離散最適化基礎論

第 8 回

最大流問題：Push-Relabel 法 (計算量評価)

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2023 年 11 月 28 日

最終更新：2023 年 11 月 28 日　10:33

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)

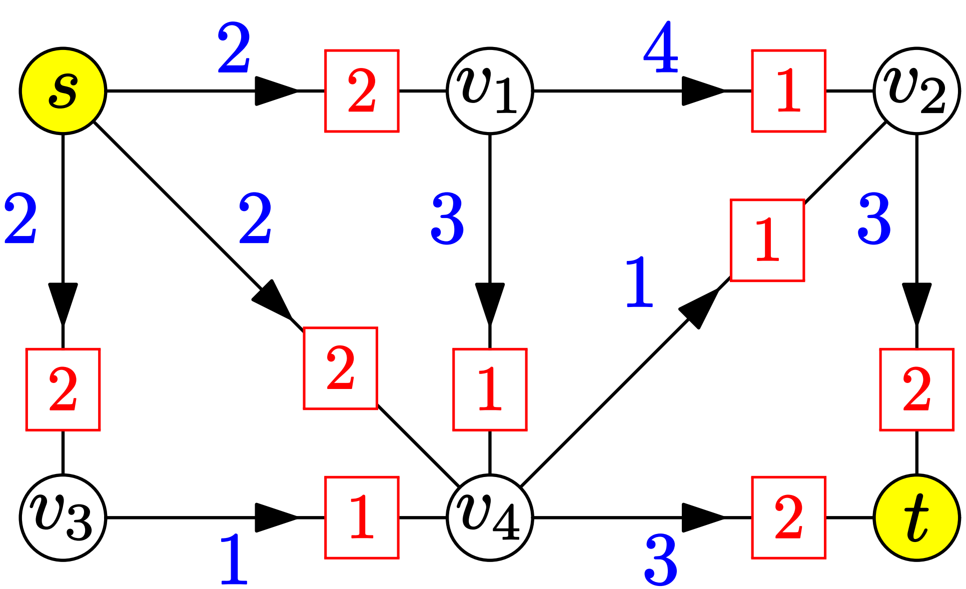
前流：定義

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，$f^-(v)\ge f^+(v)$



s-t 流なら

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

$\therefore$ s-t 流は s-t 前流

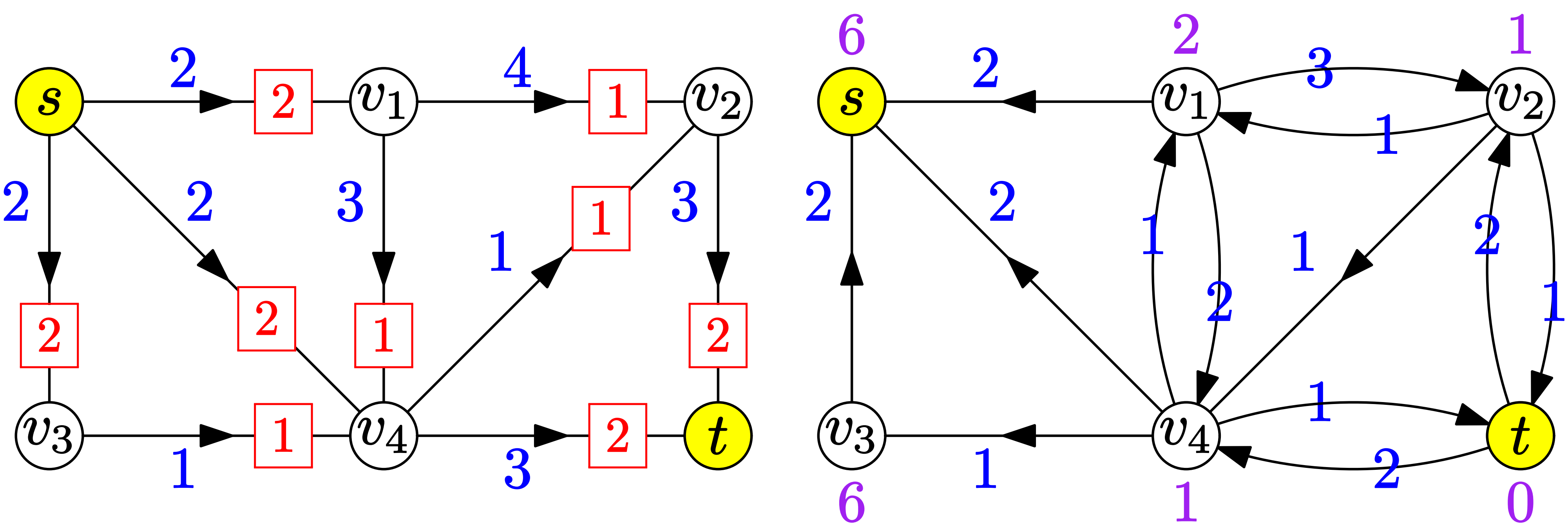
距離ラベル

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，$d(u) \leq d(v) + 1$



Push-Relabel 法の考え方

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

より詳細に述べると

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

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

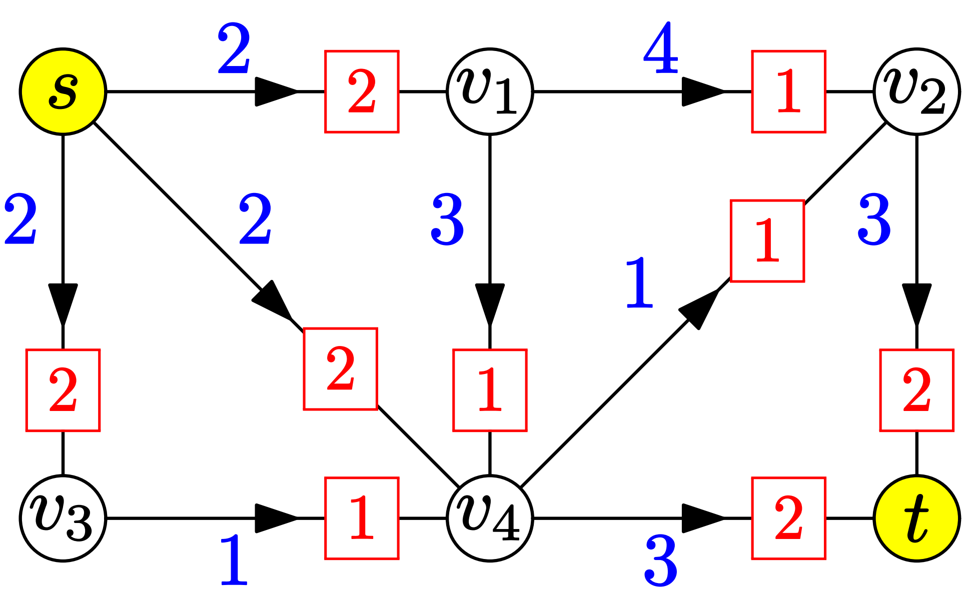
Push 操作 (1)

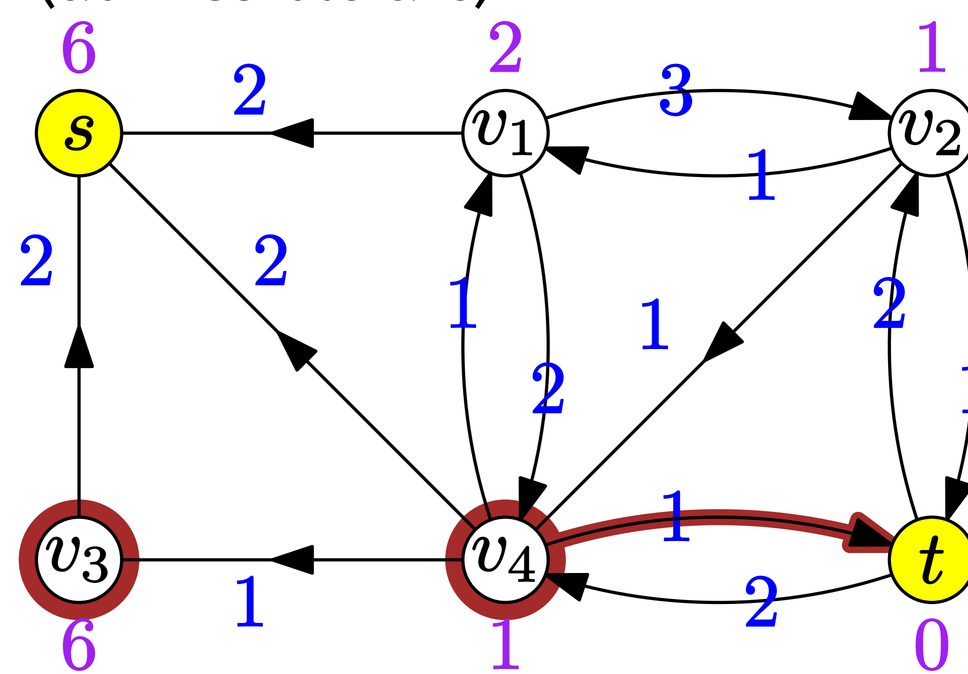
補助ネットワーク G_f の弧 $a = (u, v)$

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)

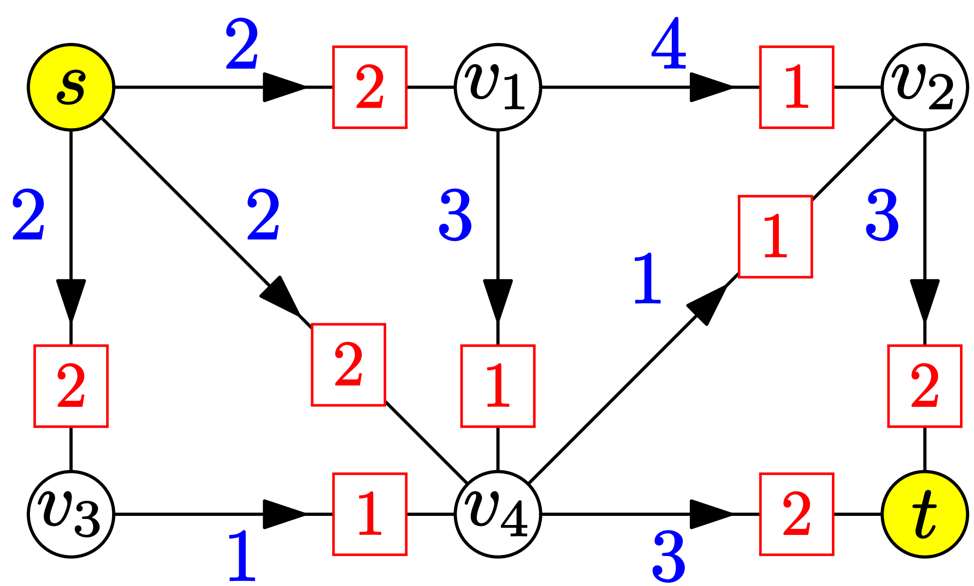
弧 $a = (u, v)$

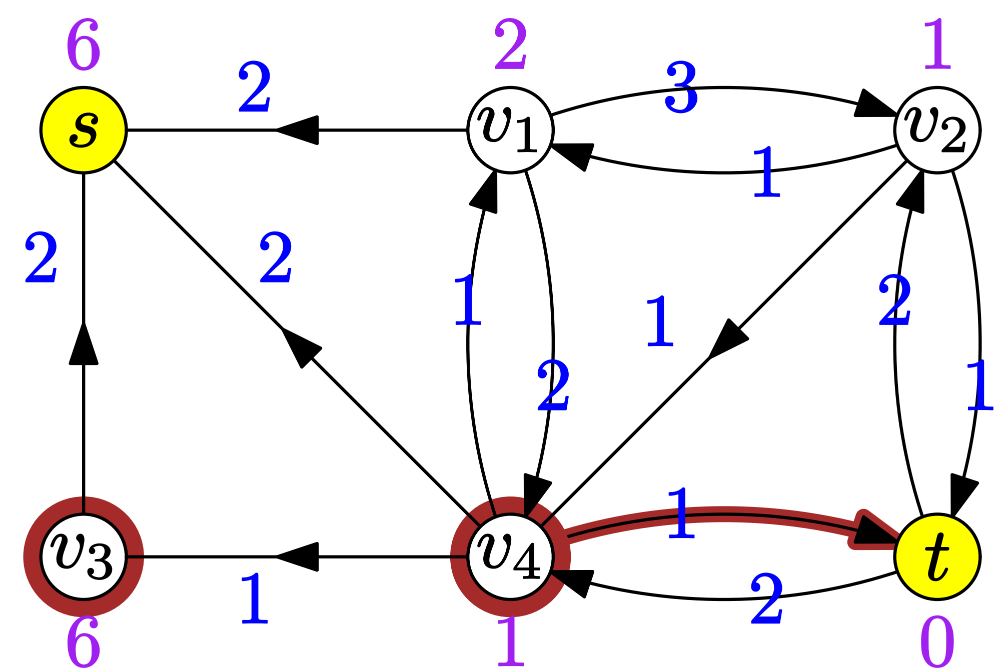
Push(a)

$f(a)$ 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

飽和 Push　　非飽和 Push





Push 操作 (2)

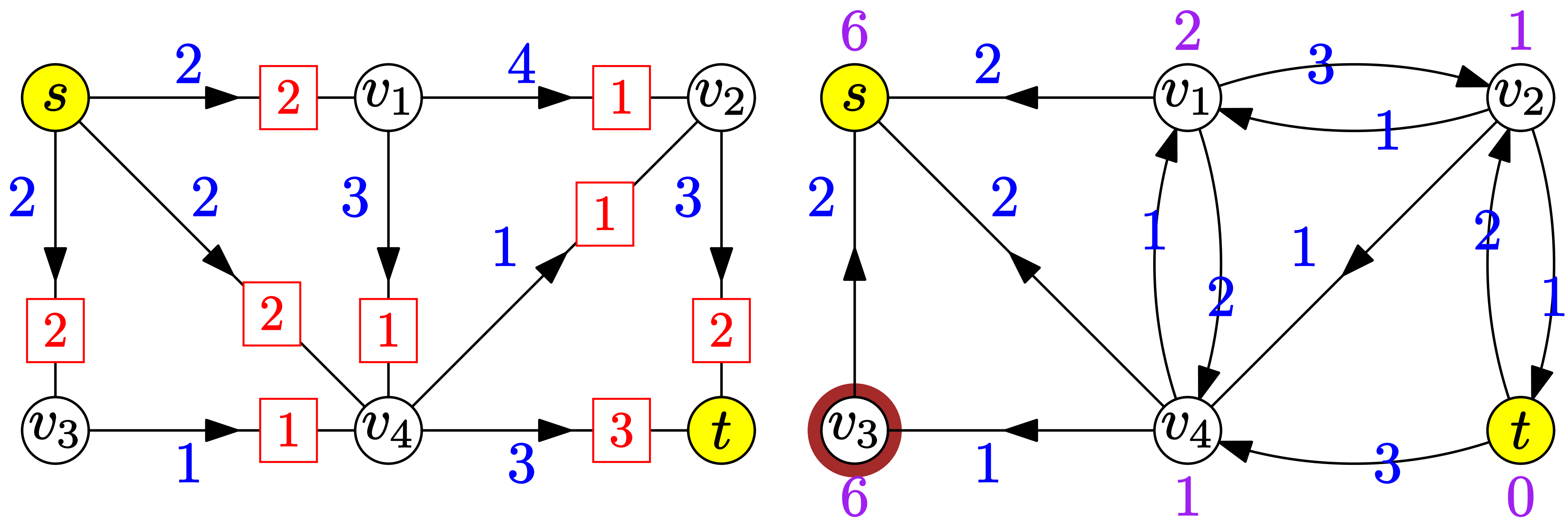
弧 $a = (u, v)$

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

飽和 Push　非飽和 Push



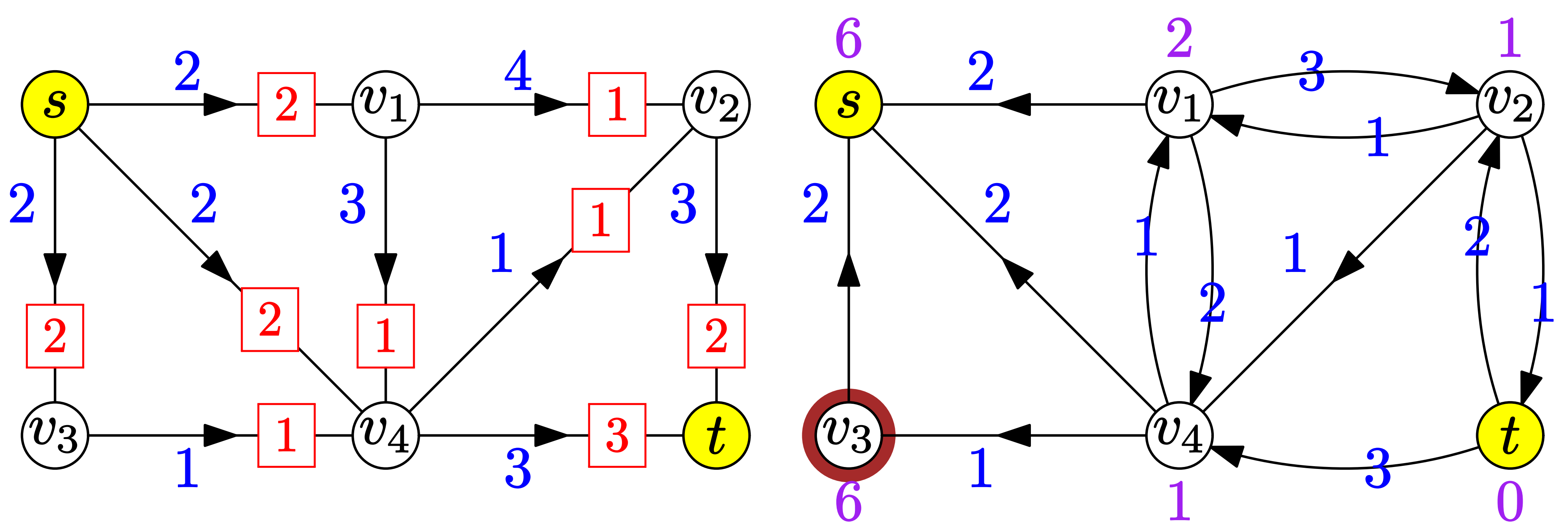
Relabel 操作 (1)

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 $d(u) \leq d(v)$

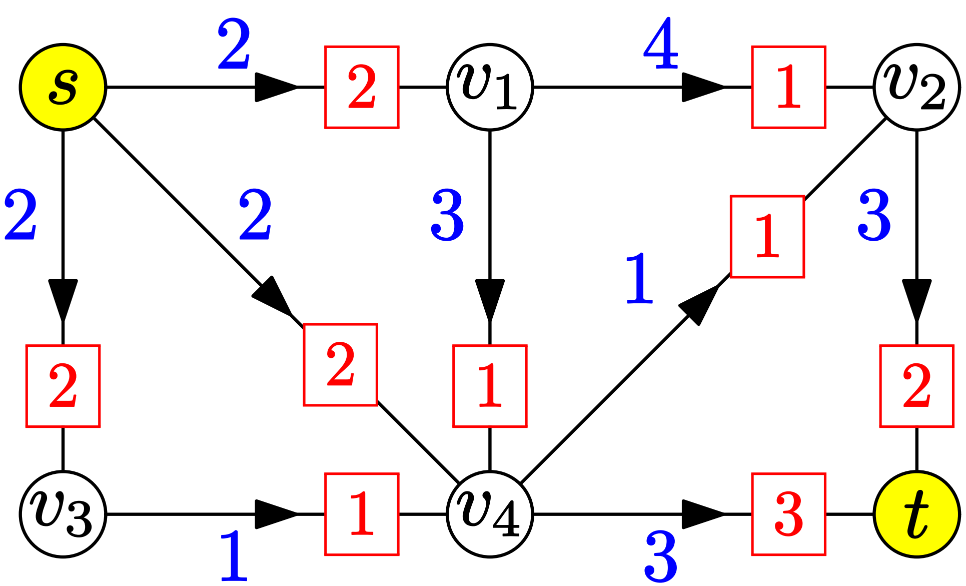


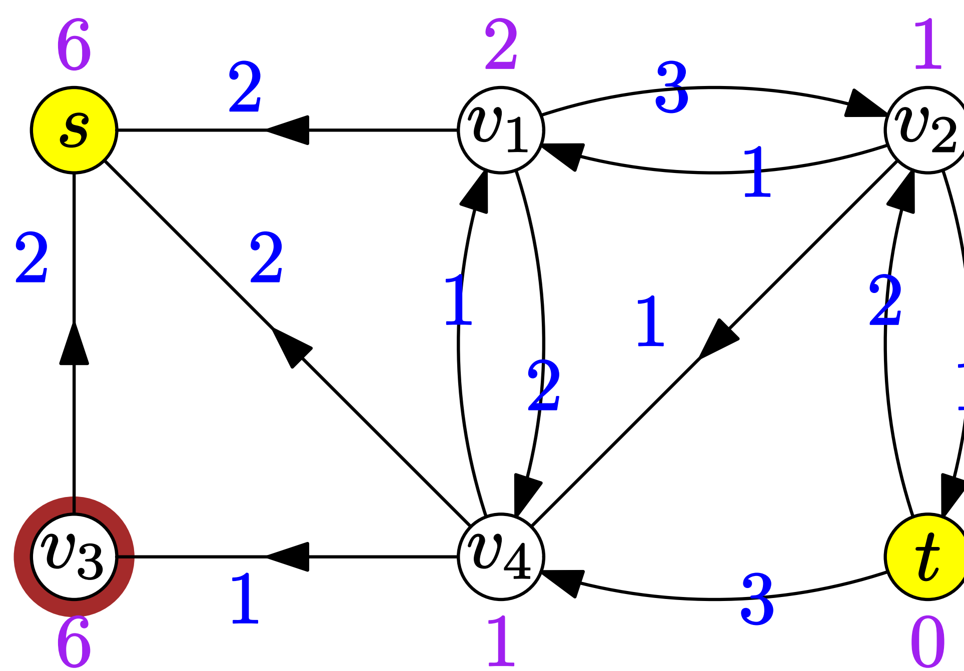
Relabel 操作 (2)

頂点 $u \in V - \{s, t\}$

Relabel(u)

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





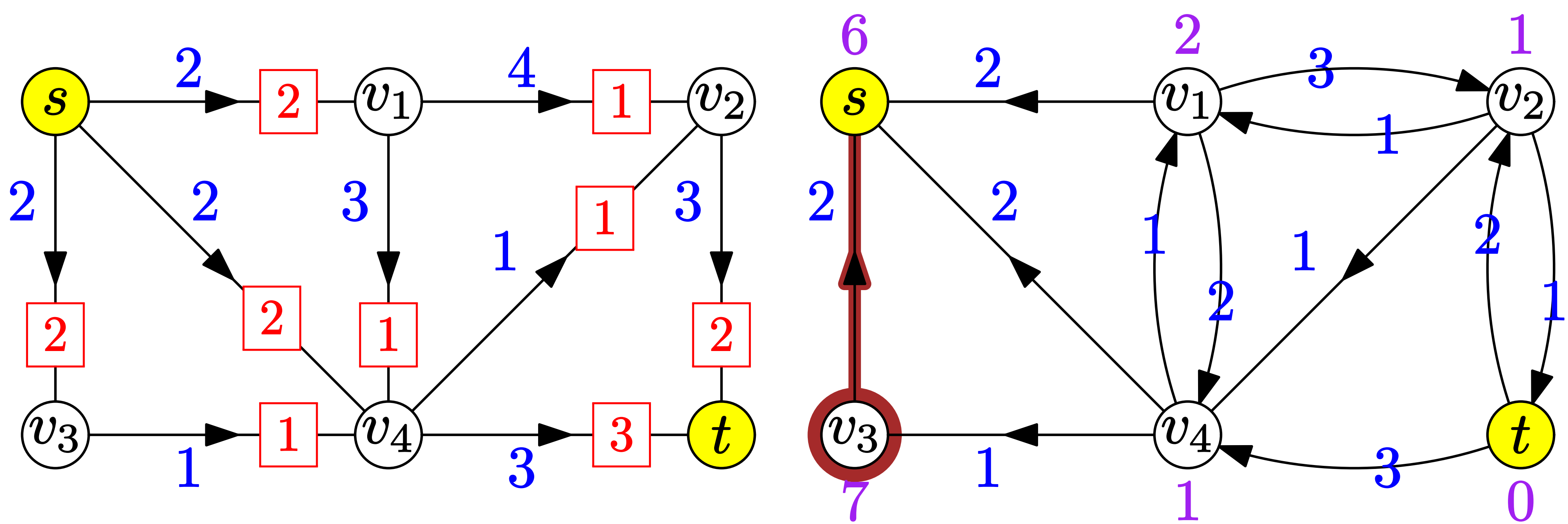
Relabel 操作 (2)

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



Push-Relabel 法

(Preflow-Push 法とも呼ばれる)　(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
3. f を出力

前回の結論：

Push-Relabel 法が停止する $\Rightarrow$ 出力 f は最大 s-t 流

今日の目標

Push-Relabel 法の計算量評価

まとめ

	回数	1 回あたりの計算量
Relabel(u)	$O(n)$	$O(\lvert\delta^{\pm}(u)\rvert)$
飽和 Push	$O(nm)$	$O(1)$
非飽和 Push	$O(n^2m)$	$O(1)$

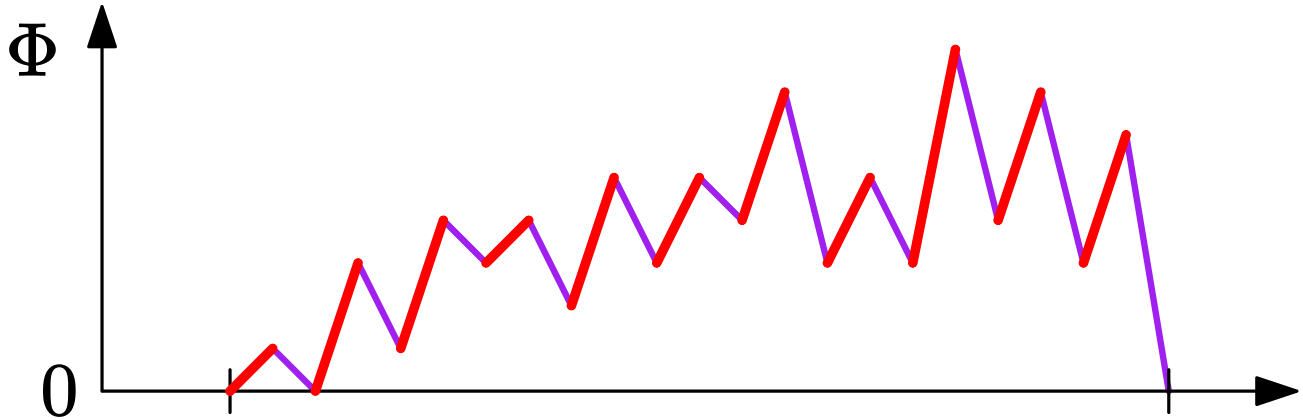
$n = |V|$ (頂点数)

$m = |A|$ (弧数)

$\Rightarrow$ 全体の計算量 $O(n^2m)$

本日の内容

1. **Relabel 操作**
2. 飽和 Push 操作
3. 非飽和 Push 操作
4. 計算量評価



Relabel 操作の回数

頂点 $u \in V - \{s, t\}$

性質：Relabel の回数

Relabel(u) を実行する回数 $= O(n)$

Relabel(u) が行える条件 $\Rightarrow uv \in A_f$ ならば $d(u) \leq d(v)$

Relabel(u) : $d(u) \leftarrow \min\{d(v) \mid uv \in A_f\} + 1$

Relabel 操作の回数

頂点 $u \in V - \{s, t\}$

性質：Relabel の回数

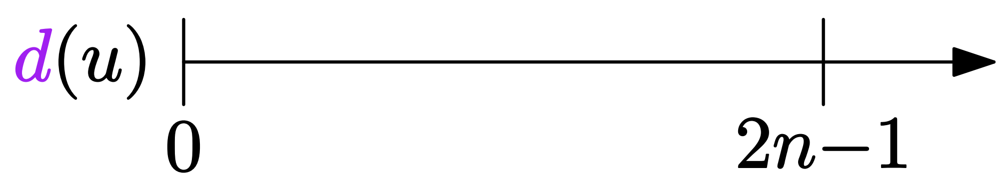
Relabel(u) を実行する回数 $= O(n)$

Relabel(u) が行える条件 $\Rightarrow$ $uv \in A_f$ ならば $d(u) \leq d(v)$

Relabel(u) : $d(u) \leftarrow \min\{d(v) \mid uv \in A_f\} + 1$

$\therefore$ Relabel(u) を行うと，$d(u)$ が 1 以上増加

$d(u)$ の初期値 ≥ 0



よって，次を証明すれば十分

$d(u) \leq 2n - 1$

頂点ラベルの上界

よって，次を証明すれば十分

$d(u) \leq 2n - 1$

Relabel(u) が行える条件 $\Rightarrow f^-(u) - f^+(u) > 0$

頂点ラベルの上界

よって，次を証明すれば十分

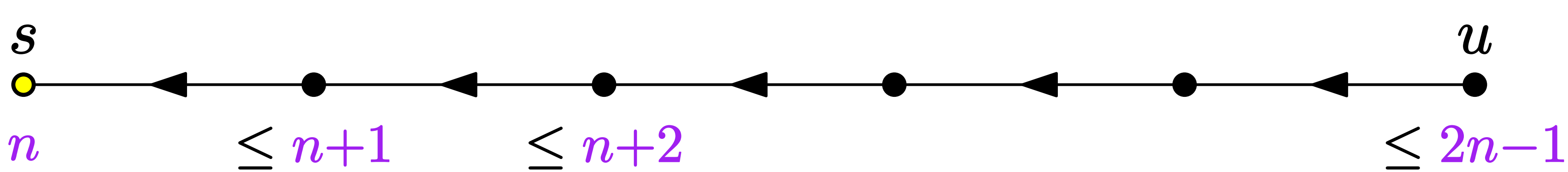
$d(u) \leq 2n - 1$

Relabel(u) が行える条件 $\Rightarrow$ $f^-(u) - f^+(u) > 0$

よって，次を証明すれば十分

$f^-(u) - f^+(u) > 0 \Rightarrow G_f$ において，u から s に到達可能

なぜか？



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

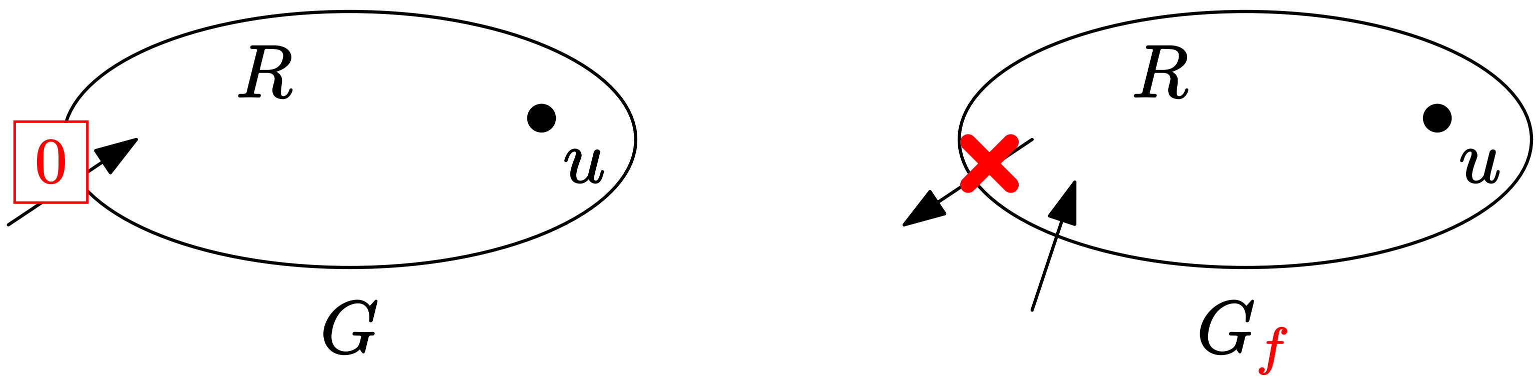
残っている頂点から s に到達可能 (1)

s-t 前流 f

よって，次を証明すれば十分

$f^-(u) - f^+(u) > 0 \Rightarrow G_f$ において，u から s に到達可能

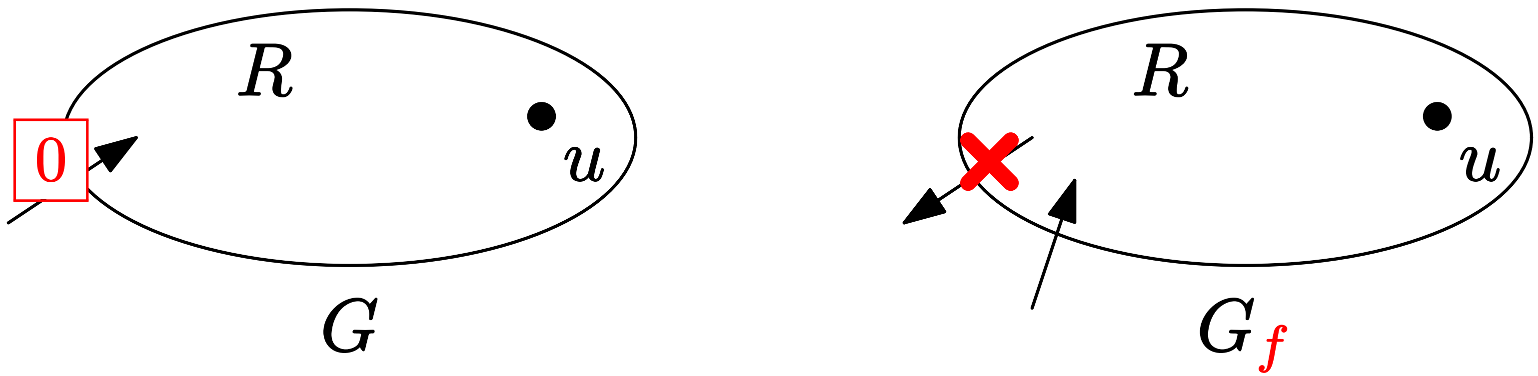
証明：$R = \{G_f$ において u から到達可能な頂点 $\}$ とする



$$\sum_{v \in R} (f^-(v) - f^+(v)) = \sum_{a \in \delta^+_G(\overline{R})} f(a) - \sum_{a \in \delta^+_G(R)} f(a) \leq 0$$

残っている頂点から s に到達可能 (2)

証明：$R = \{G_f$ において u から到達可能な頂点 $\}$ とする



$$\sum_{v \in R}(f^-(v) - f^+(v)) = \sum_{a \in \delta_G^+(\overline{R})} f(a) - \sum_{a \in \delta_G^+(R)} f(a) \leq 0$$

$u \in R$ で，$f^-(u) - f^+(u) > 0$ なので，

ある $v \in R$ において，$f^-(v) - f^+(v) < 0$ である

f は s-t 前流なので，そのような v は s に限る □

Relabel 操作の回数：まとめ

頂点 $u \in V - \{s, t\}$

性質：Relabel の回数

Relabel(u) を実行する回数 $= O(n)$

Relabel 操作の回数：まとめ

頂点 $u \in V - \{s, t\}$

性質：Relabel の回数

Relabel(u) を実行する回数 $= O(n)$

性質：Relabel の回数 (合算)

Relabel を実行する回数 $= O(n^2)$

証明：

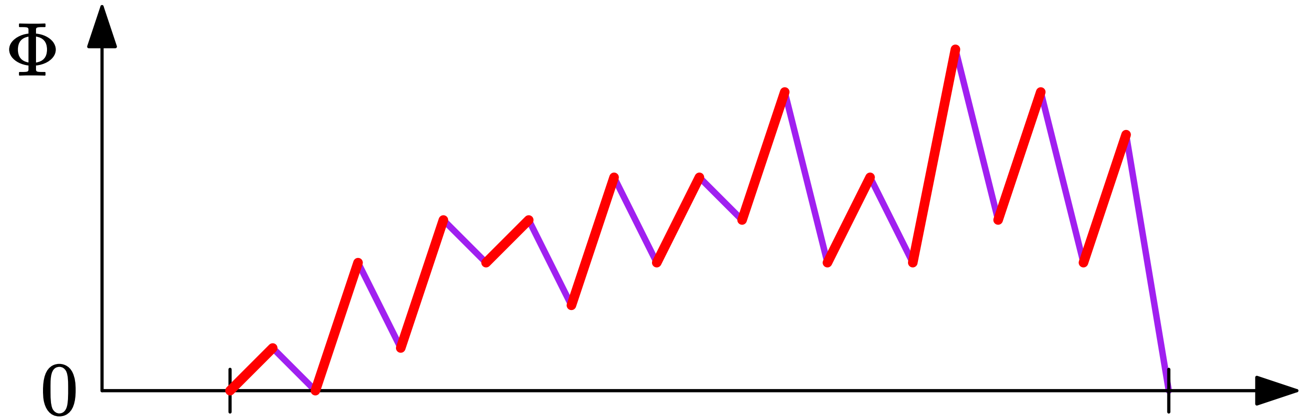
$$\text{Relabel の回数} = \sum_{u \in V - \{s,t\}} (\text{Relabel}(u) \text{ の回数})$$

$$= (n-2) \cdot O(n) = O(n^2)$$

□

本日の内容

1. Relabel 操作
2. **飽和 Push 操作**
3. 非飽和 Push 操作
4. 計算量評価



飽和 Push 操作の回数

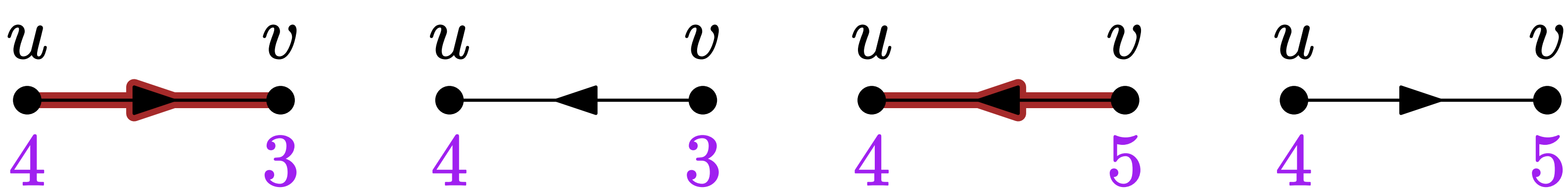
弧 $a = uv$

飽和 Push(a) : $f(a) \leftarrow f(a) + (u(a) - f(a)) = u(a)$

性質：飽和 Push の回数

飽和 Push(a) の回数 $= O(n)$

Push(a) が行える条件 $\Rightarrow d(u) = d(v) + 1$



飽和 Push 操作の回数 (続)

弧 $a = uv$

飽和 Push(a) : $f(a) \leftarrow f(a) + (u(a) - f(a)) = u(a)$

性質：飽和 Push の回数

飽和 Push(a) の回数 $= O(n)$

Push(a) が行える条件 $\Rightarrow d(u) = d(v) + 1$

飽和 Push(a) を行うと，a が G_f から消える

ふたたび飽和 Push(a) を行うためには，a が現れる必要あり

そのとき，必ず $d(v)$ が増加する

$d(u), d(v) \leq 2n - 1$ なので，飽和 Push(a) の回数 $\leq 2n - 1$

□

飽和 Push 操作の回数：まとめ

弧 $a = uv$

性質：飽和 Push の回数

飽和 Push(a) の回数 $= O(n)$

飽和 Push 操作の回数：まとめ

弧 $a = uv$

性質：飽和 Push の回数

飽和 Push(a) の回数 $= O(n)$

性質：飽和 Push の回数 (合算)

飽和 Push の回数 $= O(nm)$

補助ネットワークの弧の数 $= O(m)$ (合算)

$\therefore$ 飽和 Push の回数 $= O(m) \cdot O(n) = O(nm)$ (合算) $\square$

今日の目標：再掲

今日の目標

Push-Relabel 法の計算量評価

まとめ

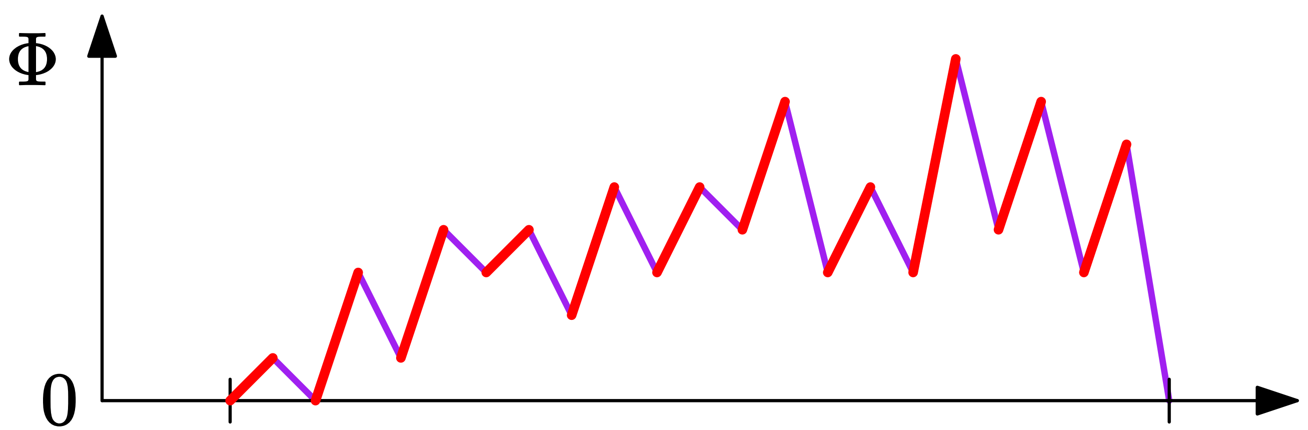
	回数	1 回あたりの計算量
Relabel(u)	$O(n)$	$O(\|\delta^{\pm}(u)\|)$
飽和 Push	$O(nm)$	$O(1)$
非飽和 Push	$O(n^2m)$	$O(1)$

$n = |V|$ (頂点数)

$m = |A|$ (弧数)

$\Rightarrow$ 全体の計算量　$O(n^2m)$

1. Relabel 操作
2. 飽和 Push 操作
3. **非飽和 Push 操作**
4. 計算量評価



非飽和 Push 操作の回数

非飽和 Push$(uv) \Rightarrow f(uv) \leftarrow f(uv) + (f^-(u) - f^+(u))$

性質：非飽和 Push の回数 (合算)

非飽和 Push の回数 $= O(n^2m)$

非飽和 Push 操作の回数

非飽和 Push(uv) $\Rightarrow$ $f(uv) \leftarrow f(uv) + (f^-(u) - f^+(u))$

性質：非飽和 Push の回数 (合算)

非飽和 Push の回数 $= O(n^2m)$

Push-Relabel 法の実行中に,

次の量 Φ がどう変化するのか，考える

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

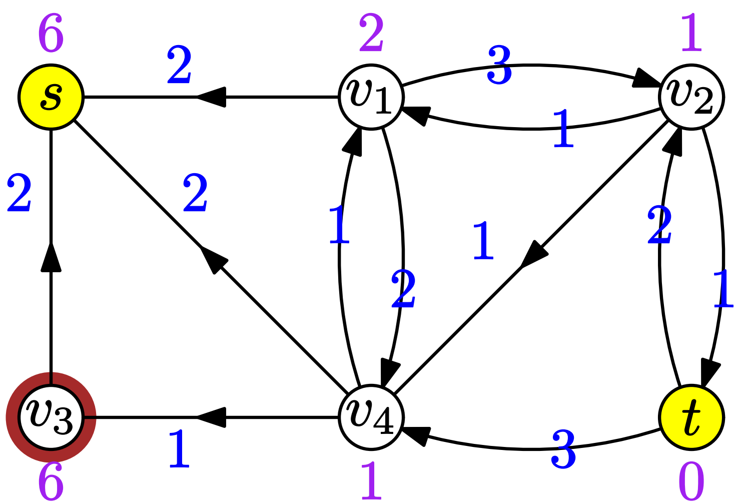
流が残っている頂点における

距離ラベルの和

注：Φ は非負整数

Φ の変化：Relabel 操作 (例)

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

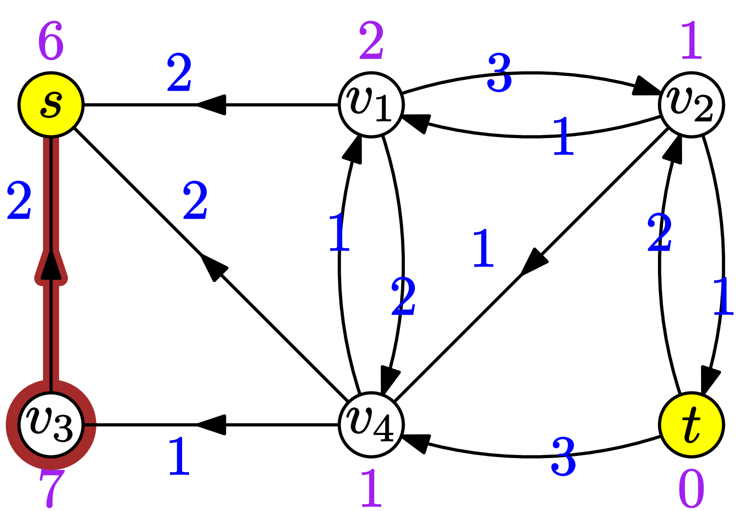


$\Phi = 6$

Relabel(v_3)

Relabel 操作で

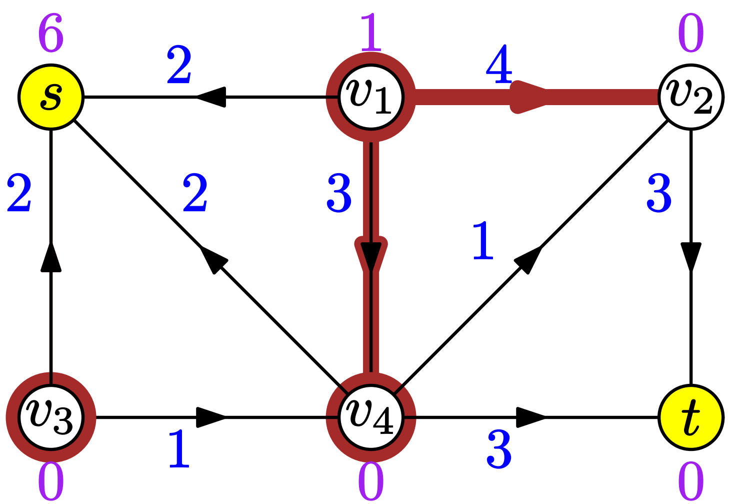
Φ は **必ず増加** する



$\Phi = 7$

Φ の変化：非飽和 Push 操作 (例)

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

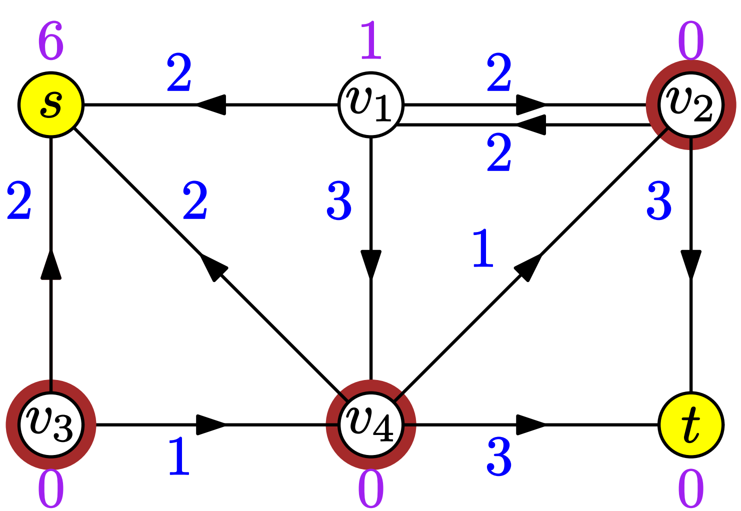


$\Phi = 1$

Push$(v_1 v_2)$

非飽和 Push 操作で

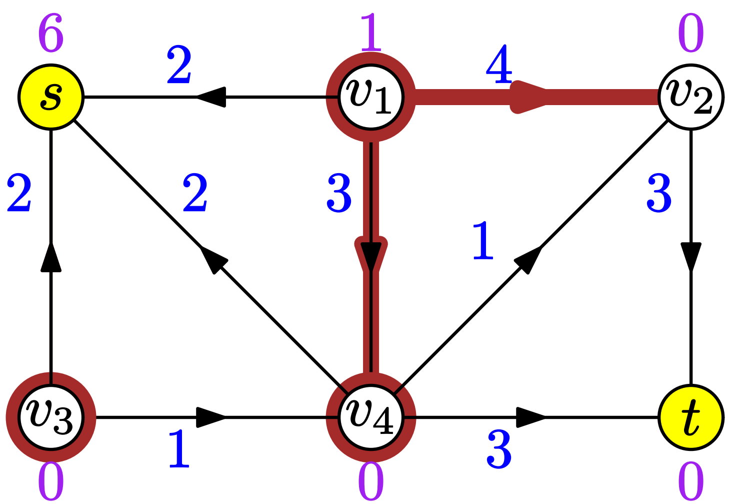
Φ は **必ず減少** する



$\Phi = 0$

Φ の変化：非飽和 Push 操作 (例)

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

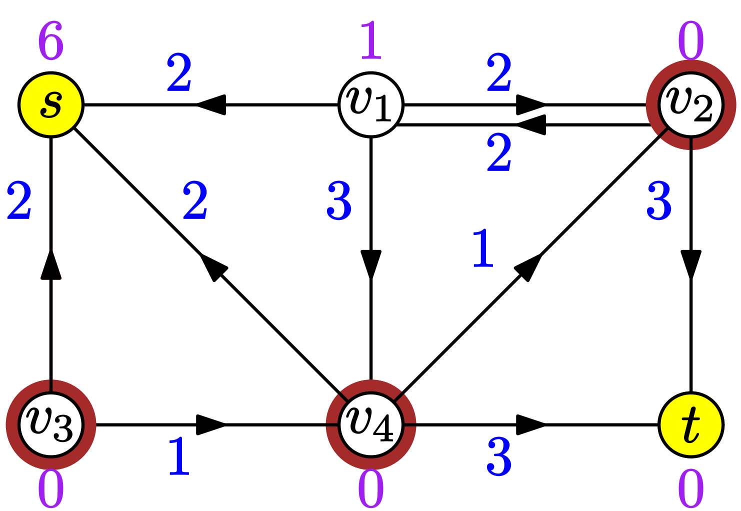


$\Phi = 1$

注：Push(uv) が行える ⇒

$d(u) = d(v) + 1$

Push(v_1v_2)



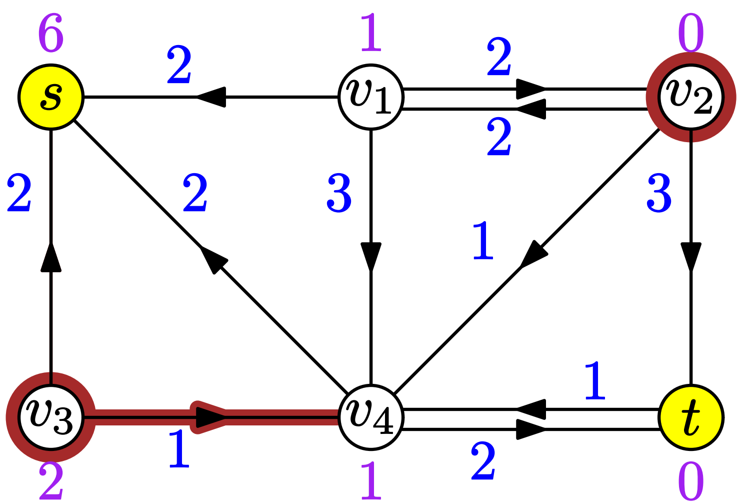
非飽和 Push 操作で

Φ は **必ず減少** する

$\Phi = 0$

Φ の変化：飽和 Push 操作 (例)

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

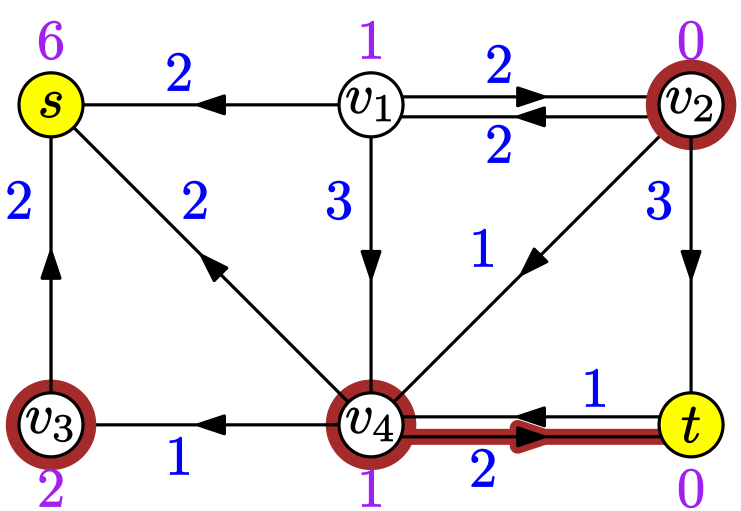


$\Phi = 2$

Push$(v_3 v_4)$

飽和 Push 操作で

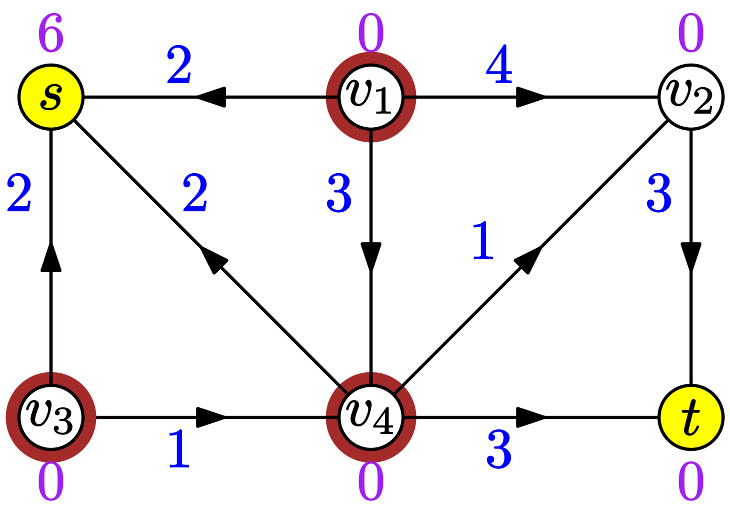
Φ は **増加しうる**



$\Phi = 3$

Φ の変化：アルゴリズム開始時 (例)

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

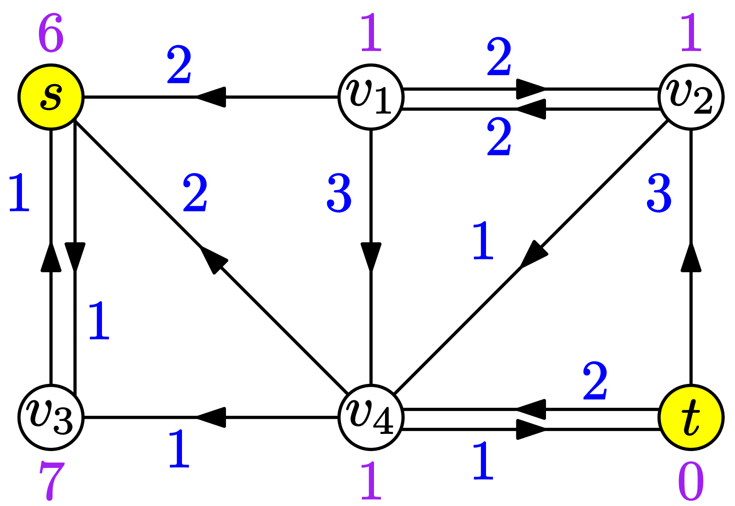


$\Phi = 0$

アルゴリズム開始時

$\Phi = 0$

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$



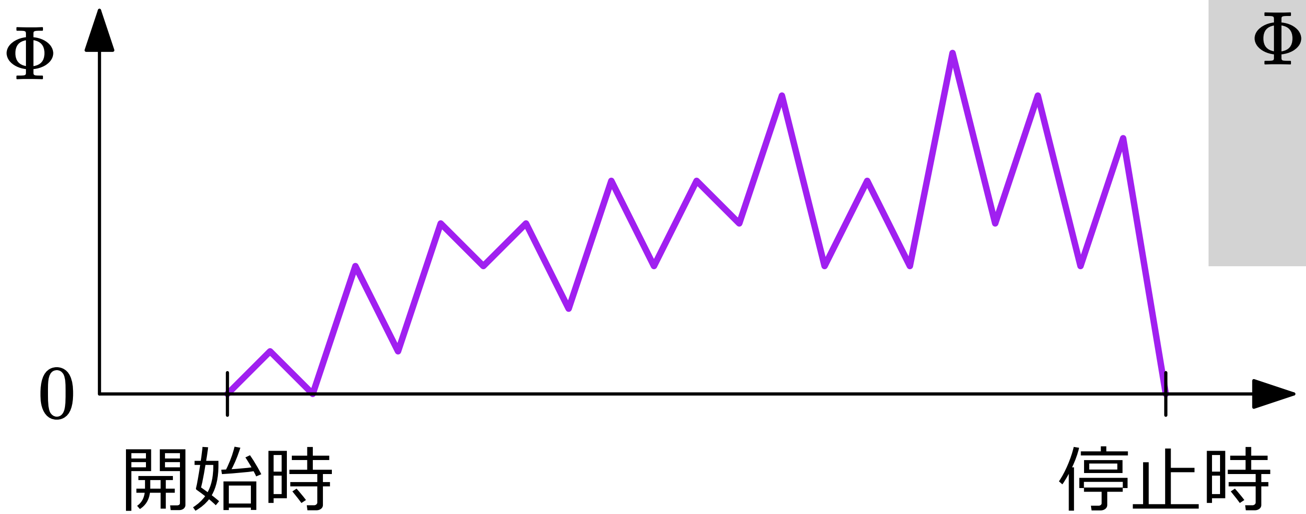
$\Phi = 0$

アルゴリズム停止時

$\Phi = 0$

Φ の変化：増加の積算

$$\Phi = \sum_{\substack{v\in V-\{s,t\},\\ f^-(v)-f^+(v)>0}} d(v)$$



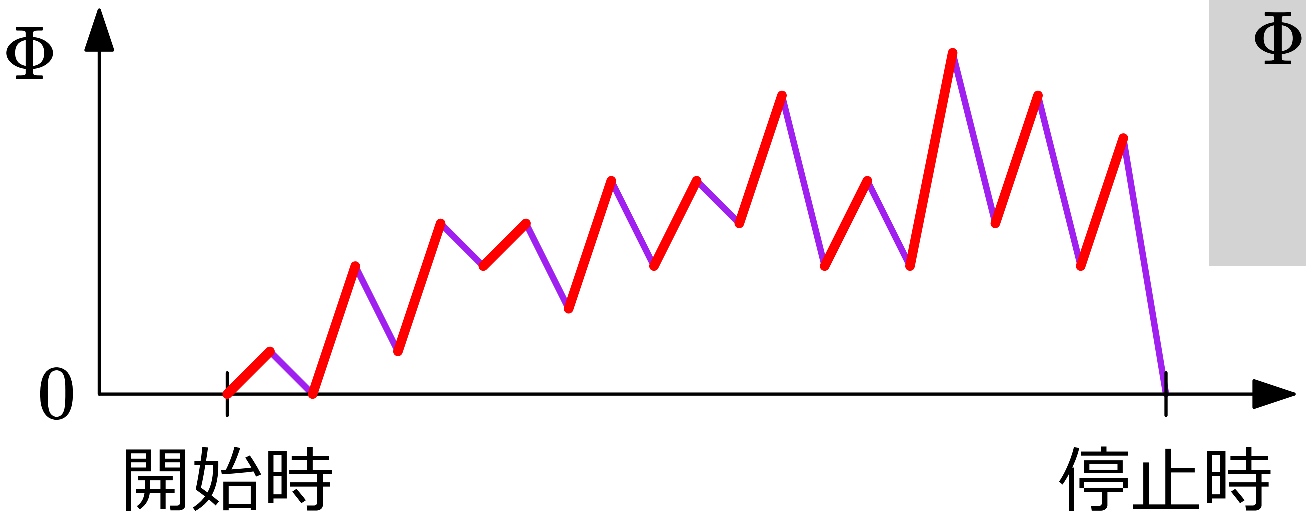
Φ の増加

Relabel，飽和 Push

Φ の減少

非飽和 Push，飽和 Push

Φ の変化：増加の積算

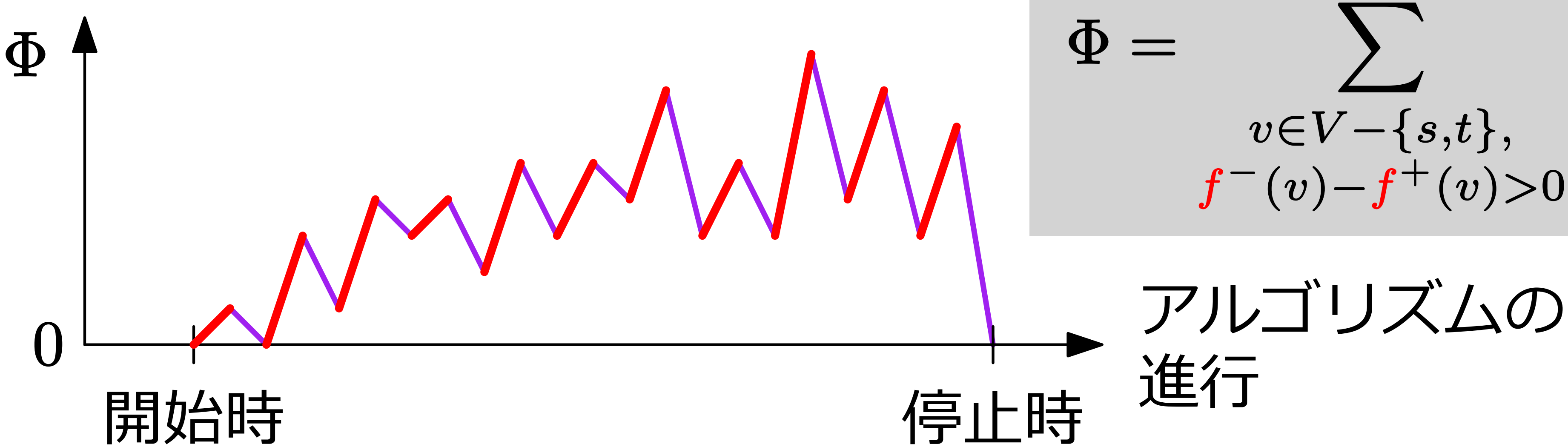


$$\Phi = \sum_{\substack{v \in V - \{s,t\},\\ f^-(v) - f^+(v) > 0}} d(v)$$

$$\sum \underline{(\Phi \text{ の増加})} \leq \sum_{v} (d(v) \text{ の最大値})$$
$$+ \sum_{uv} (\text{飽和 Push}(uv) \text{ の回数}) \cdot (d(v) \text{ の最大値})$$

Φ の増加	Φ の減少
Relabel，飽和 Push	非飽和 Push，飽和 Push

Φ の変化：増加の積算



$$\Phi = \sum_{\substack{v \in V-\{s,t\},\\ f^-(v)-f^+(v)>0}} d(v)$$

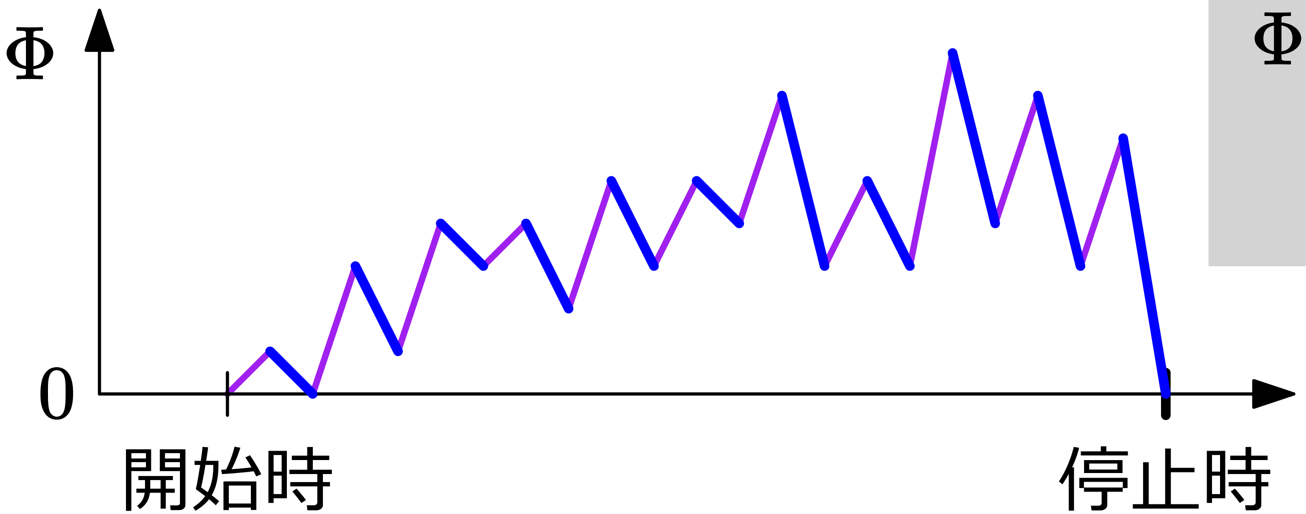
$$\sum \underline{(\Phi\ \text{の増加})} \le \sum_v (d(v)\ \text{の最大値})$$

$$+ \sum_{uv} (\text{飽和 Push}(uv)\ \text{の回数}) \cdot (d(v)\ \text{の最大値})$$

$$\le (n-2)\cdot(2n-1) + O(nm)\cdot(2n-1)$$

$$= O(n^2 m)$$

Φ の変化：減少の積算



アルゴリズムの進行

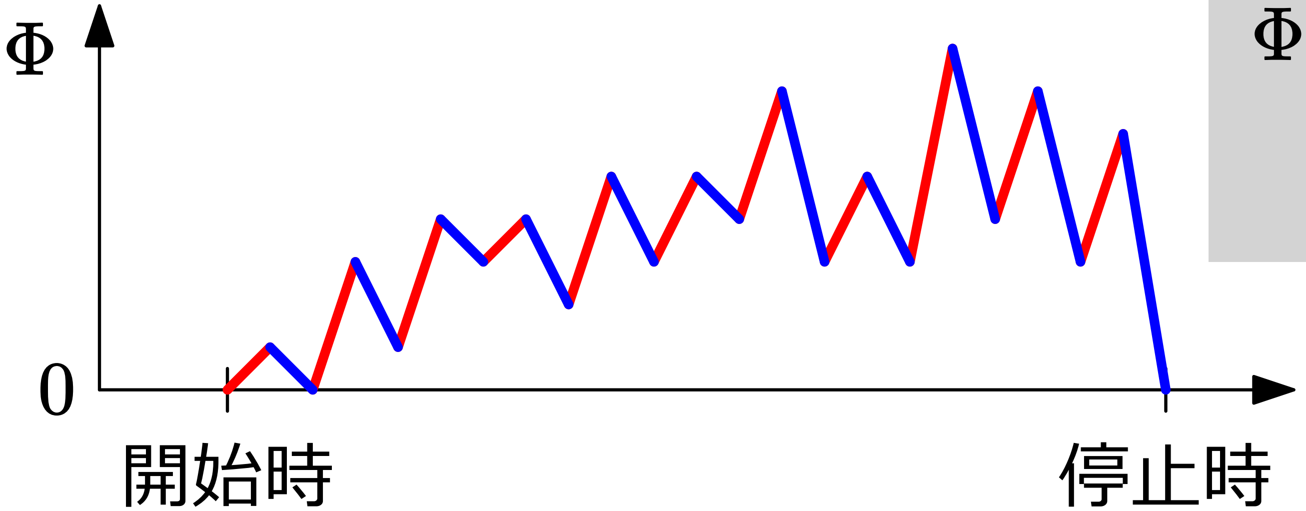
$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

$\sum \underline{(\Phi \text{ の減少})} \geq \sum_{uv} (\text{非飽和 Push}(uv) \text{ による減少})$

$\geq$ 非飽和 Push の総回数

Φ の増加	Φ の減少
Relabel，飽和 Push	非飽和 Push，飽和 Push

Φ の変化：増加と減少の相殺

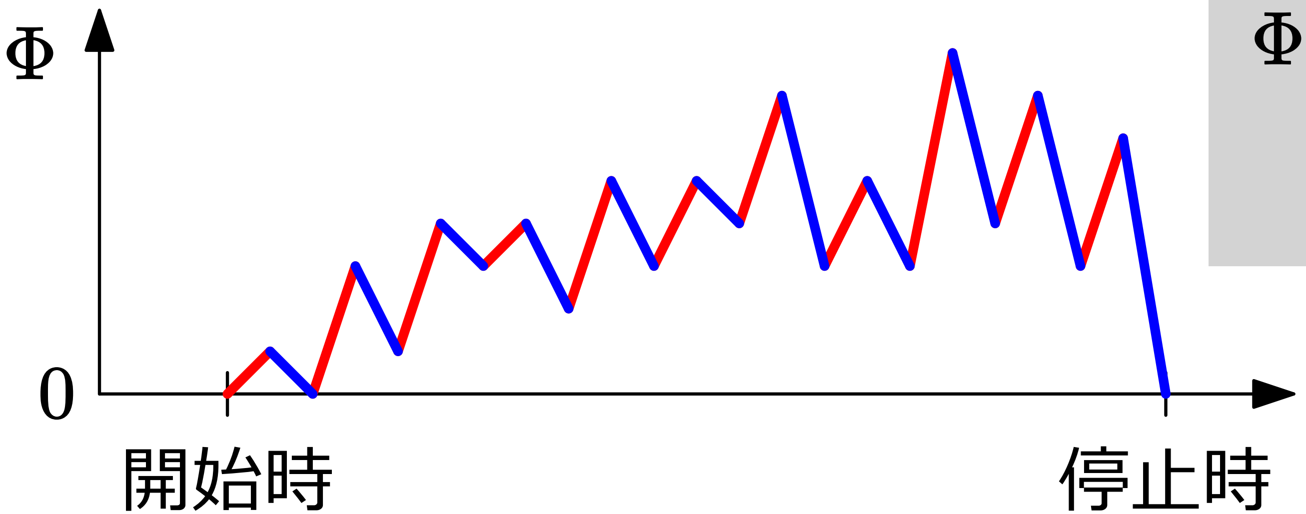


$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

開始時と停止時に，$\Phi = 0$ であるから

$$\sum \underline{(\Phi \text{ の減少})} = \sum \underline{(\Phi \text{ の増加})}$$

$$\Phi = \sum_{\substack{v\in V-\{s,t\},\\ f^-(v)-f^+(v)>0}} d(v)$$



開始時と停止時に，$\Phi = 0$ であるから

非飽和 Push の総回数

$$\le \sum \underline{(\Phi \text{ の減少})} = \sum \underline{(\Phi \text{ の増加})} = O(n^2 m)$$

非飽和 Push 操作の回数：まとめ

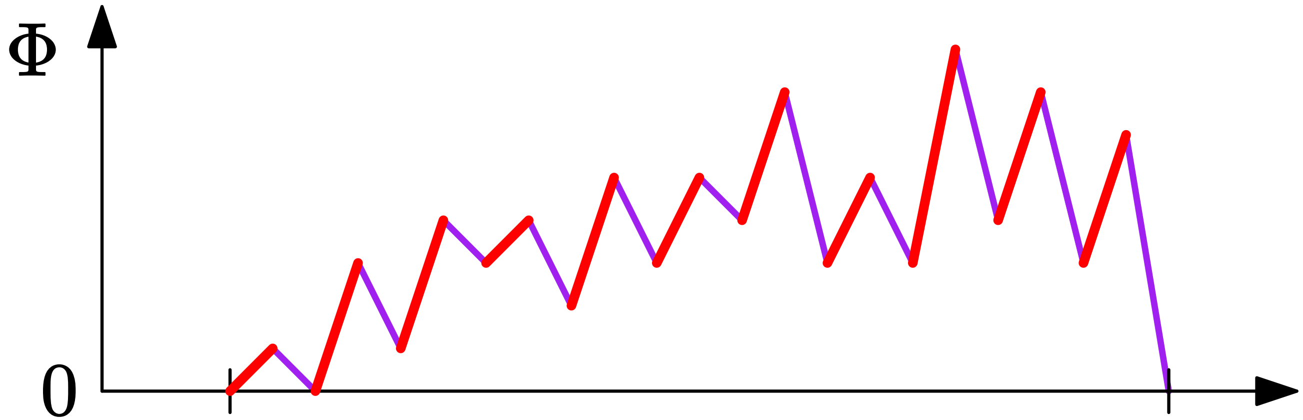
非飽和 Push$(uv) \Rightarrow f(uv) \leftarrow f(uv) + (f^-(u) - f^+(u))$

性質：非飽和 Push の回数 (合算)

非飽和 Push の回数 $= O(n^2m)$

証明：付録を参照 (考え方は既に述べたとおり)

1. Relabel 操作
2. 飽和 Push 操作
3. 非飽和 Push 操作
4. **計算量評価**



今日の目標：再掲

今日の目標

Push-Relabel 法の計算量評価

まとめ

	回数	1 回あたりの計算量
Relabel(u)	$O(n)$	$O(\lvert\delta^{\pm}(u)\rvert)$
飽和 Push	$O(nm)$	$O(1)$
非飽和 Push	$O(n^2m)$	$O(1)$

$n = \lvert V\rvert$ (頂点数)

$m = \lvert A\rvert$ (弧数)

$\Rightarrow$ 全体の計算量 $O(n^2m)$

データ構造

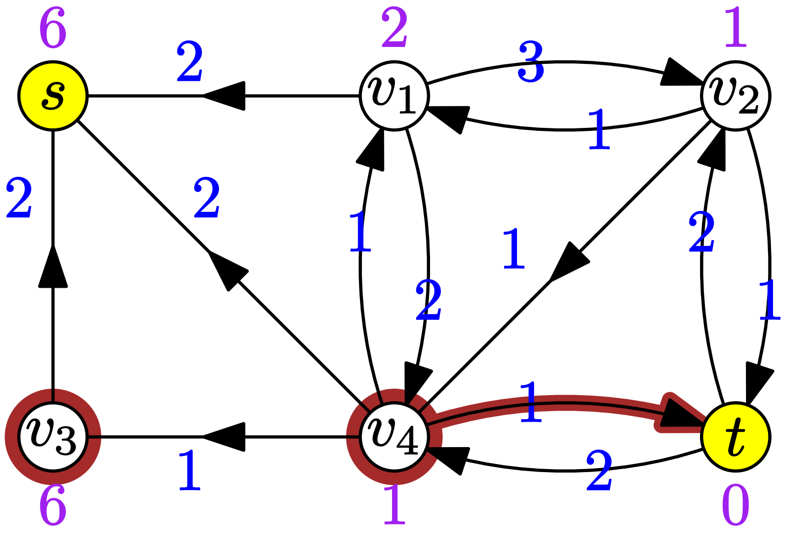
アルゴリズムでは，次を保持 (例えば，二重連結リストで)

- 流が残っている頂点の集合 (を含む集合)

$$E \supseteq \{u \in V - \{s,t\} \mid f^-(u) - f^+(u) > 0\}$$

- Push できる弧の集合

$$P = \{uv \in A_f \mid f^-(u) - f^+(u) > 0, d(u) = d(v) + 1\}$$



$E = \{v_3, v_4\}, P = \{v_4 t\}$

初期化において，
$E = \{v \mid sv \in A\}$ とする

データ構造を使ってアルゴリズムを実行

アルゴリズムでは，次を保持 (例えば，二重連結リストで)

- 流が残っている頂点の集合 (を含む集合)

$$E \supseteq \{u \in V - \{s,t\} \mid f^-(u) - f^+(u) > 0\}$$

- Push できる弧の集合

$$P = \{uv \in A_f \mid f^-(u) - f^+(u) > 0, d(u) = d(v) + 1\}$$

- $P \neq \emptyset$ である限り
  - Push(uv) を実行
  - P を更新
- $E \neq \emptyset$ である限り
  - E, P を更新
  - $P \neq \emptyset$ になったら，上に戻る

$uv \in P$ に対して

Push(uv) で起きうる更新

- $u \notin E$ となる　　E に 1 単位だけ借金
- $v \in E$ となる　　$O(1)$ ステップで可能
- $uv \notin P$ となる　　$O(1)$ ステップで可能

$\therefore O(1)$ ステップ $+$ 1 単位の借金 で更新可能

借金額 = この集合の差の要素数

データ構造：

- $E \supseteq \{u \in V - \{s,t\} \mid f^-(u) - f^+(u) > 0\}$
- $P = \{uv \in A_f \mid u \in E, d(u) = d(v) + 1\}$

データ構造：Relabel による更新 (1)

$P = \emptyset$ のとき，$u \in E$ に対して

$f^-(u) - f^+(u) > 0$ のとき，Relabel(u) を実行

Relabel(u) で起きうる更新

- $uv \in P$ となる

$O(|\delta^+(u)| + |\delta^-(u)|)$ ステップで可能

$\therefore O(|\delta^+(u)| + |\delta^-(u)|)$ ステップで更新可能

データ構造：
- $E \supseteq \{u \in V - \{s,t\} \mid f^-(u) - f^+(u) > 0\}$
- $P = \{uv \in A_f \mid u \in E, d(u) = d(v) + 1\}$

データ構造：Relabel による更新 (2)

$P = \emptyset$ のとき，$u \in E$ に対して

$f^-(u) - f^+(u) = 0$ のとき，u を E から削除

<u>u の削除で起きる更新</u>

- $u \notin E$ となる　　　1 単位の借金返済

$\therefore$ 1 単位の借金返済で可能 $(+1-1 = 0)$

借金額 = この集合の差の要素数

データ構造：

- $E \supseteq \{u \in V - \{s, t\} \mid f^-(u) - f^+(u) > 0\}$
- $P = \{uv \in A_f \mid u \in E, d(u) = d(v) + 1\}$

Push にかかる総ステップ数

$$= O(n^2 m) \cdot O(1)$$
$$= O(n^2 m)$$

Relabel にかかる総ステップ数

$$= \sum_u O(n) \cdot O(|\delta^+(u)| + |\delta^-(u)|)$$
$$= O(n) \sum_u O(|\delta^+(u)| + |\delta^-(u)|)$$
$$= O(n) \cdot O(m)$$
$$= O(nm)$$

性質：Push-Relabel 法の計算量評価

Push-Relabel 法は $O(nm^2)$ ステップで停止する

	計算量	分類	注意
増加道法	$O(m^2U)$	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	$O(n^2m)$	強多項式時間	実数 OK

補足：Relabel を行える条件

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 $d(u) \le d(v)$

次のように変えても，アルゴリズムは正しく動く

Relabel(u) を行える条件 (より局所的なバージョン)

1. $f^-(u) - f^+(u) > 0$　　(u に残っている)
2. 任意の $v \in V$ に対して，$uv \in A_f$ ならば $d(u) \le d(v)$

(Goldberg, Tarjan '88)

アルゴリズム：より局所的な Push-Relabel 法

1. s-t 前流 f，距離ラベル d を初期化
2. $f^-(u) - f^+(u) > 0$ を満たす $u \in V - \{s, t\}$ がある限り，次を実行
   (a) Push できる弧 $uv \in A_f$ があれば，Push(uv) を行う
   (b) Push できる弧 $uv \in A_f$ がなければ，Relabel(u) を行う
3. f を出力

ふつうは，この「より局所的な」Push-Relabel 法の方を Push-Relabel 法 (または，Preflow-Push 法) と呼ぶ

補足：u の選び方

アルゴリズム：より局所的な Push-Relabel 法

1. s-t 前流 f，距離ラベル d を初期化
2. $f^-(u) - f^+(u) > 0$ を満たす $u \in V - \{s, t\}$ がある限り，次を実行
   (a) Push できる弧 $uv \in A_f$ があれば，Push(uv) を行う
   (b) Push できる弧 $uv \in A_f$ がなければ，Relabel(u) を行う
3. f を出力

選び方に，自由度 (任意性) がある

(Cheriyan, Maheshwari ’89)

アルゴリズム：最大ラベル Push-Relabel 法

u を選ぶとき，条件を満たす頂点の中で
距離ラベルが最大のものを選ぶ

↓ 計算量の解析

非飽和 Push の回数 $= O(n^2\sqrt{m})$

↓ データ構造の工夫

計算量 $= O(n^2\sqrt{m})$

前半のまとめ

枠組

最大流問題 → 最適性条件

線形計画法

アルゴリズム

視点

- 相補性定理の使い方 $\leadsto$ アルゴリズムの多様性
- アルゴリズムにおける任意性 $\leadsto$ 計算量削減の可能性

後半の予告

枠組

最小費用流問題 → 最適性条件

線形計画法

アルゴリズム

注意

最小費用流問題は 最大流問題に比べて
圧倒的に 研究しにくい

講義では，細かい議論をあまりしないようにしていく予定

- 非飽和 Push 操作：証明

非飽和 Push 操作の回数

性質：非飽和 Push の回数 (合算)

非飽和 Push の回数 $= O(n^2 m)$

証明：

Push-Relabel 法の実行中に，

次の量 Φ がどう変化するのか，考える

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

流が残っている頂点における

距離ラベルの和

注：Φ は非負整数

Φ の変化：Relabel 操作

Relabel(u) によって，

Φ が Φ' に変化するとする

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

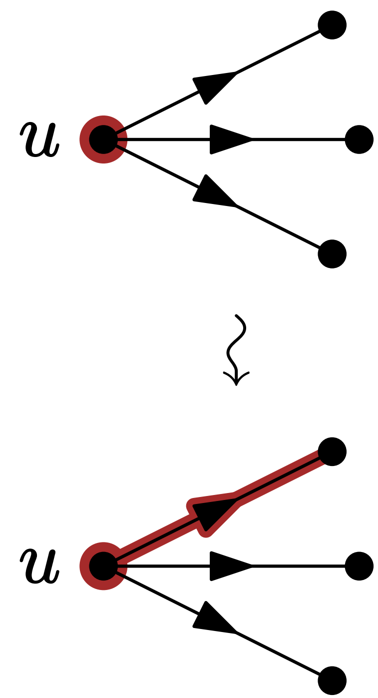
Relabel(u) を行える条件 $\Rightarrow uv \in A_f$ のとき $d(u) \le d(v)$

Relabel(u)：$d(u) \leftarrow \min\{d(v) \mid uv \in A_f\} + 1$

したがって，

$$\begin{aligned} \Phi' &= \Phi - d(u) + \min\{d(v) \mid uv \in A_f\} + 1 \\ &\ge \Phi - d(u) + d(u) + 1 = \Phi + 1 \end{aligned}$$

$\therefore$ Φは必ず増加する



Φ の変化：非飽和 Push 操作

非飽和 Push(uv) によって，

Φ が Φ' に変化するとする

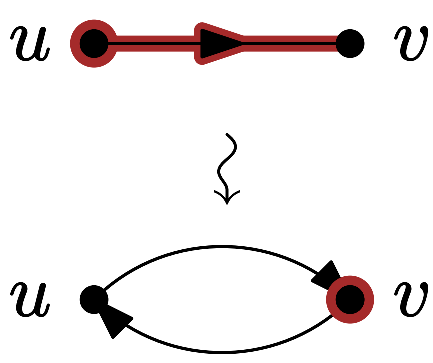
$$\Phi = \sum_{\substack{v \in V - \{s,t\},\\ f^-(v) - f^+(v) > 0}} d(v)$$

Push(uv) を行える条件 $\Rightarrow d(u) = d(v) + 1$

非飽和 Push(uv)：$f(uv) \leftarrow f(uv) + (f^-(u) - f^+(u))$

したがって，

$$\Phi' = \Phi - d(u) + d(v)$$
$$= \Phi - 1$$



$\therefore$ Φは必ず1だけ減少する

Φ の変化：飽和 Push 操作

飽和 Push(uv) によって，

Φ が Φ' に変化するとする

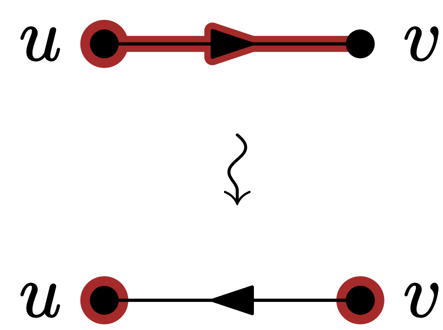
$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

Push(uv) を行える条件 $\Rightarrow d(u) = d(v) + 1$

飽和 Push(uv)：$f(uv) \leftarrow u(uv)$

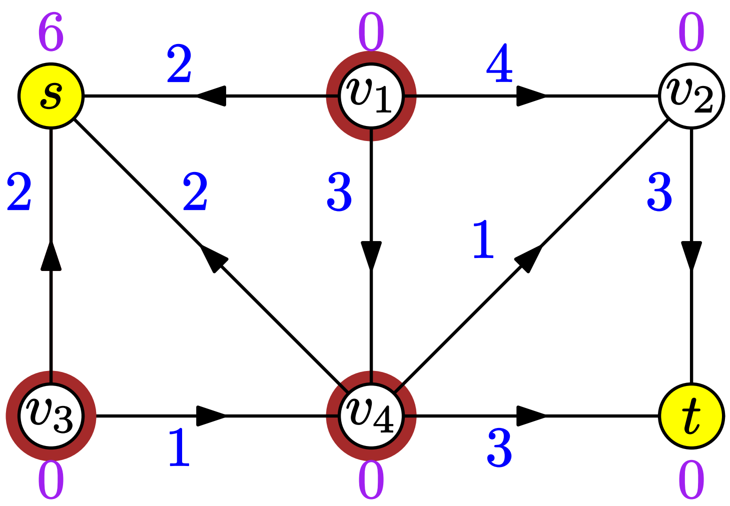
したがって，

$$\Phi' \leq \Phi + d(v)$$



$\therefore \Phi$ は増加したとしても，高々 $d(v)$

(減少するかもしれない)



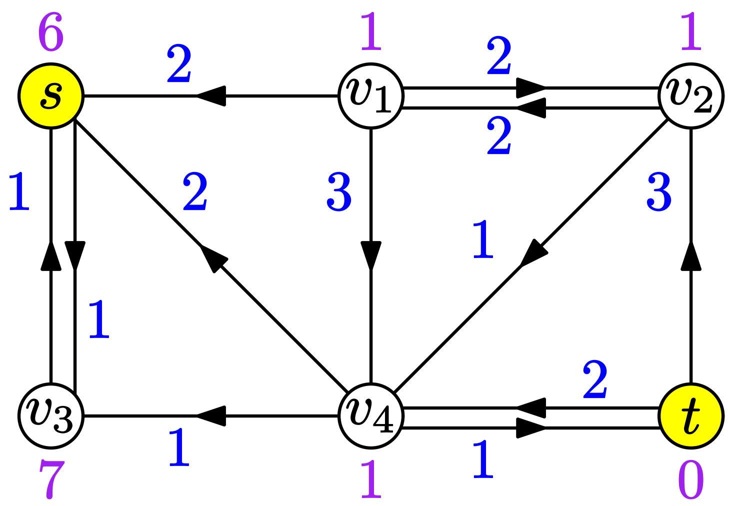
$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

アルゴリズムの開始時では

任意の頂点 $v \in V - \{s, t\}$ に対して

$d(v) = 0$

したがって，$\Phi = 0$



$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

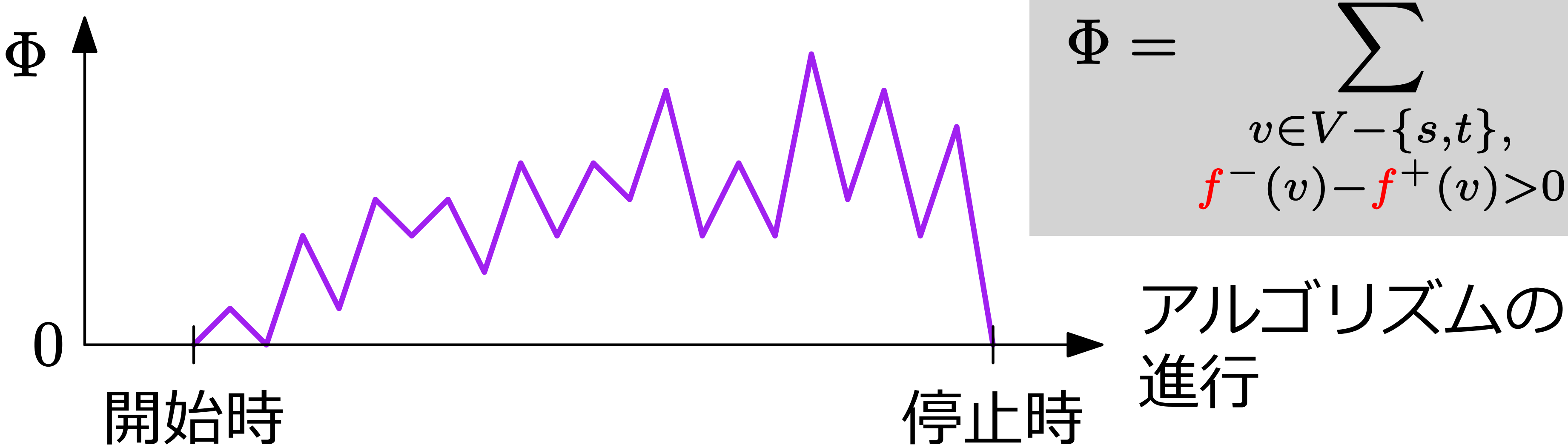
アルゴリズムの停止時では

任意の頂点 $v \in V - \{s, t\}$ に対して

$f^-(v) - f^+(v) = 0$

したがって，$\Phi = 0$

Φ の変化：増加の積算



$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

Φ の増加

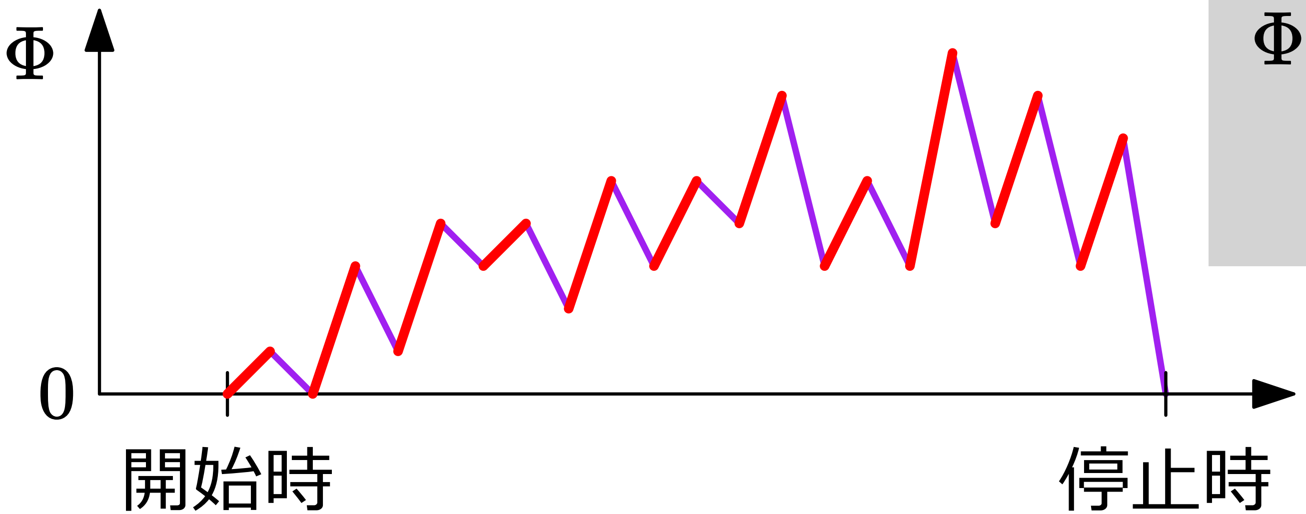
Relabel，飽和 Push

Φ の減少

非飽和 Push，飽和 Push

Φ の変化：増加の積算

$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$



アルゴリズムの進行

$$\sum \underline{(\Phi \text{ の増加})} \leq \sum_{v} (d(v) \text{ の最大値})$$
$$+ \sum_{uv} (\text{飽和 Push}(uv) \text{ の回数}) \cdot (d(v) \text{ の最大値})$$

Φ の増加

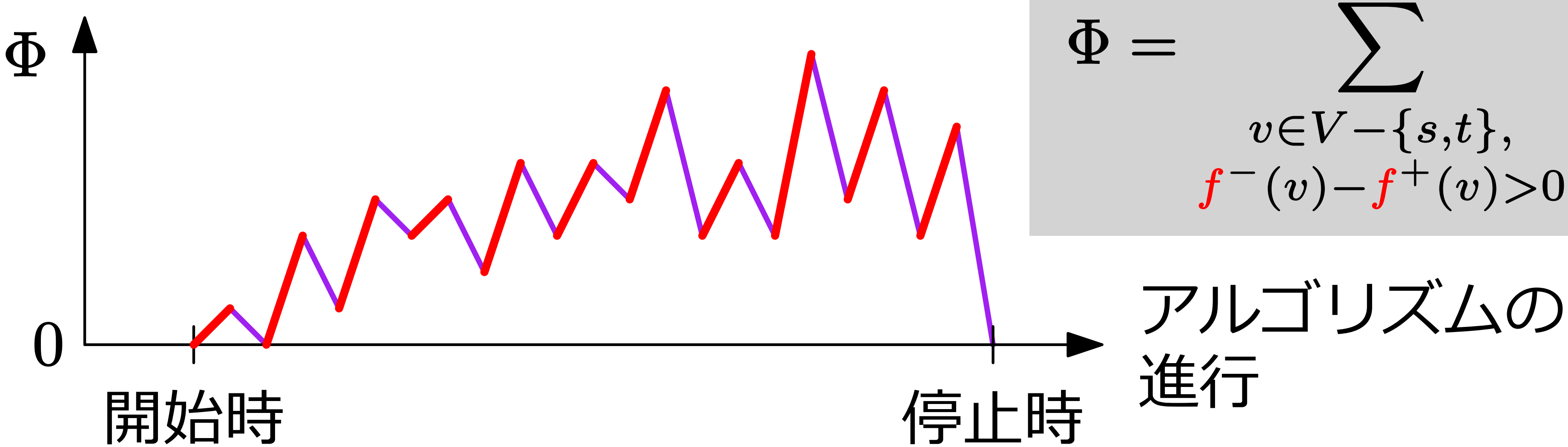
Relabel，飽和 Push

Φ の減少

非飽和 Push，飽和 Push

Φ の変化：増加の積算

$$\Phi = \sum_{\substack{v \in V - \{s,t\},\\ f^-(v) - f^+(v) > 0}} d(v)$$

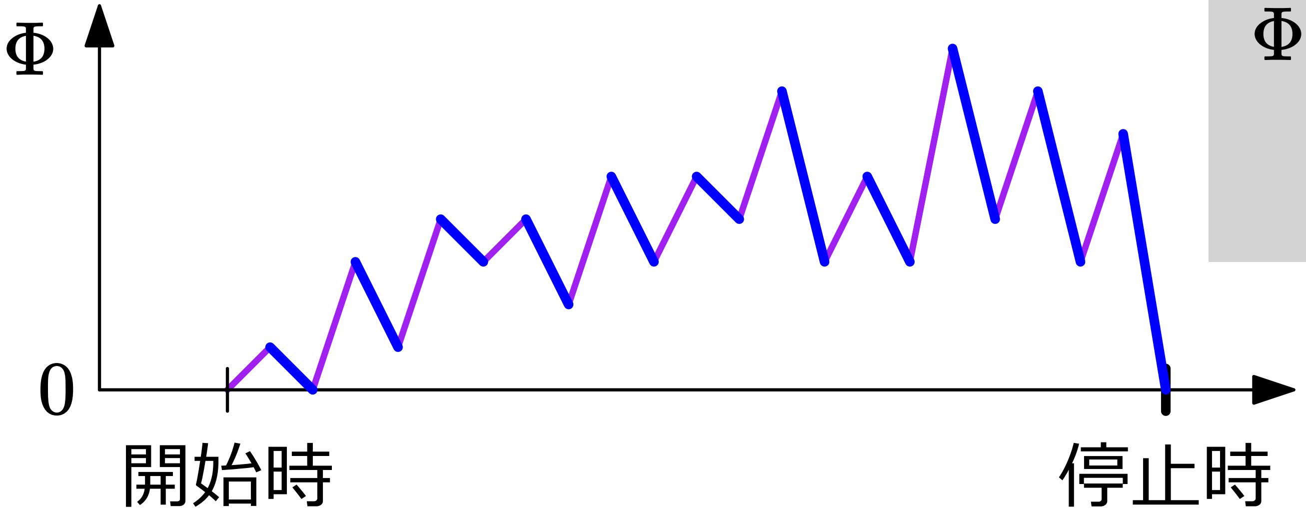


$$\sum \underline{(\Phi \text{ の増加})} \le \sum_v (d(v) \text{ の最大値})$$

$$+ \sum_{uv} (\text{飽和 Push}(uv) \text{ の回数}) \cdot (d(v) \text{ の最大値})$$

$$\le (n-2)\cdot(2n-1) + O(nm)\cdot(2n-1)$$

$$= O(n^2 m)$$



$$\Phi = \sum_{\substack{v \in V - \{s,t\},\\ f^-(v) - f^+(v) > 0}} d(v)$$

$\sum$ (Φ の減少) $\geq \sum_{uv}$(非飽和 Push(uv) による減少)

$\geq$ 非飽和 Push の総回数

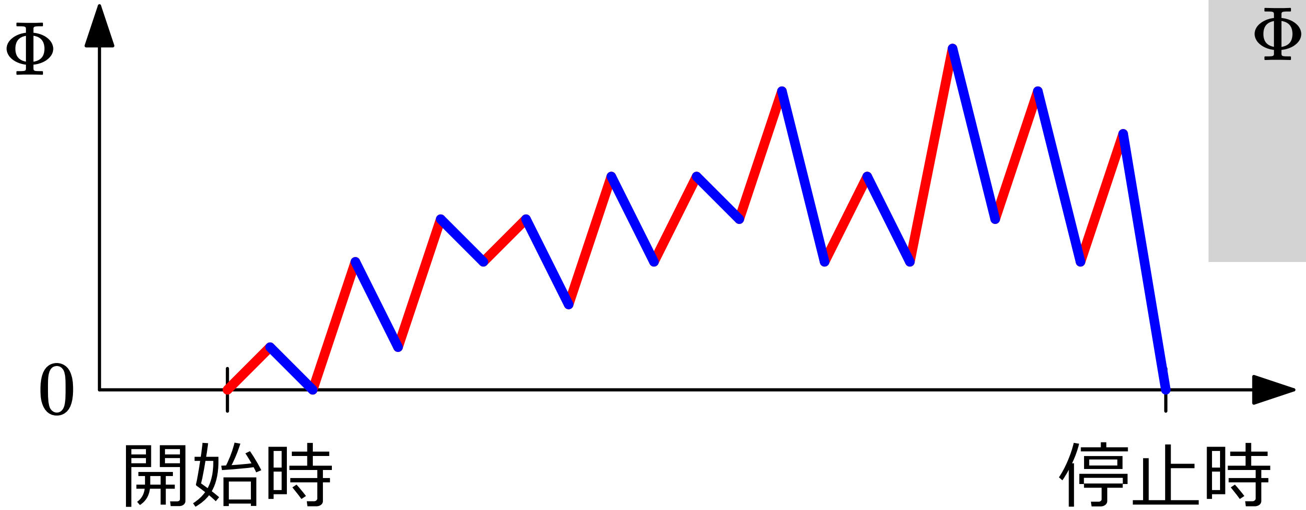
Φ の増加

Relabel，飽和 Push

Φ の減少

非飽和 Push，飽和 Push

Φ の変化：増加と減少の相殺



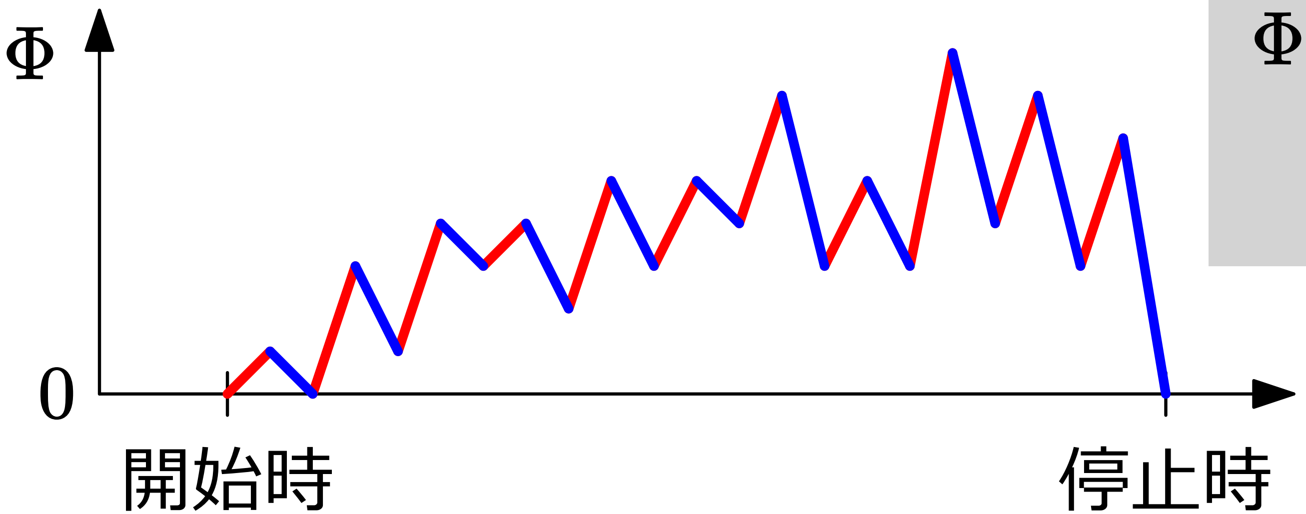
$$\Phi = \sum_{\substack{v \in V - \{s,t\}, \\ f^-(v) - f^+(v) > 0}} d(v)$$

開始時と停止時に，$\Phi = 0$ であるから

$$\sum \underline{(\Phi \text{ の減少})} = \sum \underline{(\Phi \text{ の増加})}$$

Φ の変化：増加と減少の相殺

$$\Phi = \sum_{\substack{v \in V-\{s,t\},\\ f^-(v)-f^+(v)>0}} d(v)$$



アルゴリズムの進行

開始時と停止時に，$\Phi = 0$ であるから

非飽和 Push の総回数

$$\leq \sum \underbrace{(\Phi \text{ の減少})} = \sum \underbrace{(\Phi \text{ の増加})} = O(n^2 m)$$

$\square$