最大流

北海道大学　修士1年　小畠教寛

本日の流れ

- 最大流の説明
- 最大流アルゴリズムの説明
- 最大流の応用
    - 二部グラフのマッチング
    - 最小流量制約付き最大流

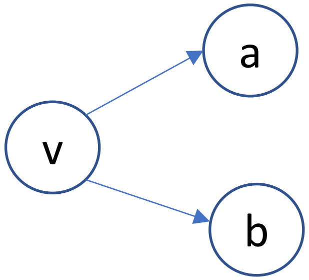
本日の流れ

- 最大流の説明
- 最大流アルゴリズムの説明
- 最大流の応用
  - 二部グラフのマッチング
  - 最小流量制約付き最大流

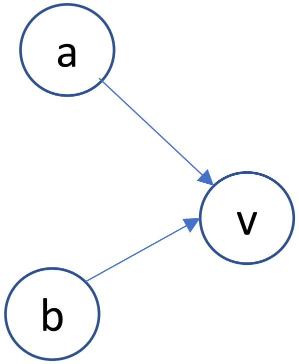
前提知識

有向グラフを$G = (V, E)$に対して以下を定める.

- $\delta_+(v)$を頂点$v \in V$から出て行く辺の集合とする.



- $\delta_-(v)$を頂点$v \in V$に入ってくる辺の集合とする.



最大フローとは

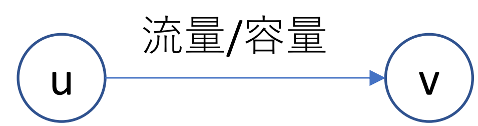
有向グラフ$G=(V,E)$において，各辺$e \in E$に非負実数$c(e)$が与えられる．

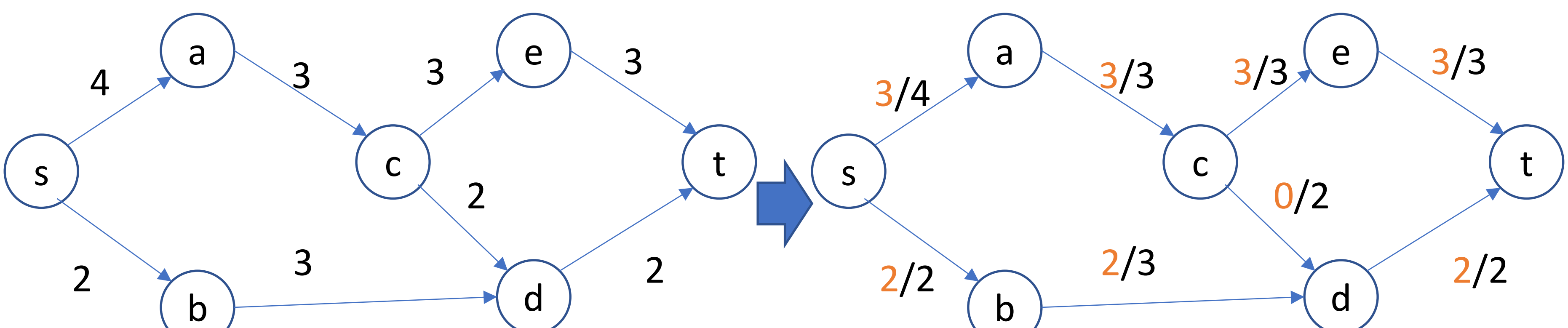
[定義]
各辺に対して，以下の性質を満たす非負実数$f(e)$を割り当てる．
- $e \in E$に対して，$f(e) \leq c(e)$
- $v \in V-\{\mathrm{s},\mathrm{t}\}$に対して，$\sum_{e \in \delta_{-}(v)} f(e) = \sum_{e \in \delta_{+}(v)} f(e)$

$\sum_{e \in \delta_{+}(s)} f(e)$ をフロー値といい，これが最大になるものを最大フローという．

cを辺の**容量**，fを**流量**，

sを**始点(source)**，tを**終点(sink)**という．





本日の流れ

- 最大流の説明
- 最大流アルゴリズムの説明
- 最大流の応用
  - 二部グラフのマッチング
  - 最小流量制約付き最大流

アルゴリズムの説明の流れ

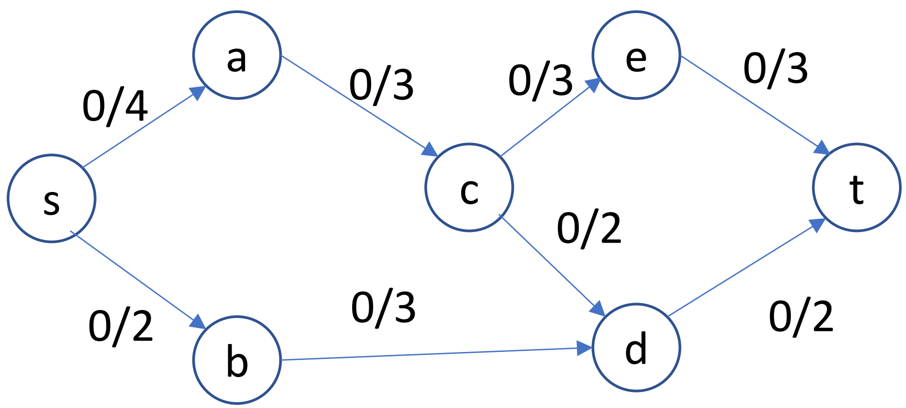
- 貪欲アルゴリズムではだめか?
- 最大フローを求めるための基本アイデア
- Ford-Fulkersonのアルゴリズムの簡単な説明
- Dinicのアルゴリズム

アルゴリズムの説明の流れ

- 貪欲アルゴリズムではだめか?
- 最大フローを求めるための基本アイデア
- Ford-Fulkersonのアルゴリズムの簡単な説明
- Dinicのアルゴリズム

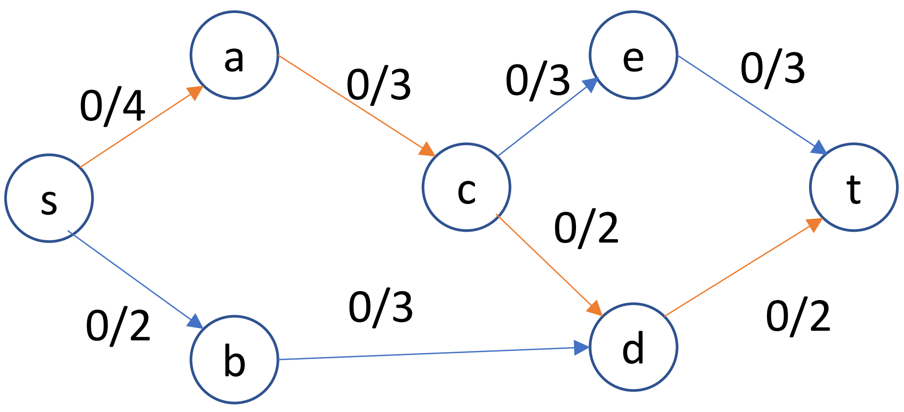
貪欲アルゴリズム(1/7)

1. フローを流せる辺のみで，s-tパスを探す.
2. s-tパスに流せるだけ流す.
3. 見つからなければ終了



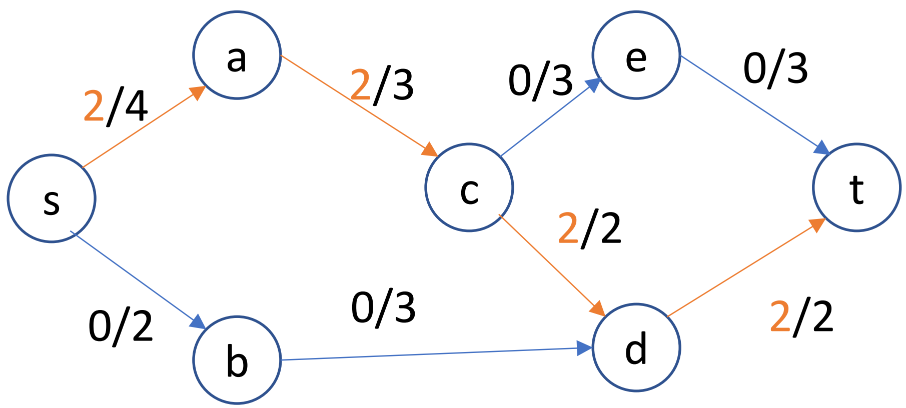
貪欲アルゴリズム(2/7)

1. フローを流せる辺のみで，s-tパスを探す.
2. s-tパスに流せるだけ流す.
3. 見つからなければ終了



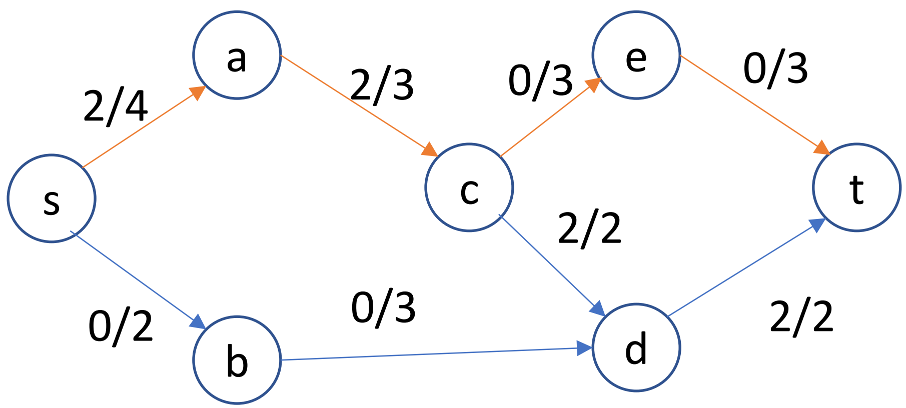
貪欲アルゴリズム(3/7)

1. フローを流せる辺のみで，s-tパスを探す.
2. s-tパスに流せるだけ流す.
3. 見つからなければ終了



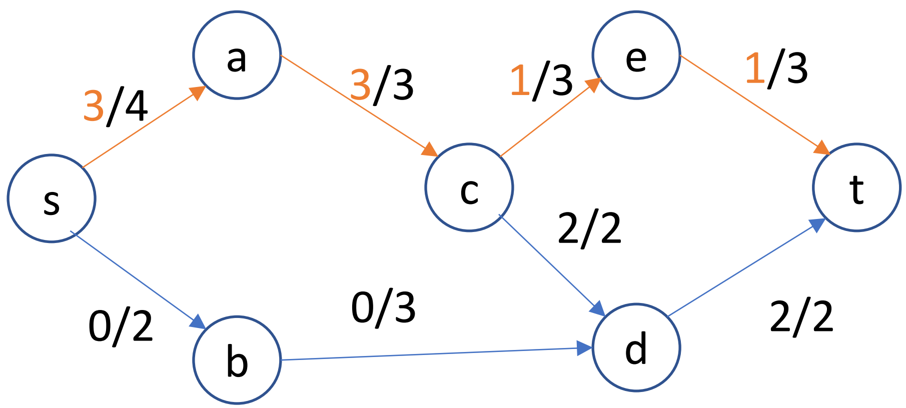
貪欲アルゴリズム(4/7)

1. フローを流せる辺のみで，s-tパスを探す.
2. s-tパスに流せるだけ流す.
3. 見つからなければ終了



貪欲アルゴリズム(5/7)

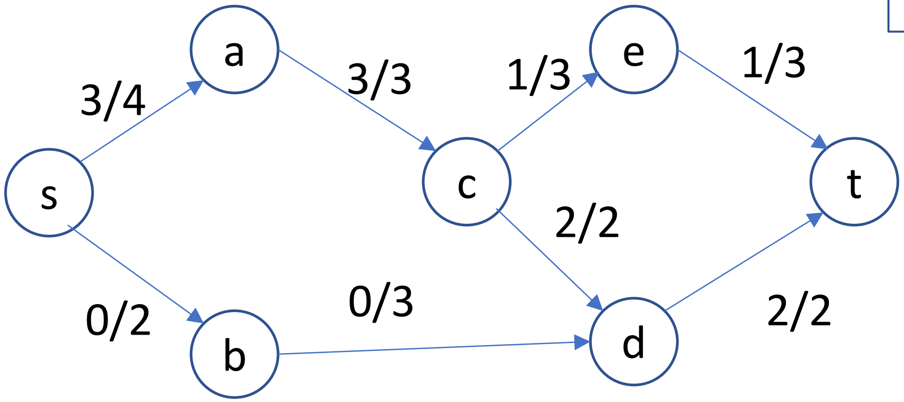
1. フローを流せる辺のみで，s-tパスを探す.
2. s-tパスに流せるだけ流す.
3. 見つからなければ終了



貪欲アルゴリズム(6/7)

1. フローを流せる辺のみで，s-tパスを探す．
2. s-tパスに流せるだけ流す．
3. 見つからなければ終了

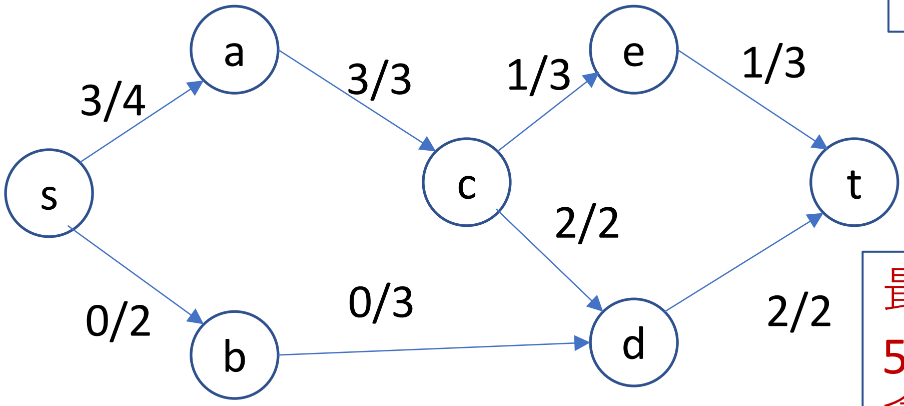
パスがないので，
終了する．
答えは，3になる．



貪欲アルゴリズム(7/7)

1. フローを流せる辺のみで，s-tパスを探す．
2. s-tパスに流せるだけ流す．
3. 見つからなければ終了

パスがないので，
終了する．
答えは，3になる．



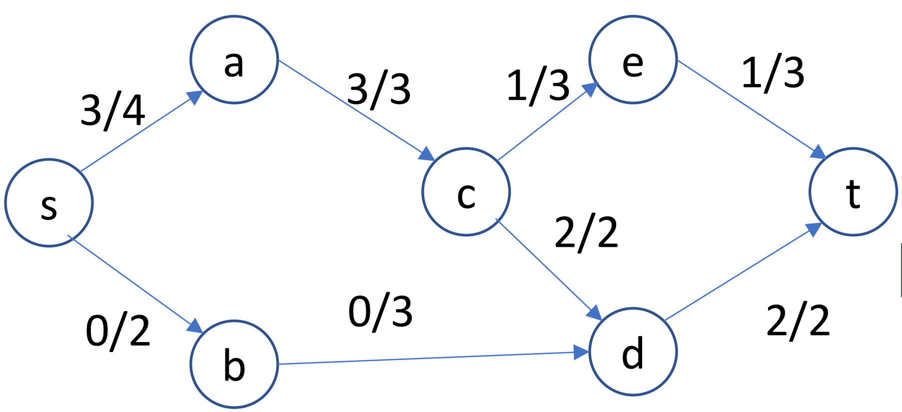
最初に示した通り，
5のフローが存在する．
貪欲では，最大フロー
が求められない

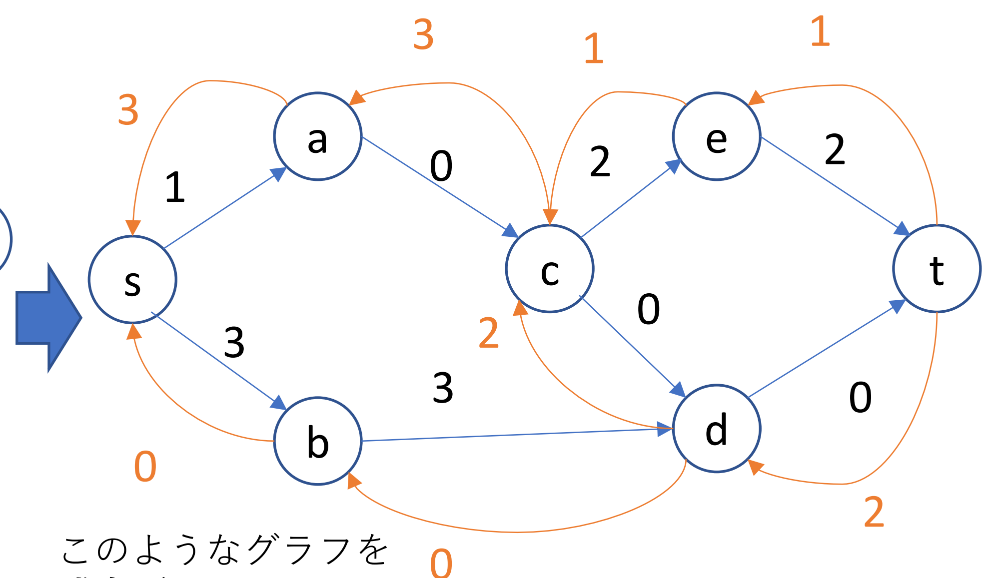
アルゴリズムの説明の流れ

- 貪欲アルゴリズムではだめか?
- 最大フローを求めるための基本アイデア
- Ford-Fulkersonのアルゴリズムの簡単な説明
- Dinicのアルゴリズム

改善アイデア

流した流量分の容量を減らし，逆辺にその容量の辺をはる．逆辺も含めてフローを流せる辺のみでs-tパスを探す．





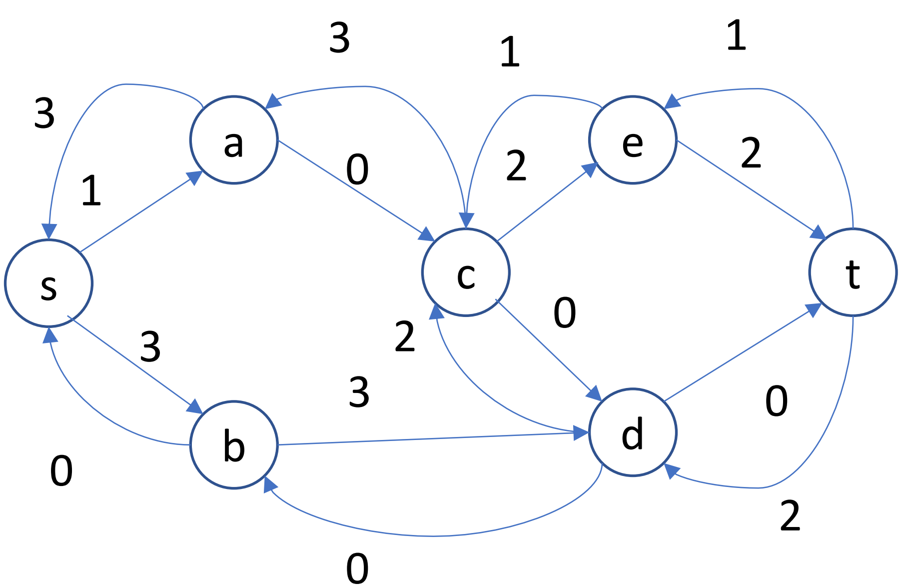
このようなグラフを
残余グラフという

アルゴリズムの説明の流れ

- 貪欲アルゴリズムではだめか?
- 最大フローを求めるための基本アイデア
- Ford-Fulkersonのアルゴリズムの簡単な説明
- Dinicのアルゴリズム

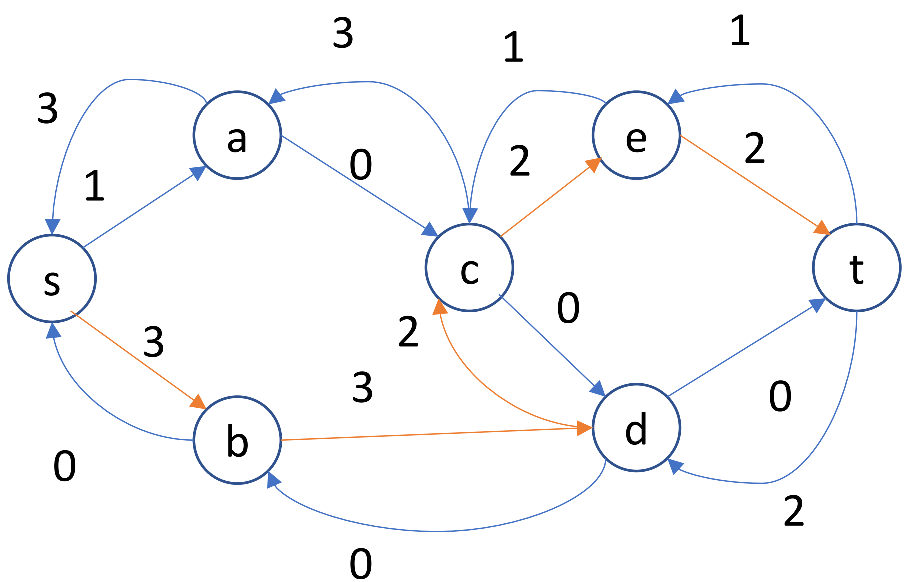
Ford-Fulkersonのアルゴリズム(1/4)

1. 容量のある辺のみで，s-tパスを探す．
2. 見つかれば，流せるだけ流し，逆辺の容量をその分増やす．
3. 見つからなければ終了する．



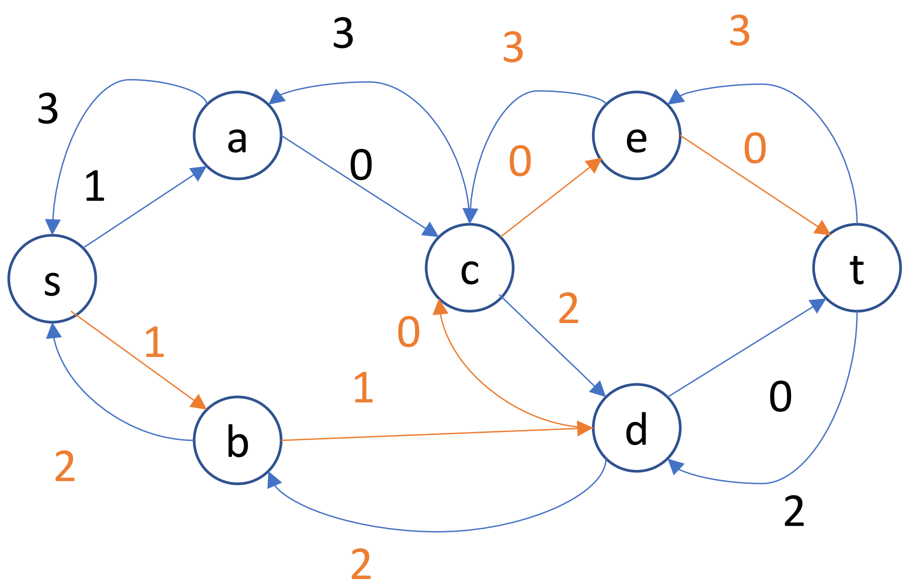
Ford-FulkersonのアルゴリズムM(2/4)

1. 容量のある辺のみで，s-tパスを探す.
2. 見つかれば，流せるだけ流し，逆辺の容量をその分増やす.
3. 見つからなければ終了する.



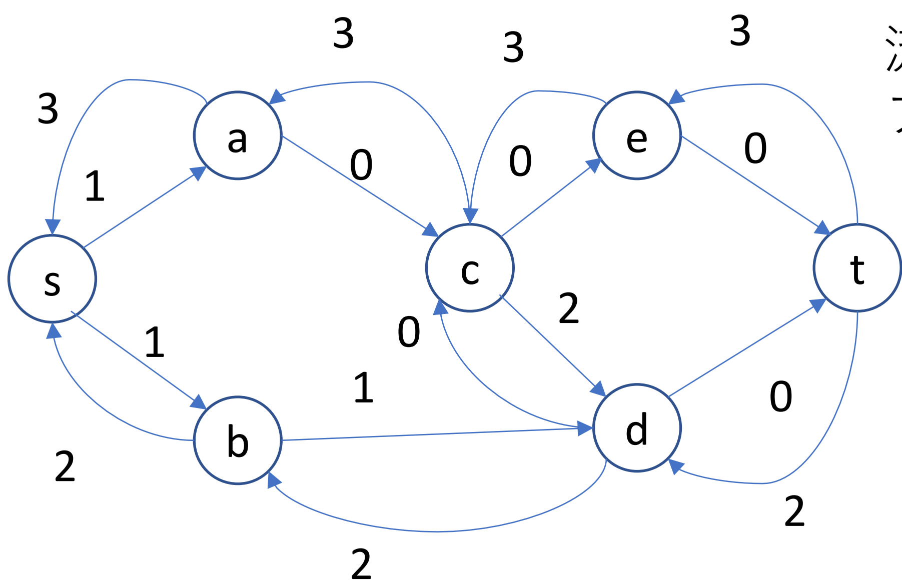
Ford-Fulkersonのアルゴリズム(3/4)

1. 容量のある辺のみで，s-tパスを探す．
2. 見つかれば，流せるだけ流し，逆辺の容量をその分増やす．
3. 見つからなければ終了する．



Ford-Fulkersonのアルゴリズム(4/4)

1. 容量のある辺のみで，s-tパスを探す.
2. 見つかれば，流せるだけ流し，逆辺の容量をその分増やす.
3. 見つからなければ終了する.

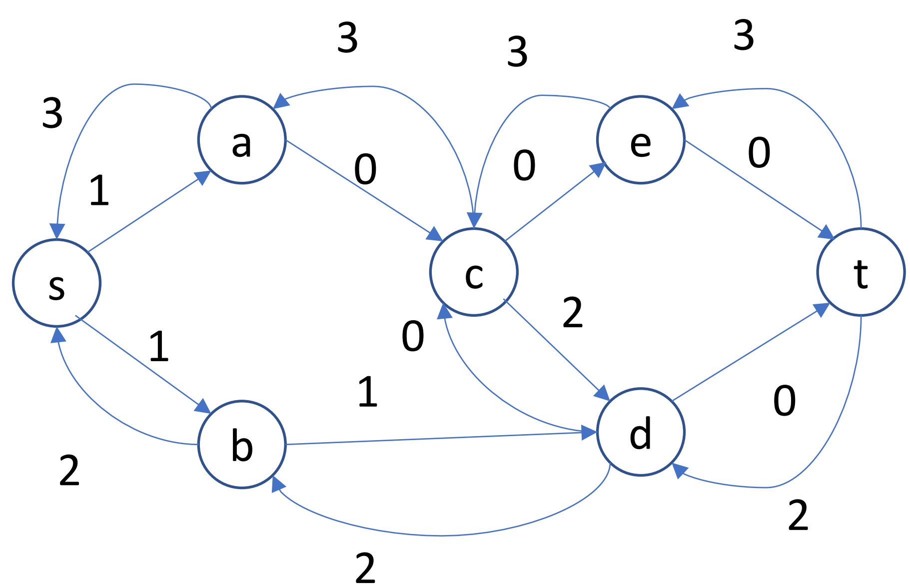


流せるパスがないので，フローは5になる

Ford-Fulkersonの時間計算量

最大フローの流量をFとすると高々F回DFSが行われるので,

時間計算量は, $O(F(|V|+|E|))$時間である.



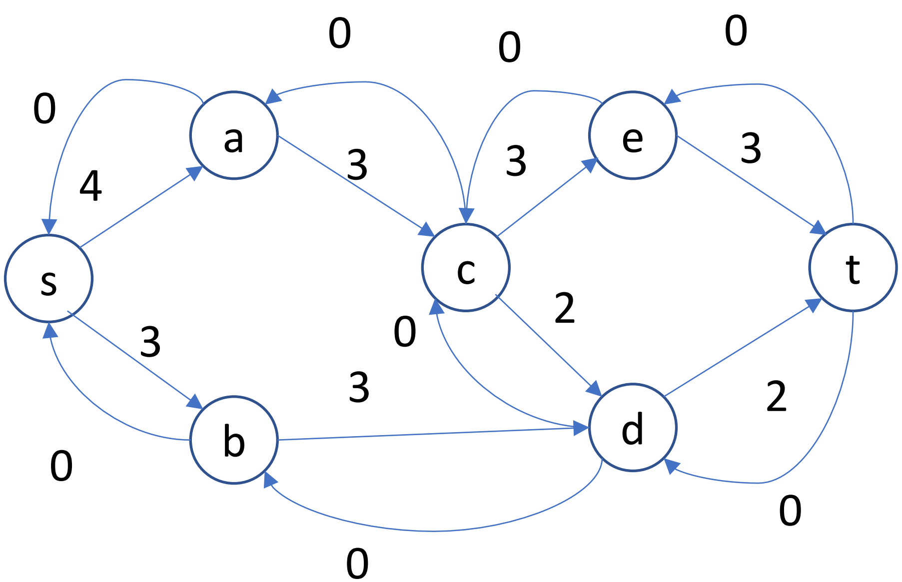
アルゴリズムの説明の流れ

- 貪欲アルゴリズムではだめか?
- 最大フローを求めるための基本アイデア
- Ford-Fulkersonのアルゴリズムの簡単な説明
- Dinicのアルゴリズム

Dinicのアルゴリズム(1/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す.

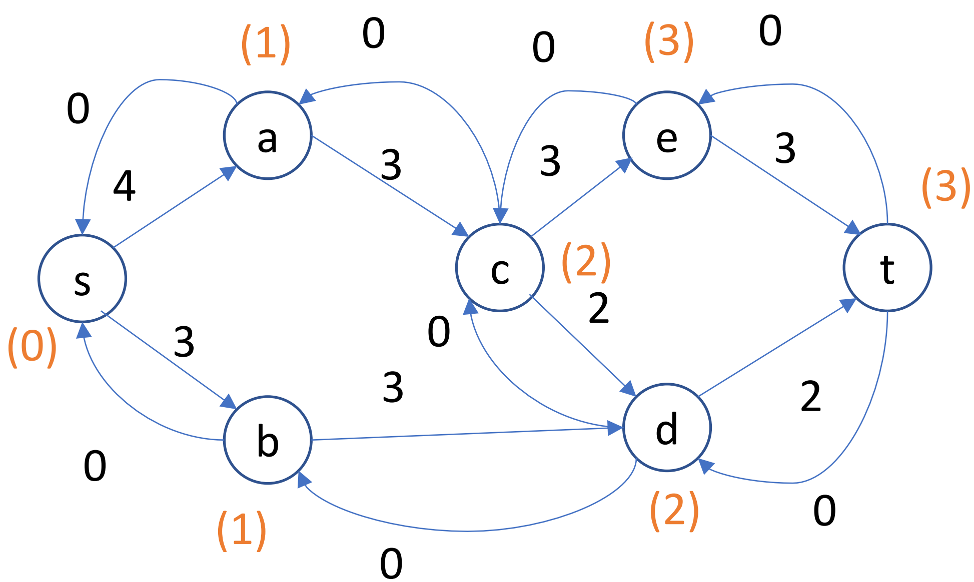
1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める.
2. 終点へ到達しなければ終了する.
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す.



Dinicのアルゴリズム(2/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す.

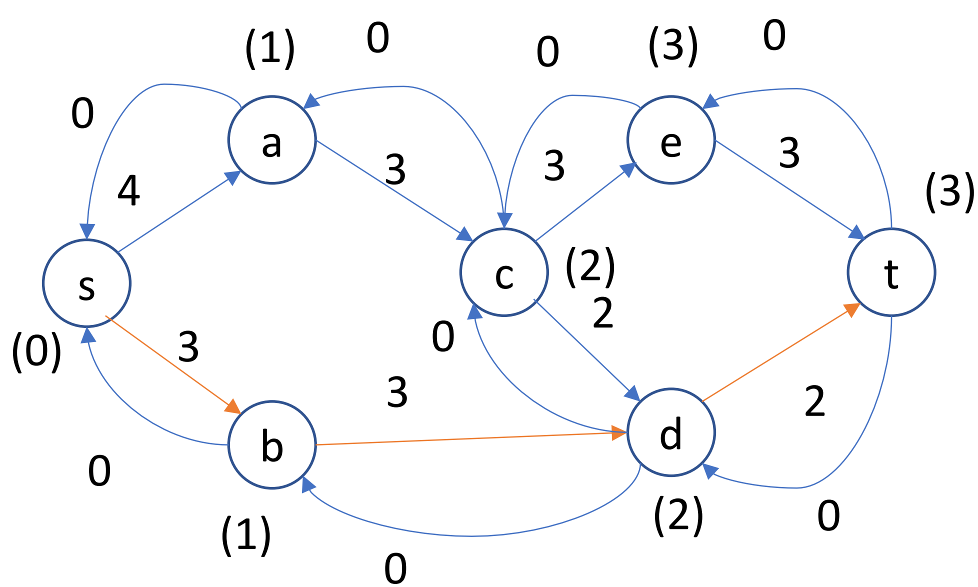
1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める.
2. 終点へ到達しなければ終了する.
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す.



Dinicのアルゴリズム(3/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す.

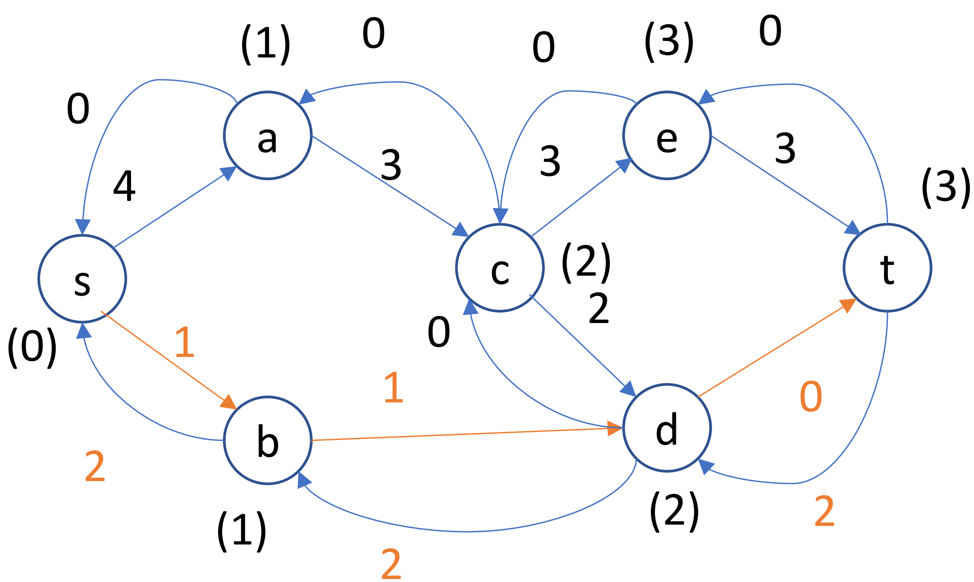
1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める.
2. 終点へ到達しなければ終了する.
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す.



Dinicのアルゴリズム(4/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す.

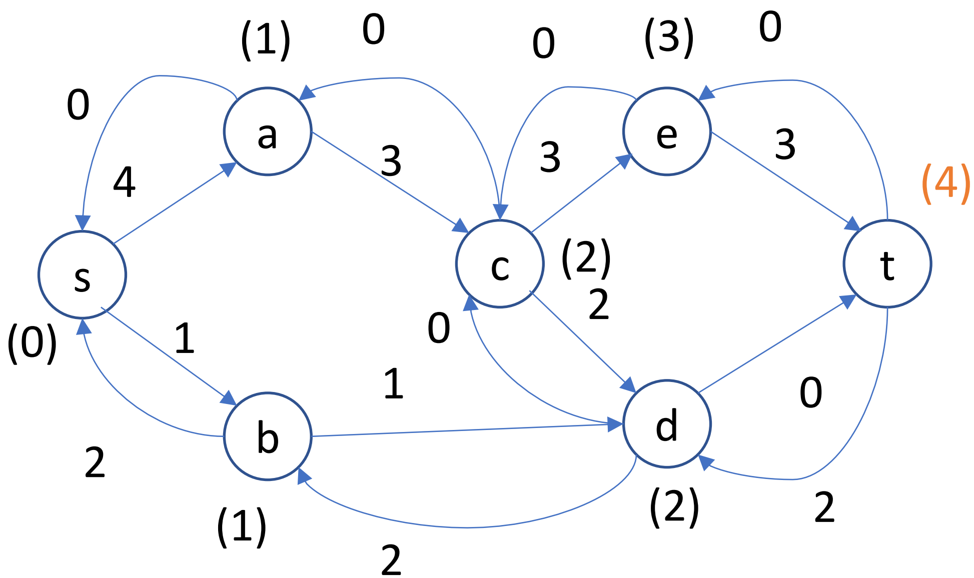
1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める.
2. 終点へ到達しなければ終了する.
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す.



Dinicのアルゴリズム(5/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す.

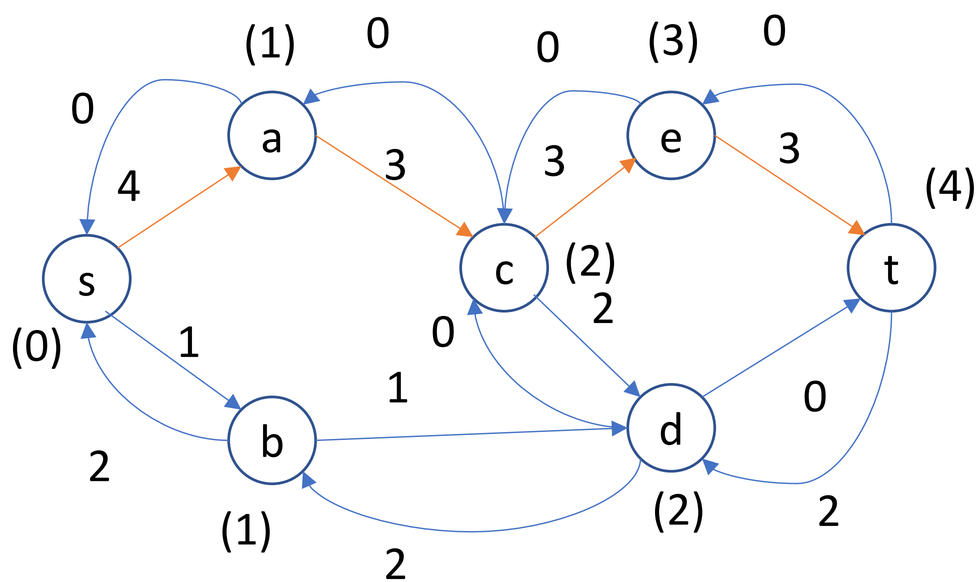
1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める.
2. 終点へ到達しなければ終了する.
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す.



Dinicのアルゴリズム(6/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す.

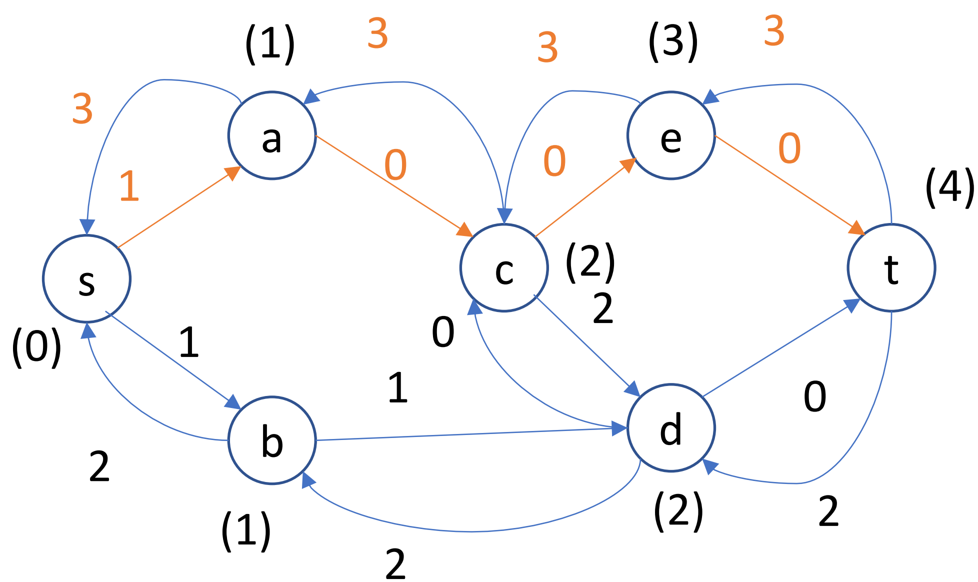
1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める.
2. 終点へ到達しなければ終了する.
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す.



Dinicのアルゴリズム(7/7)

残余グラフG_f(容量正の辺のみ見る)に対して以下を繰り返す．

1. 始点sからBFSを行い，始点から各頂点$v \in V$への距離$d(v)$を求める．
2. 終点へ到達しなければ終了する．
3. $d(w) = d(v) + 1$となる辺vwのみを用いたs-tパスが存在する間，そのパスに対して，フローをできる限り流す．

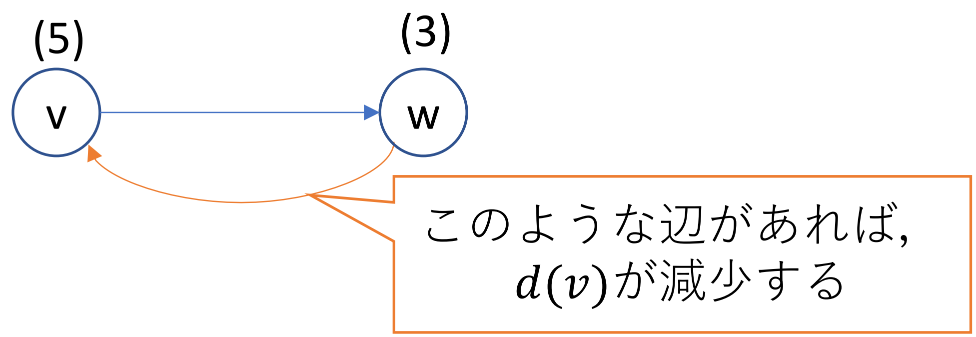


繰り返し回数が制限できる

- フローを流すとき，$d(v)$は減らない.

証明)

もし，$d(v)$が減ったとすると，そのいづれかの頂点vはフロー上のパスに含まれる．フロー上に含まれる頂点vで，$d(v)$が減ったとすると，$d(w) < d(v)$を満たす辺vwにフローを流したことになり，これは，Dinicのアルゴリズムに反する.

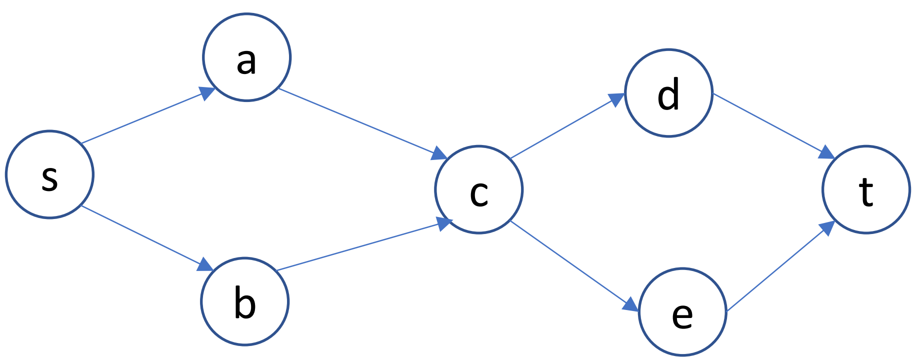


- 一度の繰り返しで，$d(t)$が1以上増えるので,

繰り返し回数は，最大で，$|V|$回

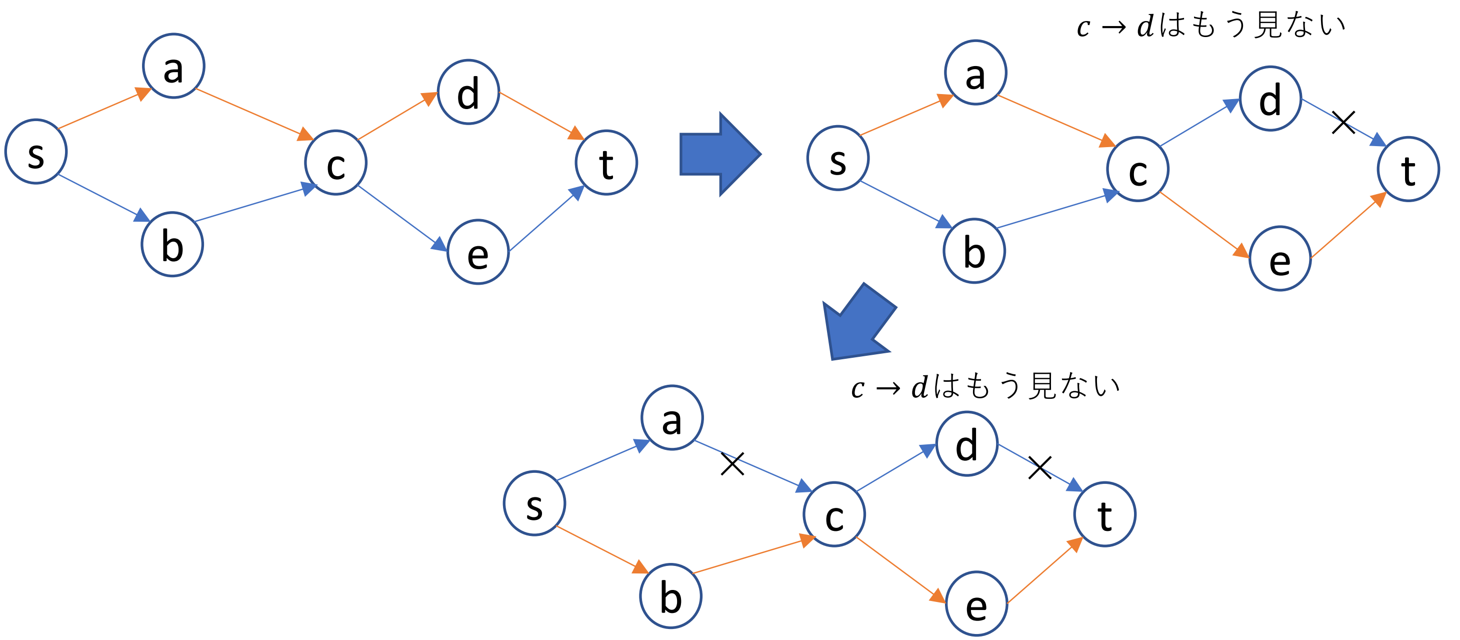
Dinicの時間計算量(繰り返し部分)

- BFSは$O(|E|)$時間で実行できる．
- フローを流す毎にいづれかの辺が使えなくなるので，フローを更新する回数は最大で，$|E|$回である．なので，フローの更新は$O(|V||E|)$時間で実行できる．



Dinicの時間計算量(DFS部分)

- フローを流す新しいパスを探すのは，DFSで行う．ただし，容量が0になった辺は見ないようにする．また，各頂点は，どの行き先までみたかを記憶しておくことで，同じ頂点を何度も見るのを防ぐ．



- 単純に，DFSを毎回やると，$O(|E|^2)$時間かかるが，上記の方法だと，同じパスを辿るのは，フローを更新したときだけなので，更新するパスを探すのは，$O(|V||E| + |E|) = O(|V||E|)$時間で実行できる．

Dinicの時間計算量

- 繰り返し回数は，最大で$|V|$回
- 繰り返し部分の時間計算量は，$O(|V||E|)$時間である．

以上のことから，Dinicの時間計算量は，
$O(|V|^2|E|)$時間である．

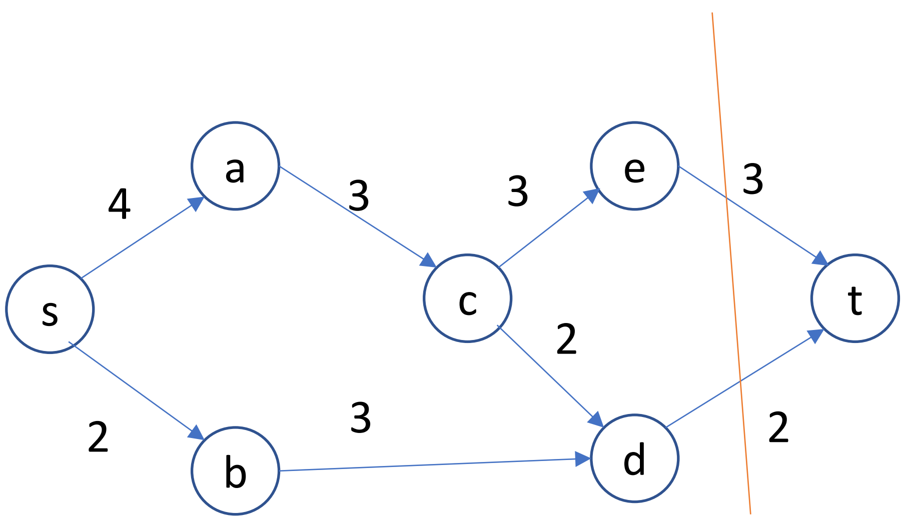
Dinicのアルゴリズムの正当性

- Ford-Fulkersonのアルゴリズムと同様の手順で証明できる.

(https://hcpc-hokudai.github.io/archive/graph_flow_002.pdf)

カット

- カットとは，頂点集合の任意の分割$S, V-S$で，Sから$V-S$に出て行く辺の集合．
- $s \in S, t \in V-S$をs-tカットといい，最小カットとは，任意のs-tカットの中で，カット容量が最小のもの．



Dinicの正当性

証明の手順

1. 任意のフローf ≤最小カットの容量
2. Dinicで得られるフロー値f' がカットの容量と等しいs-tカットが存在する.

これを示すと，f'が最大フローであることを示せる.

さらに，最大フロー・最小カットの定理も示される.

任意のフローf ≤最小カットの容量

任意のフローfと任意のカット$(S, V-S)$について考える.

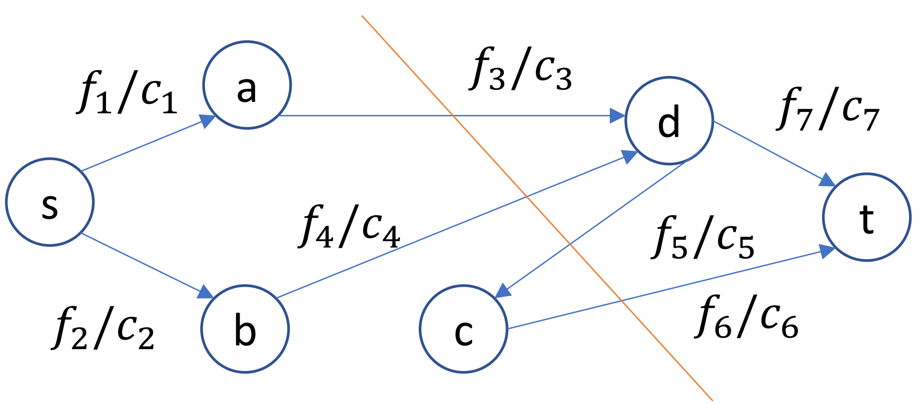
$f = \sum_{e\in\delta_+(s)} f(e)$なので,

$v \in S-\{s\}$に対して，$\sum_{e\in\delta_-(v)} f(e) = \sum_{e\in\delta_+(v)} f(e)$であるので,

(fの流量) = (Sから出ていく辺の流量)-(Sに入ってくる辺の流量)

≤ (Sから出ていく辺の流量)

≤ (Sから出ていく辺の容量)　= カットの容量



フローの流量

$f_3 + f_4 - f_5 + f_6$

カットの容量

$c_3 + c_4 + c_5 + c_6$

f'の流量= 最小カットの容量

Dinicのアルゴリズムが終了したとき，残余グラフにs-tパスは，存在しない．

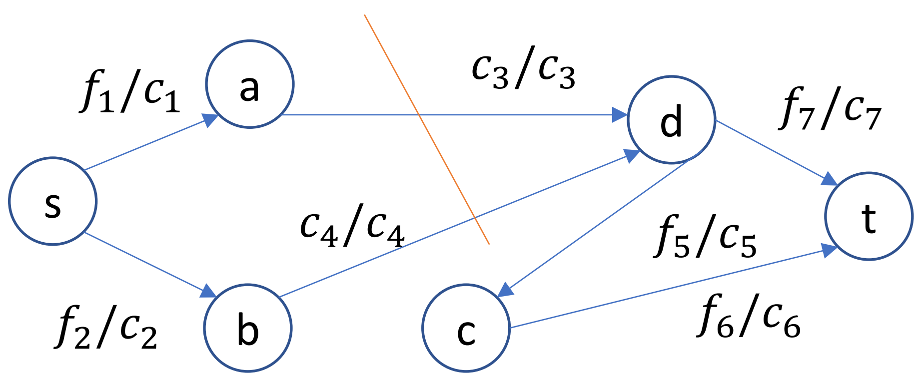
s-vパスが存在するような頂点の集合をSとする．このとき，$(S, V - S)$は，s-tカットである．

また，Sから$V - S$に向かう辺eでは，$f'(e) = c(e)$が成り立ち，

$V - S$からSに向かう辺eでは，$f'(e) = 0$が成り立つ．

よって，f'の流量=(Sから出ていく辺の流量)-(Sに入る辺の流量)

= (カットの容量)

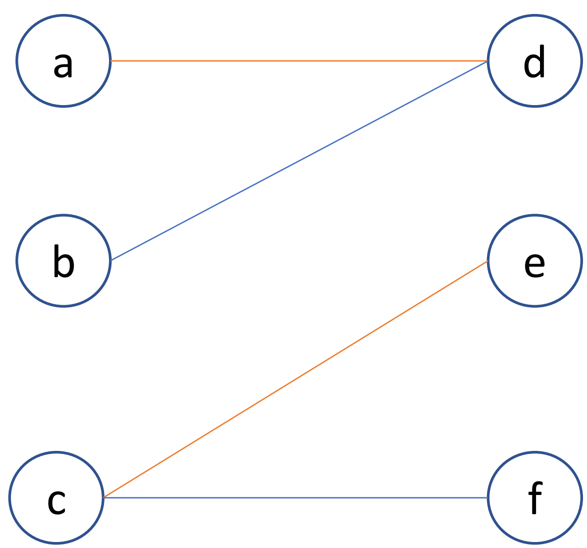


本日の流れ

- 最大流の説明
- 最大流アルゴリズムの説明
- 最大流の応用
  - 二部グラフのマッチング
  - 最小流量制約付き最大流

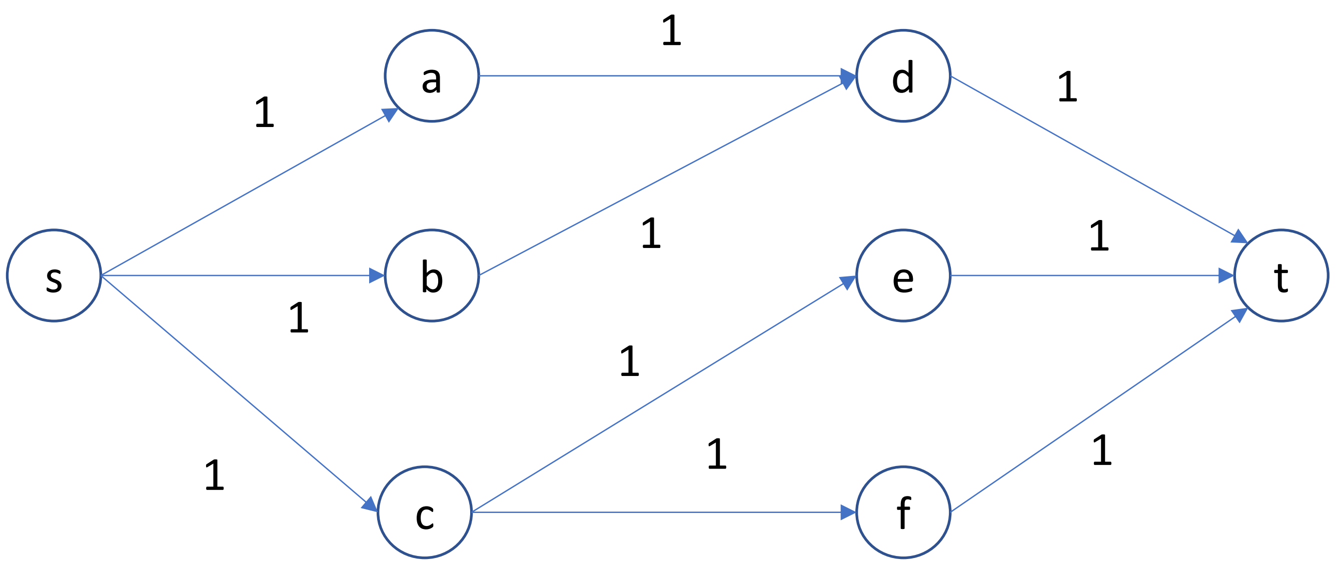
二部グラフのマッチング

- 二部グラフ
  - グラフを2つの部分集合に分割したとき，各集合内の頂点の間に辺が存在しないグラフ.
- マッチング
  - 辺の集合であって，どの 2 辺も端点を共有しないもの



二部グラフの最大流の解法

- 二部グラフのマッチングは最大流で求められる.

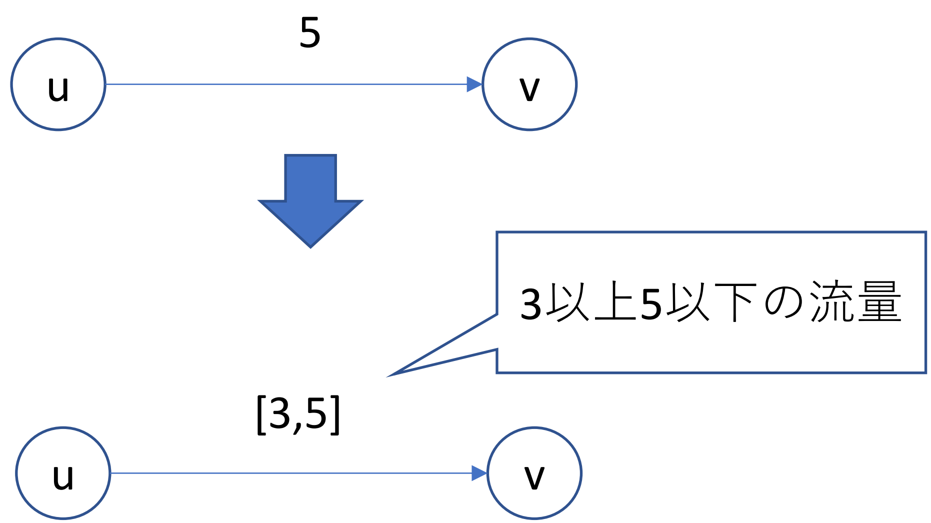


本日の流れ

- 最大流の説明
- 最大流アルゴリズムの説明
- 最大流の応用
  - 二部グラフのマッチング
  - 最小流量制約付き最大流

最小流量制約付き最大流

- 各辺に最大流量の制約だけでなく，最小流量$b(e)$の制約も考える.

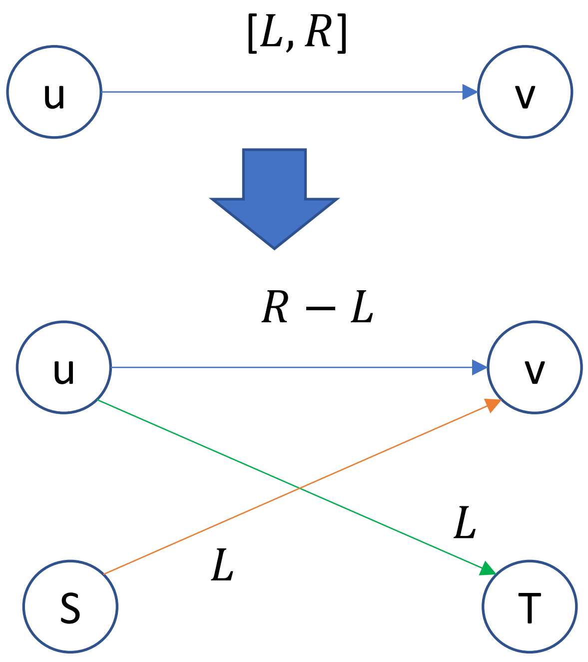


基本アイデア(1/2)

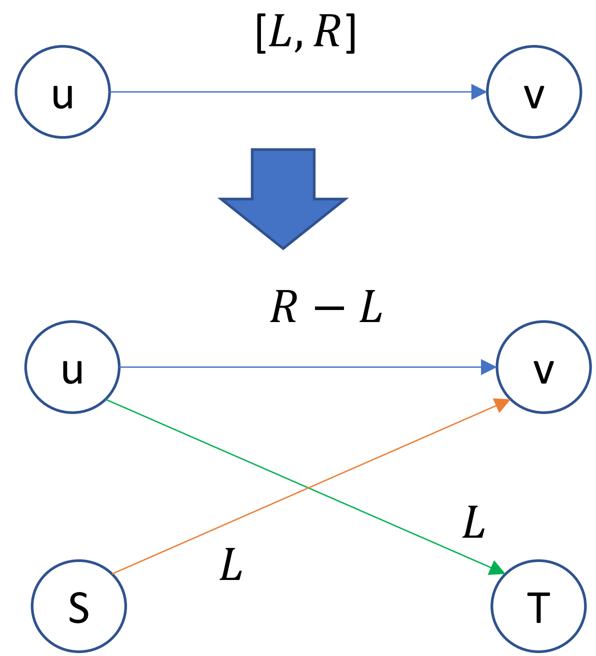
s, tとは別にS, Tを用意する.

uからvへの容量を$[L, R]$とすると,

- $u \to v$の容量を$R - L$とする.
- $u \to T$の容量をLとする.
- $S \to v$の容量をLとする.



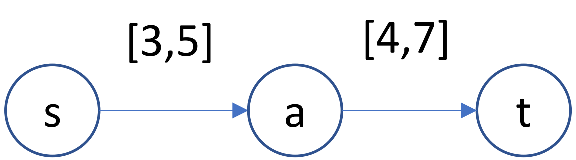
基本アイデア(2/2)

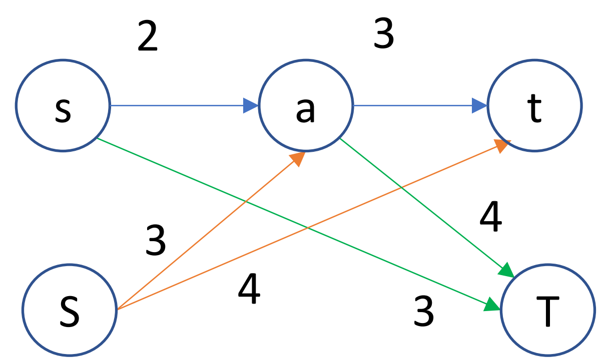


s,tとは別にS,Tを用意する.

uからvへの容量を$[L,R]$とすると,

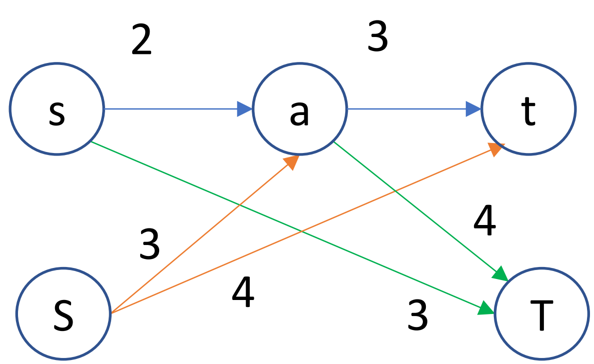
- $u \to v$の容量を$R-L$とする.
- $u \to T$の容量をLとする.
- $S \to v$の容量をLとする.





解法

$S \to T,\ s \to T,\ S \to t,\ s \to t$の順に流した結果を,
f_a, f_b, f_c, f_dとする.
Sから出た辺とTへ行く辺がすべて使われていたら,
条件を満たす最大フローが存在し，$f_b + f_d$である.
($f_a + f_c = f_a + f_b = \sum_{e \in E} b(e)$ を満たせばよい)



最大流問題

- 二部グラフのマッチング
  - AOJ 1163 カードゲーム(http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=1163&lang=en)
- 最小流量制約付き最大流
  - AOJ 1615 プレゼント交換会(http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=1615&lang=en)