิ่มการไหลให้มากขึ้น ๆ โดยรักษากฎการไหลเข้าเท่าไหลออกตามปมต่าง ๆ ไว้ตลอดเวลา (maintain feasibility, aim for optimality)
- preflow push method เพิ่มการไหลให้มากที่สุด ๆ ไปทาง sink แล้วค่อย ๆ ผลักให้ไหลกลับมาทาง source เพื่อให้ถูกกฎการไหลเข้าเท่าไหลออก (start with super optimality, aim for feasibility)



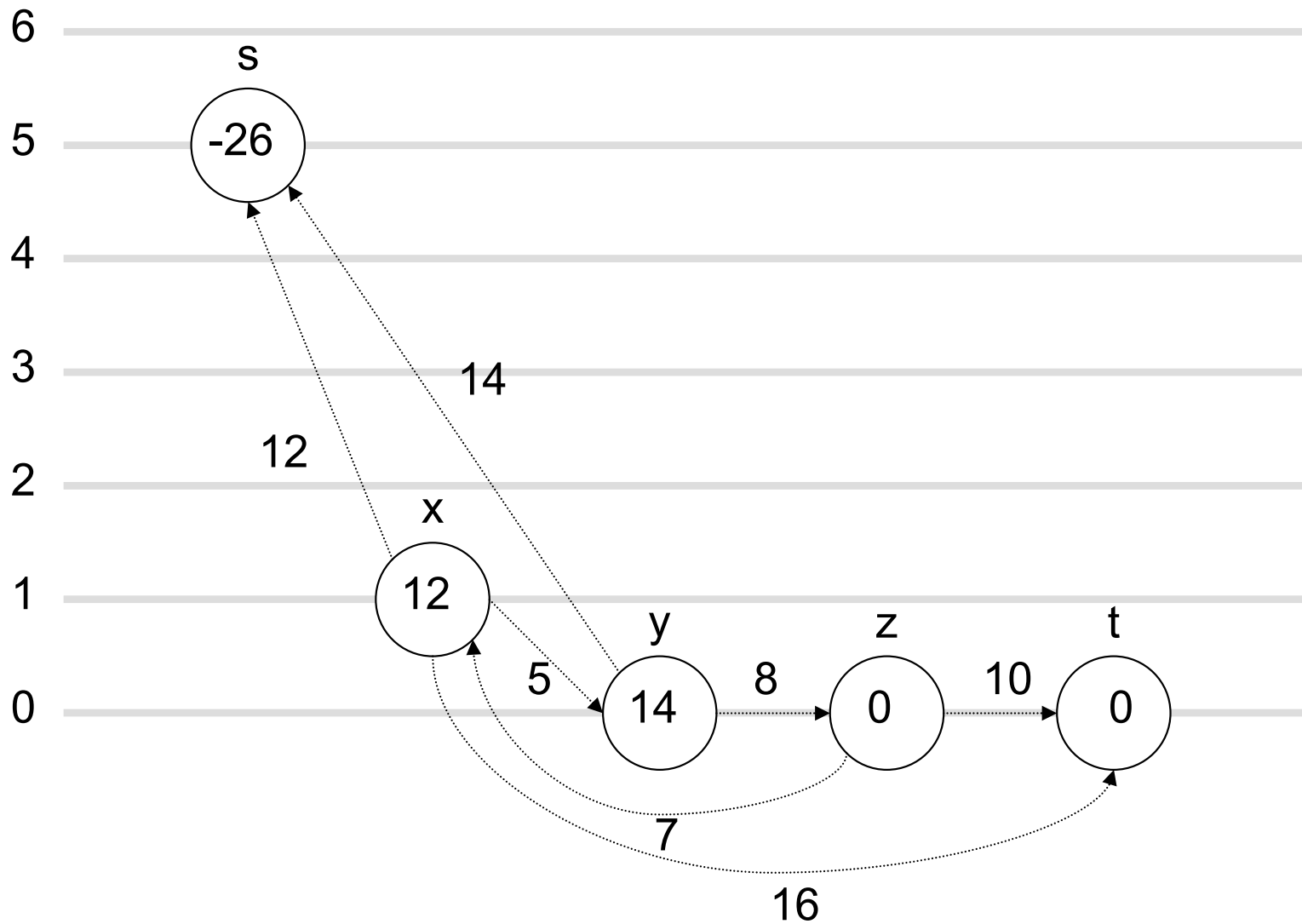
Original Network



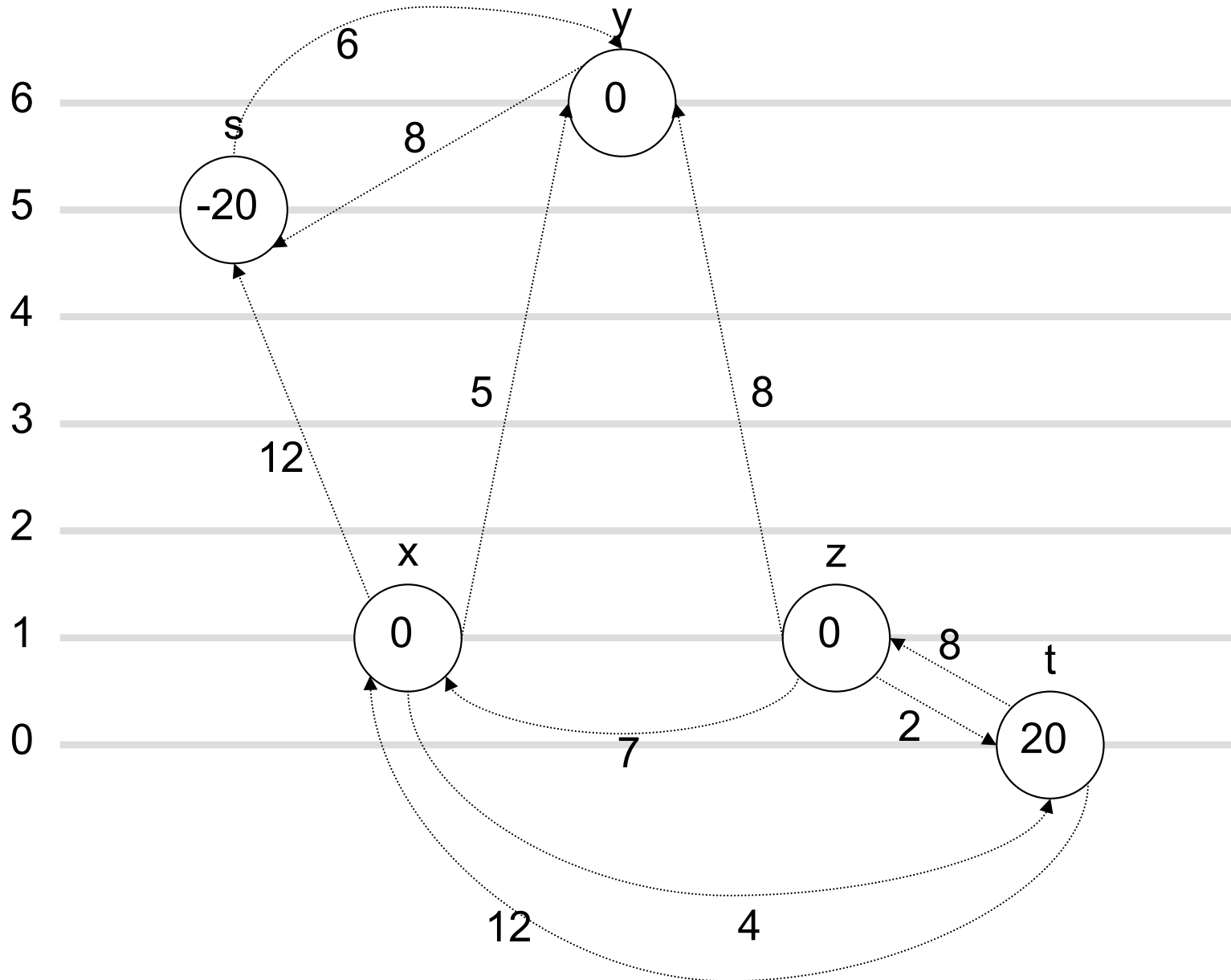
Residual network : step 1 - initialize



Residual network : step 2 - relabel



Residual network : step 13 - push



แนวคิด

- $e(u)$ คือ excess ของปม u
ปริมาณการไหลเข้า u ลบด้วยปริมาณการไหลออก
- active node คือปมที่มี $e(u) > 0$
(ไม่สนใจ s กับ t , $e(s) \leq 0$, $e(t) \geq 0$ แน่ๆ)
- แต่ละปมมีความสูงจำกัด
- ให้ flow ไหลจากที่สูงลงสู่ที่ต่ำ (sink อยู่ต่ำสุด)
- มีเส้น (u, v) , $d(u) \leq d(v) + 1$
- เลือก active node u แล้ว push การไหล $e(u)$
ออกไปให้ปม "เพื่อน" v โดยที่ $d(u) = d(v) + 1$
(ปมใกล้สุดที่เข้าหา sink)

Preflow-Push Method

Algorithm:

initialization

while there exists an active node

select an active node i

Push-Relabel(i)

Initialization:

Set $f = 0$

Compute distance label $d(i)$ for all $i \in \mathbf{V}$

$f(s,j) = c(s,j)$ for all $(s,j) \in \mathbf{E}$

$d(s) = n$, where $n = |\mathbf{V}|$

Push-Relabel(i):

if there is an admissible edge (i,j)

push $\min(e(i), c_{f(i,j)})$ units of flow on (i,j)

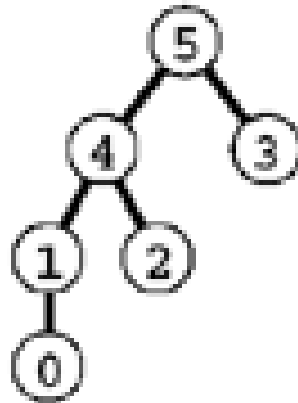
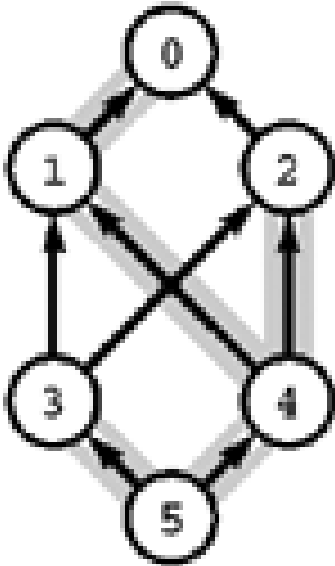
else

$d(i) = 1 + \min_{(i,j) \in E_f} \{ d(j) \mid c_f(i,j) > 0 \}$

Running Time $\mathbf{O(V^2E)}$

ให้ distance label อย่างไร

- ให้ $d(i) = 0$ หมด
- ทำ reverse breadth-first search จาก t
ให้ $d(t) = 0$ และให้ $d(i) = k$ โดยที่ k คือ
หมายเลขระดับของ i ใน breadth-first tree



	0	1	2	3	4	5
d	3	2	2	1	1	0

จะเลือก active nodes อย่างไร

- FIFO preflow-push algorithm

$$O(V^3)$$

- Highest label preflow-push algorithm

$$O(V^2E^{1/2})$$

- Excess scaling algorithm

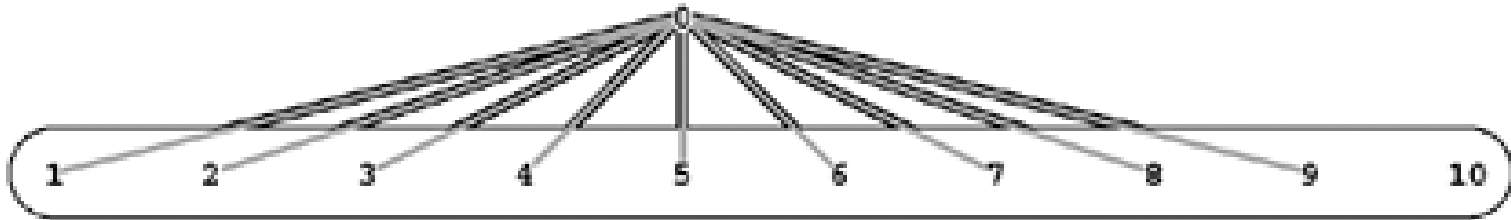
$$O(VE + V^2\log U)$$

FIFO Preflow-Push

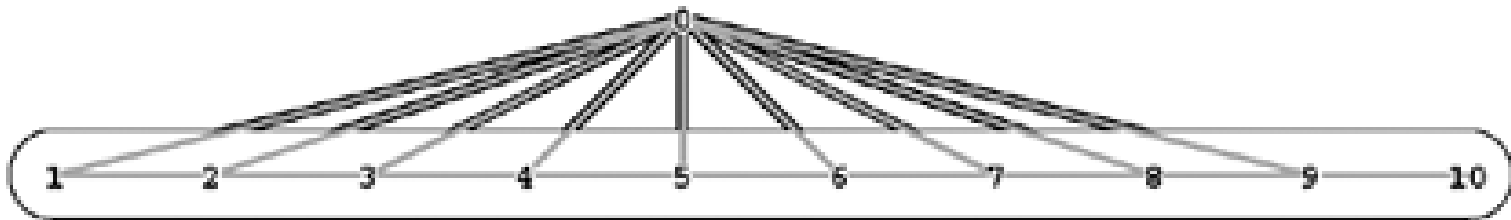
- ใช้ queue เก็บรายการของ active nodes ที่จะถูกนำออกมา push flow
- ปมที่ถูก relabel จะถูกไปต่อท้าย queue
- ถ้า u ถูกเลือกมา push, แล้วยังเป็น active node อยู่ ก็จะมี push ต่อไปจนกว่า $e(u) = 0$ หรือจนกว่า relabel

Highest-Label Preflow-Push

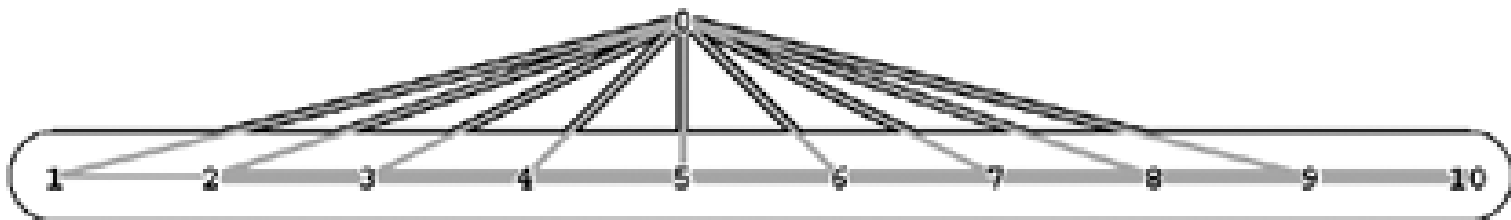
- เลือก active node ที่อยู่สูงที่สุดมา push



9 8 7 6 5 4 3 2 1

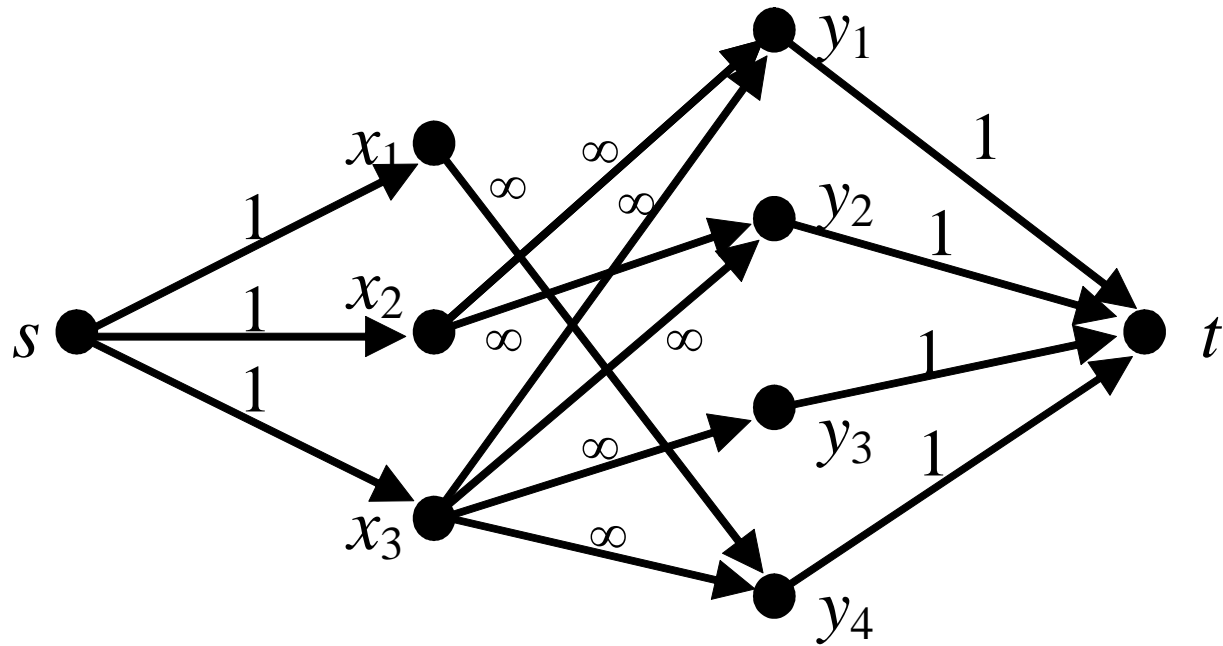
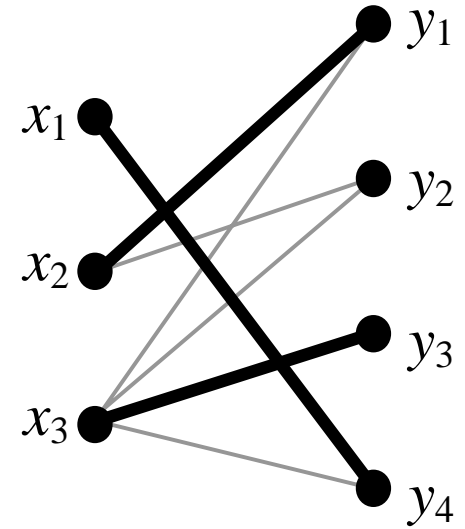
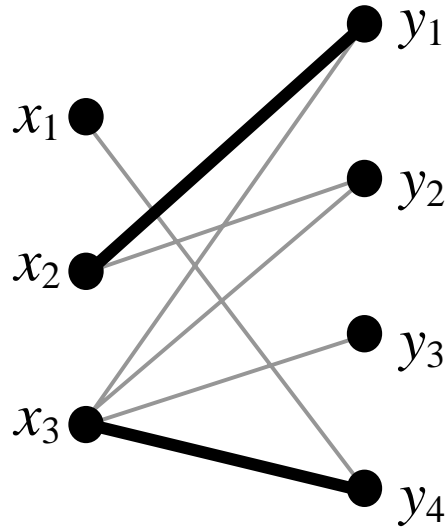
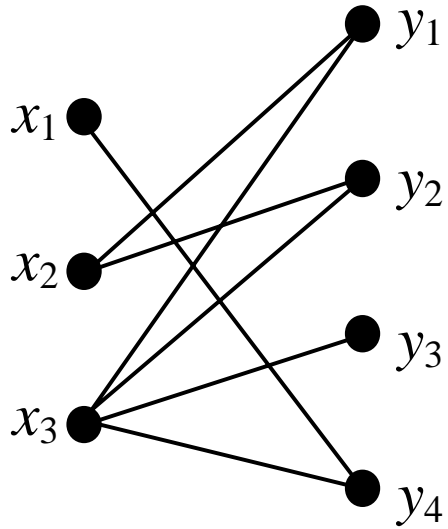


9 8 7 6 5 4 3 2

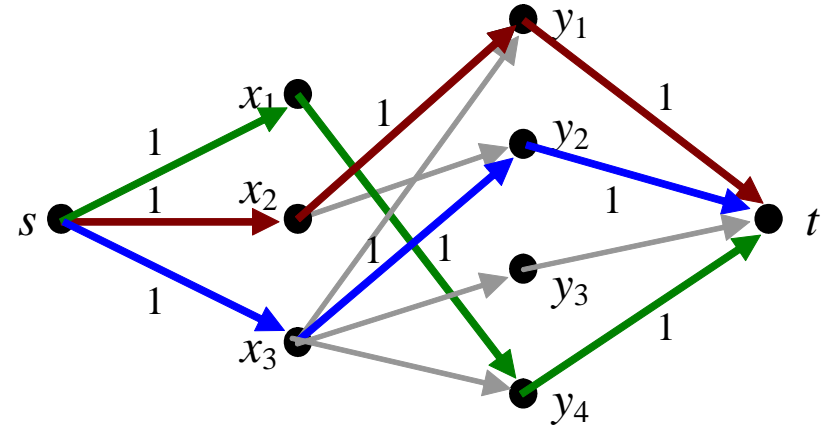
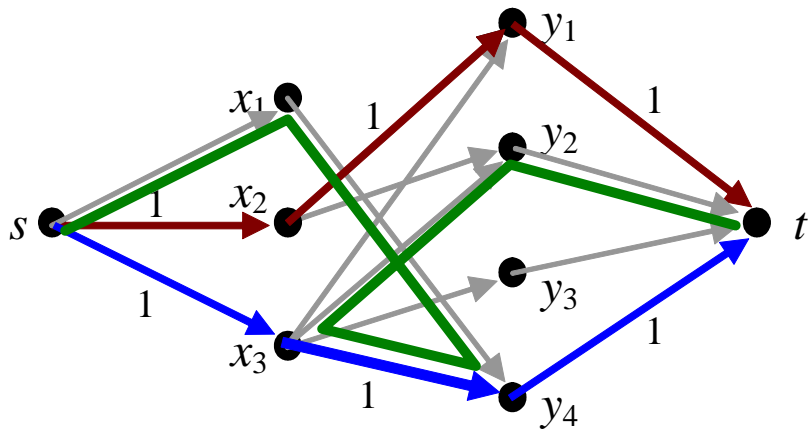
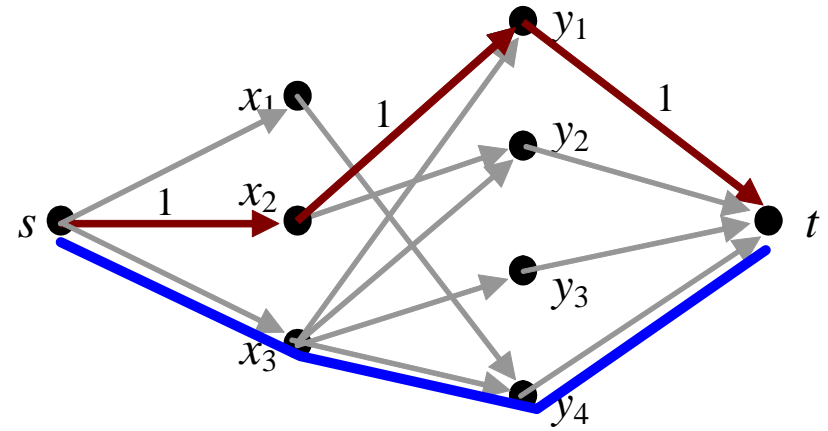
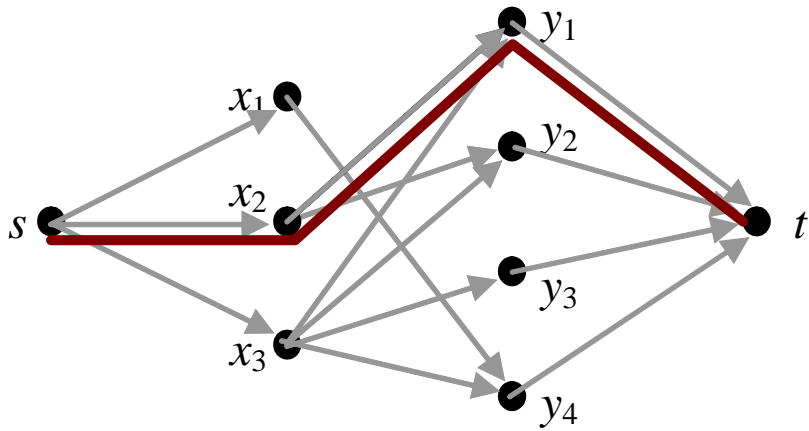


9 8 7 6 5 4 3

การจับคู่

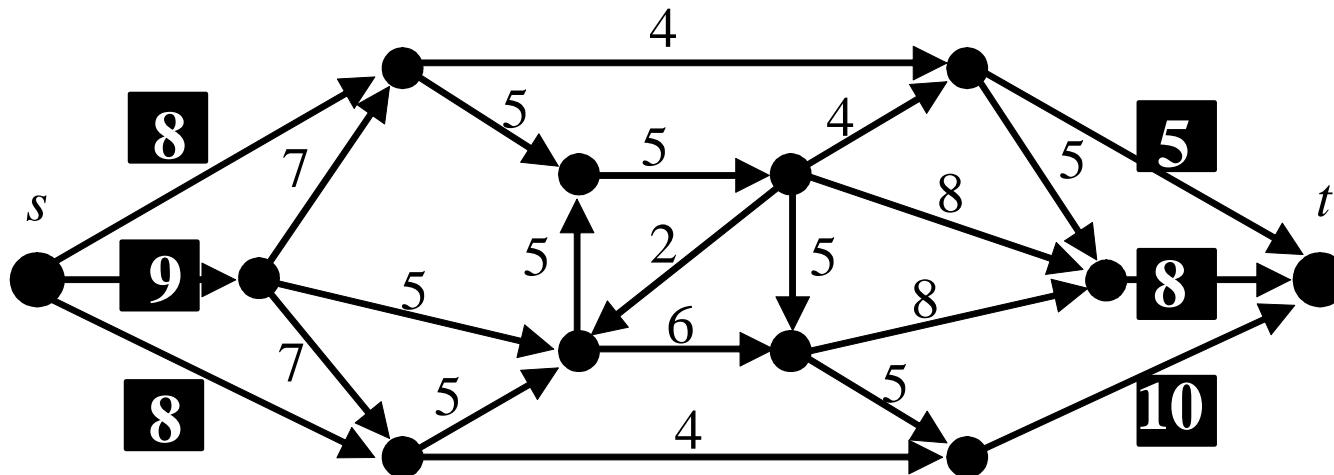
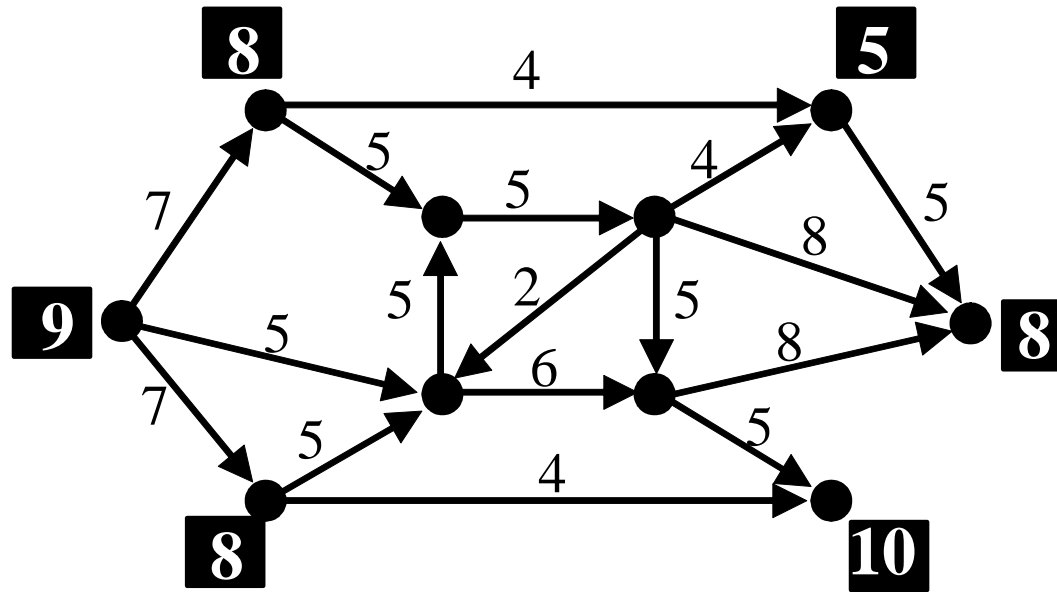


การจับคู่

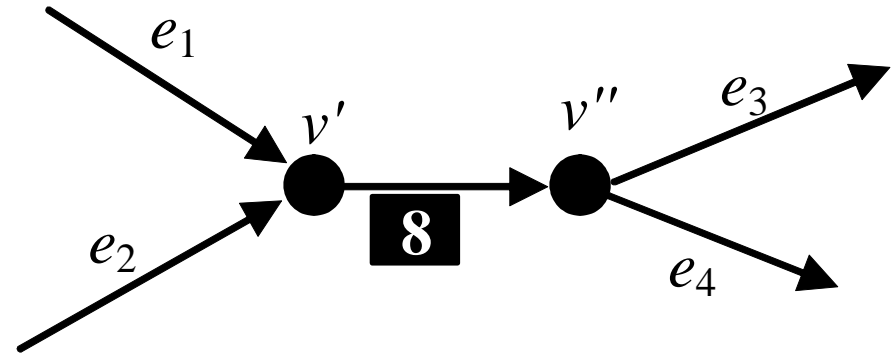
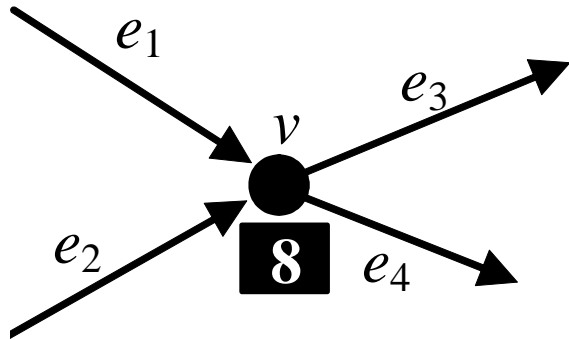


	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	1	1	0	0
x_3	1	1	1	1

ข่ายงานที่มีหลาย sources และ sinks



ข่ายงานที่ปมมีความจุจำกัด



การหาจำนวนวิถีในกราฟที่ไม่มีเส้นร่วมกัน

