離散最適化基礎論

第 7 回

最大流問題：Push-Relabel 法 (概要)

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2023 年 11 月 21 日

最終更新：2023 年 11 月 28 日　10:34

スケジュール (前半)

1. 最大流と最小費用流：定義 (10/3)
2. 最大流問題：増加道法 (10/10)

＊ 休み (10/17)

3. 線形計画法の復習 (10/24)
4. 最大流問題：線形計画問題として (10/31)
5. 最大流問題：Edmonds-Karp のアルゴリズム (11/7)
6. 最大流問題：容量スケーリング法 (11/14)
7. 最大流問題：Push-Relabel 法 (概要) (11/21)
8. 最大流問題：Push-Relabel 法 (計算量評価) (11/28)

スケジュール (後半)

*	休み	(12/5)
9.	最小費用流問題：線形計画問題として	(12/12)
10.	最小費用流問題：負閉路消去法	(12/19)
11.	最小費用流問題：正カット消去法	(12/26)
*	休み	(1/2)
12.	最小費用流問題：逐次最短路法	(1/9)
13.	最小費用流問題：容量スケーリング法	(1/16)
14.	最小費用流問題：費用スケーリング法	(1/23)
*	休み	(1/30)

ここまでの話 と 今回の話

最大流問題

増加道法 (’56)

$\downarrow$ 工夫

Edmonds-Karp (’72)

容量スケーリング (’85)

ここまでの話 と 今回の話

最大流問題

増加道法 ('56)

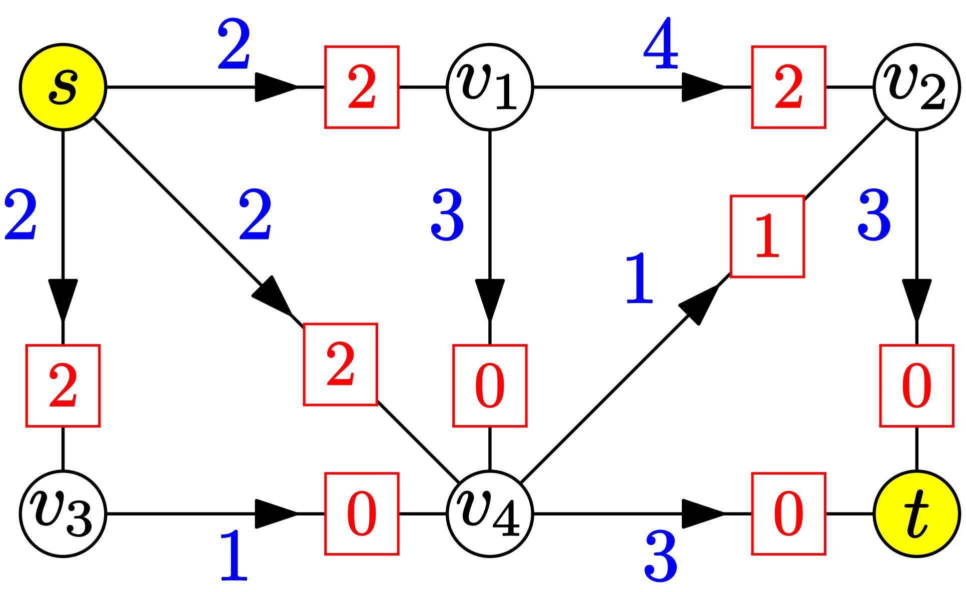
工夫

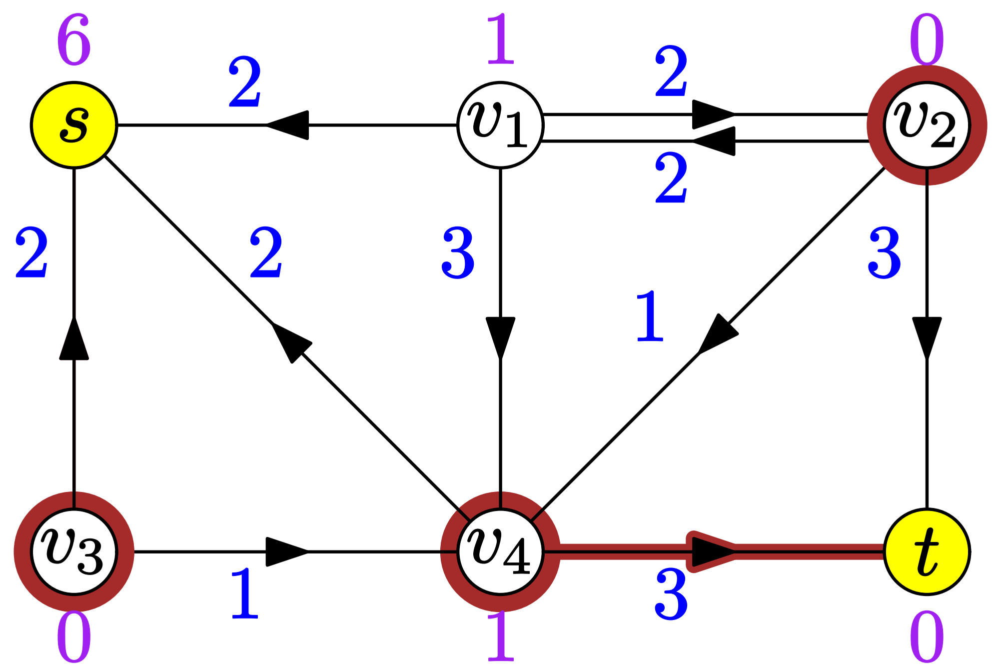
Edmonds-Karp ('72)

容量スケーリング ('85)

Push-Relabel 法 ('88)

1. **増加道法と線形計画法**
2. 前流と距離ラベル
3. Push-Relabel 法：概要
4. Push-Relabel 法：性質





最大流問題と線形計画問題

(P) 最大流問題

$$
\begin{aligned}
\text{max.} \quad & \sum_{a\in\delta^+(s)} f_a - \sum_{a\in\delta^-(s)} f_a \\
\text{s.t.} \quad & f_a \le u_a \quad \forall\, a \in A, \\
& \sum_{a\in\delta^-(v)} f_a - \sum_{a\in\delta^+(v)} f_a = 0 \\
& \qquad\qquad \forall\, v \in V - \{s,t\}, \\
& f_a \ge 0 \quad \forall\, a \in A
\end{aligned}
$$

(D) 双対問題

$$
\begin{aligned}
\text{min.} \quad & \sum_{a\in A} u_a z_a \\
\text{s.t.} \quad & z_{uv} \ge y_u - y_v \quad \forall\, uv \in A, \\
& y_s = 1, y_t = 0, \\
& z_a \ge 0 \quad \forall\, a \in A
\end{aligned}
$$

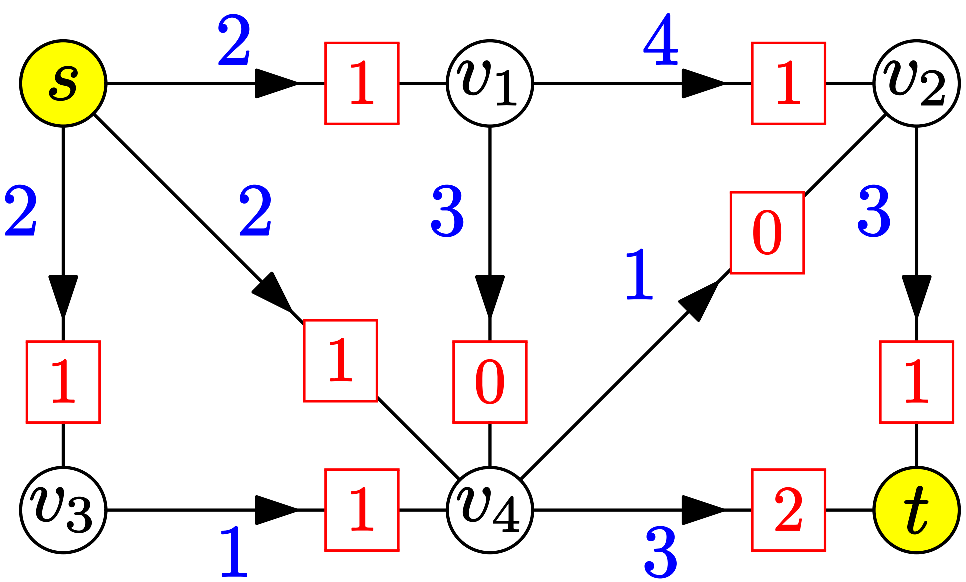
相補性定理：f が (P) の最適解，y, z が (D) の最適解 $\Leftrightarrow$

- f が (P) の許容解，y, z が (D) の許容解
- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$

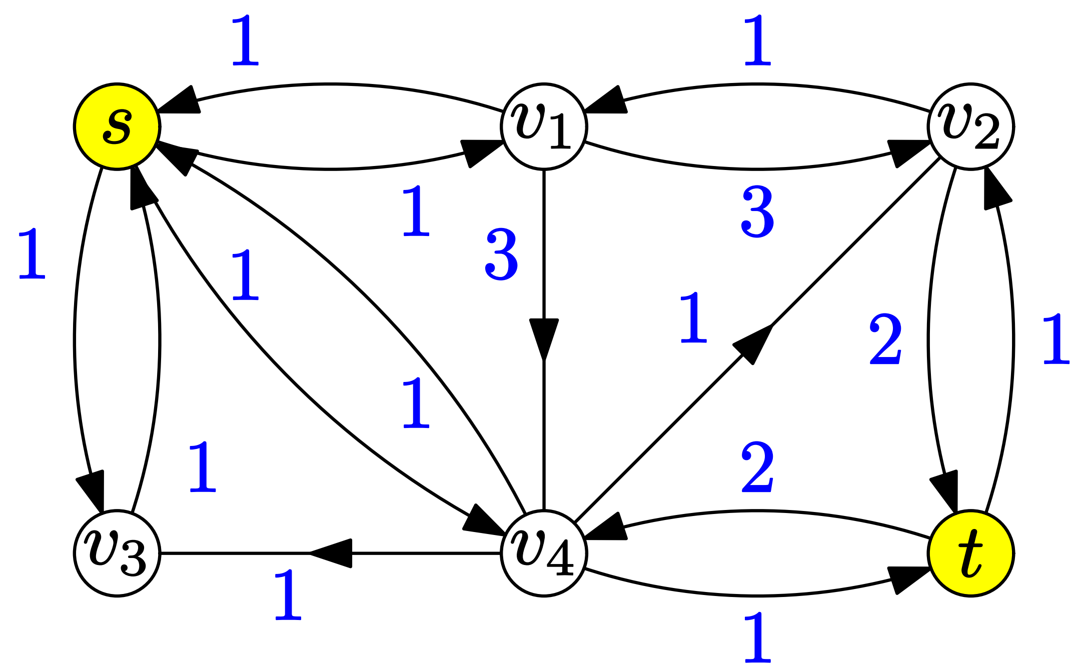
相補性条件と補助ネットワーク：例

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$



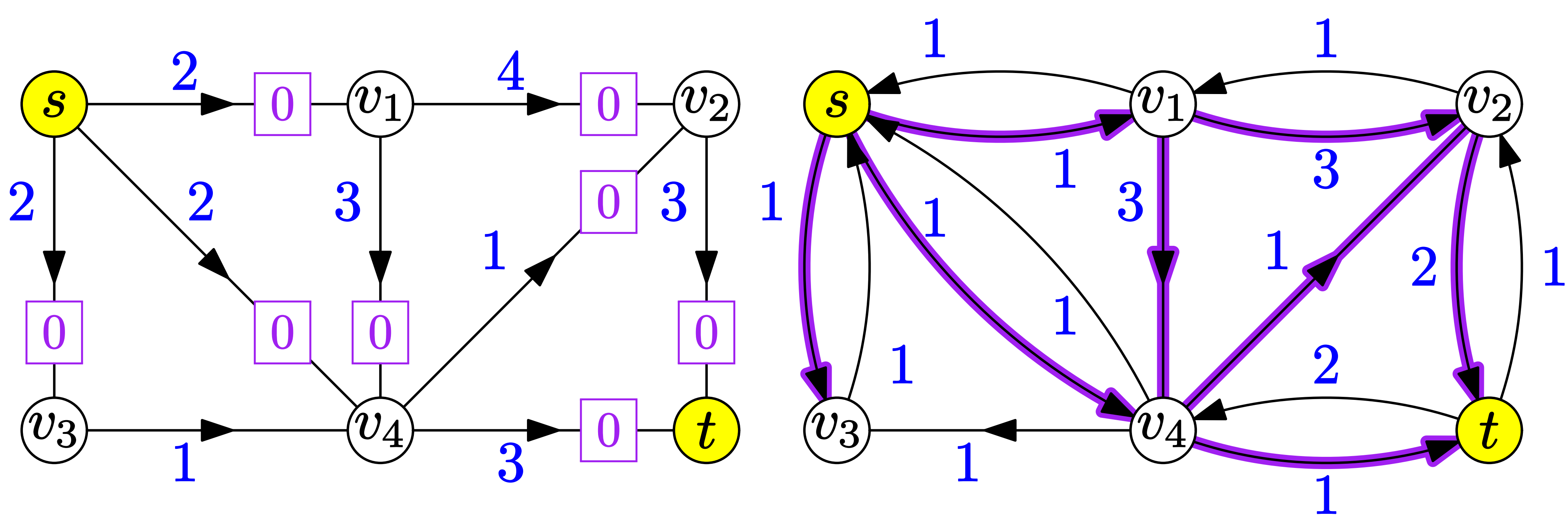
$(G, u), f$



(G_f, u_f)

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$

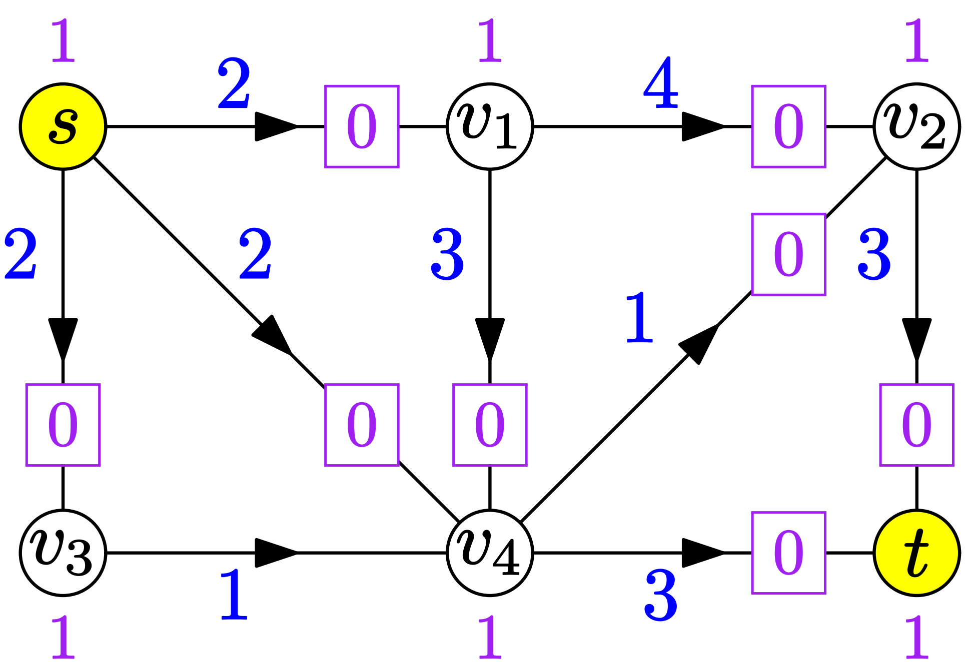


$(G, u), f$ $\qquad$ (G_f, u_f)

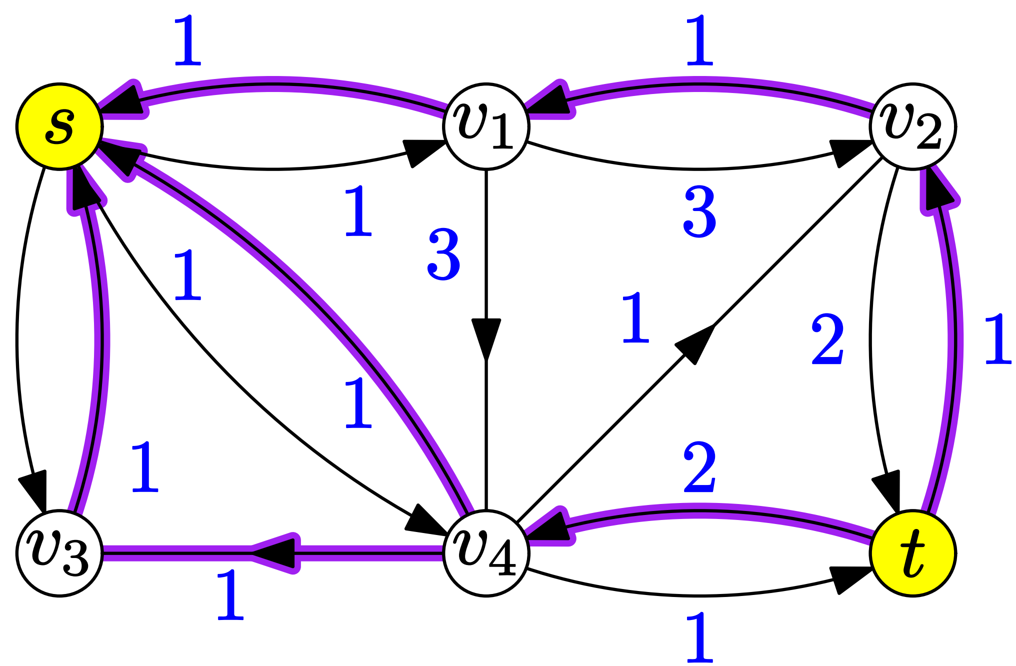
相補性条件と補助ネットワーク：例

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$



$(G, u), f$

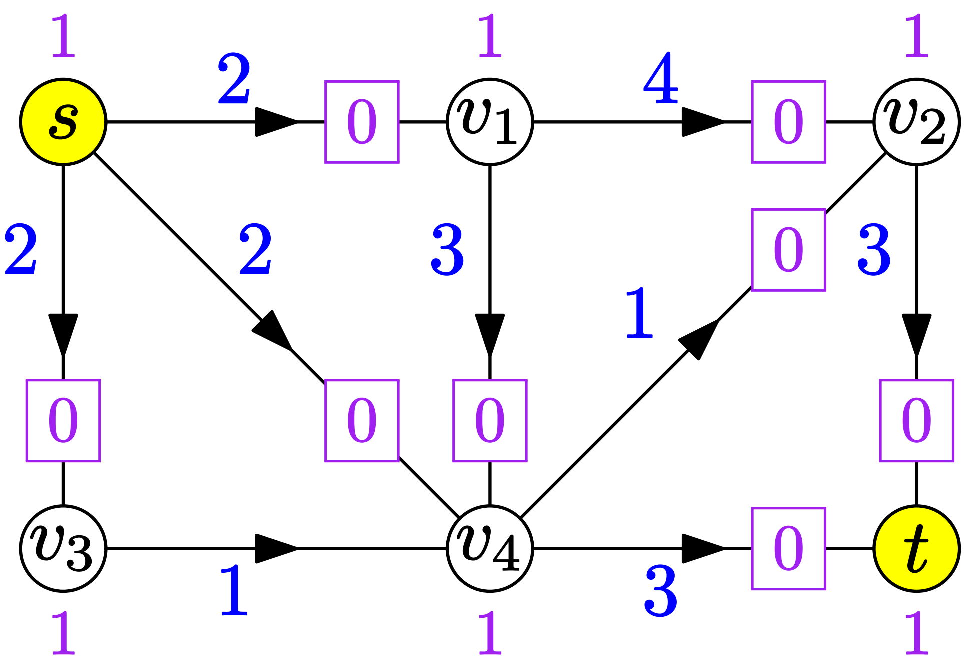


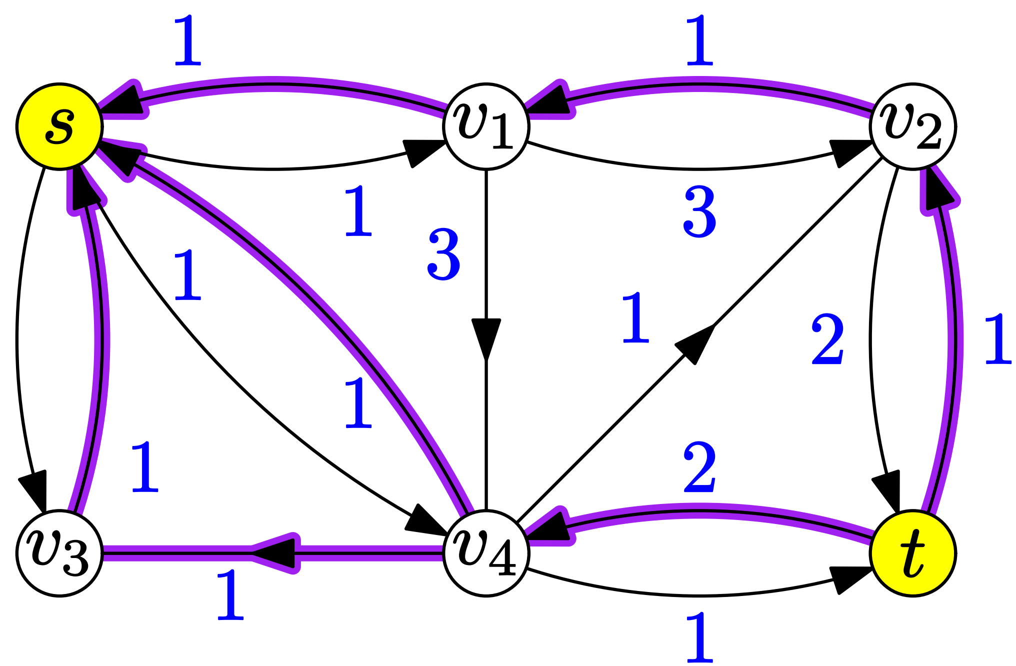
(G_f, u_f)

相補性条件と補助ネットワーク：例

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a) z_a = 0 \quad \forall\, a \in A$





双対許容解ではない

$(G, u), f$

(G_f, u_f)

相補性条件と補助ネットワーク

設定：ネットワーク (G, u), s-t 流 f,
相補性条件を満たす $y \in \mathbb{R}^V, z \in \mathbb{R}^A$

性質：相補性条件と補助ネットワーク

補助ネットワーク (G_f, u_f) において，増加道が存在 ⇒ y, z は双対問題 (D) の許容解ではない

(D) の制約

$$
\begin{aligned}
z_{uv} &\geq y_u - y_v & \forall\, uv \in A,\\
y_s &= 1, y_t = 0, &\\
z_a &\geq 0 & \forall\, a \in A
\end{aligned}
$$

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0$
- $(u_a - f_a) z_a = 0$

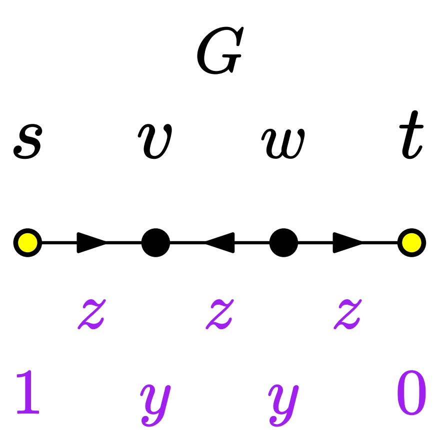
相補性条件と補助ネットワーク

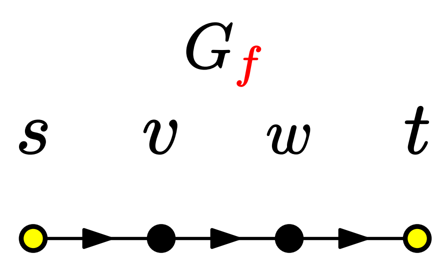
設定：ネットワーク (G, u), s-t 流 f,
相補性条件を満たす $y \in \mathbb{R}^V, z \in \mathbb{R}^A$

性質：相補性条件と補助ネットワーク

補助ネットワーク (G_f, u_f) において，増加道が存在 ⇒ y, z は双対問題 (D) の許容解ではない

証明 (概略)：





(D) の制約

$$\begin{aligned} & z_{uv} \geq y_u - y_v && \forall\, uv \in A, \\ & y_s = 1, y_t = 0, && \\ & z_a \geq 0 && \forall\, a \in A \end{aligned}$$

相補性条件：
- $f_{uv}(z_{uv} - y_u + y_v) = 0$
- $(u_a - f_a) z_a = 0$

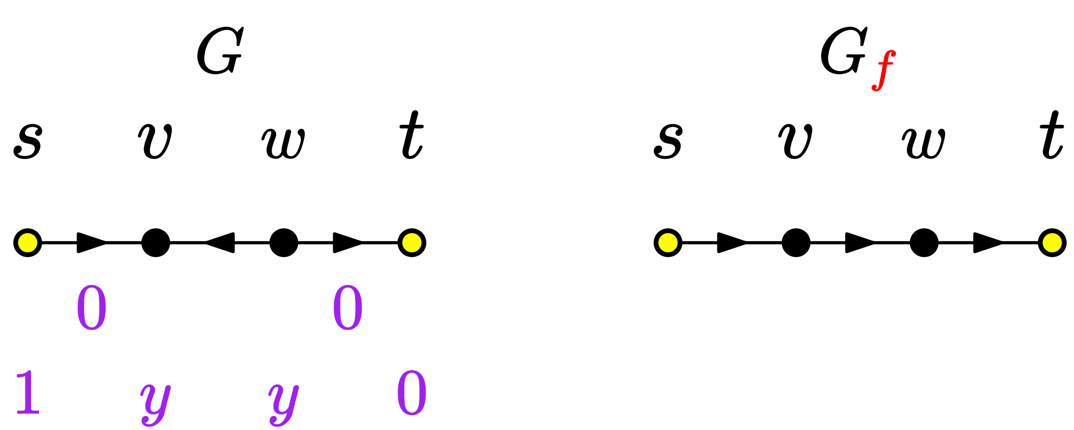
相補性条件と補助ネットワーク

設定：ネットワーク (G, u)，s-t 流 f，
相補性条件を満たす $y \in \mathbb{R}^V, z \in \mathbb{R}^A$

性質：相補性条件と補助ネットワーク

補助ネットワーク (G_f, u_f) において，増加道が存在 $\Rightarrow$ y, z は双対問題 (D) の許容解ではない

証明 (概略)：



(D) の制約

$$
\begin{aligned}
&z_{uv} \geq y_u - y_v && \forall\, uv \in A,\\
&y_s = 1, y_t = 0,\\
&z_a \geq 0 && \forall\, a \in A
\end{aligned}
$$

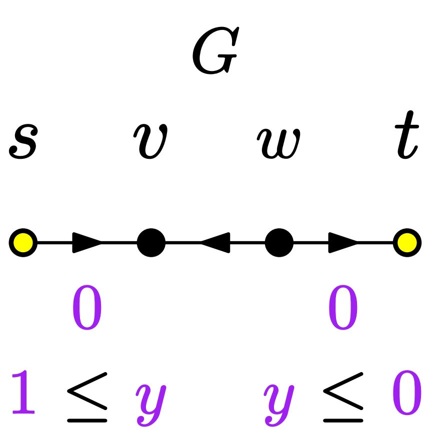
相補性条件：
- $f_{uv}(z_{uv} - y_u + y_v) = 0$
- $(u_a - f_a)z_a = 0$

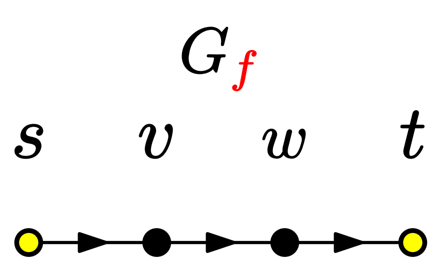
設定：ネットワーク (G, u)，s-t 流 f，
相補性条件を満たす $y \in \mathbb{R}^V, z \in \mathbb{R}^A$

性質：相補性条件と補助ネットワーク

補助ネットワーク (G_f, u_f) において，増加道が存在 $\Rightarrow$ y, z は双対問題 (D) の許容解ではない

証明 (概略)：





(D) の制約

$$
\begin{aligned}
& z_{uv} \ge y_u - y_v && \forall\, uv \in A, \\
& y_s = 1, y_t = 0, && \\
& z_a \ge 0 && \forall\, a \in A
\end{aligned}
$$

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0$
- $(u_a - f_a) z_a = 0$

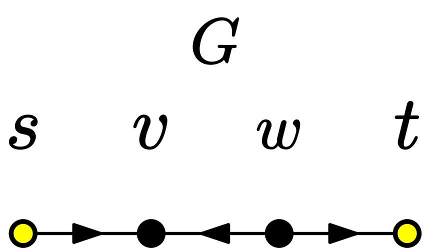
相補性条件と補助ネットワーク

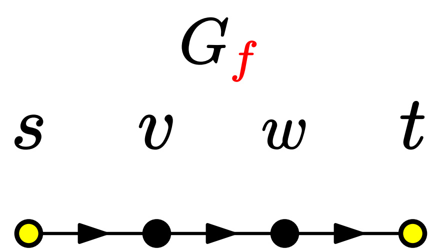
設定：ネットワーク (G, u)，s-t 流 f，
相補性条件を満たす $y \in \mathbb{R}^V, z \in \mathbb{R}^A$

性質：相補性条件と補助ネットワーク

補助ネットワーク (G_f, u_f) において，増加道が存在 $\Rightarrow$
y, z は双対問題 (D) の許容解ではない

証明 (概略)：　$y_w = y_v + z_{uv} \geq y_v$





$0 \qquad 0$

$1 \leq y \leq y \leq 0$

(D) の制約

$$
\begin{aligned}
& z_{uv} \geq y_u - y_v && \forall\, uv \in A, \\
& y_s = 1, y_t = 0, && \\
& z_a \geq 0 && \forall\, a \in A
\end{aligned}
$$

相補性条件：
- $f_{uv}(z_{uv} - y_u + y_v) = 0$
- $(u_a - f_a) z_a = 0$

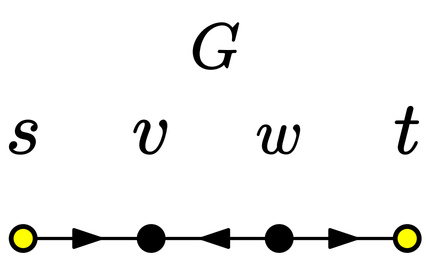
相補性条件と補助ネットワーク

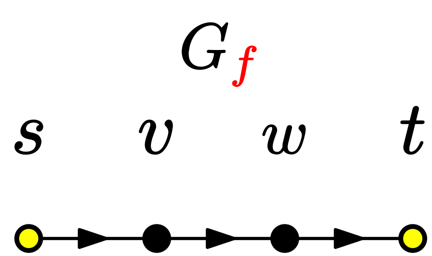
設定：ネットワーク (G, u)， s-t 流 f，
相補性条件を満たす $y \in \mathbb{R}^V, z \in \mathbb{R}^A$

性質：相補性条件と補助ネットワーク

補助ネットワーク (G_f, u_f) において，増加道が存在 $\Rightarrow$ y, z は双対問題 (D) の許容解ではない

証明 (概略)： $y_w = y_v + z_{uv} \geq y_v$





$0 \qquad 0$

$1 \leq y \leq y \leq 0$

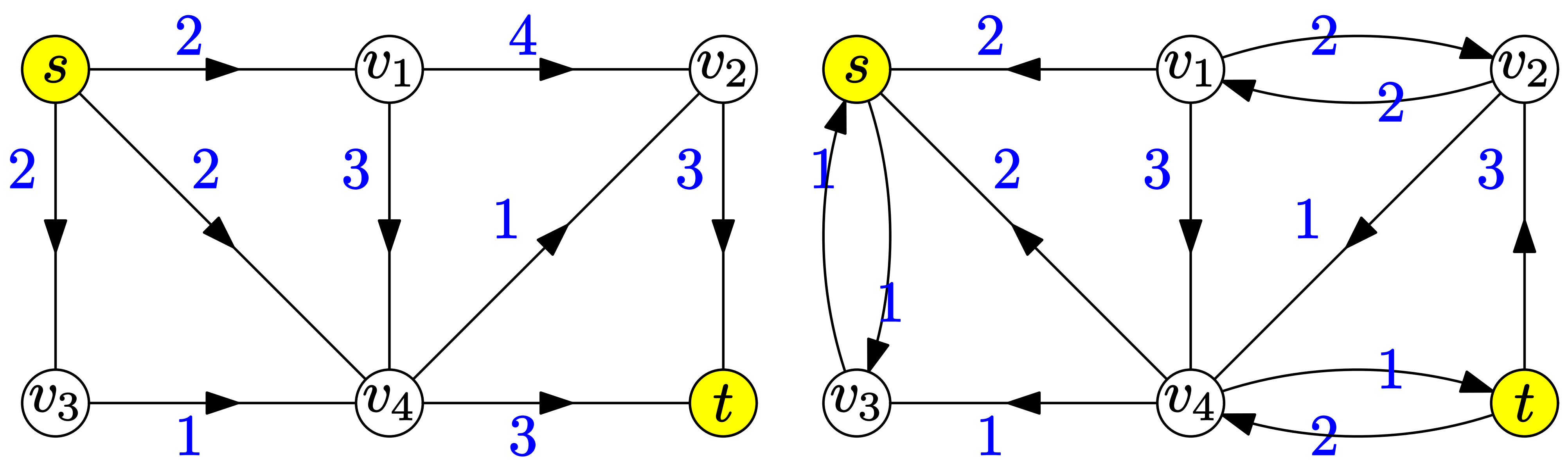
(D) の制約

$$z_{uv} \geq y_u - y_v \quad \forall\, uv \in A,$$
$$y_s = 1, y_t = 0,$$
$$z_a \geq 0 \quad \forall\, a \in A$$

相補性条件：
- $f_{uv}(z_{uv} - y_u + y_v) = 0$
- $(u_a - f_a)z_a = 0$

相補性条件：

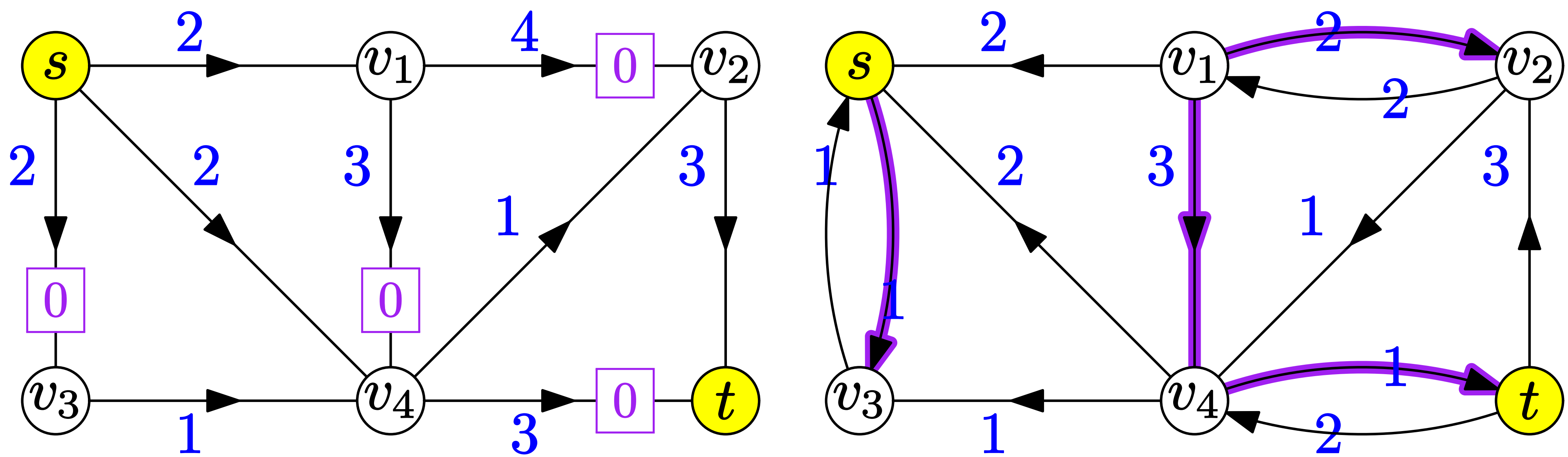
- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$



相補性条件と補助ネットワーク

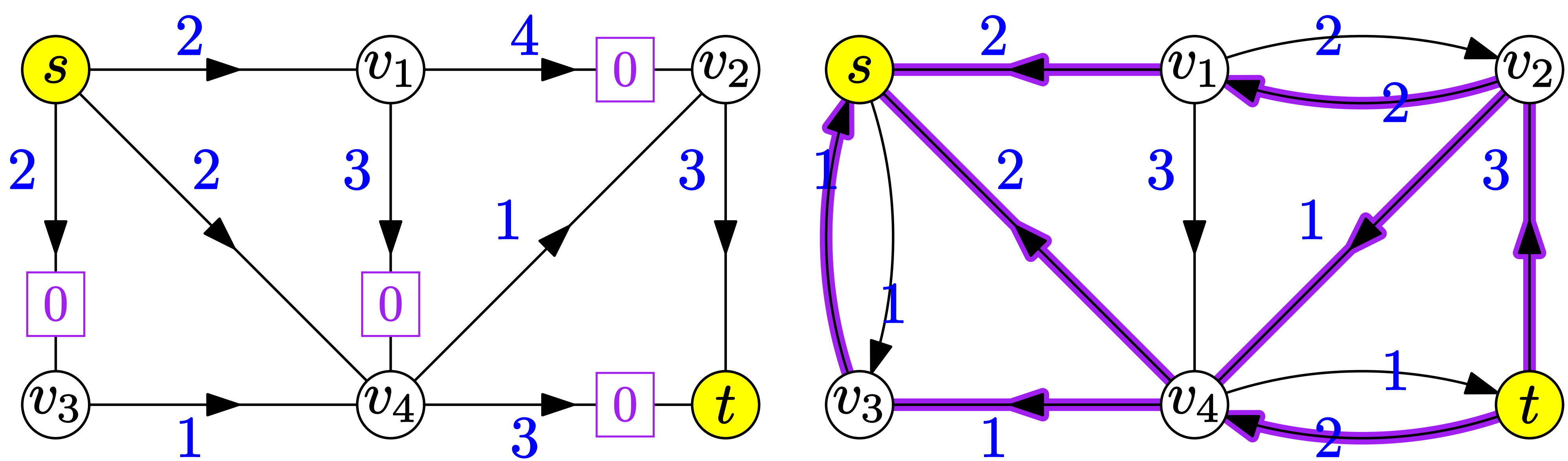
相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a) z_a = 0 \quad \forall\, a \in A$



相補性条件：

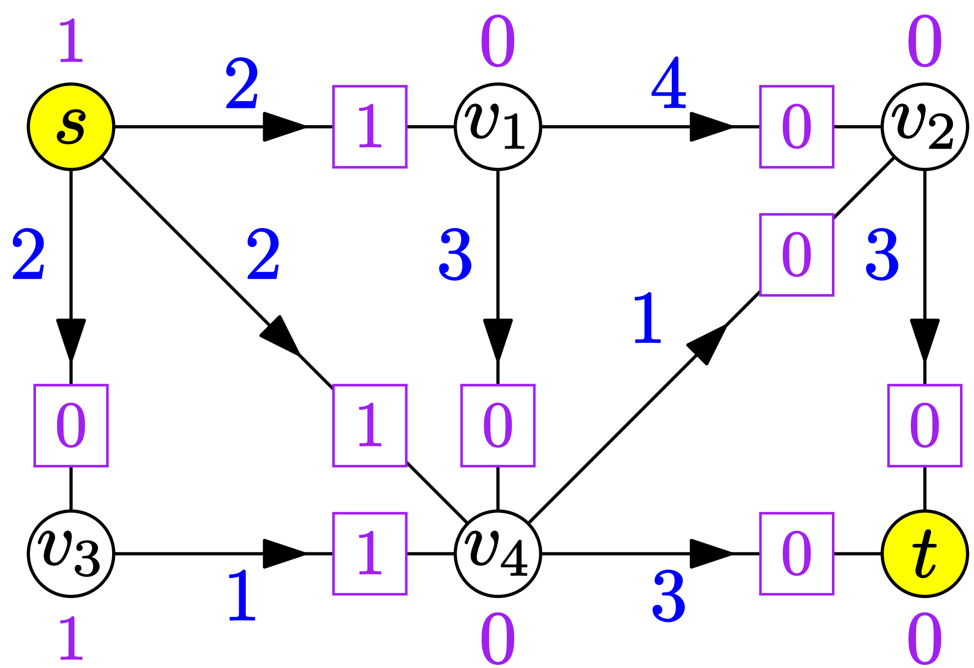
- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$



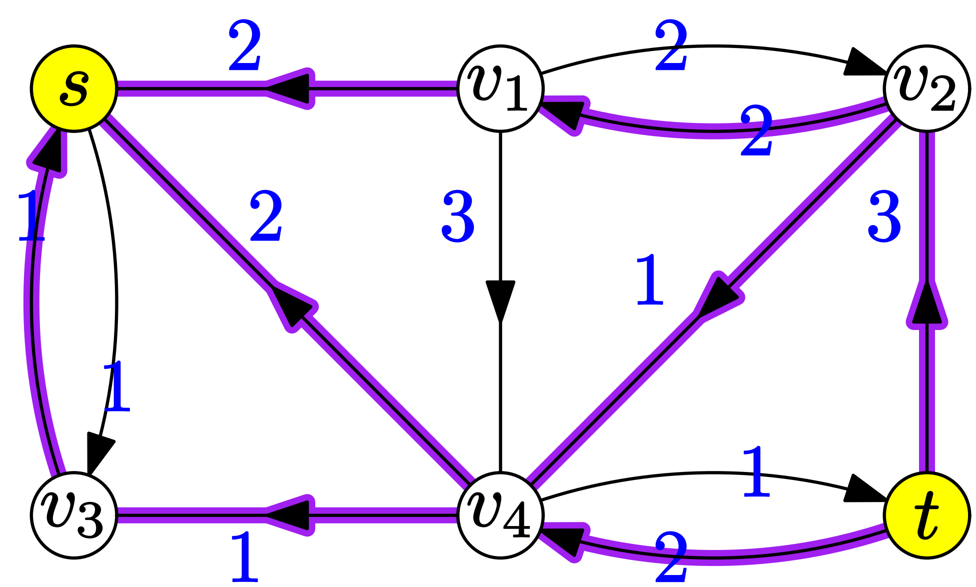
相補性条件と補助ネットワーク

相補性条件：

- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$



これは双対許容解



相補性定理より，これは双対最適解

増加道法と相補性定理

相補性定理：f が (P) の最適解，y, z が (D) の最適解 $\Leftrightarrow$

- f が (P) の許容解，y, z が (D) の許容解
- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a) z_a = 0 \quad \forall\, a \in A$

増加道法 は 次を満たす f, y, z を考える

- アルゴリズムの実行中は常に
  主許容性 と **相補性** を満たすようにする
- アルゴリズムが停止したときに
  双対許容性 も満たすようにする

相補性定理より，停止したときの f, y, z は最適解！

Push-Relabel 法と相補性定理

相補性定理：f が (P) の最適解，y, z が (D) の最適解 $\Leftrightarrow$

- f が (P) の許容解，y, z が (D) の許容解
- $f_{uv}(z_{uv} - y_u + y_v) = 0 \quad \forall\, uv \in A$
- $(u_a - f_a)z_a = 0 \quad \forall\, a \in A$

Push-Relabel 法 は 次を満たす f, y, z を考える

- アルゴリズムの実行中は常に
  双対許容性 と **相補性** を満たすようにする
- アルゴリズムが停止したときに
  主許容性 も満たすようにする

相補性定理より，停止したときの f, y, z は最適解！

増加道法と Push-Relabel 法

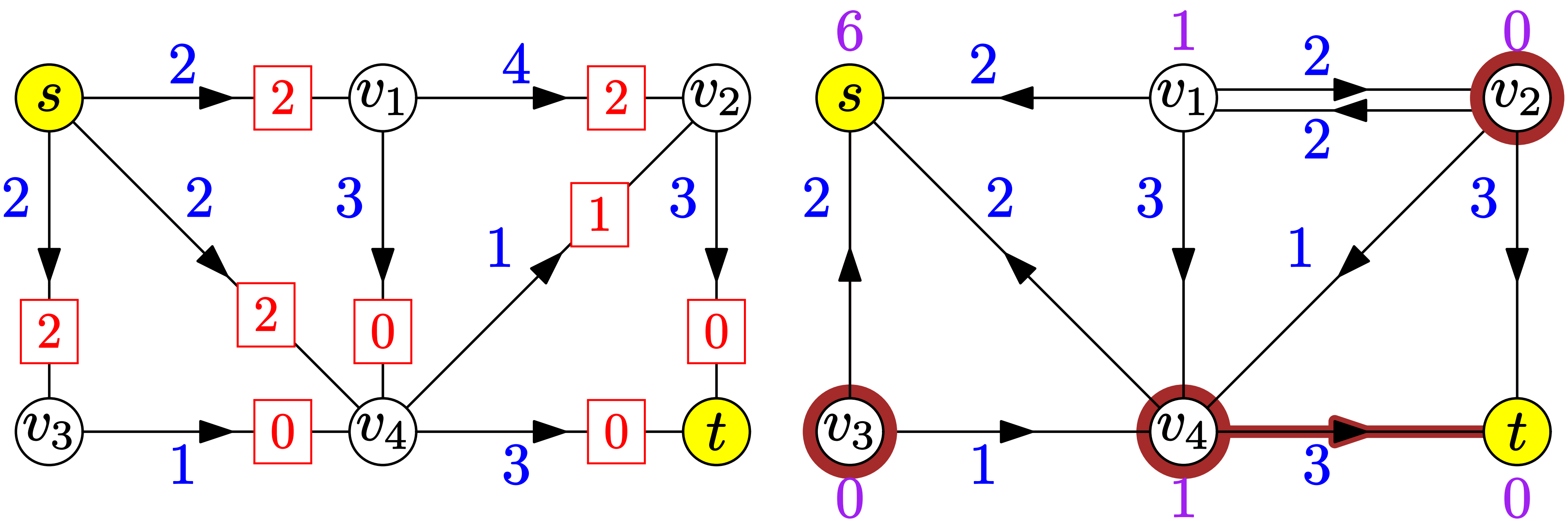
増加道法

主許容性
を満たしたまま **双対許容性** を満たしに行く
相補性

Push-Relabel 法

双対許容性
を満たしたまま **主許容性** を満たしに行く
相補性

1. 増加道法と線形計画法
2. **前流と距離ラベル**
3. Push-Relabel 法：概要
4. Push-Relabel 法：性質



前流：定義

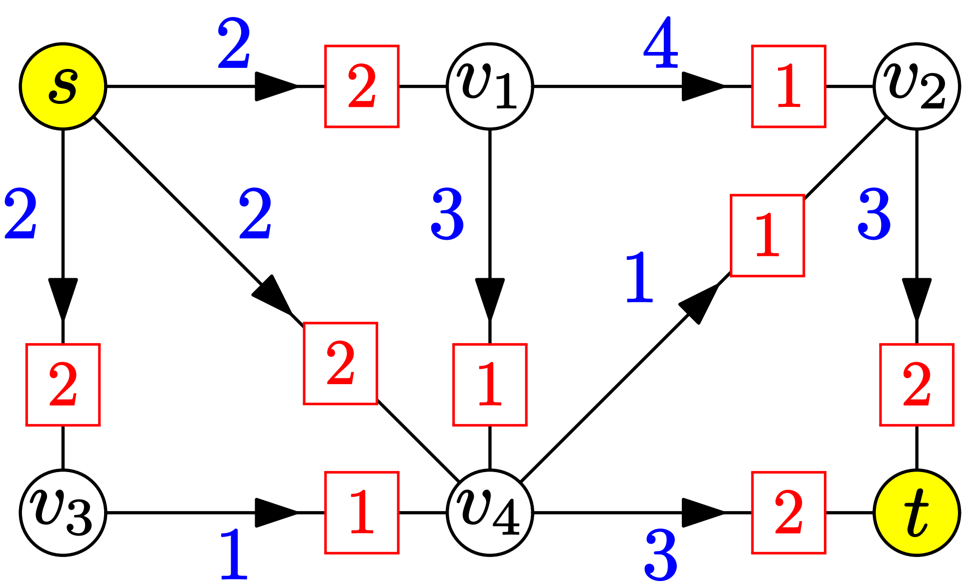
設定：有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，

弧容量関数 $u\colon A \to \mathbb{R}_+$

定義：前流 (preflow，準流，準フロー，プリフロー)

次を満たす関数 $f\colon A \to \mathbb{R}_+$ を
(G, u) における s-t **前流** と呼ぶ

- 任意の弧 $a \in A$ に対して，$0 \le f(a) \le u(a)$
- 任意の頂点 $v \in V - \{s, t\}$ に対して，$f^-(v) \ge f^+(v)$

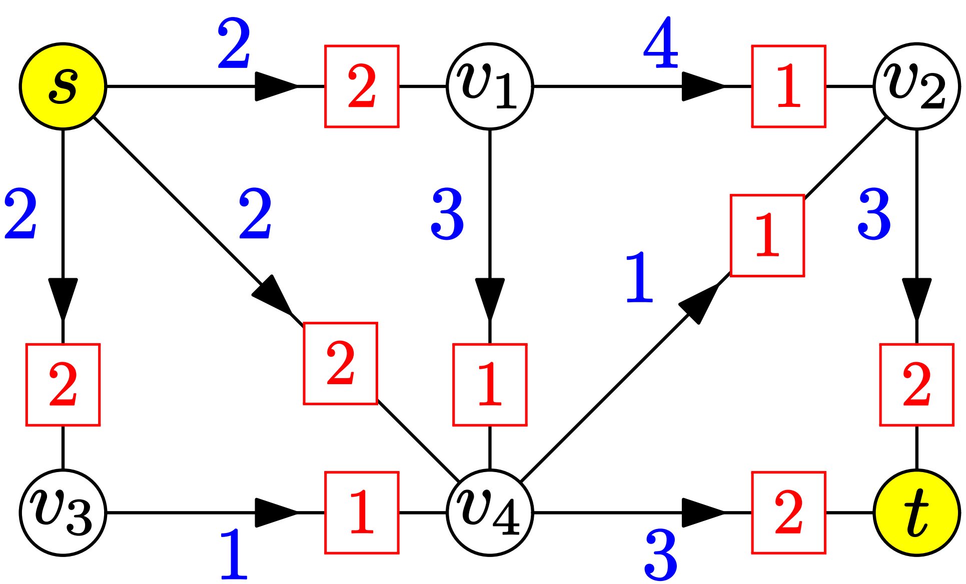


設定：有向グラフ $G=(V,A)$，2 頂点 $s,t\in V$，
弧容量関数 $u\colon A\to\mathbb{R}_+$

定義：前流 (preflow，準流，準フロー，プリフロー)

次を満たす関数 $f\colon A\to\mathbb{R}_+$ を
(G,u) における s-t **前流** と呼ぶ

- 任意の弧 $a\in A$ に対して，$0\le f(a)\le u(a)$
- 任意の頂点 $v\in V-\{s,t\}$ に対して，$f^-(v)\ge f^+(v)$



s-t 流なら

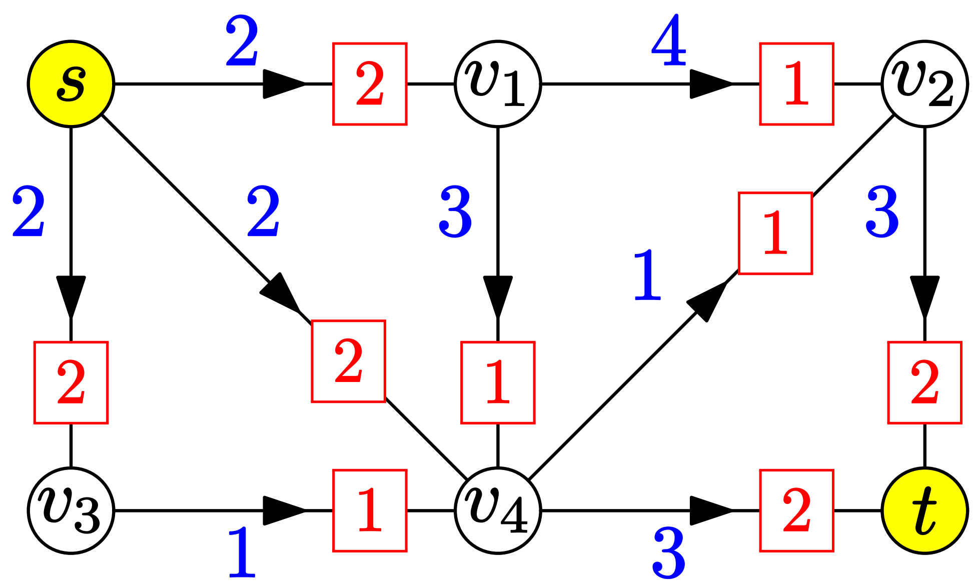
$$f^-(v)=f^+(v)$$

$\therefore$ s-t 流は s-t 前流

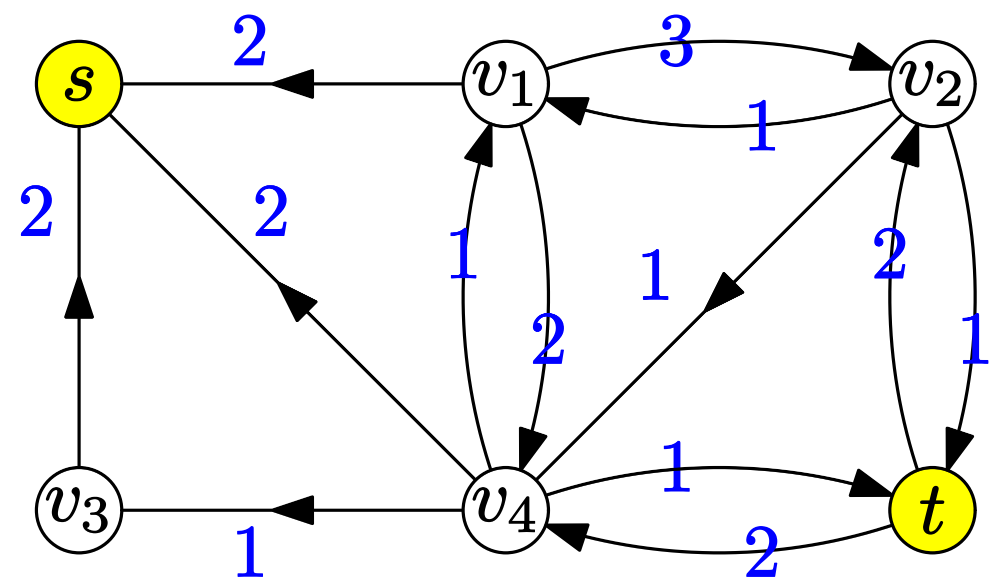
前流と補助ネットワーク

s-t 流のときと同様に,

s-t 前流に対して補助ネットワークを定義する



$(G, u), f$



(G_f, u_f)

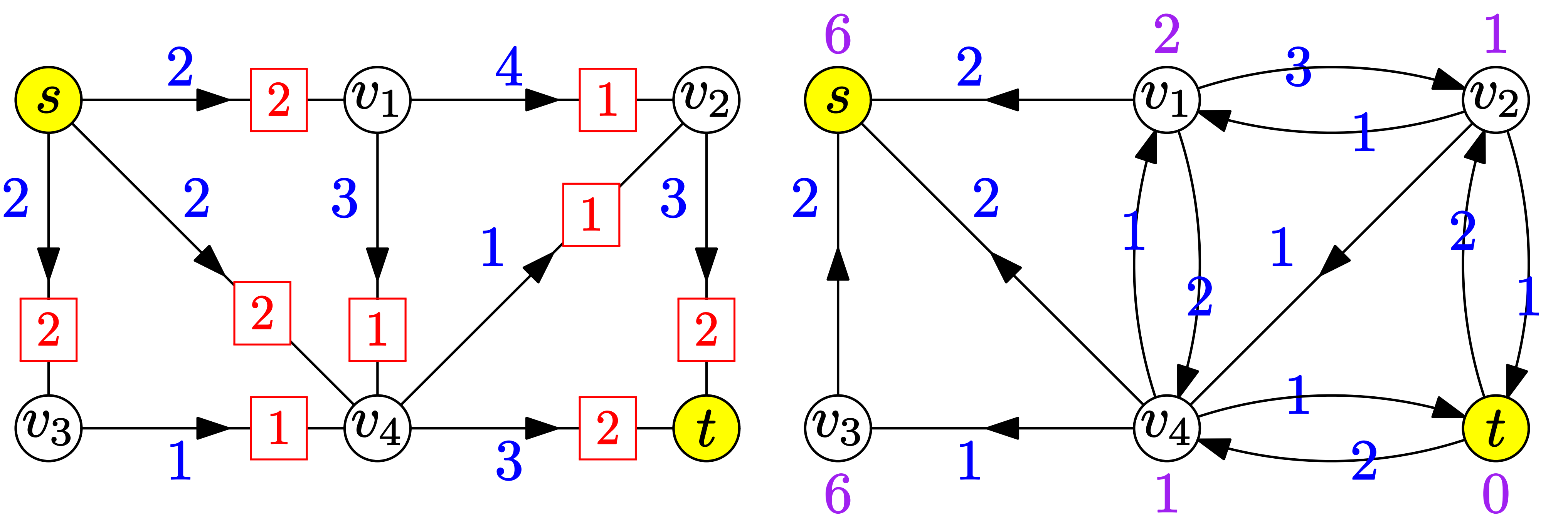
距離ラベル

設定：前と同様，s-t 前流 f，$n = |V|$

定義：距離ラベル (distance label)

次を満たす関数 $d\colon V \to \mathbb{Z}_+$ を
f に関する **距離ラベル** と呼ぶ

- $d(s) = n$，$d(t) = 0$
- 任意の弧 $(u, v) \in A_f$ に対して，$d(u) \leq d(v) + 1$

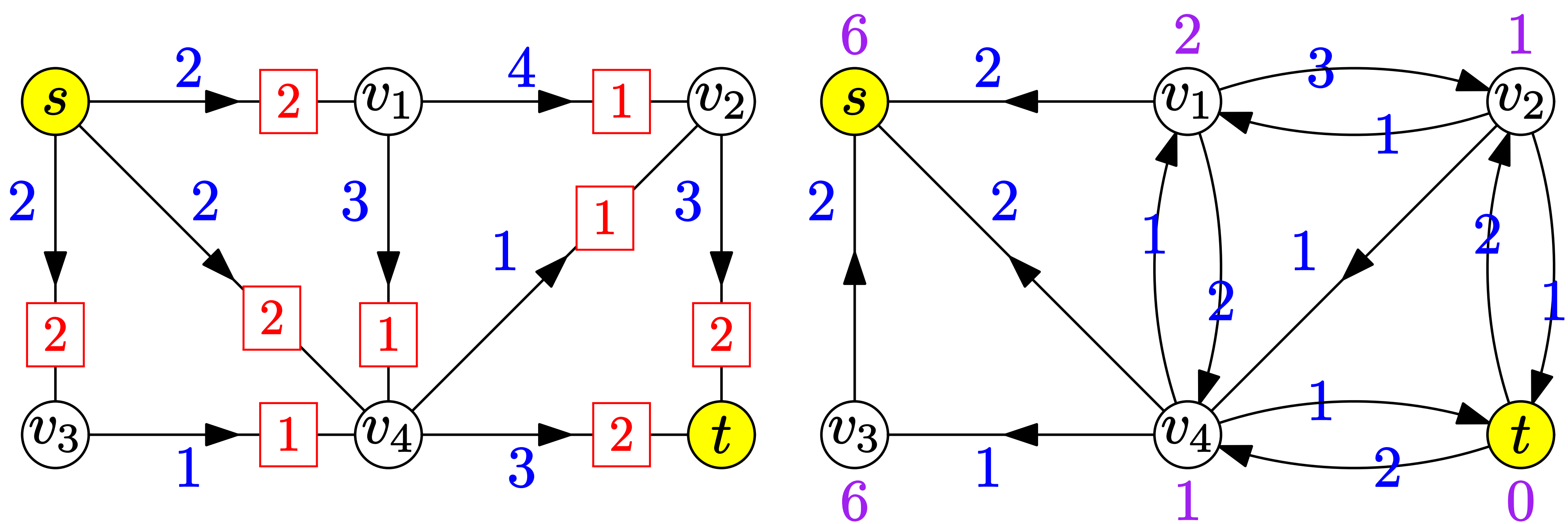


距離ラベルと補助ネットワーク

設定：前と同様，s-t 前流 f

性質：距離ラベルと補助ネットワーク

f に関する距離ラベル d が存在 ⇒
補助ネットワーク G_f には，増加道が存在しない

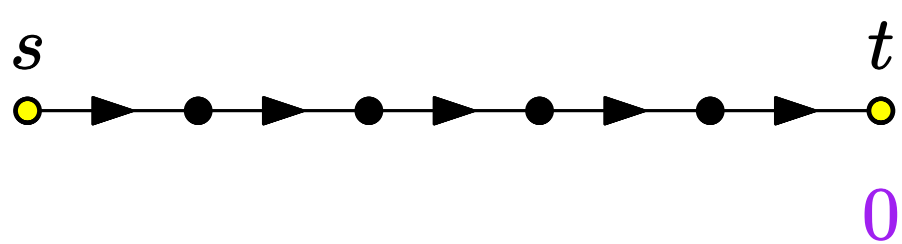


設定：前と同様，s-t 前流 f

性質：距離ラベルと補助ネットワーク

f に関する距離ラベル d が存在 $\Rightarrow$
補助ネットワーク G_f には，増加道が存在しない

証明：増加道が存在すると仮定



d が距離ラベル：

$d(s) = n$, $d(t) = 0$,

$d(u) \leq d(v) + 1 \quad \forall\, uv \in A_f$

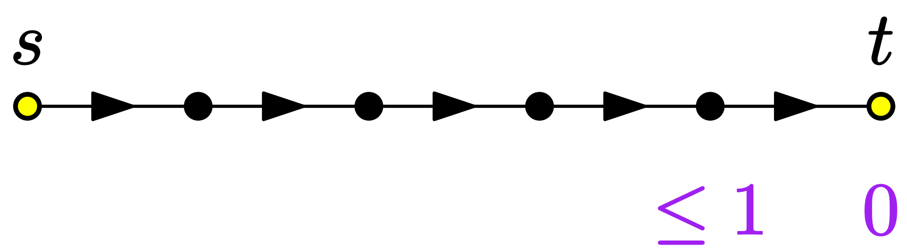
距離ラベルと補助ネットワーク：証明

設定：前と同様，s-t 前流 f

性質：距離ラベルと補助ネットワーク

f に関する距離ラベル d が存在 $\Rightarrow$
補助ネットワーク G_f には，増加道が存在しない

証明：増加道が存在すると仮定



d が距離ラベル：

$d(s) = n$, $d(t) = 0$,

$d(u) \leq d(v) + 1 \quad \forall\, uv \in A_f$

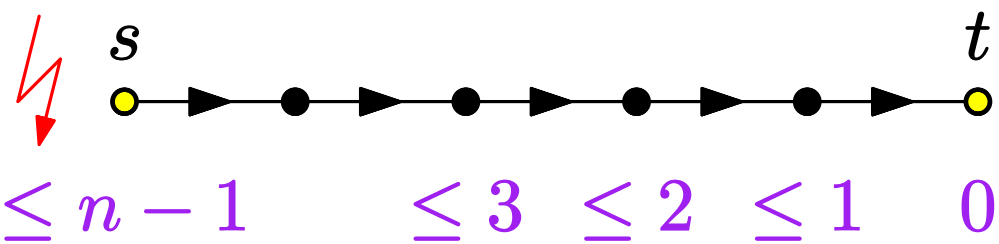
距離ラベルと補助ネットワーク：証明

設定：前と同様，s-t 前流 f

性質：距離ラベルと補助ネットワーク

f に関する距離ラベル d が存在 $\Rightarrow$
補助ネットワーク G_f には，増加道が存在しない

証明：増加道が存在すると仮定



d が距離ラベル：

$d(s) = n$, $d(t) = 0$,

$d(u) \leq d(v) + 1 \ \ \forall\, uv \in A_f$

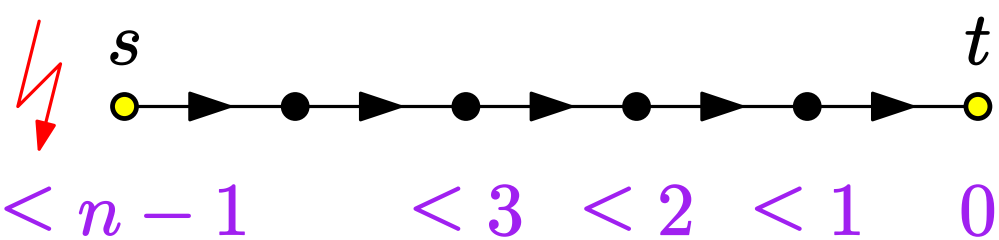
距離ラベルと補助ネットワーク：証明

<u>設定</u>：前と同様，s-t 前流 f

性質：距離ラベルと補助ネットワーク

f に関する距離ラベル d が存在 $\Rightarrow$
補助ネットワーク G_f には，増加道が存在しない

<u>証明</u>：増加道が存在すると仮定



d が距離ラベル：
$d(s) = n$, $d(t) = 0$,
$d(u) \leq d(v) + 1 \quad \forall\, uv \in A_f$

- 増加道の弧数 $\leq n-1$
- $\therefore d(s) \leq d(t) + n - 1 = n - 1$
- $d(s) = n$ に矛盾 $\square$

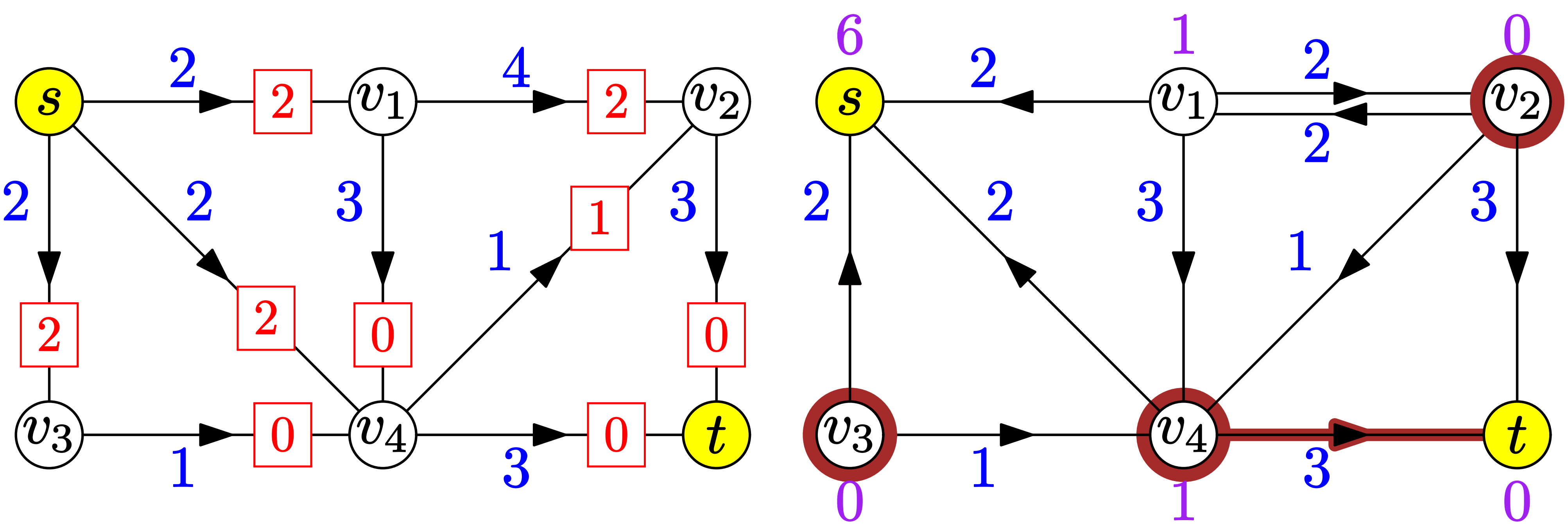
Push-Relabel 法

双対許容性
相補性
を満たしたまま **主許容性** を満たしに行く

<u>より詳細に述べると</u>

$$\begin{tikzcd}
\text{(双対許容解)} \quad \text{距離ラベル } d \arrow[r, "\text{更新}"] \arrow[d, leftrightarrow, "\text{(相補性)}"'] & \text{距離ラベル } d \arrow[d, leftrightarrow, "\text{(相補性)}"] \\
\text{(主非許容解)} \quad s\text{-}t \text{ 前流 } f \arrow[r, "\text{更新}"] & s\text{-}t \text{ 流 } f \quad \text{(主許容解)}
\end{tikzcd}$$

1. 増加道法と線形計画法
2. 前流と距離ラベル
3. **Push-Relabel 法：概要**
4. Push-Relabel 法：性質



Push-Relabel 法とは？

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

アルゴリズムの実行中は 常に次の 2 つを保持

- f： s-t 前流
- d： f に関する距離ラベル

Push(a)	$a \in A_f$	f の更新
Relabel(v)	$v \in V - \{s, t\}$	d の更新

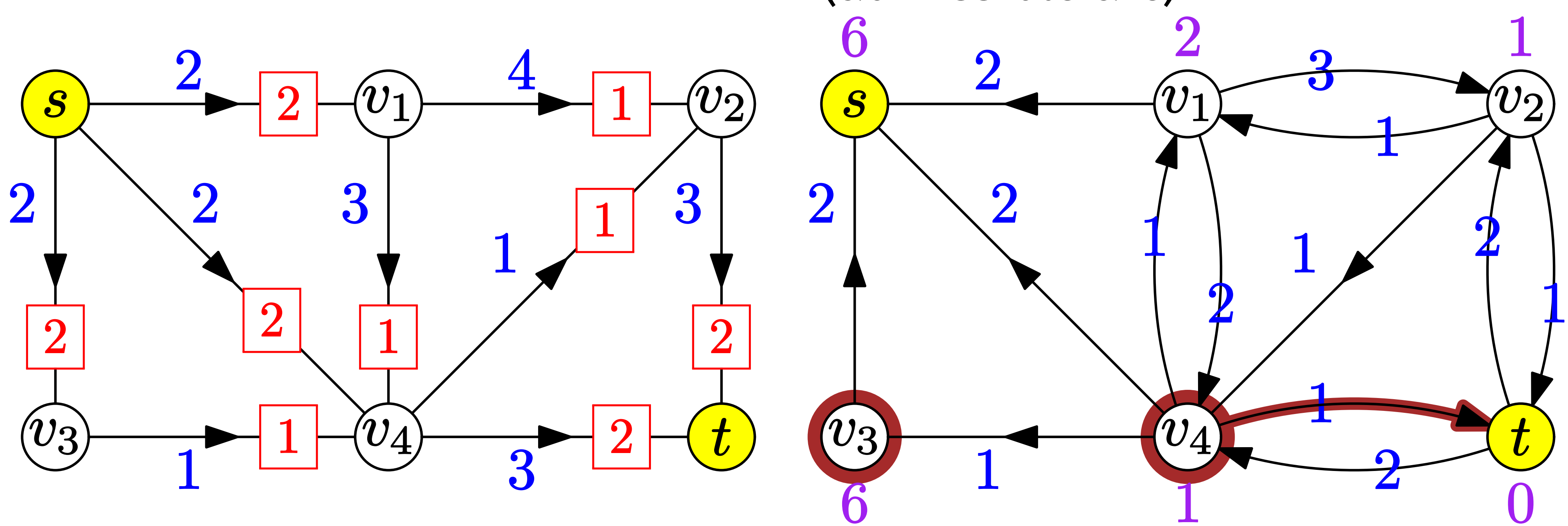
Push 操作 (1)

補助ネットワーク G_{f} の弧 $a = (u, v)$

Push(a) を行える条件

1. $f^{-}(u) - f^{+}(u) > 0$ 　　(u に残っている)
2. $d(u) = d(v) + 1$

このような弧 $a = (u, v)$ を **可能弧** と呼ぶことにする
(admissible arc)



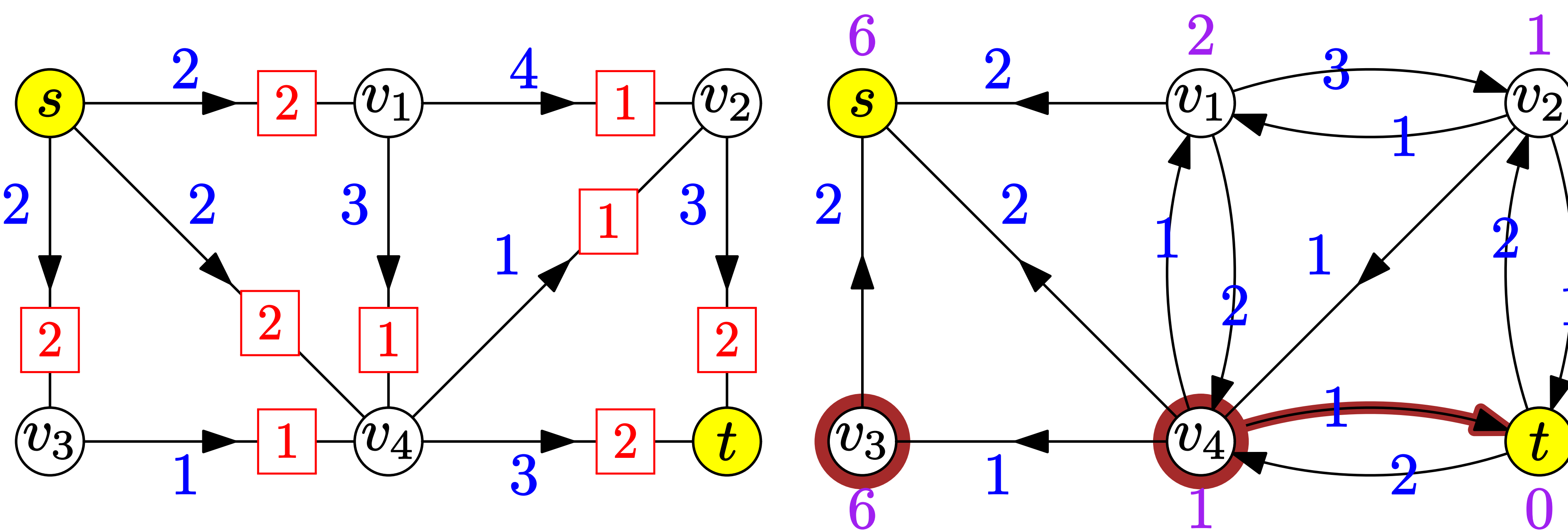
Push 操作 (2)

弧 $a = (u, v)$

Push(a)

$f(a)$ を $f(a) + \min\{u(a) - f(a), f^{-}(u) - f^{+}(u)\}$ に更新

容量まで流す　残りを流す



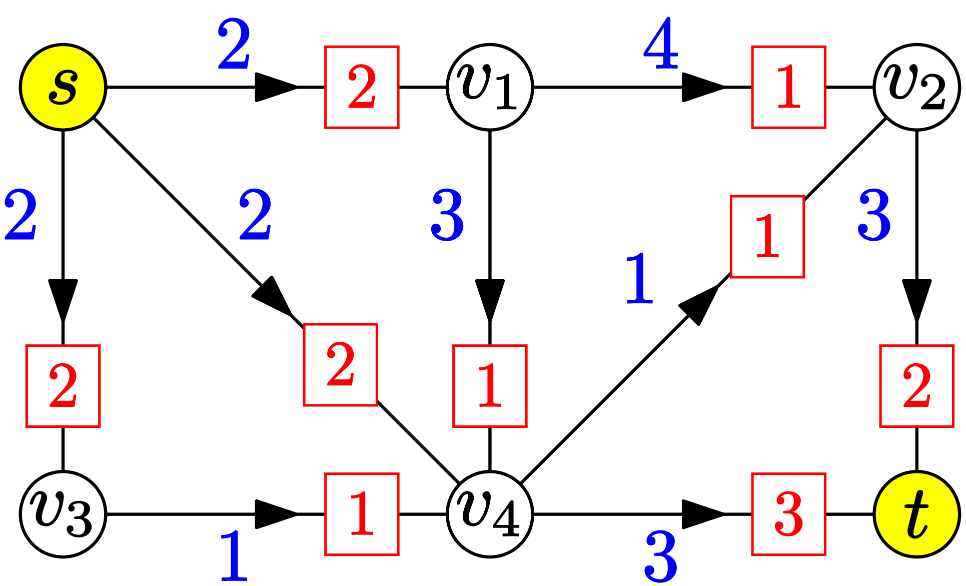
Push 操作 (2)

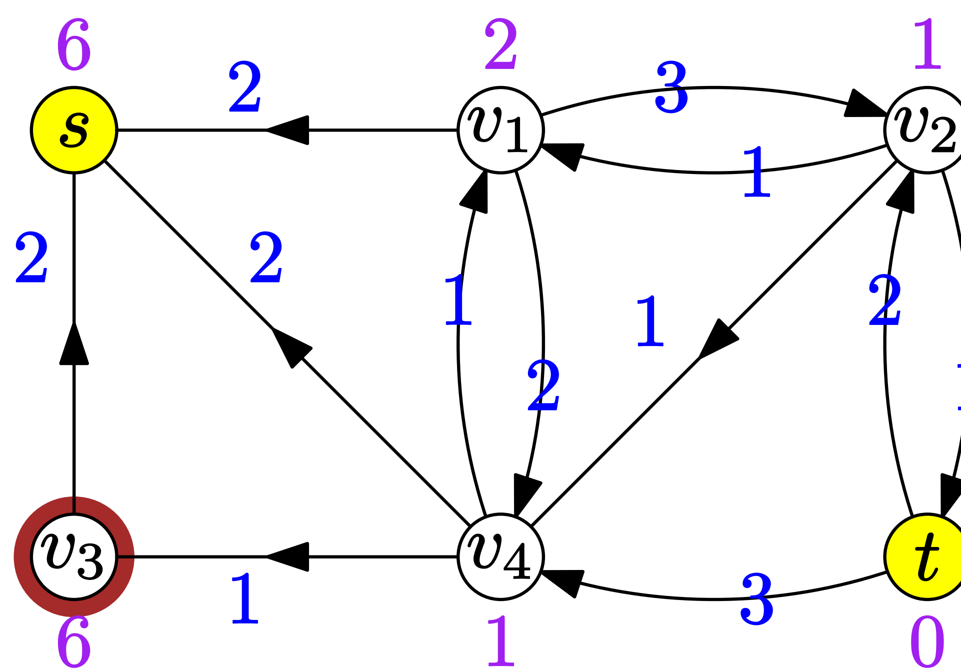
弧 $a = (u, v)$

Push(a)

$f(a)$ を $f(a) + \min\{u(a) - f(a), f^-(u) - f^+(u)\}$ に更新

容量まで流す　残りを流す





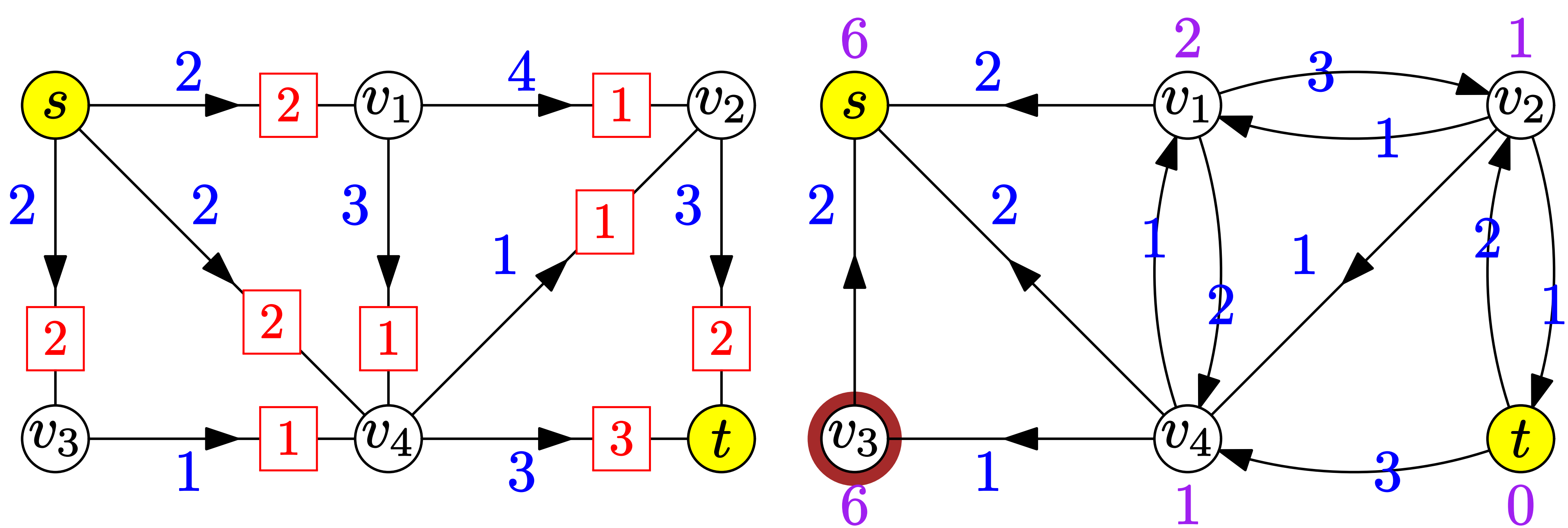
Relabel 操作 (1)

頂点 $u \in V - \{s, t\}$

Relabel(u) を行える条件

1. $f^{-}(u) - f^{+}(u) > 0$　　　　(u に残っている)
2. Push(a) を行える弧 $a \in A_f$ がない

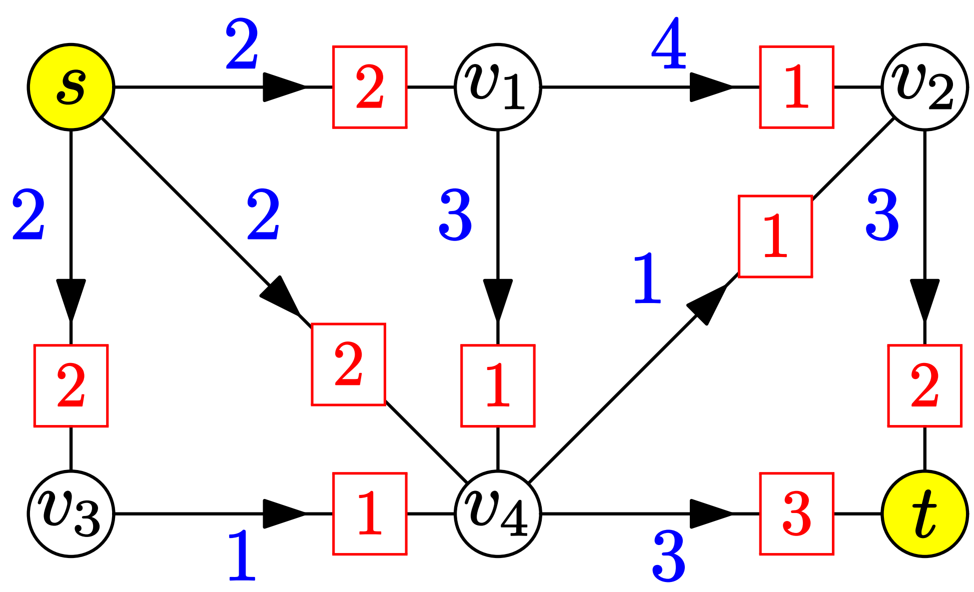
$\therefore$ 任意の $v \in V$ に対して，$uv \in A_f$ ならば $d(u) \leq d(v)$

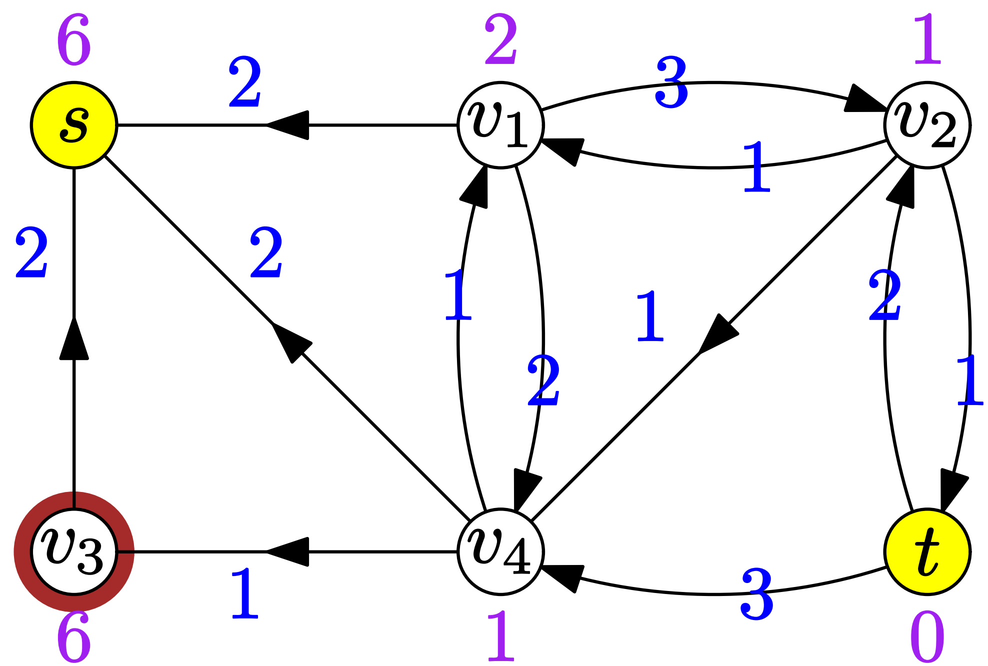


頂点 $u \in V - \{s, t\}$

Relabel(u)

$d(u)$ を $\min\{d(v) \mid uv \in A_f\} + 1$ に変更





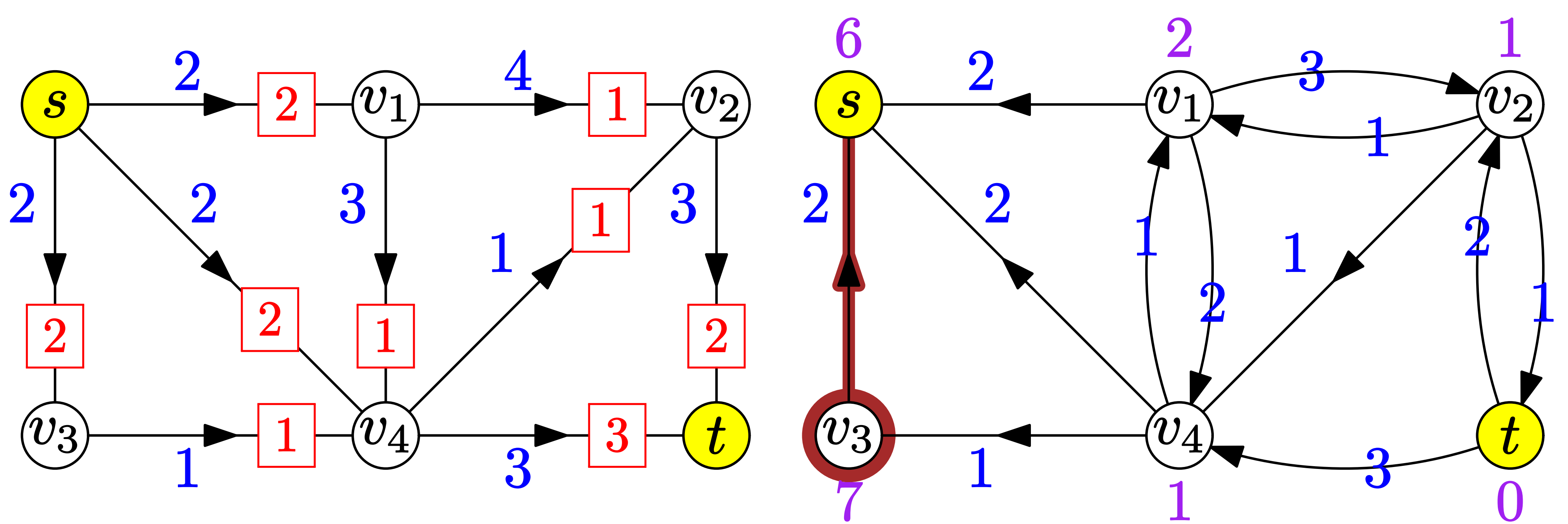
Relabel 操作 (2)

頂点 $u \in V - \{s, t\}$

Relabel(u)

$d(u)$ を $\min\{d(v) \mid uv \in A_f\} + 1$ に変更

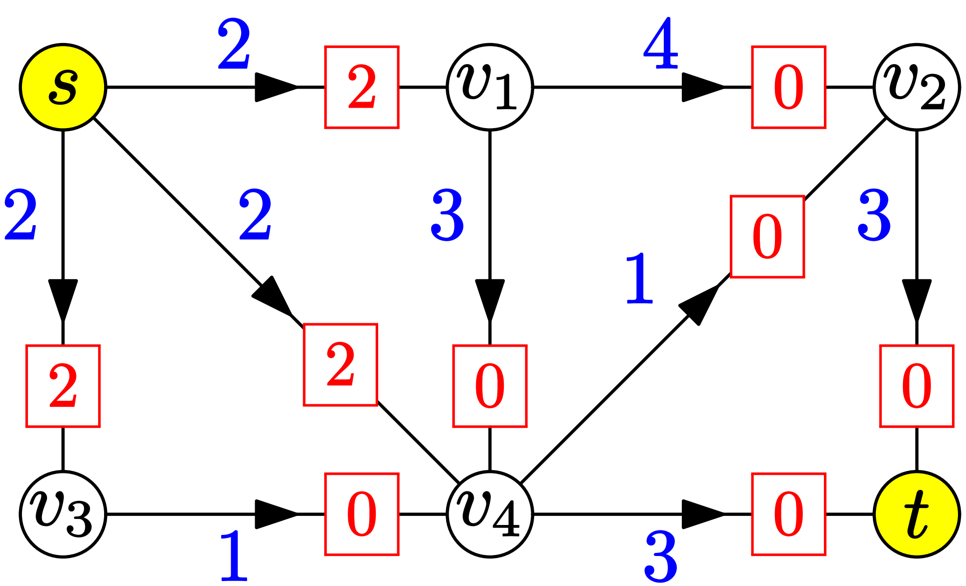
Relabel の後で，Push ができるようになる

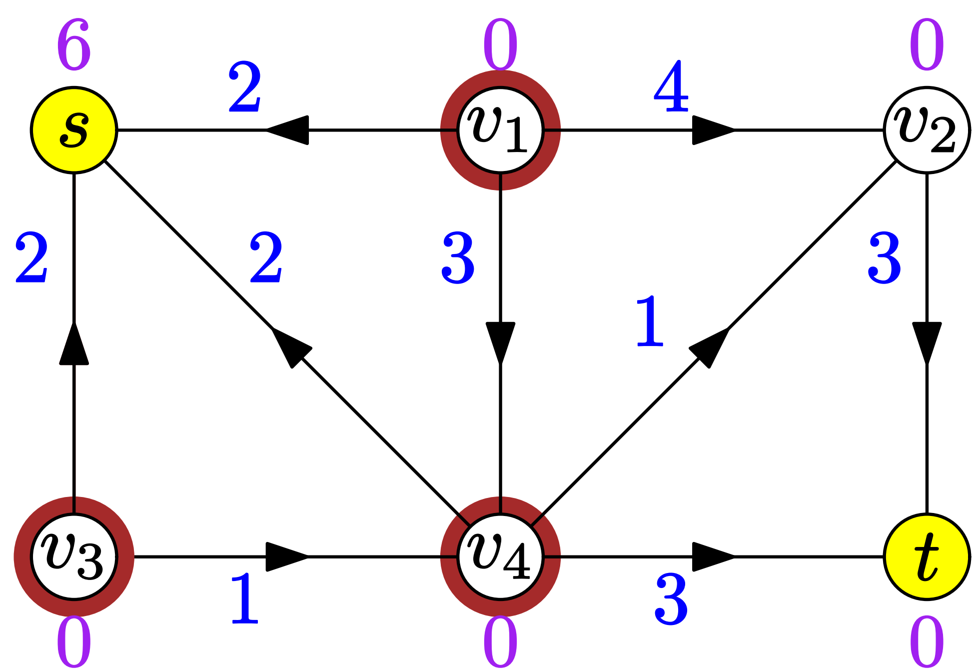


次のように，s-t 前流 f と距離ラベル d を初期化

- a が s を始点とするとき
  $f(a) = u(a)$
- その他のとき
  $f(a) = 0$

- $d(s) = n$
- その他の頂点 v に対して
  $d(v) = 0$



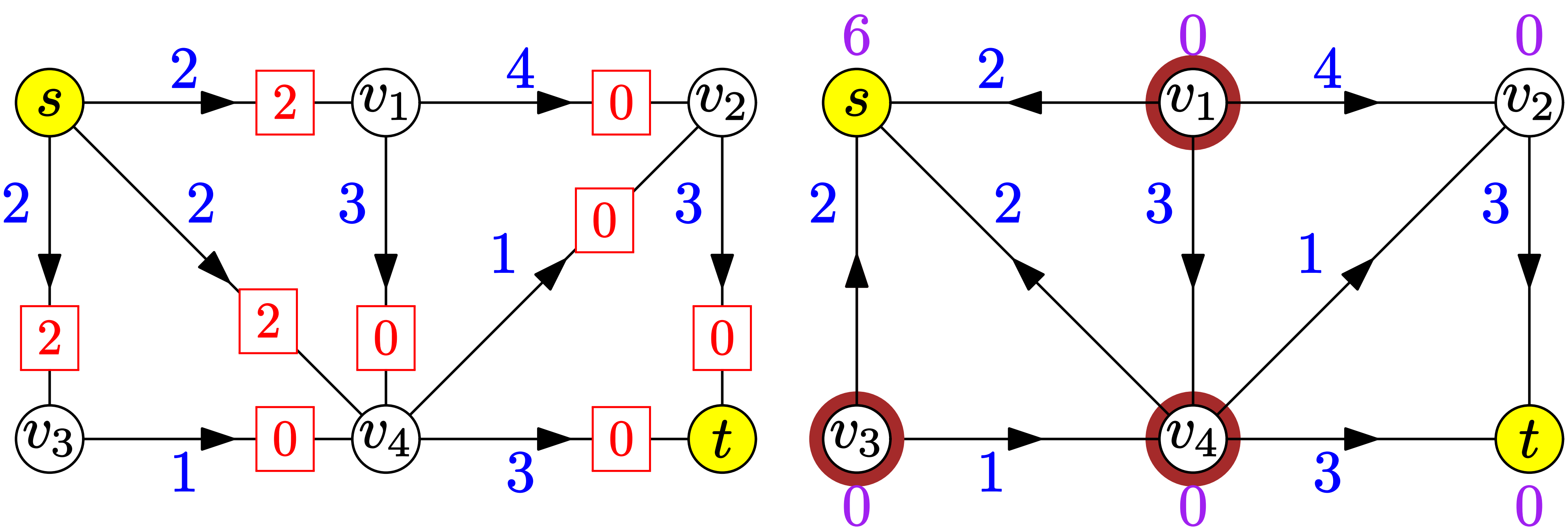


Push-Relabel 法：例 (1)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

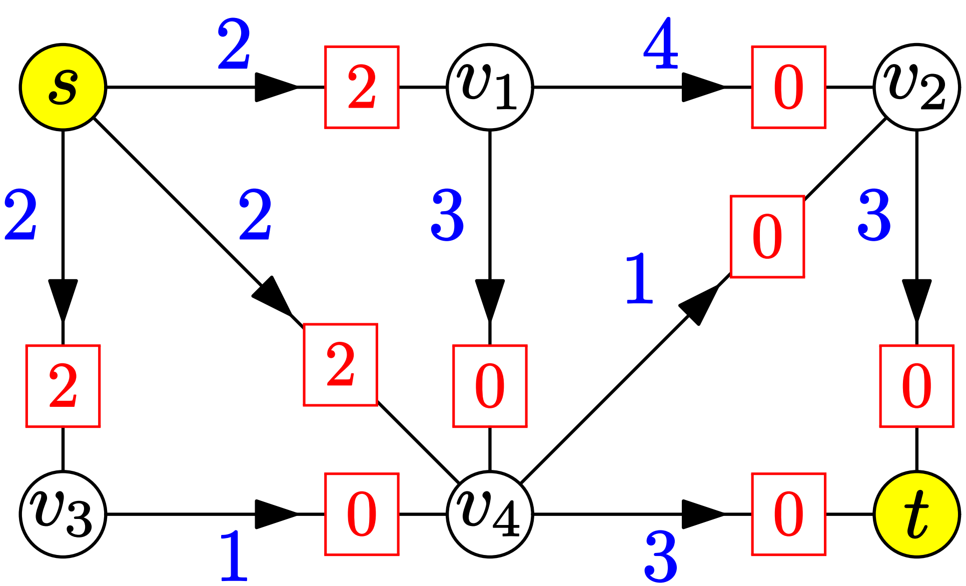
Relabel(v_1)

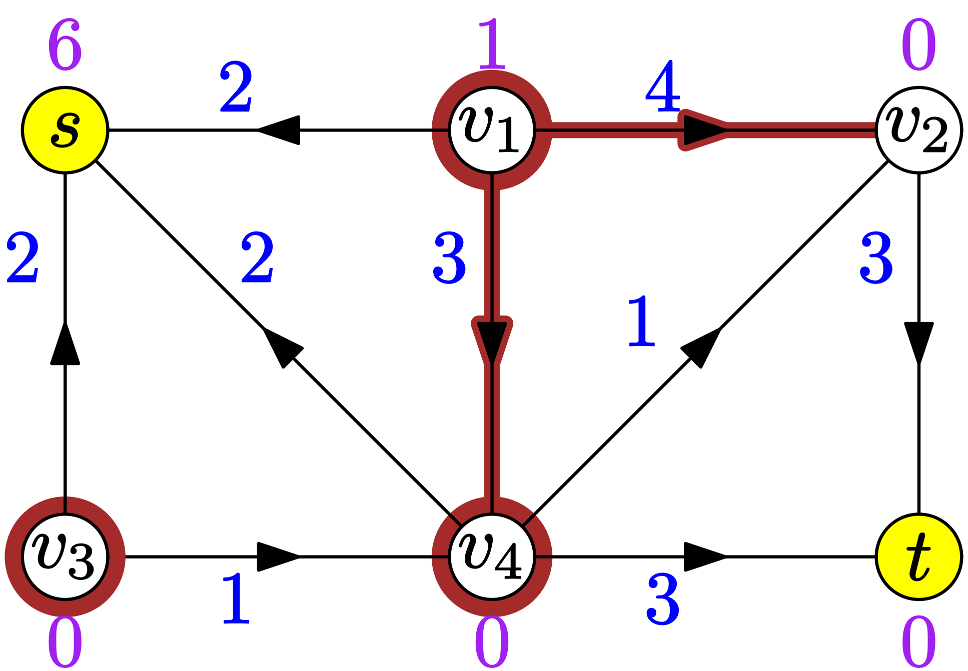


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Relabel(v_1)



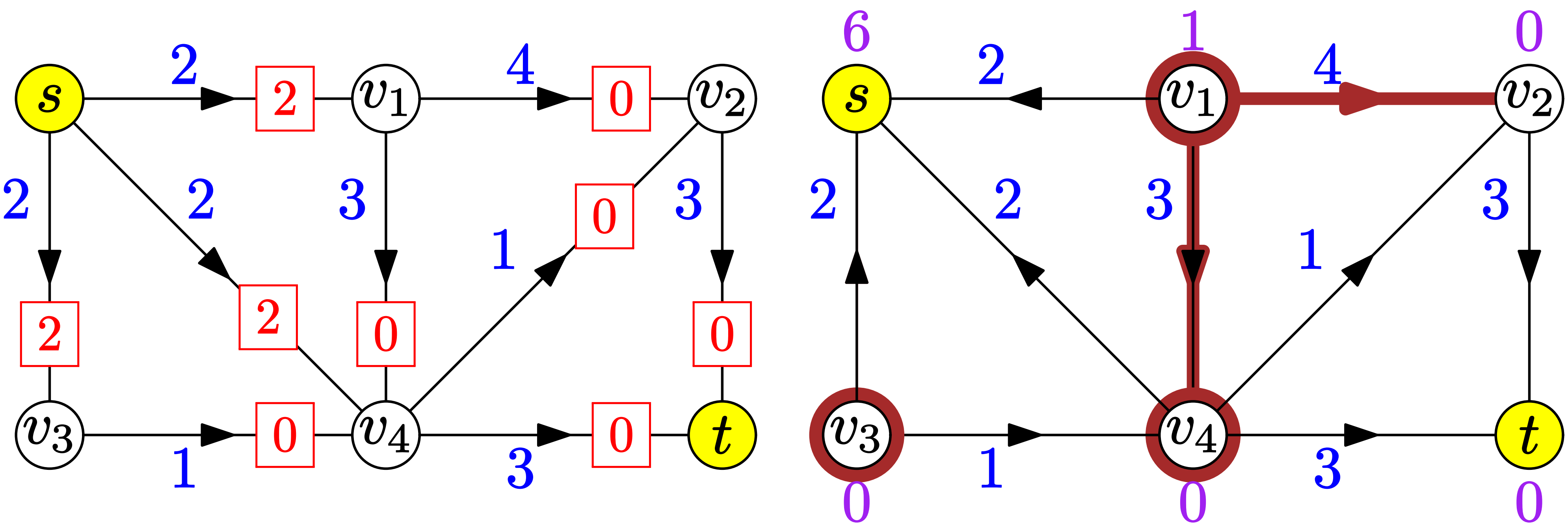


Push-Relabel 法：例 (2)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push($v_1 v_2$)

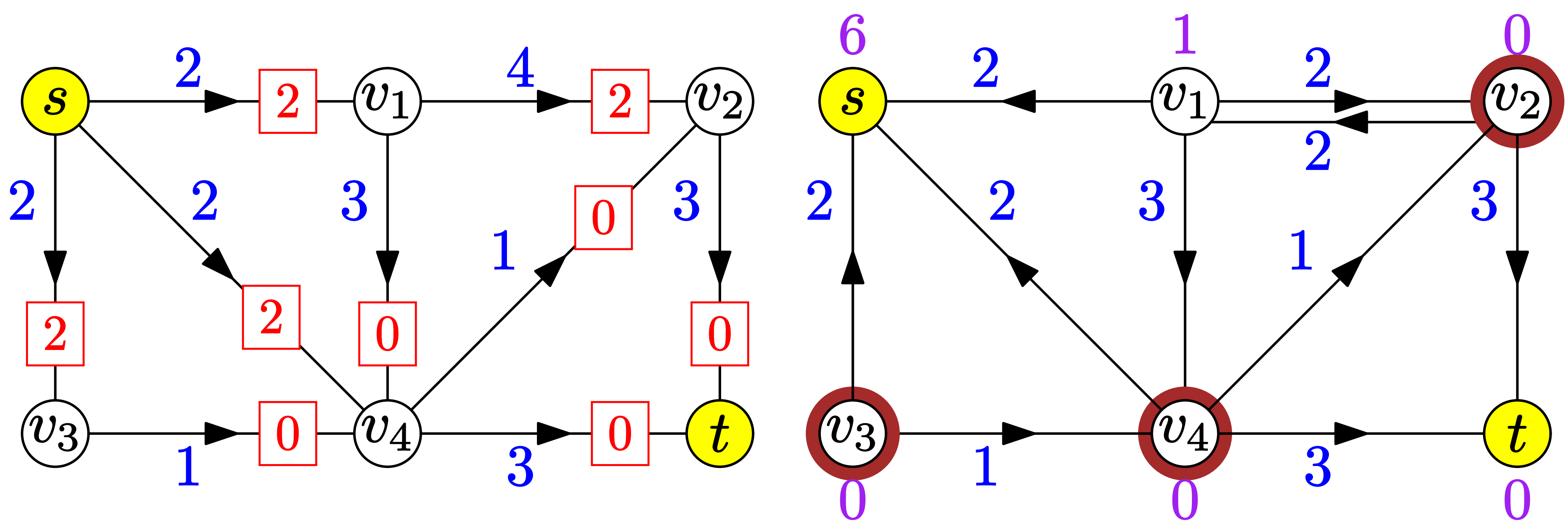


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_1 v_2)$

非飽和 Push
(unsaturated push)

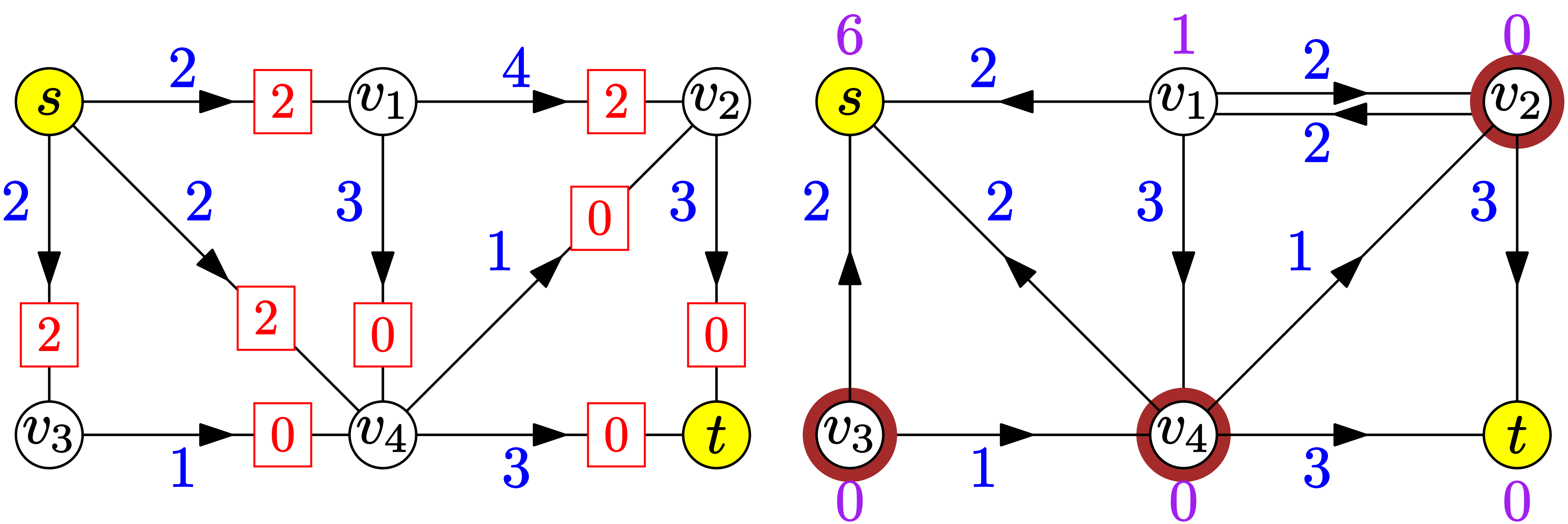


Push-Relabel 法：例 (3)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

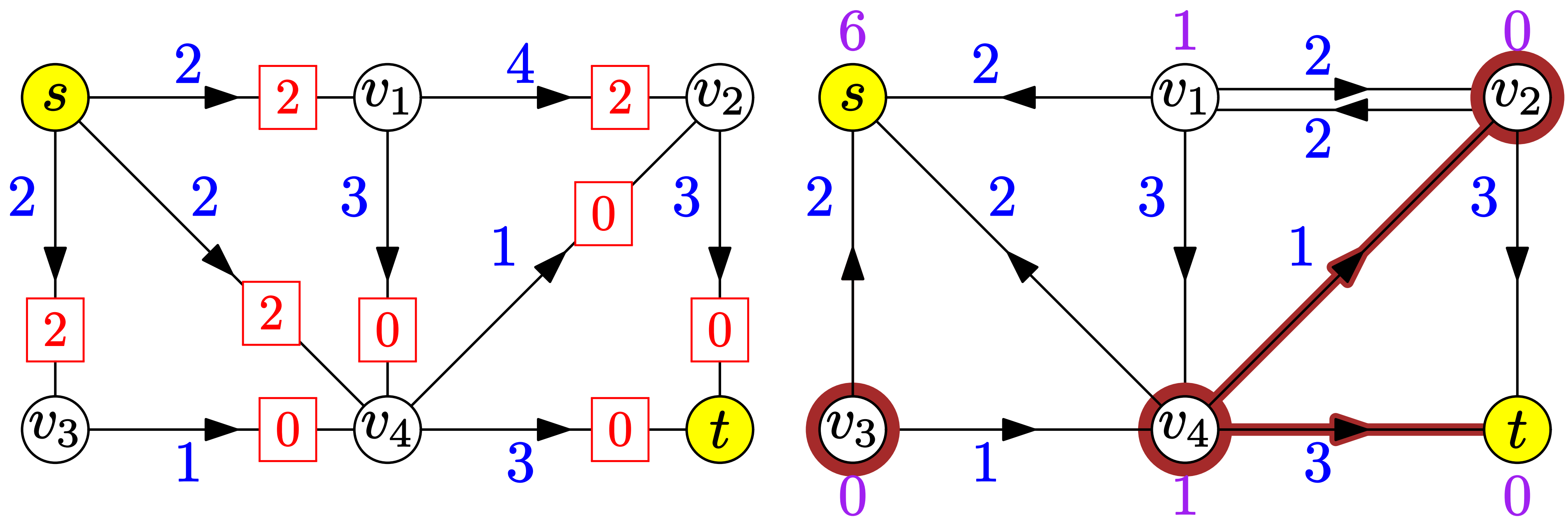
Relabel(v_4)



Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

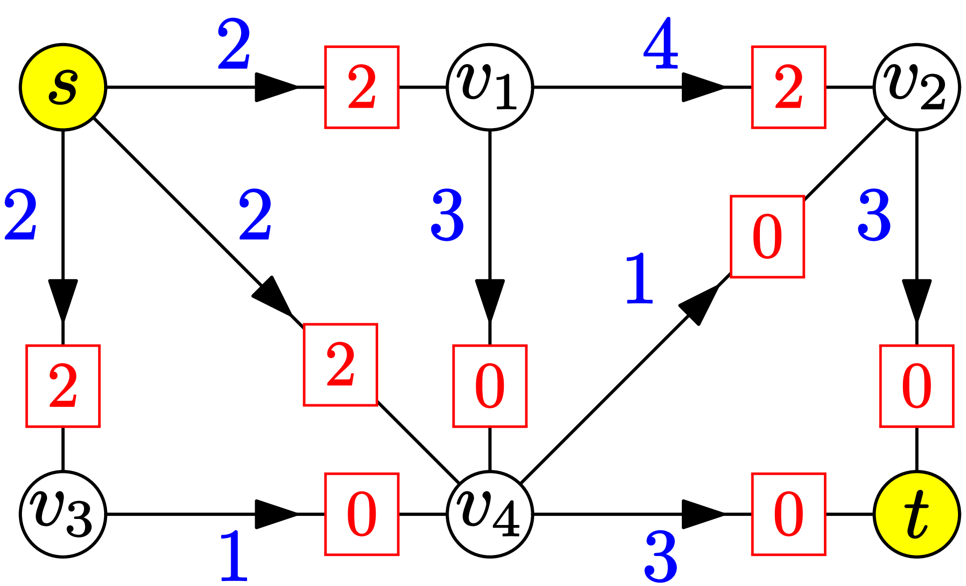
Relabel(v_4)

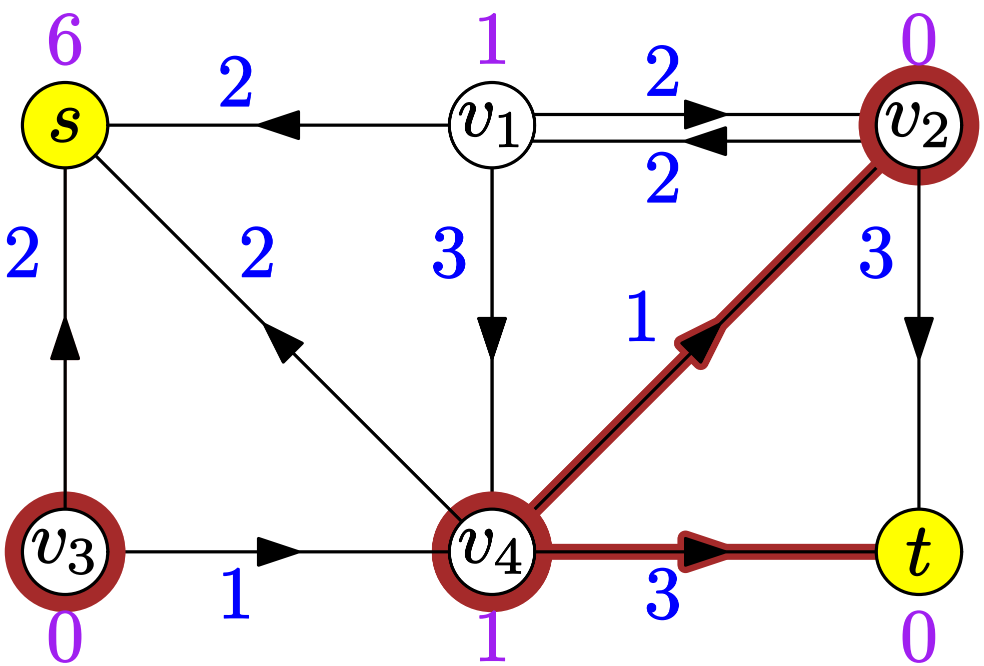


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_4 v_2)$



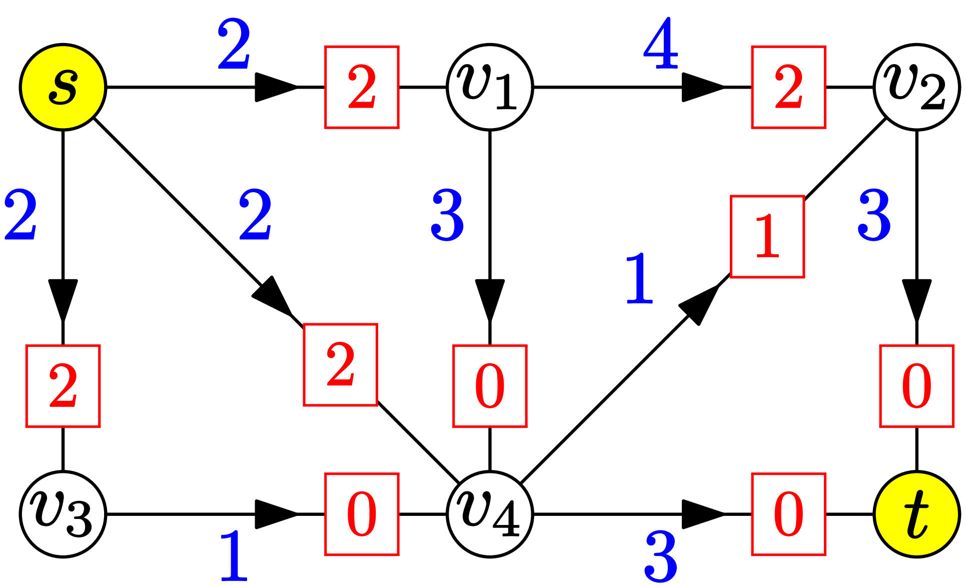


Push-Relabel 法：例 (4)

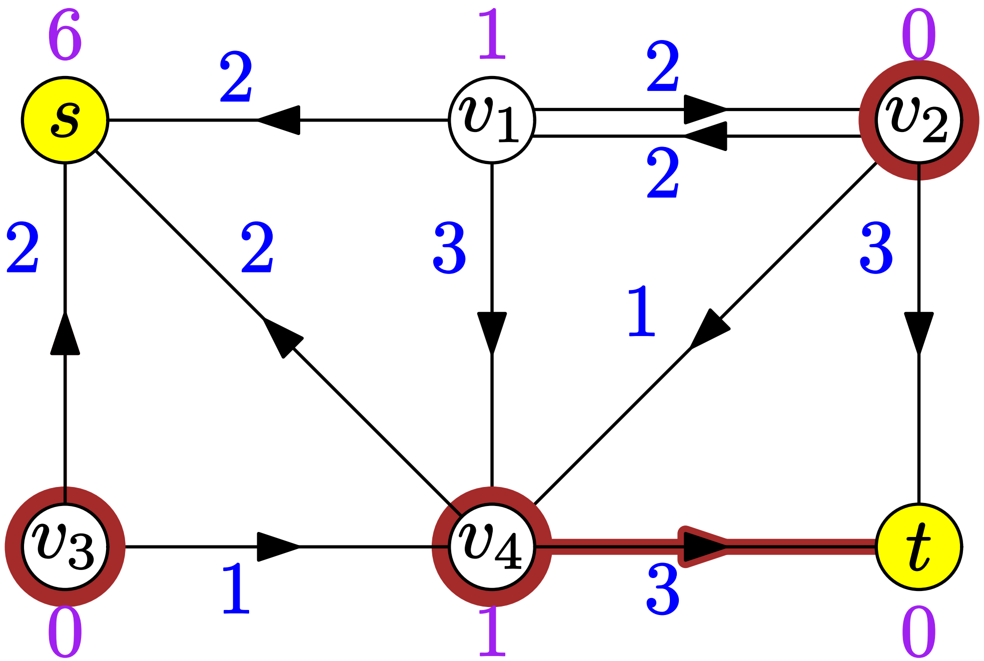
Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_4 v_2)$



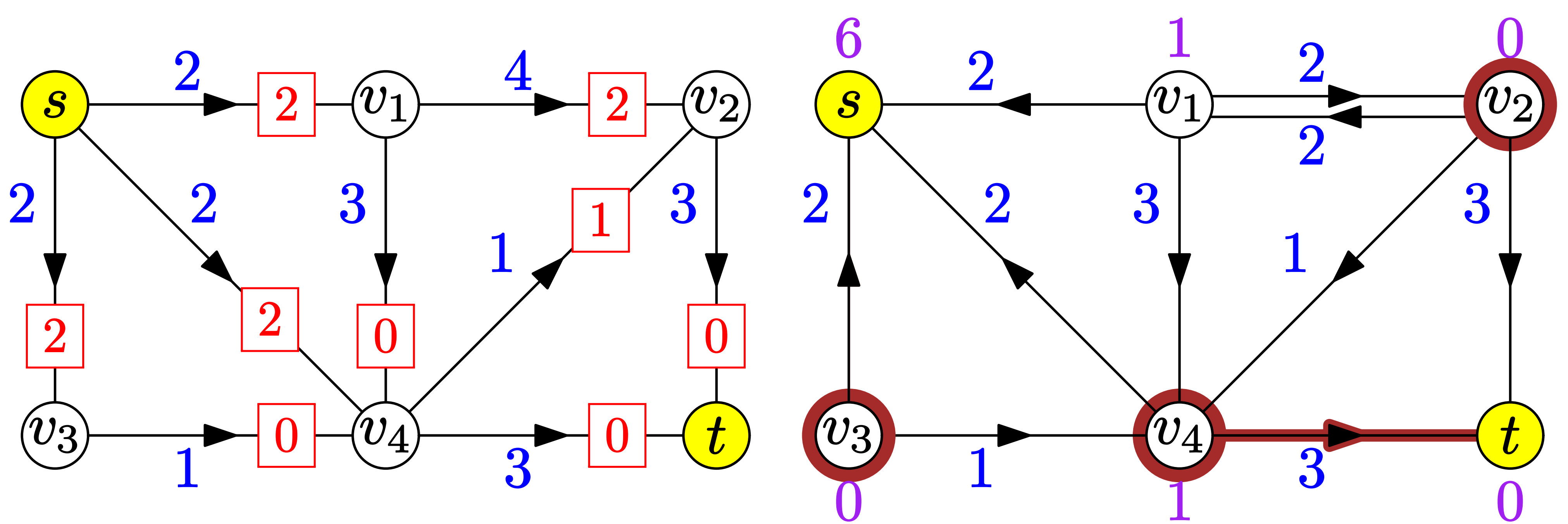
飽和 Push
(saturated push)



Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push($v_4 t$)

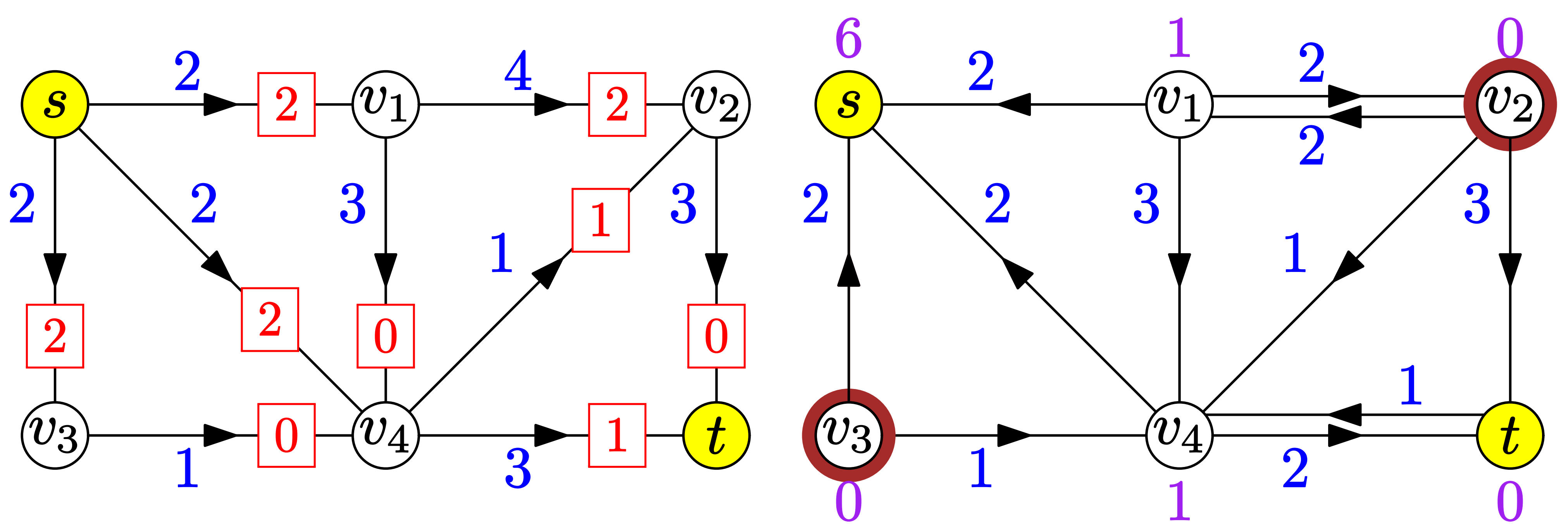


Push-Relabel 法：例 (5)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push($v_4 t$)

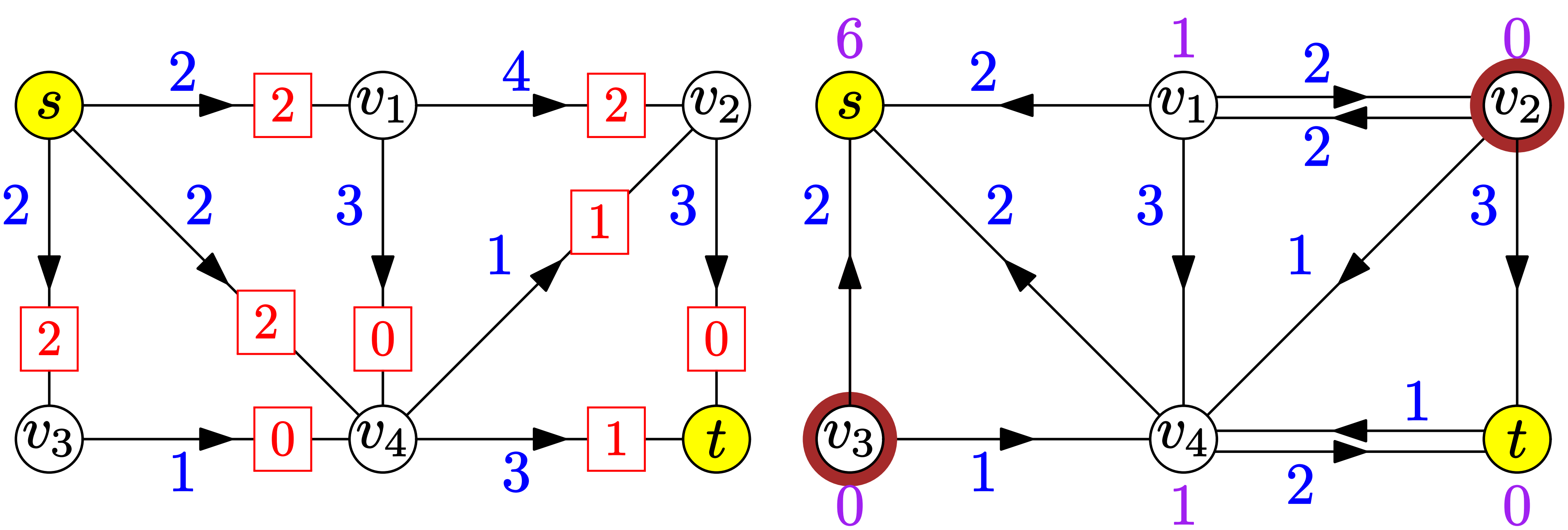


Push-Relabel 法：例 (6)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Relabel(v_3)

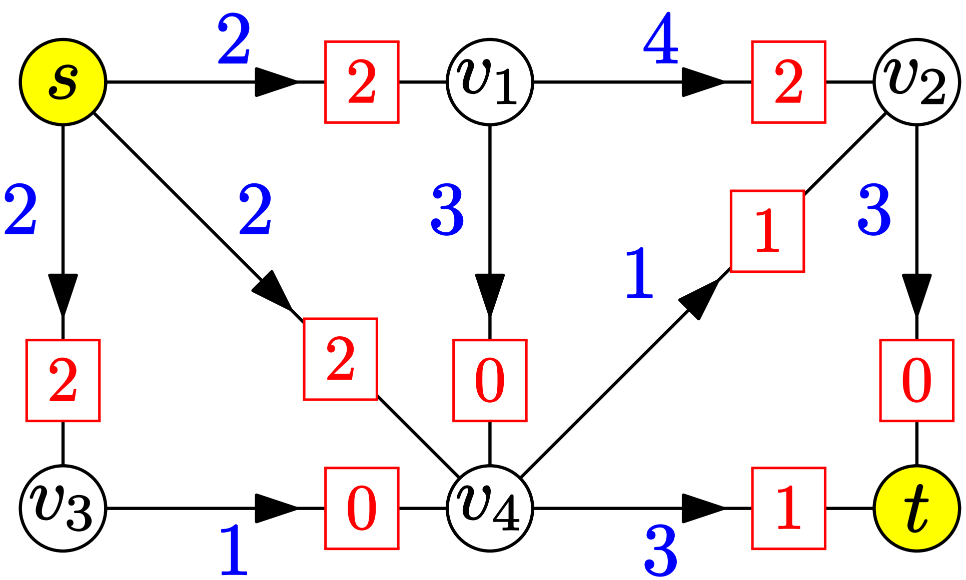


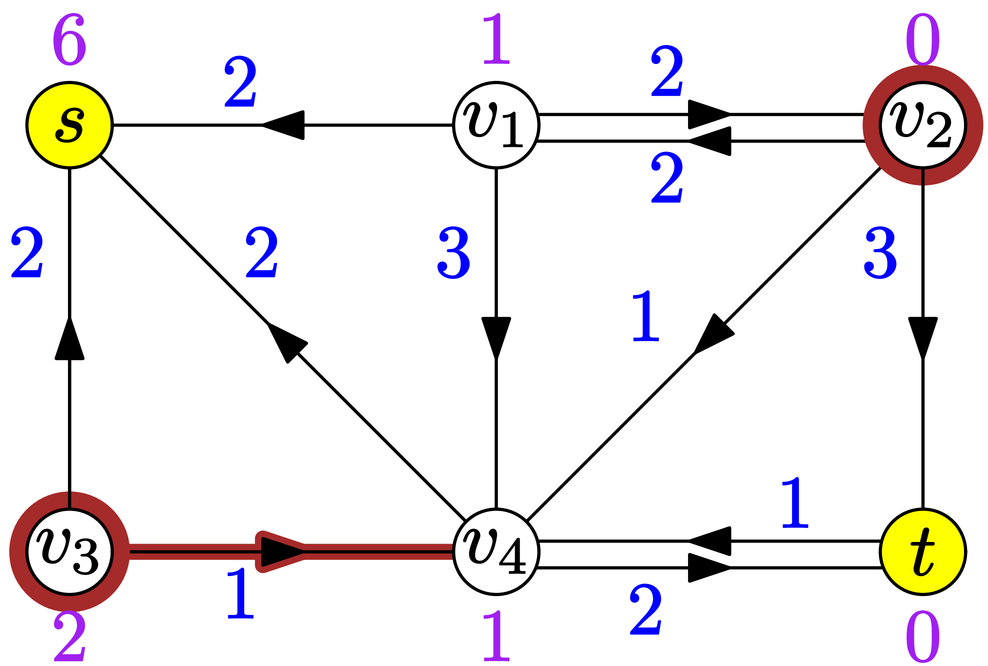
Push-Relabel 法：例 (6)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Relabel(v_3)

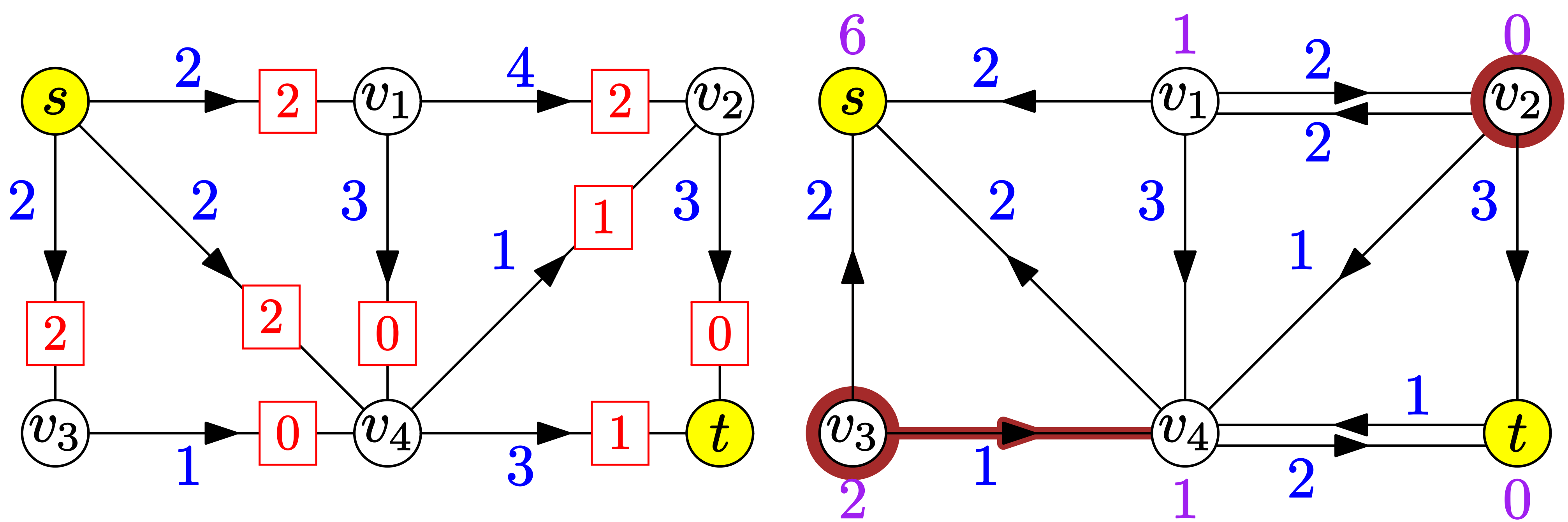




Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_3 v_4)$

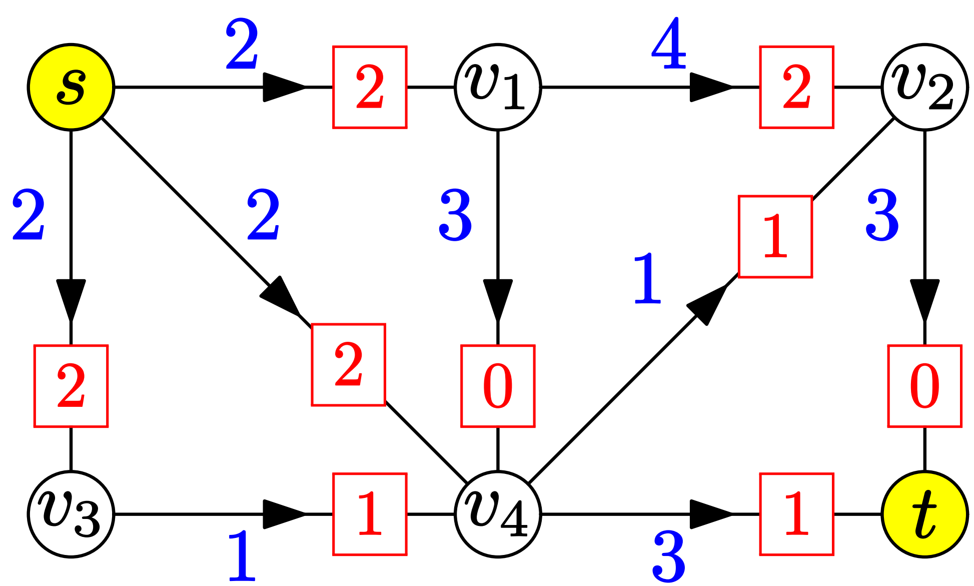


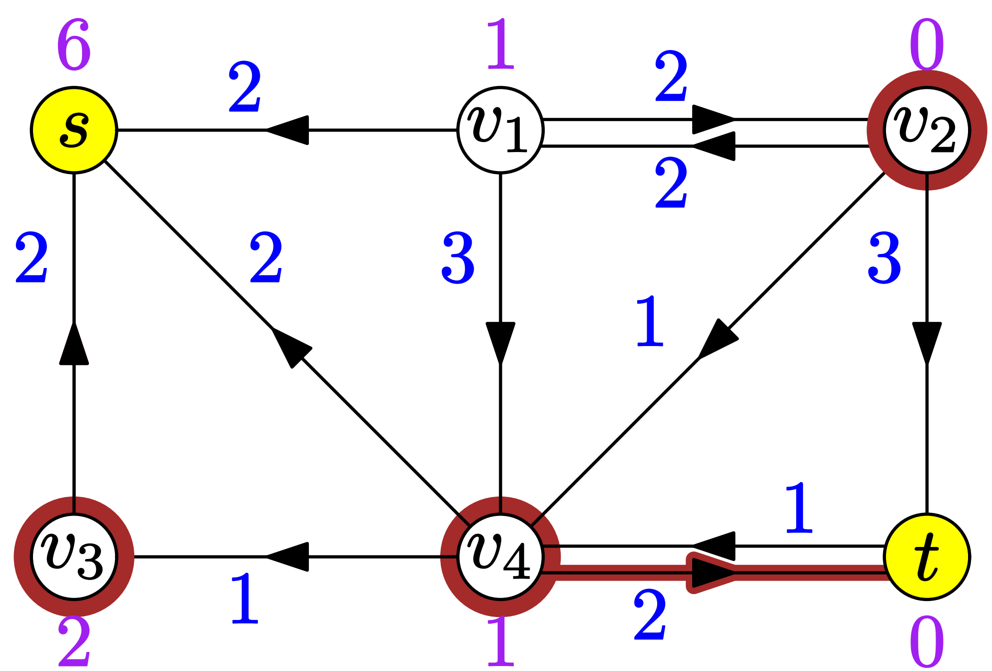
Push-Relabel 法：例 (7)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push(v_3v_4)



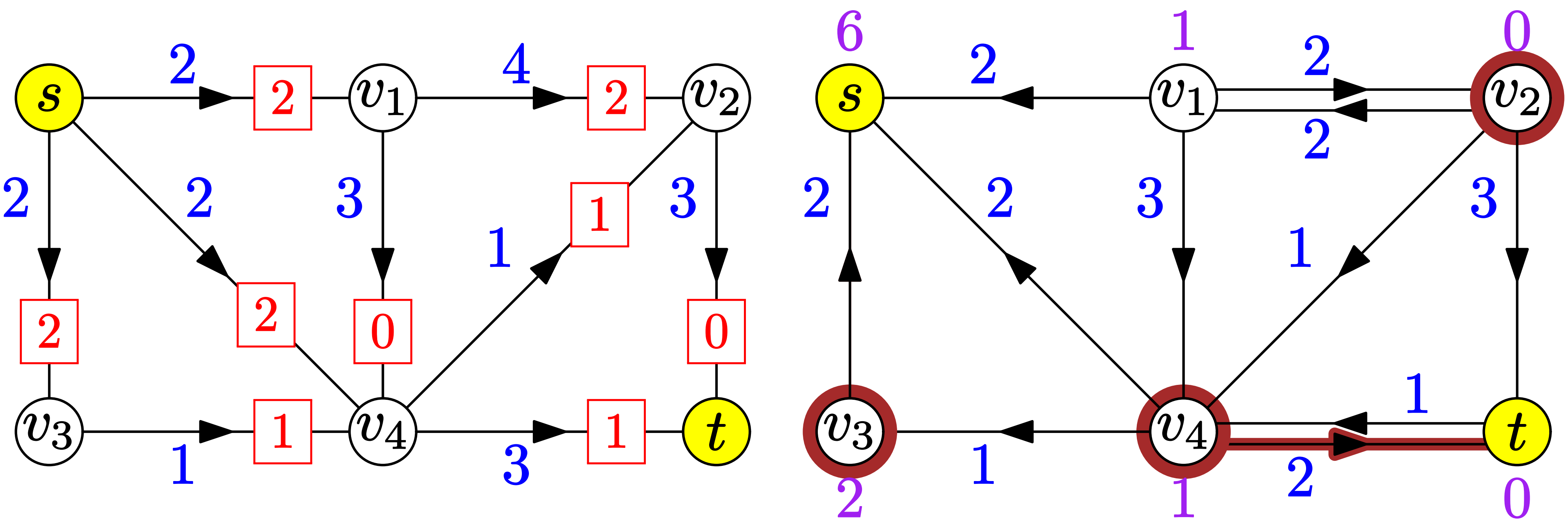


Push-Relabel 法：例 (8)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

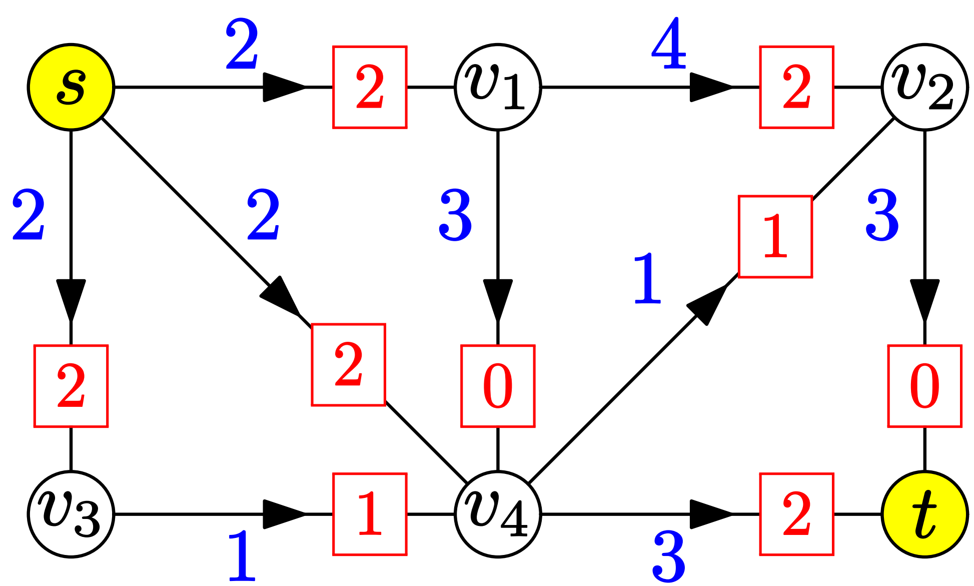
Push$(v_4 t)$

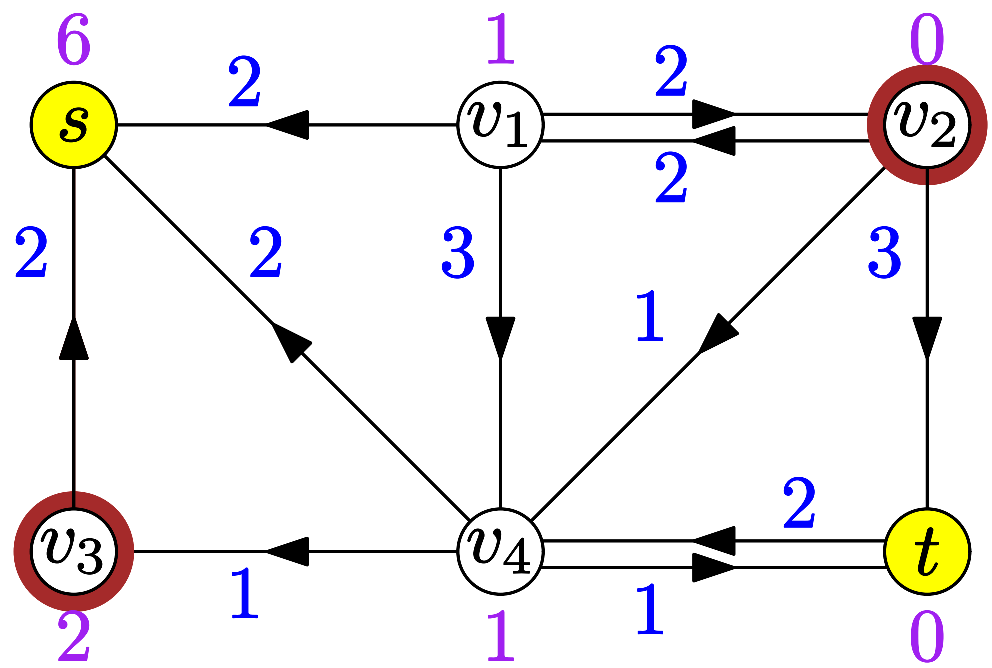


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_4 t)$

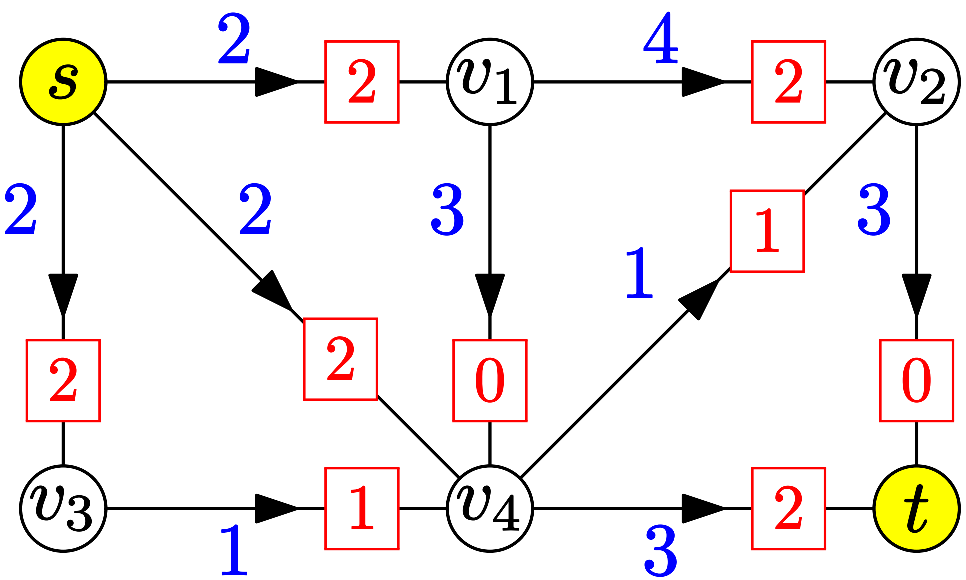


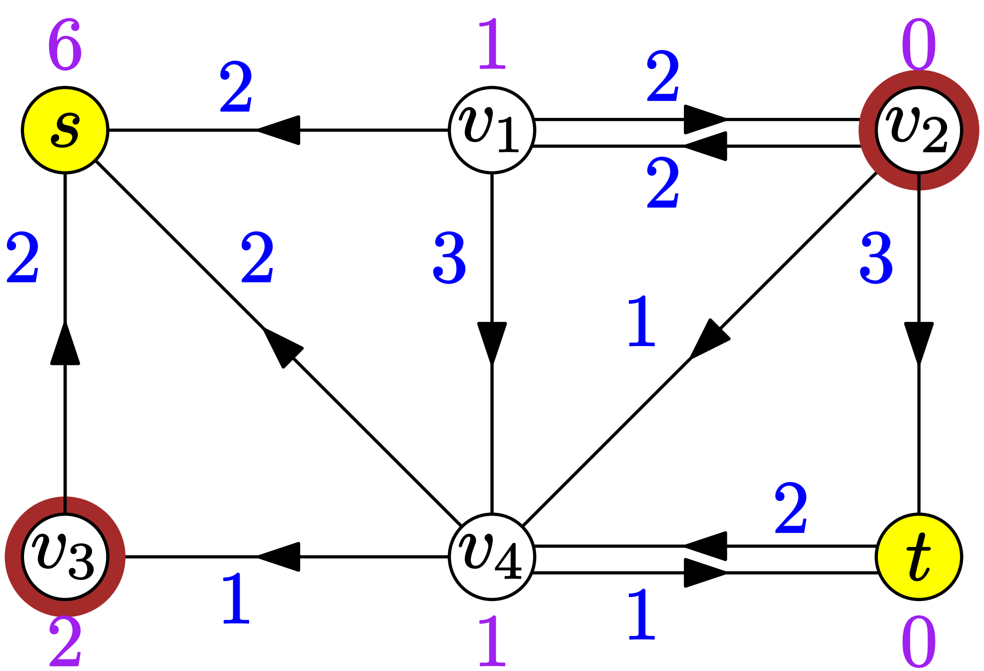


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Relabel(v_3)

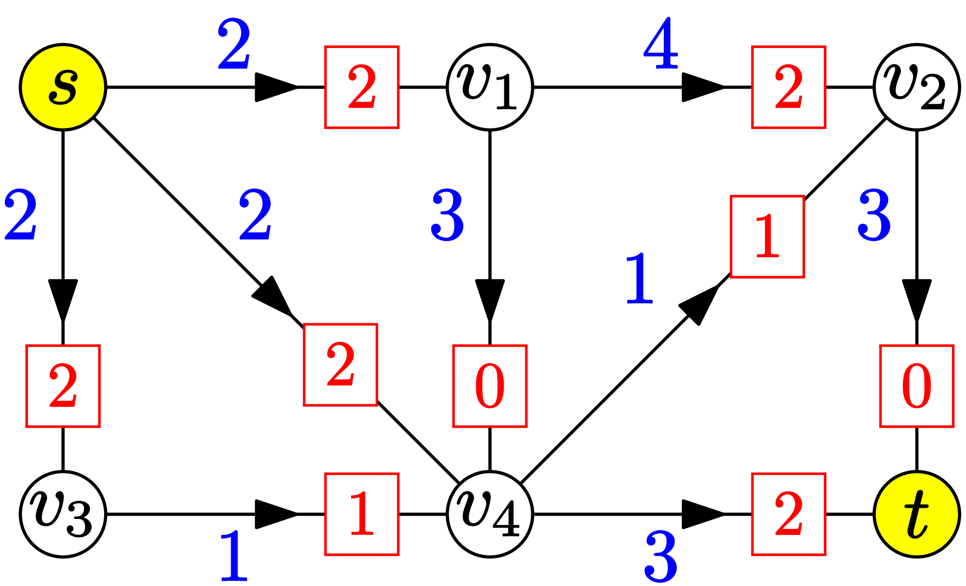


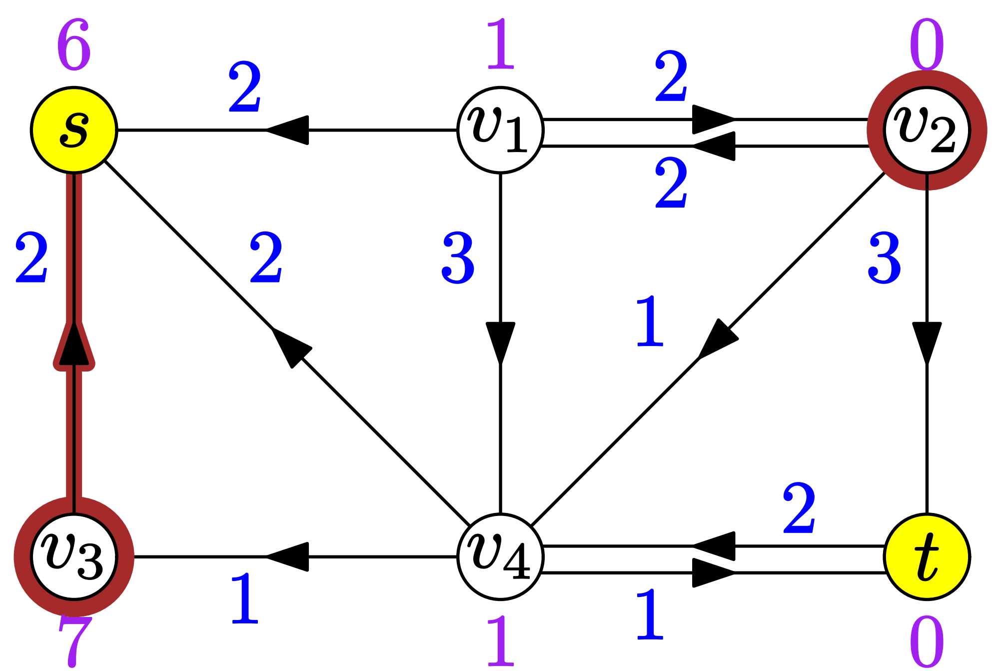


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Relabel(v_3)



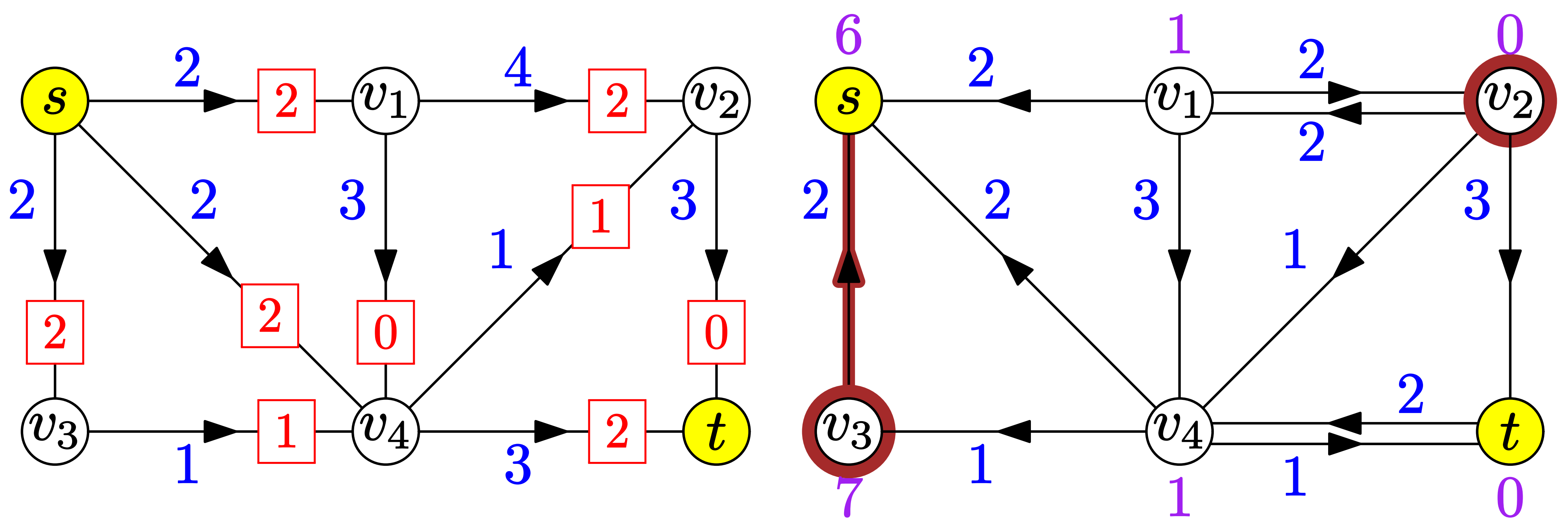


Push-Relabel 法：例 (10)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

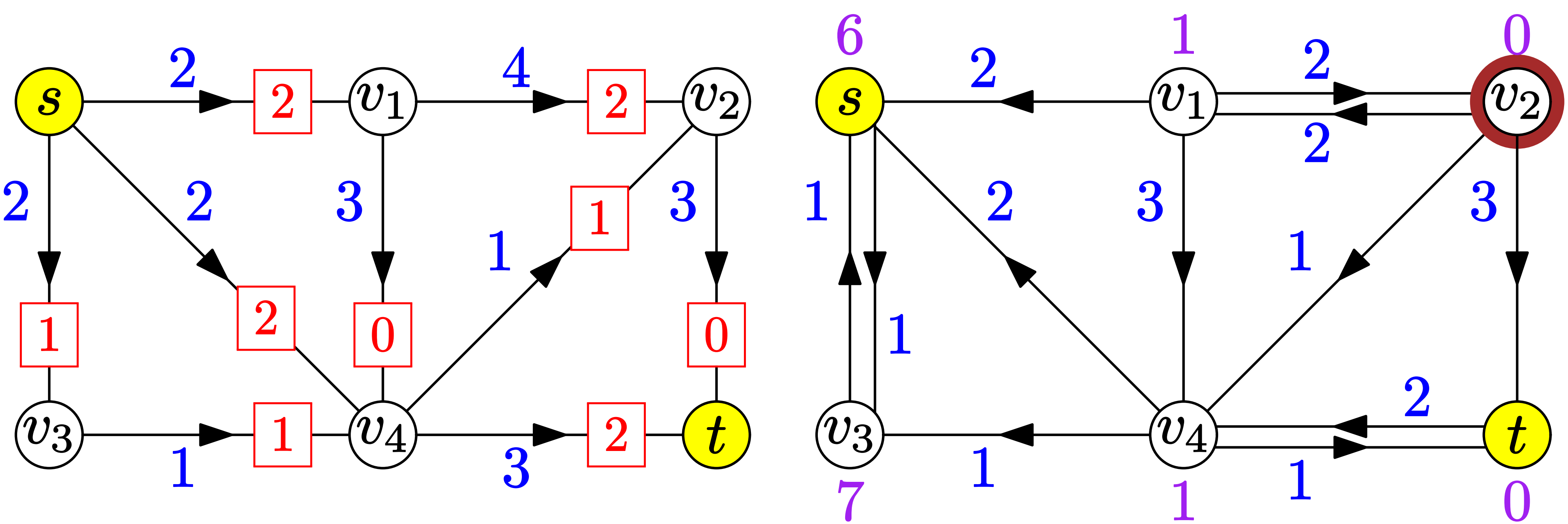
Push$(v_3 s)$



Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_3 s)$

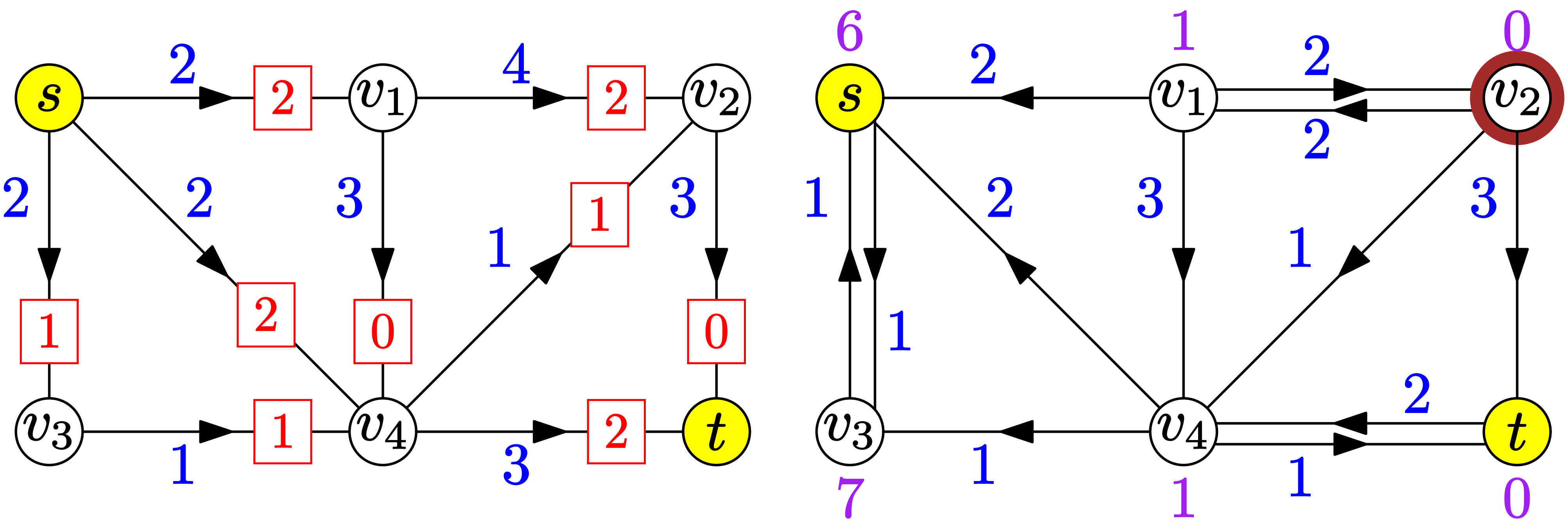


Push-Relabel 法：例 (11)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Relabel(v_2)

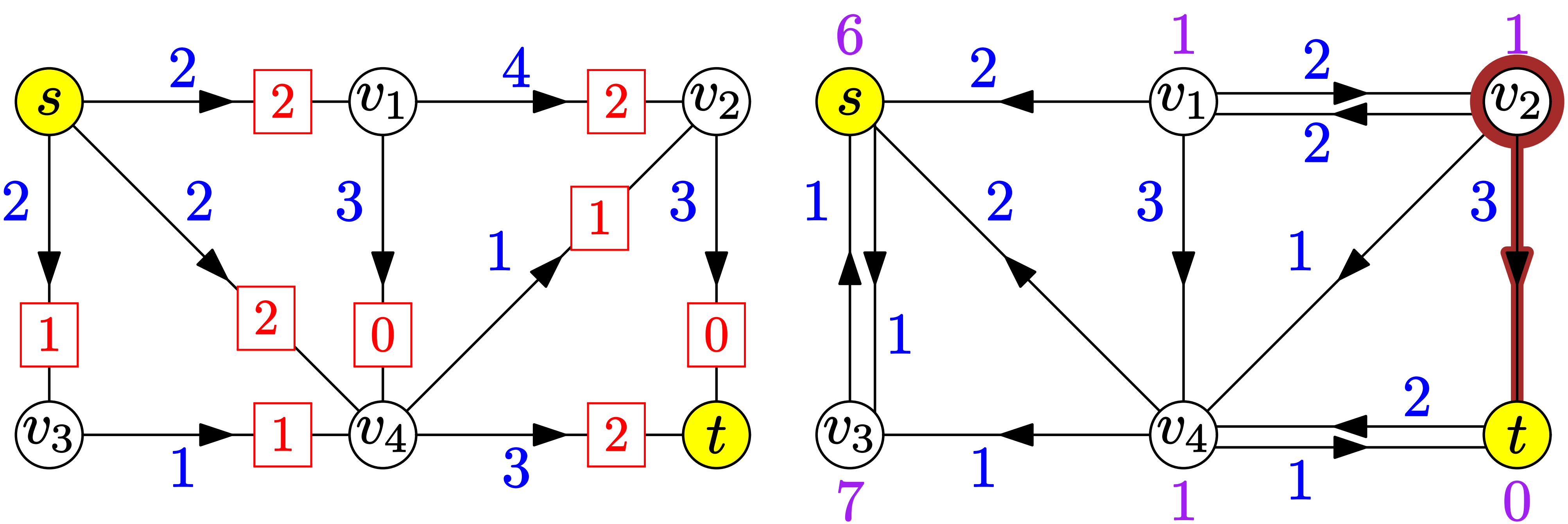


Push-Relabel 法：例 (11)

Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

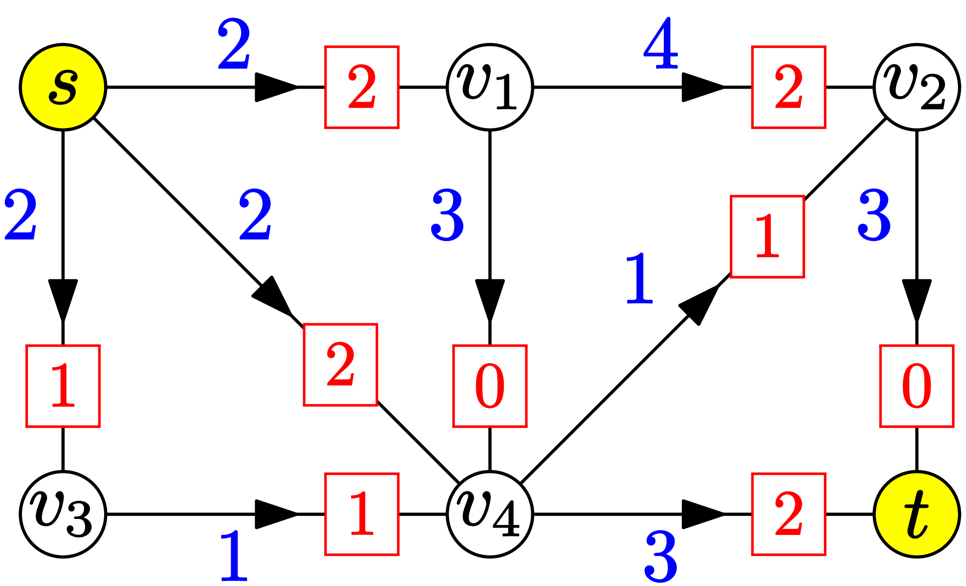
Relabel(v_2)

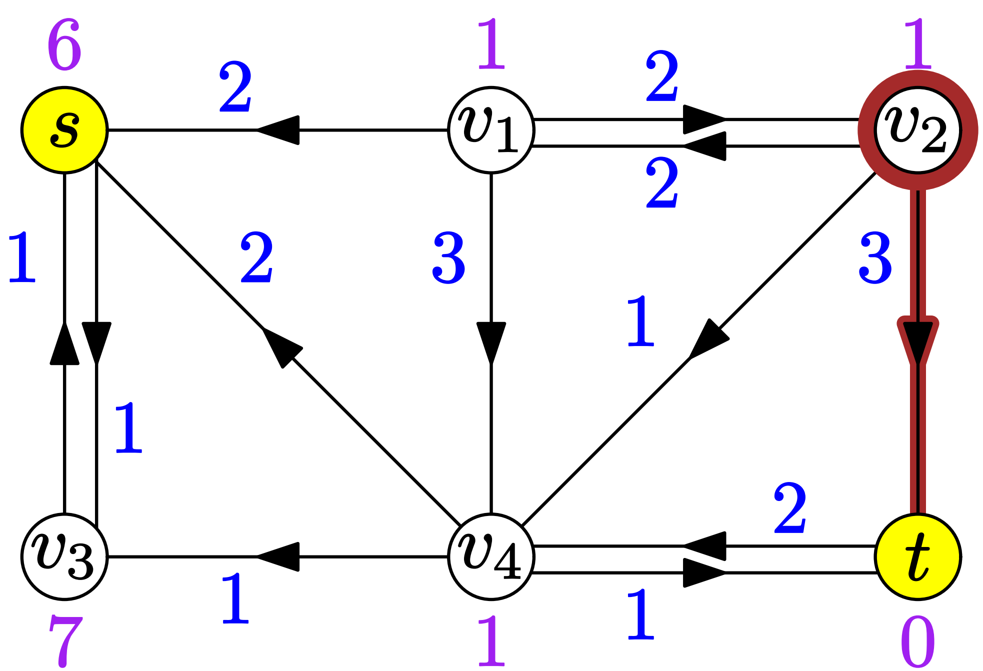


Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_2 t)$





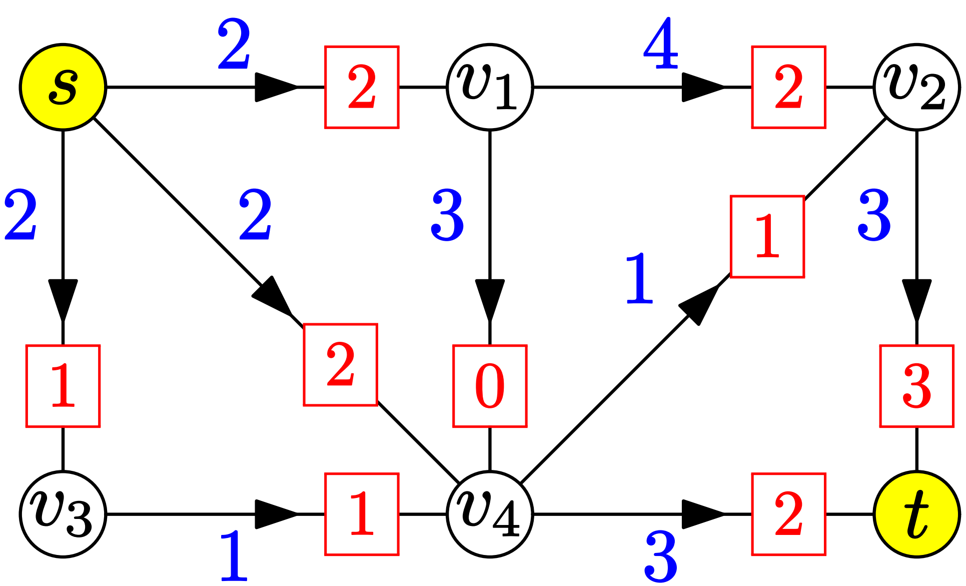
Push-Relabel 法：例 (12)

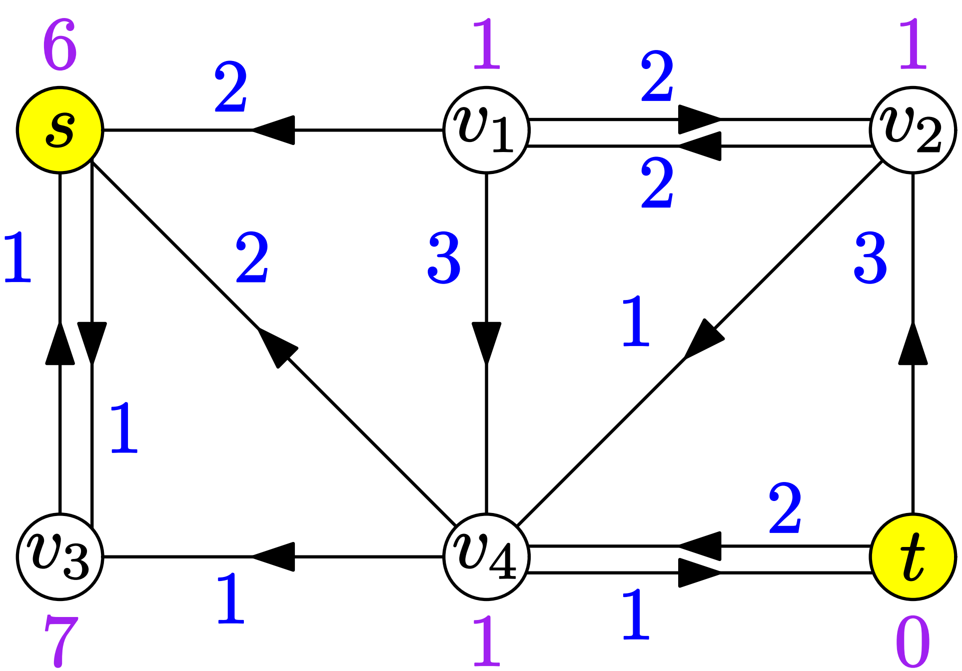
Push-Relabel 法を一言で述べると？

Push 操作と Relabel 操作を繰り返すアルゴリズム

Push$(v_2 t)$

アルゴリズムは停止





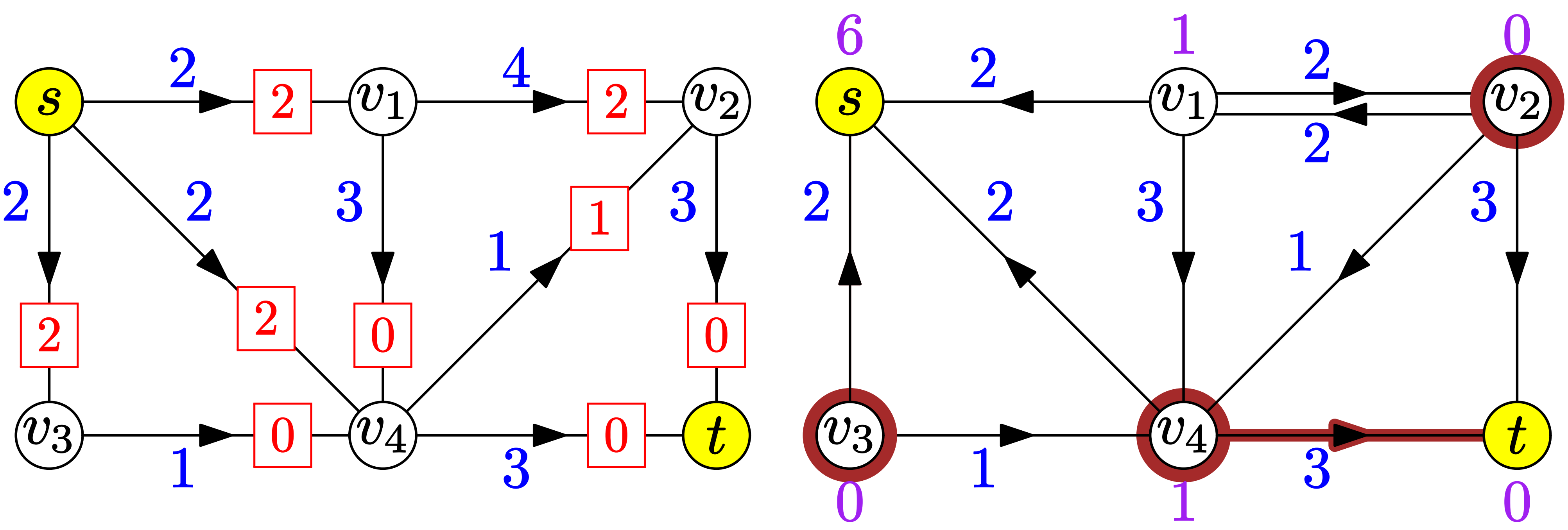
(Goldberg, Tarjan '88)

アルゴリズム：Push-Relabel 法

1. s-t 前流 f，距離ラベル d を初期化
2. f が s-t 流になるまで反復
   (a) Push できる弧 $a \in A_f$ があれば，Push(a) を行う
   (b) Push できる弧がなければ，
   Relabel できる頂点 v に対して，Relabel(v) を行う

(Preflow-Push 法とも呼ばれる)

1. 増加道法と線形計画法
2. 前流と距離ラベル
3. Push-Relabel 法：概要
4. **Push-Relabel 法：性質**



アルゴリズム：Push-Relabel 法

1. s-t 前流 f，距離ラベル d を初期化
2. f が s-t 流になるまで反復
   (a) Push できる弧 $a \in A_f$ があれば，Push(a) を行う
   (b) Push できる弧がなければ，
   Relabel できる頂点 v に対して，Relabel(v) を行う

確認すべき事項

- 初期化による f が s-t 前流で，d が距離ラベルであること (レポート問題)
- Push 操作と Relabel 操作によって，
  f が s-t 前流，d が距離ラベルであるという性質が保たれること

アルゴリズム：Push-Relabel 法

1. s-t 前流 f，距離ラベル d を初期化
2. f が s-t 流になるまで反復
   (a) Push できる弧 $a \in A_f$ があれば，Push(a) を行う
   (b) Push できる弧がなければ，
   Relabel できる頂点 v に対して，Relabel(v) を行う

性質：Push-Relabel 法と最大 s-t 流

Push-Relabel 法が停止する ⇒
出力 f は (G, u) に対する最大 s-t 流

<u>証明</u>：f が s-t 前流 ⇒ G_f は増加道を持たない

$\therefore$ 出力は増加道を持たない s-t 流（つまり，最大 s-t 流） $\square$

まとめ

Push-Relabel 法

双対許容性 を満たしたまま **主許容性** を満たしに行く

相補性

$$
\begin{array}{llcl}
\text{(双対許容解)} & \text{距離ラベル } d & \xrightarrow{\text{更新}} & \text{距離ラベル } d \\
\text{(相補性)} & \quad\Big\updownarrow & & \quad\Big\updownarrow \ \text{(相補性)} \\
\text{(主非許容解)} & s\text{-}t \text{ 前流 } f & \xrightarrow{\text{更新}} & s\text{-}t \text{ 流 } f \quad \text{(主許容解)}
\end{array}
$$

増加道法

主許容性 を満たしたまま **双対許容性** を満たしに行く

相補性

次回予告

次回の予告

Push-Relabel 法の計算量評価

- Relabel 操作の回数
- Push 操作の回数

	計算量	分類	注意
増加道法	$O(m^2 U)$	擬多項式時間	整数のみ
容量スケーリング	$O(m^2 \log U)$	弱多項式時間	整数のみ
Edmonds-Karp	$O(nm^2)$	強多項式時間	実数 OK
Push-Relabel	$O(n^2 m)$	強多項式時間	実数 OK

残した証明

Relabel と距離ラベル

設定：f に関する距離ラベル $d \xrightarrow{\text{Relabel}(u)} d'$

性質：Relabel と距離ラベル

d' も f に関する距離ラベル

d が距離ラベル：

$d(s) = n,\ \ d(t) = 0,$

$d(u) \leq d(v) + 1 \ \ \forall\, uv \in A_f$

Relabel(u)：

$d'(u) = \min_{uv \in A_f} d(v) + 1$

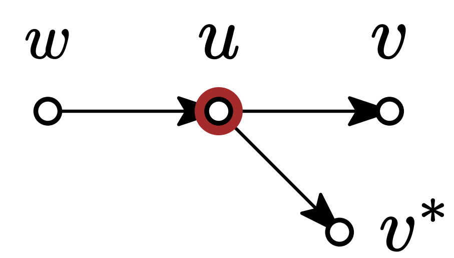
設定：f に関する距離ラベル $d \xrightarrow{\text{Relabel}(u)} d'$

性質：Relabel と距離ラベル

d' も f に関する距離ラベル

証明：$uv^* \in A_f$ で，$d'(u) = d(v^*) + 1$ となる v^* を取る

- $uv \in A_f$ のとき
  $$d'(u) = d(v^*) + 1 \le d(v) + 1$$
- $wu \in A_f$ のとき
  $$d(w) \le d(u) + 1 \le d(v^*) + 2 = d'(u) + 1$$



□

d が距離ラベル：

$$d(s) = n, \quad d(t) = 0,$$
$$d(u) \le d(v) + 1 \quad \forall\, uv \in A_f$$

Relabel(u)：

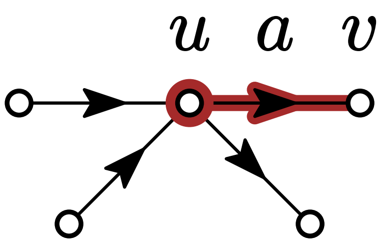
$$d'(u) = \min_{uv \in A_f} d(v) + 1$$

設定：s-t 前流 $f \xrightarrow{\text{Push}(a)} f'$

性質：Push と s-t 前流

f' は s-t 前流

証明：$a = uv$ とする



- $f'(a) \leq f(a) + (u(a) - f(a)) = u(a)$
- $f'(a) \geq f(a) \geq 0$

- $f'^-(u) - f'^+(u) \geq f^-(u) - (f^+(u) + (f^-(u) - f^+(u))) = 0$
- $f'^-(v) - f'^+(v) \geq f^-(v) - f^+(v) \geq 0$ □

Push(a)：$f'(a) = f(a) + \min\{u(a) - f(a), f^-(u) - f^+(u)\}$

Push と距離ラベル

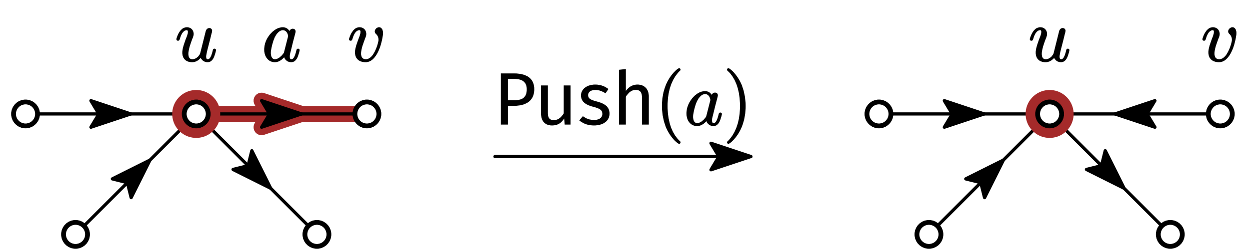
設定：s-t 前流 f $\xrightarrow{\text{Push}(a)}$ f'

性質：Push と距離ラベル

d は f' に関する距離ラベル

証明：$a = uv$ とする

- Push(a) によって追加されうる弧は vu のみ
- Push(a) を行う条件から，$d(u) = d(v) + 1$
- $\therefore d(v) = d(u) - 1 \leq d(u) + 1$ □



d が距離ラベル：$d(s) = n$，$d(t) = 0$，$d(u) \leq d(v) + 1 \; \forall \, uv \in A_f$