Computer Science and Information Engineering
National Chi Nan University

Chapter 9
Connectivity

§ 9.6 Dinic Algorithm

§ 9.6 Dinic Algorithm

- **Def:**

① **In $N = (V, E)$, e is useful from u to v w.r.t. (with respect to) f if**

(i) $e = uv \wedge f(e) < \text{Cap}(e)$ or

(ii) $e = vu \wedge f(e) > 0$.

§ 9.6 Dinic Algorithm

- **Def:**

② **layered network** $N^* = (V^*, E^*)$ of $N = (V, E)$ w.r.t f

where
$$\begin{cases} V^* = V_0 \cup V_1 \cup \ldots \cup V_\alpha \\ E^* = E_1 \cup E_2 \cup \ldots \cup E_\alpha \\ V_0 = \{s\} \\ V_i = \{y \notin V_0 \cup \ldots \cup V_{i-1}: \exists\, x \in V_{i-1} \text{ s.t. } xy \text{ is useful} \\ \quad (\underline{xy \in E \text{ with } f(x, y) < \text{Cap}(x, y)} \text{ or } \underline{yx \in E \text{ with } f(y, x) > 0})\} \\ E_i = \{xy: x \in V_{i-1} \wedge y \in V_i \wedge xy \text{ is useful} \\ \quad (\underline{xy \in E \text{ with } f(x, y) < \text{Cap}(x, y) \text{ or } yx \in E \text{ with } f(y, x) > 0})\} \\ \text{Cap}^*(x, y) = \begin{cases} \text{Cap}(x, y) - f(x, y), & \text{if } xy \in E \\ f(y, x), & \text{if } yx \in E \end{cases} \end{cases}$$

(i) (ii)

If $t \in V_\alpha$ for some α, then $V_\alpha \leftarrow \{t\}$
else f is optimal (not exist N^*)

§ 9.6 Dinic Algorithm

- 造法:

$V_0 = \{s\}$; $i \leftarrow 0$;

(♥) $V_{i+1} = \{y \notin V_0 \cup \ldots \cup V_i$: $\exists\, x \in V_i$ s.t.

($xy \in E$ with $f(x, y) < \mathrm{Cap}(x, y)$ or $yx \in E$ with $f(y, x) > 0$)};

if $V_{i+1} = \varnothing$ then STOP; /* f is optimal*/

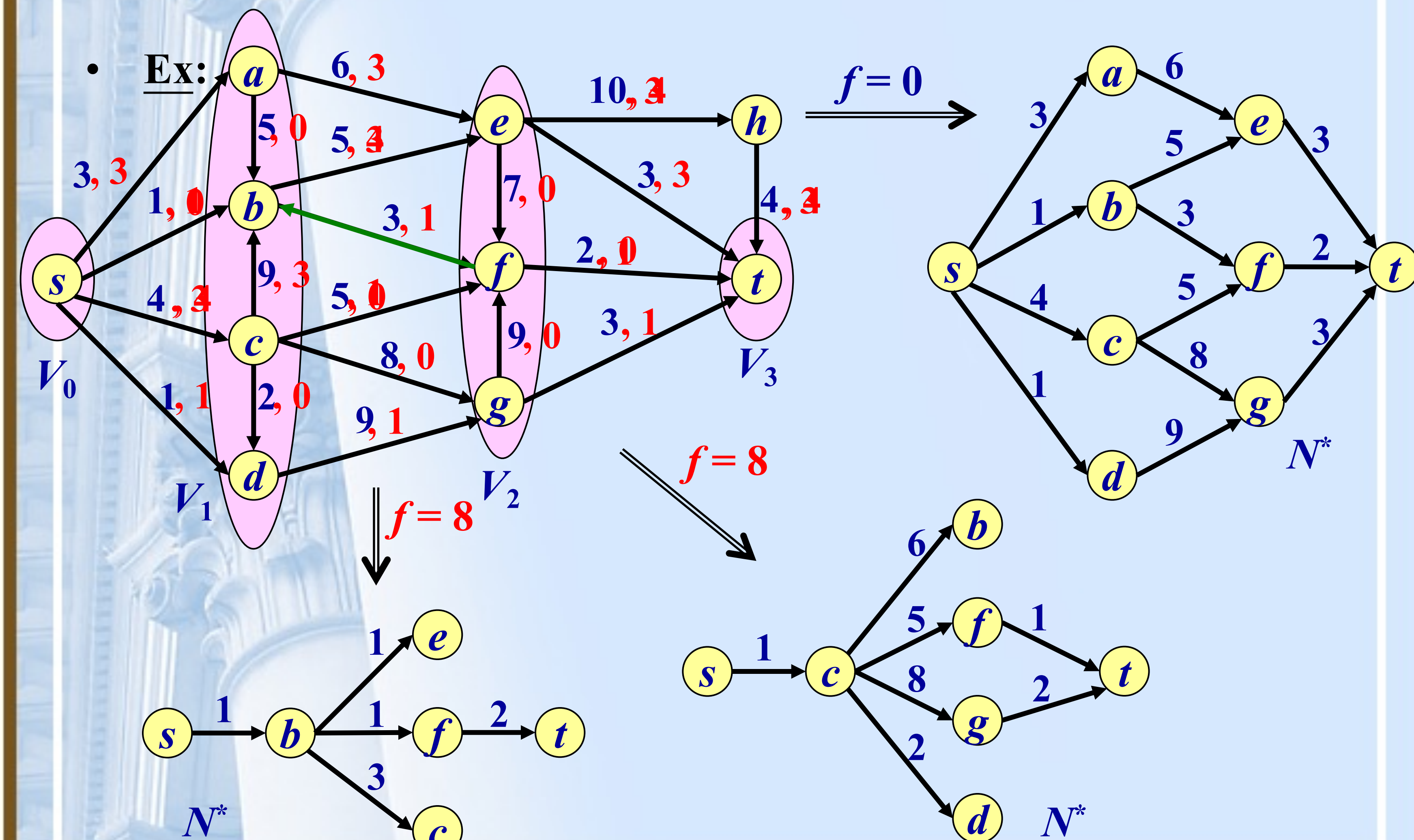
if $t \in V_{i+1}$ then $V_{i+1} = \{t\}$;

else {$i = i + 1$; go to (♥)};

xy is useful

§ 9.6 Dinic Algorithm

- Ex:



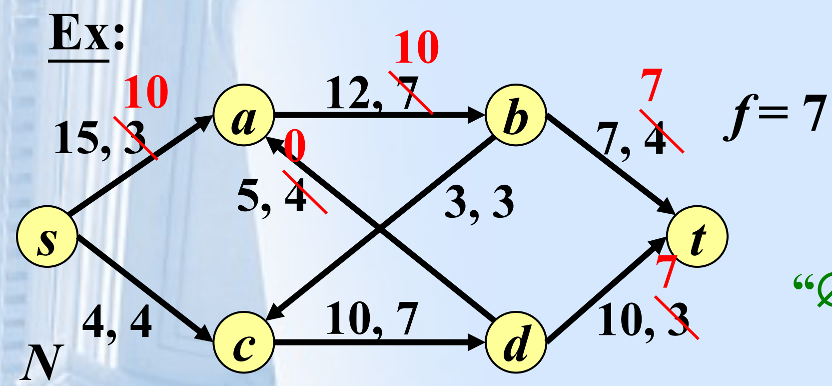
§ 9.6 Dinic Algorithm

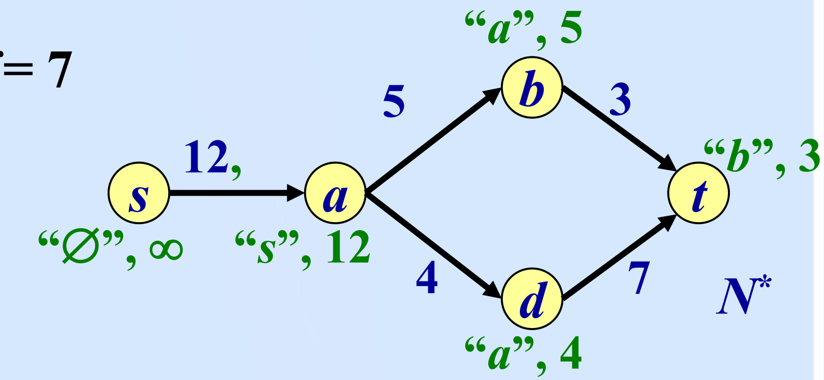
- **<u>Thm</u>: If f^* is a flow of N^*,**
  then f' is a flow of N with value(f') = value(f) + value(f^*),
  when $f'(e) = \begin{cases} f(e) + f^*(e), \text{ if } (\underline{e \in E \cap E^*}) \quad \& \text{ (i) holds;} \\ f(e) - f^*(e), \text{ if } (\underline{e \in E, e^{-1} \in E^*}) \ \& \text{ (ii) holds;} \\ f(e), \quad \text{o.w.} \end{cases}$
  (Notation: $f' = f \oplus f^*$)

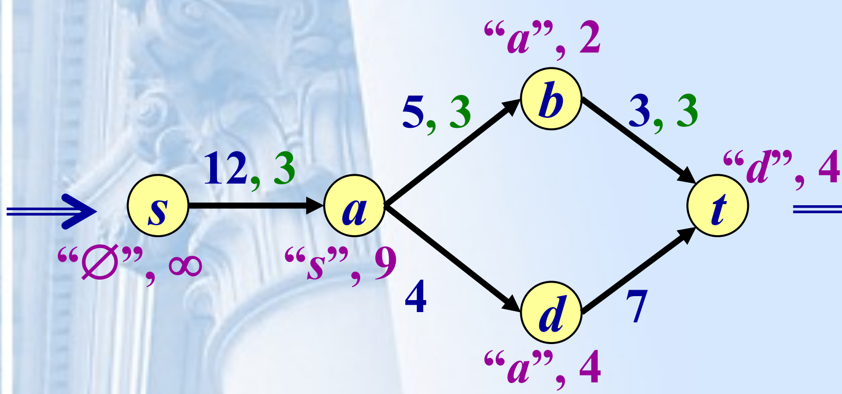
- **<u>Cor</u>: value of max. flow of N**
  = max{value(f) + value of max. flow of N^*}.

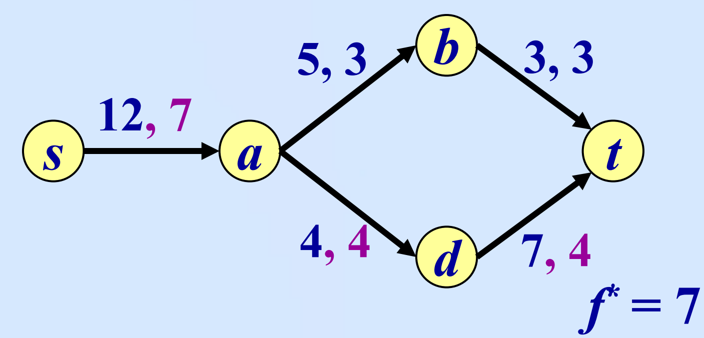
§ 9.6 Dinic Algorithm

- **Ex:**









⇒ value of max. flow of N = 7 + 7 =14.

§ 9.6 Dinic Algorithm

- 希望找 N 的 maximum flow → 找 N^* 的 maximum flow

- Def: f^* is a maximal flow in N^* if

① f^* is a flow in N^*.

② $\forall$ path in N^*: $\underset{\underset{s}{\parallel}}{x_0} \xrightarrow{e_1} \underset{\underset{V_1}{\in}}{x_1} \xrightarrow{e_2} \underset{\underset{V_2}{\in}}{x_2} \rightarrow \dots \rightarrow \underset{\underset{V_{k-1}}{\in}}{x_{k-1}} \xrightarrow{e_k} \underset{\underset{t}{\parallel}}{x_k}$

$\exists\, i \in \{1, 2, \dots, k\}$ s.t. $f^*(e_i) = \mathrm{Cap}^*(e_i)$.

§ 9.6 Dinic Algorithm

- **Dinic Algorithm:**

(1) $f \leftarrow 0$;

(2) Construct N^* **(if not exist** N^*, **STOP)**

(3) Find a maximal flow f^* **of** N^*;

(4) $f \leftarrow f \oplus f^*$; $|V||E|$

(5) Go to (2);

§ 9.6 Dinic Algorithm

- **Lemma 1: Dinic's Algorithm runs at most $|V|$ iterations.**

Proof. (1/4)

Suppose $N^*(k)$ is the kth layered network whose length is r_k, say $V^*(k) = V^*_{k,0} \cup V^*_{k,1} \cup \dots \cup V^*_{k,r_k}$.

Claim: $r_1 < r_2 < \dots < r_k < r_{k+1} < \dots < |V| - 1$

Proof of Claim.

Let P: $s = v_0 v_1 v_2 \dots v_{r_{k+1}} = t$ be an s-t dipath in $N^*(k+1)$

$\exists$ max. index a s.t. $v_i \in V^*_{k,i}$ $\forall$ $i = 0, 1, \dots, a$

if $a = r_{k+1}$, then P is also an s-t dipath in $N^*(k)$

$\Rightarrow \exists\ e = v_{j-1}v_j$ s.t. $f^*_k(e) = \text{Cap}^*_k(e)$

$$\Rightarrow f'(e) = \begin{cases} f(e) + f^*(e) = \text{Cap}(e), & \text{if (i) holds} \\ f(e) - f^*(e) = 0, & \text{if (ii) holds} \end{cases}$$

§ 9.6 Dinic Algorithm

- **<u>Lemma 1</u>: Dinic's Algorithm runs at most $|V|$ iterations.**

  Proof. (2/4)

  <u>Claim</u>: $r_1 < r_2 < \ldots < r_k < r_{k+1} < \ldots < |V| - 1$

  Proof of Claim.

  if $a = r_{k+1}$, then P is also an s-t dipath in $N^*(k)$

  $\Rightarrow \exists\, e = v_{j-1}v_j$ s.t. $f^*_k(e) = \text{Cap}^*_k(e)$

  $$\Rightarrow f'(e) = \begin{cases} f(e) + f^*(e) = \text{Cap}(e), & \text{if (i) holds} \\ f(e) - f^*(e) = 0, & \text{if (ii) holds} \end{cases}$$

  $\Rightarrow e$ is not useful when we construct $N^*(k + 1)$

  $\Rightarrow e \notin E^*(k + 1)$ $\rightarrow\leftarrow$

  $\therefore\ a < r_{k+1}$.

§ 9.6 Dinic Algorithm

- **Lemma 1: Dinic's Algorithm runs at most $|V|$ iterations.**

  Proof. (3/4)

  Claim: $r_1 < r_2 < \ldots < r_k < r_{k+1} < \ldots < |V| - 1$

  Proof of Claim.

  $P: v_0 v_1 \ldots v_a \xrightarrow{e_{a+1}} v_{a+1} \rightarrow \ldots \rightarrow v_{r_{k+1}} = t \ (v_{a+1} \notin V_{k,a+1})$

  $e_{a+1} = v_a v_{a+1}$ is useful when we construct $N^*(k+1)$

  $\Rightarrow e_{a+1}$ is useful when we construct $N^*(k)$

  (o.w. e_{a+1} isn't useful when we construct $N^*(k)$

  $\Rightarrow$ we didn't change $f(e_{a+1})$

  $\Rightarrow$ it is still not useful when we construct $N^*(k+1)$

  $\Rightarrow e_{a+1} \notin E^*(k+1) \rightarrow\leftarrow$)

§ 9.6 Dinic Algorithm

- **Lemma 1: Dinic's Algorithm runs at most $|V|$ iterations.**

  Proof. (4/4)

  Claim: $r_1 < r_2 < \ldots < r_k < r_{k+1} < \ldots < |V| - 1$

  Proof of Claim.

  $P: v_0 v_1 \ldots v_a \xrightarrow{e_{a+1}} v_{a+1} \rightarrow \ldots \rightarrow v_{r_{k+1}} = t \ (v_{a+1} \notin V_{k,a+1})$

  By the construction of $N^*(k)$: $r_k \leq a + 1$ — ①

  But $\because v_{a+1} \notin V_{k,a+1}$

  $\Rightarrow v_{a+1} \neq v_{r_{k+1}} = t$, i.e. $a + 1 \neq r_{k+1}$

  $\Rightarrow a + 1 < r_{k+1}$ — ②

  by ① ②, $r_k \leq a + 1 < r_{k+1}$

  $\Rightarrow r_k < r_{k+1}$.

§ 9.6 Dinic Algorithm

- **Question: How to find a maximal flow f^* of N^*.**

  Sol.

  一面做 DFS, 一面找 s-t aug. path 來 update f^*.

  標出 "blocked edge" ($f^*(e)$ = Cap*(e))

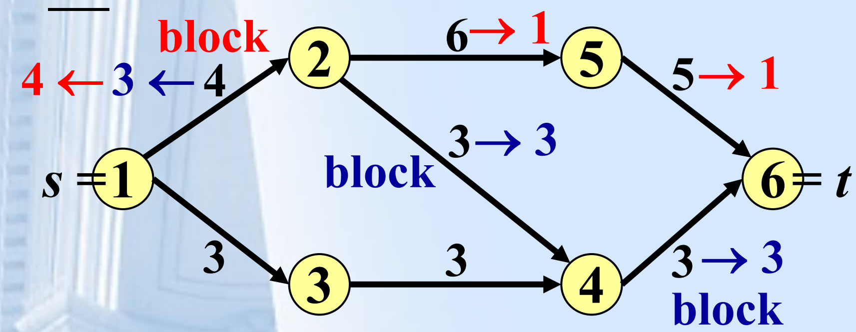
  在 DFS 過程中, 不用 blocked edges.

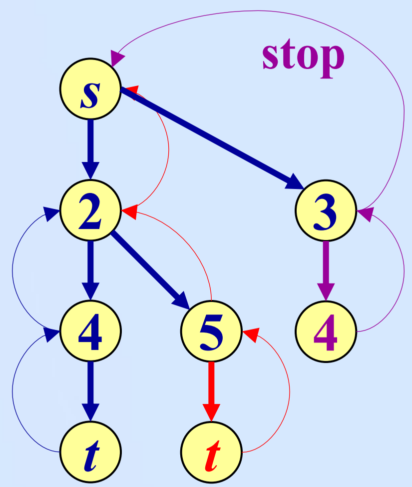
  至多做 $O(|E|)$ 次, 每次至多 $O(|V|)$

  $\Rightarrow O(|V||E|)$.

§ 9.6 Dinic Algorithm

- **Ex:**





- **Note: Dinic's Algorithm need $O(|V|^2|E|)$ time.**

§ 9.6 Dinic Algorithm

- **Compare:**

Ford & Fulkerson	**可能不停或很慢**
Edmons & Karp	$O(\lvert V\rvert^3\lvert E\rvert)$
Dinic	$O(\lvert V\rvert^2\lvert E\rvert)$
MPM	$O(\lvert V\rvert^3)$

Computer Science and Information Engineering
National Chi Nan University

Chapter 9
Connectivity

§ 9.7 MPM Algorithm

§ 9.7 MPM Algorithm

- **Def:**

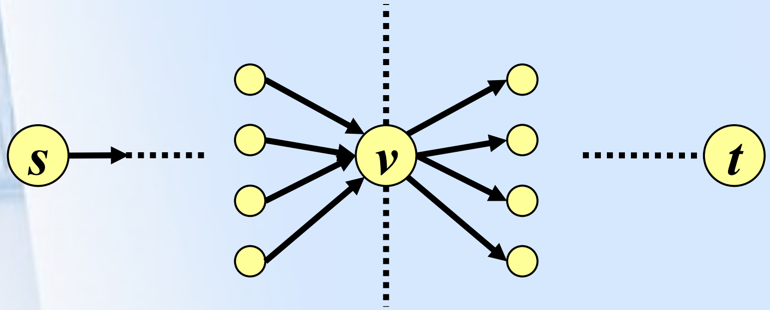
  IP(v) $= \sum_{xv \in E(G^*)} \text{Cap}^*(xv)$ (initial)

  $= \sum_{xv \in E(G^*) \text{ and } xv \text{ unsaturated}} (\text{Cap}^*(xv) - f^*(xv))$ (o.w.)

  OP(v) $= \sum_{vy \in E(G^*)} \text{Cap}^*(vy)$ (initial)

  $= \sum_{vy \in E(G^*) \text{ and } vy \text{ unsaturated}} (\text{Cap}^*(vy) - f^*(vy))$ (o.w.)

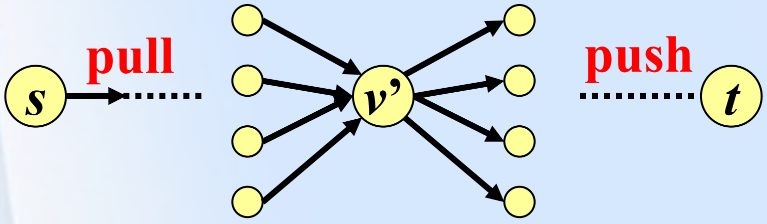
  $P(v) = \min\{\text{IP}(v), \text{OP}(v)\}$: called **Potential** of v.



§ 9.7 MPM Algorithm

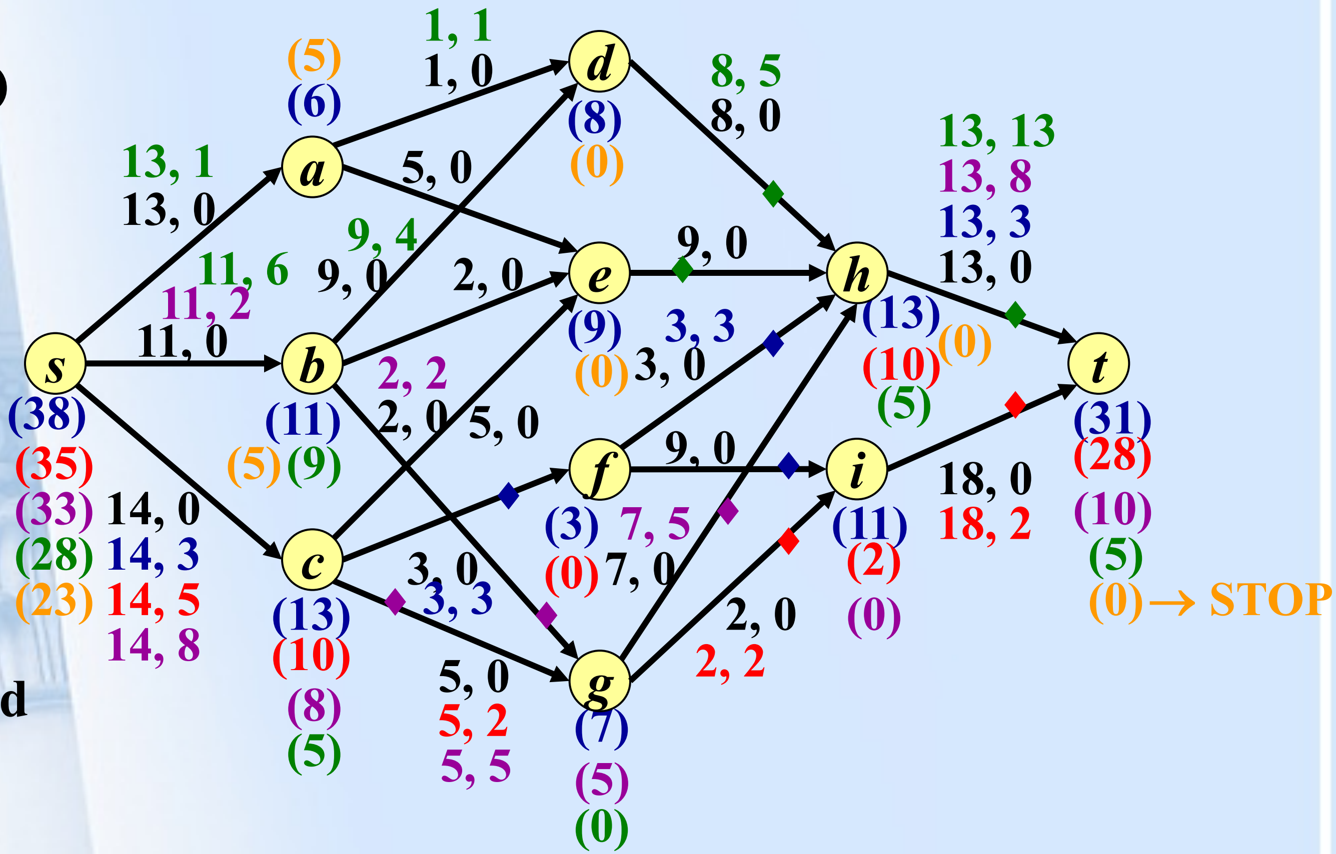
- **MPM Algorithm: (改進 Dinic Algorithm (3))**

> **(3.1) Computer $P(v^*)$ $\forall$ $v^* \in V^*$;**
> **(3.2) Find v' s.t. $P(v') = \min_{v^* \in V^*} P(v^*)$;**
> **(3.3) Pull and Push flow;**
> **(3.4) Recomputer $P(v)$ $\forall$ $v \in V$;**
> **(3.5) if $P(s)$ or $P(t) = 0$ STOP;**
> **else goto (3.2);**



§ 9.7 MPM Algorithm

- <u>Ex</u>:
  (Cap*, *f**)
  (*P*(*v*))
  (*P*(*v*))
  (*P*(*v*))
  (*P*(*v*))
  (*P*(*v*))



♦ : saturated

§ 9.7 MPM Algorithm

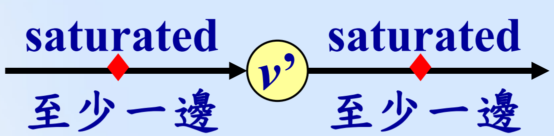
- Note: Time of MPM = $O(|V|^3)$

① N^*: $O(|V|)$ iteration

② Pull 和 Push 動作最多執行 $|V|$ 次,
每次 Pull 和 Push 至多改變 $|E|$ 邊 (執行 $|E|$ 個指令),
但是 time 不是 $O(|V| \cdot |E|)$, 而是 $O(|E| + |V|^2) = O(|V|^2)$

Why? Exercise 9 (12/19)

Hint:
∵任何一個邊只會被 saturated 一次且每次至少 saturated 2個邊



⇒ $O(|V|^3)$