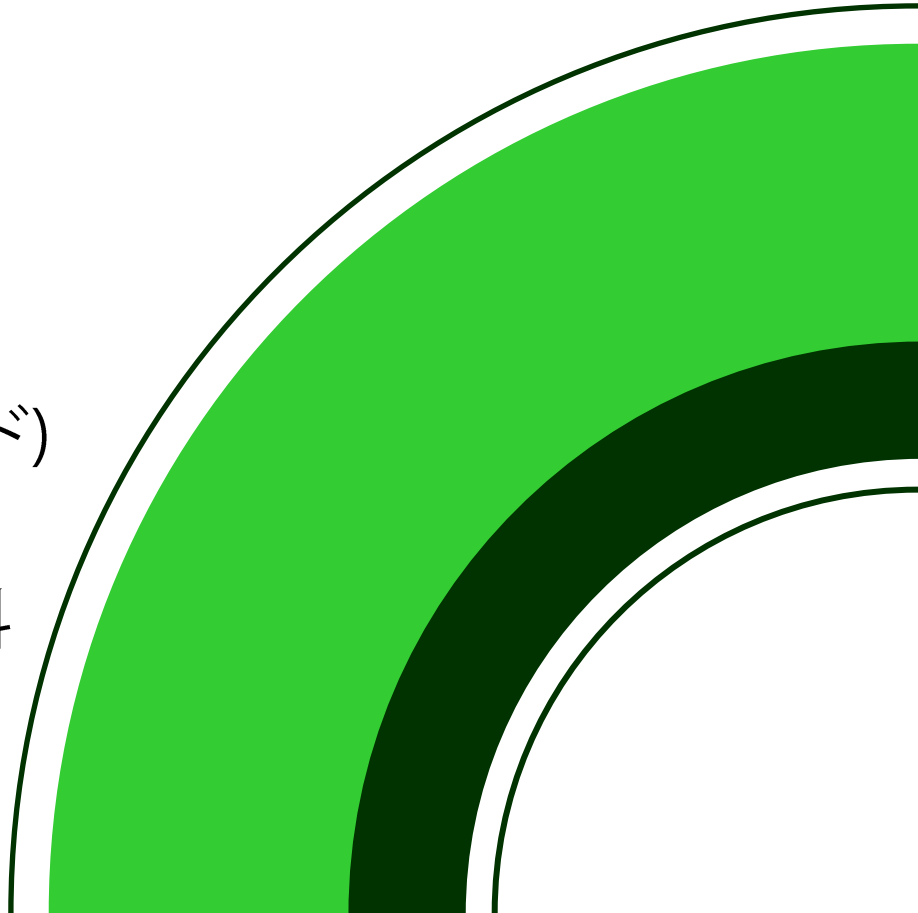
プロコン練習会Advanced フロー



2014年6月7日
(加筆・修正のうえアップロード)

京都大学工学部電気電子工学科
久留島隆史 (KMC ID: gire)

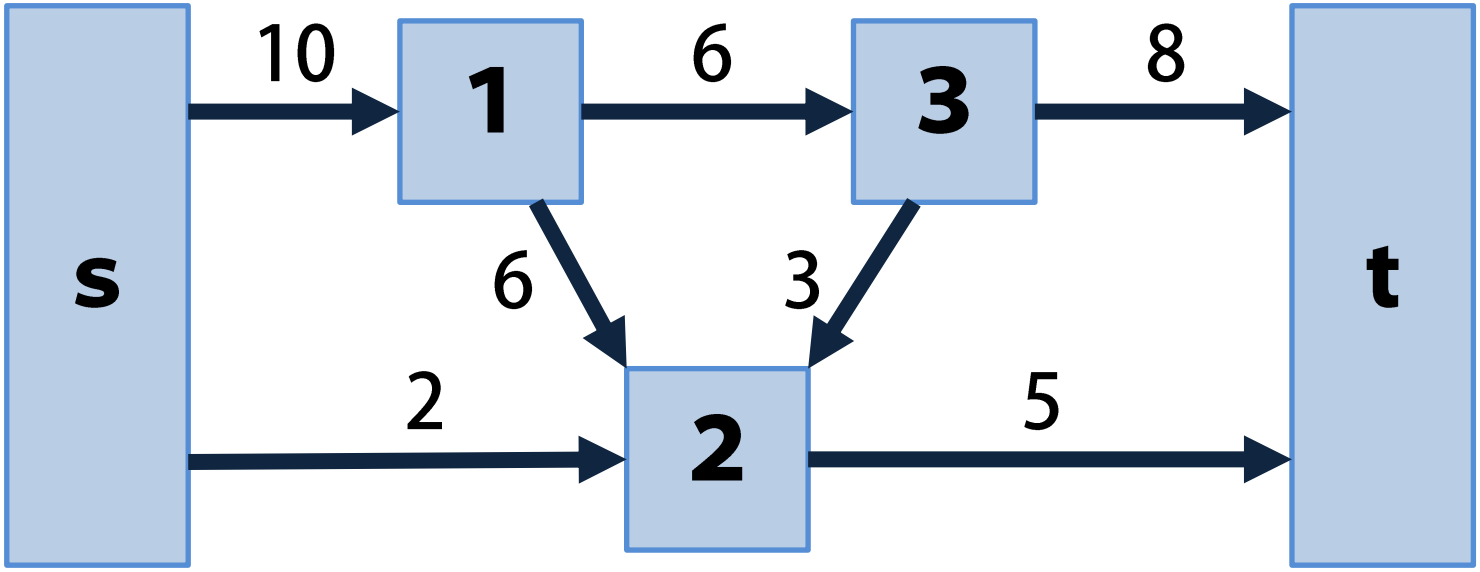
フロー

流れ(Flow)を用いる問題

- 最大流問題
- 最小カット
- 二部マッチング
- (一般マッチング)
- 最小費用問題

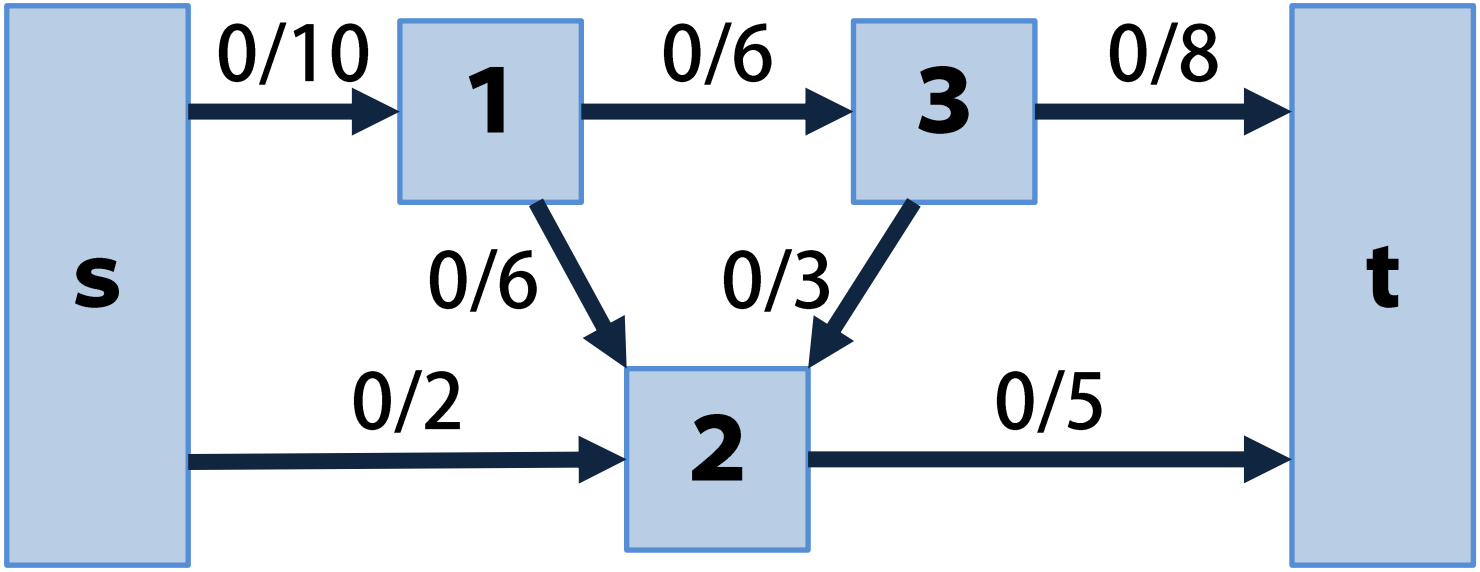
最大流問題

- 点s(Source)から点t(Sink)へ水を流す
- 最もたくさん水が流れるときの流量は？



貪欲法でやってみる

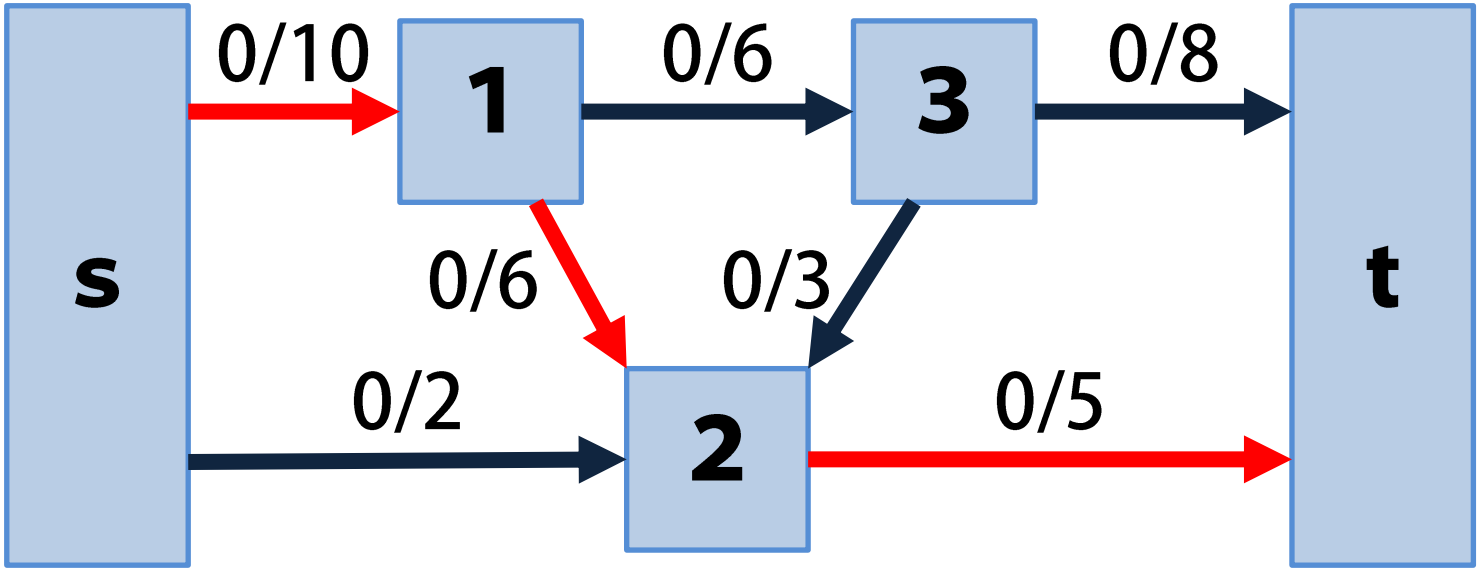
- sからtへの経路を見つけたら流す
- 見つからなくなるまでやる



貪欲法でやってみる

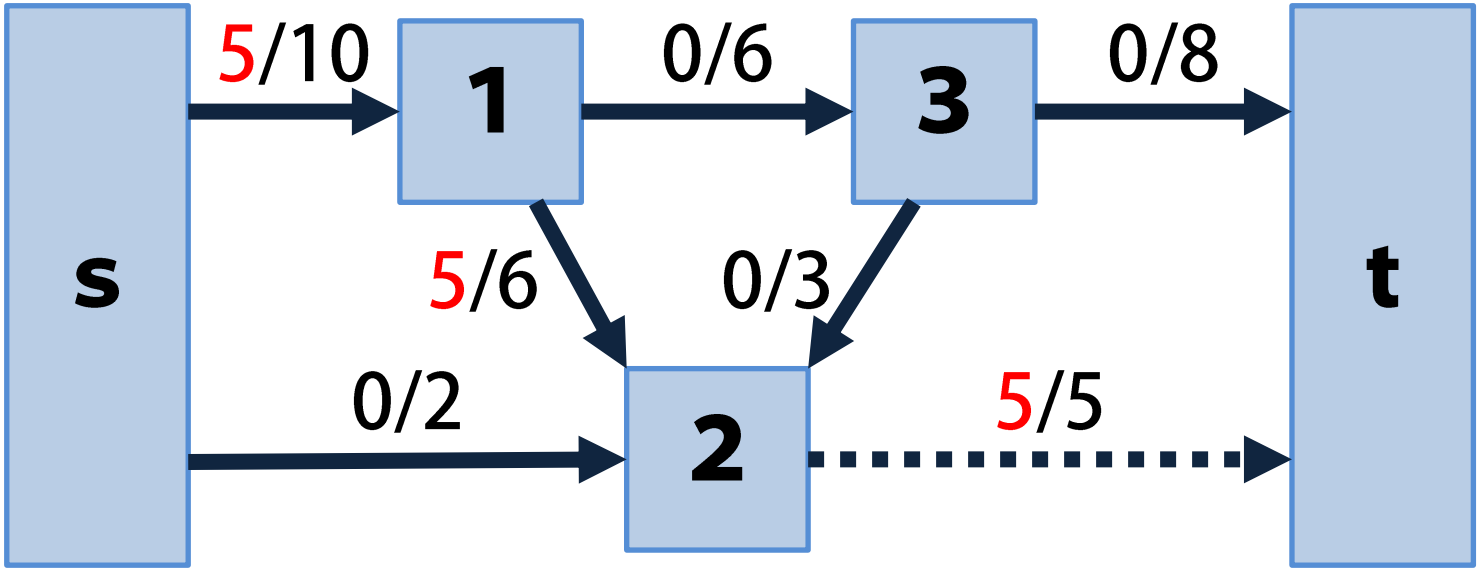
s→1→2→tという経路を発見

この経路で流せる最大量は5



貪欲法でやってみる

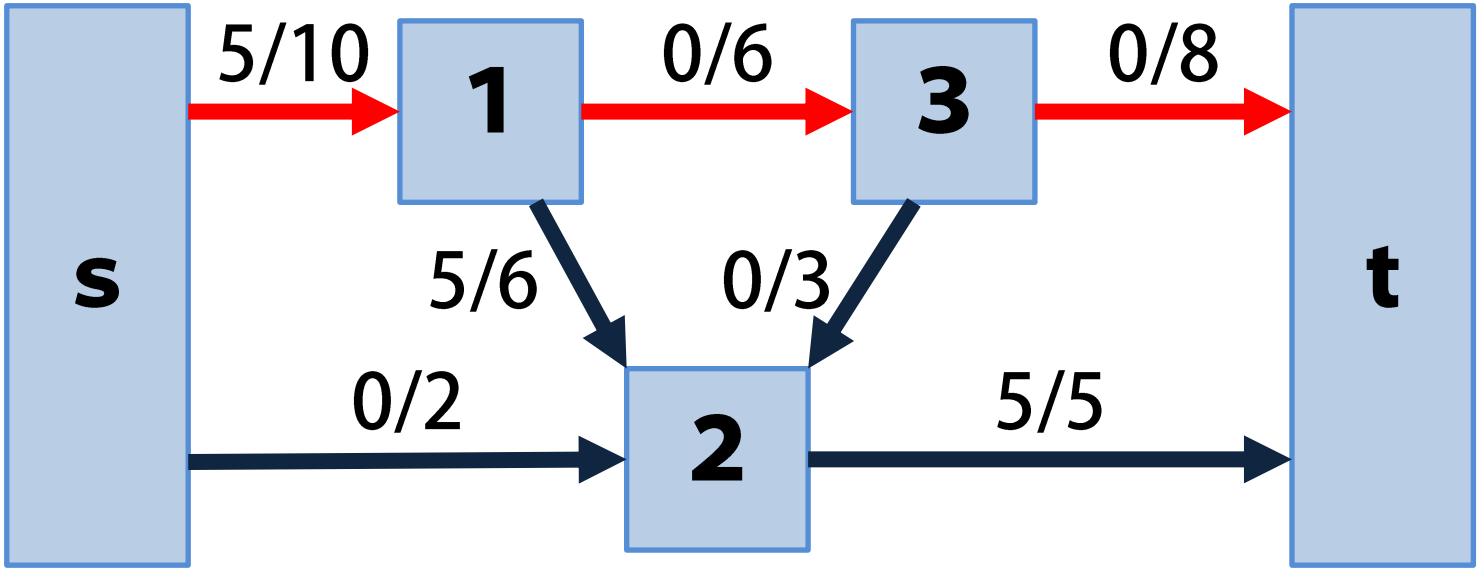
流してみた



貪欲法でやってみる

s→1→3→tという経路を発見

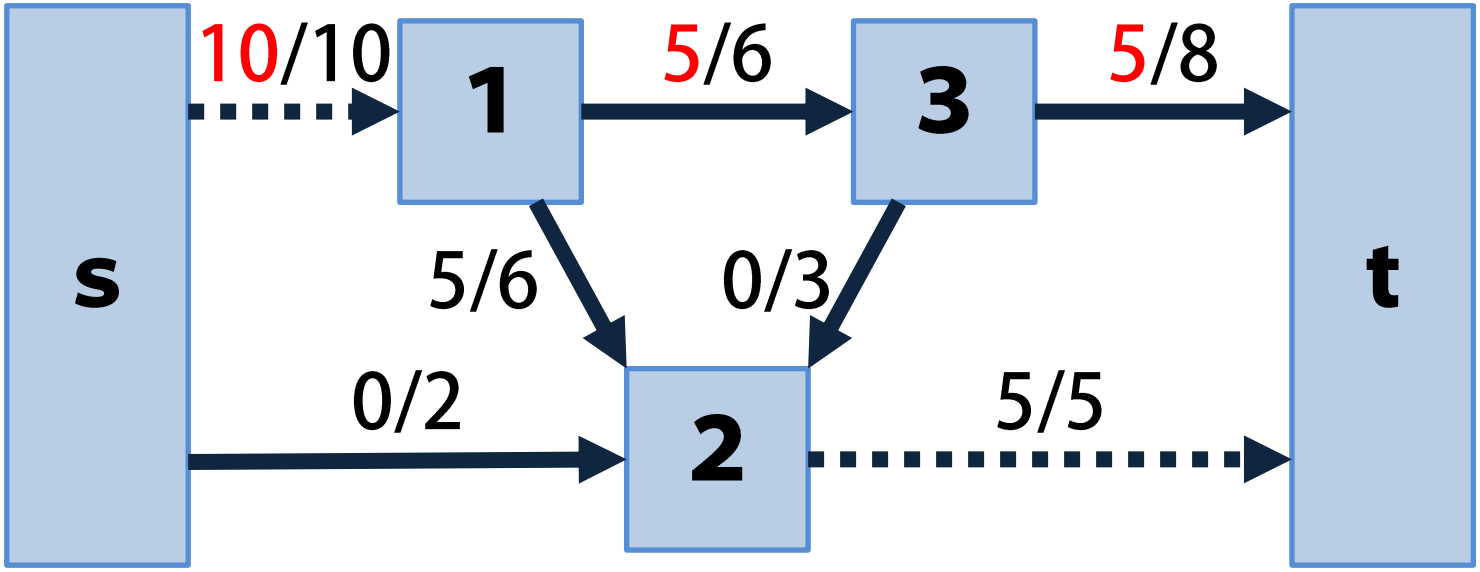
現状ではこの経路で流せる最大量は5



貪欲法でやってみる

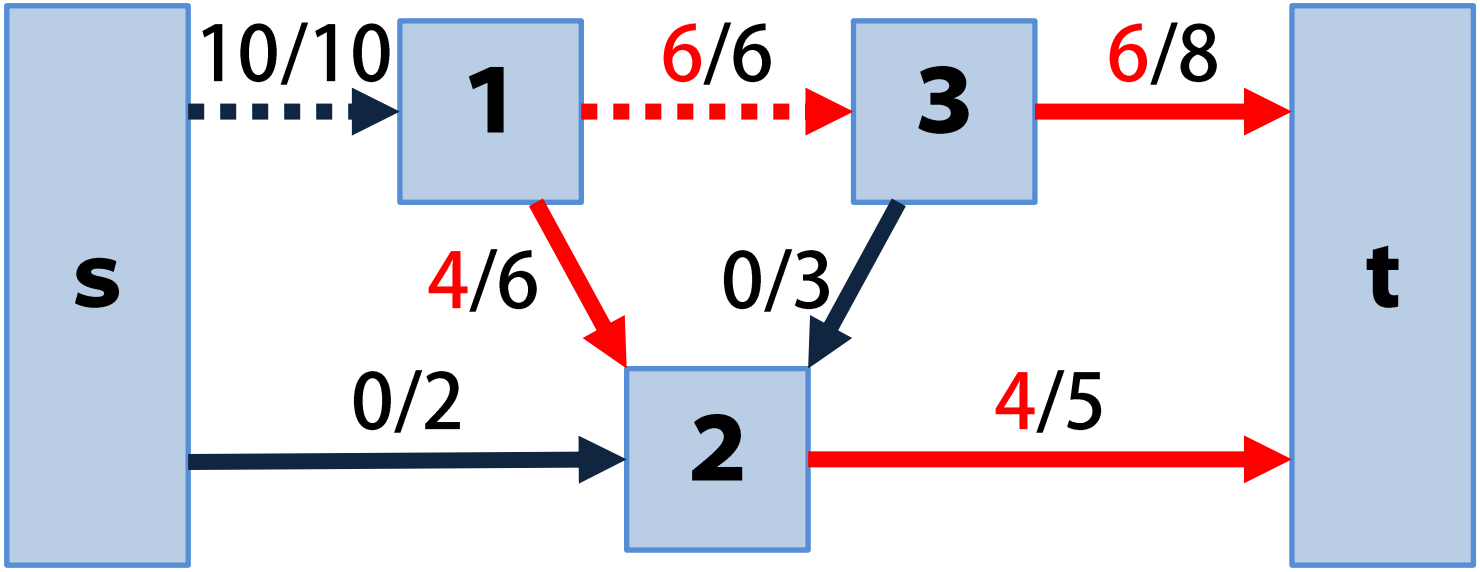
流してみた

もう流せないので最大流は10



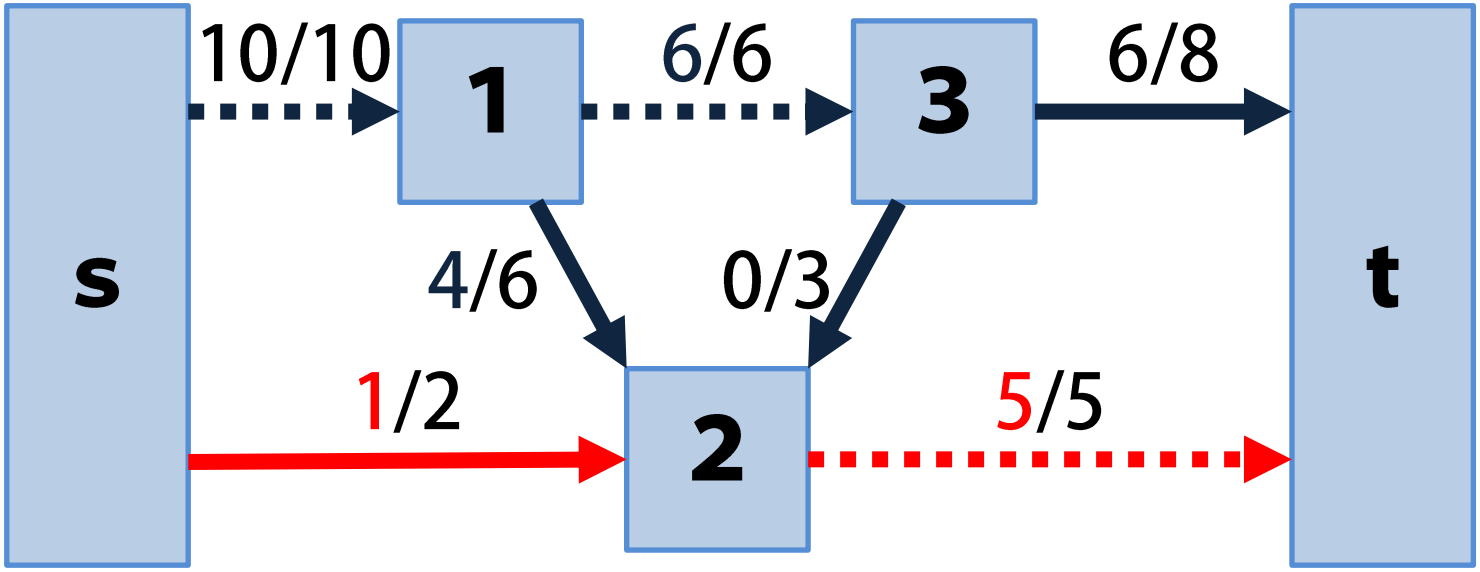
しかし

- s→1→2→tを1減、 s→1→3→tを1増



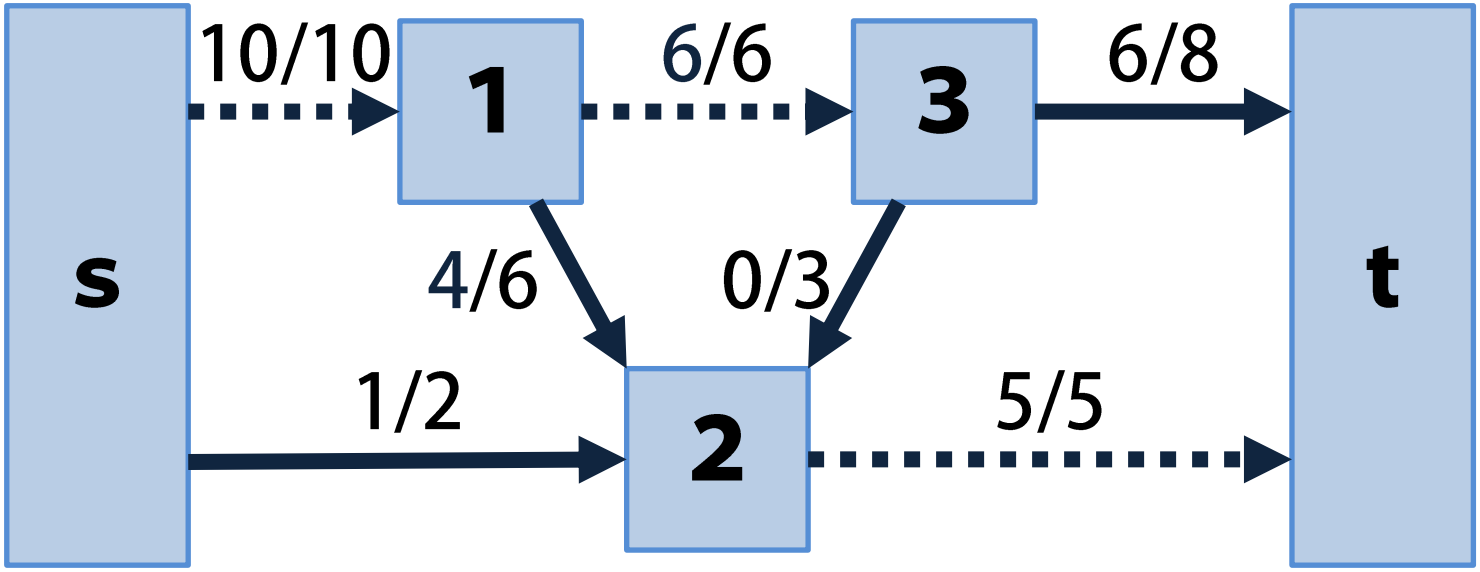
しかし

- s→1→2→tを1減、 s→1→3→tを1増
- s→2→tを1増



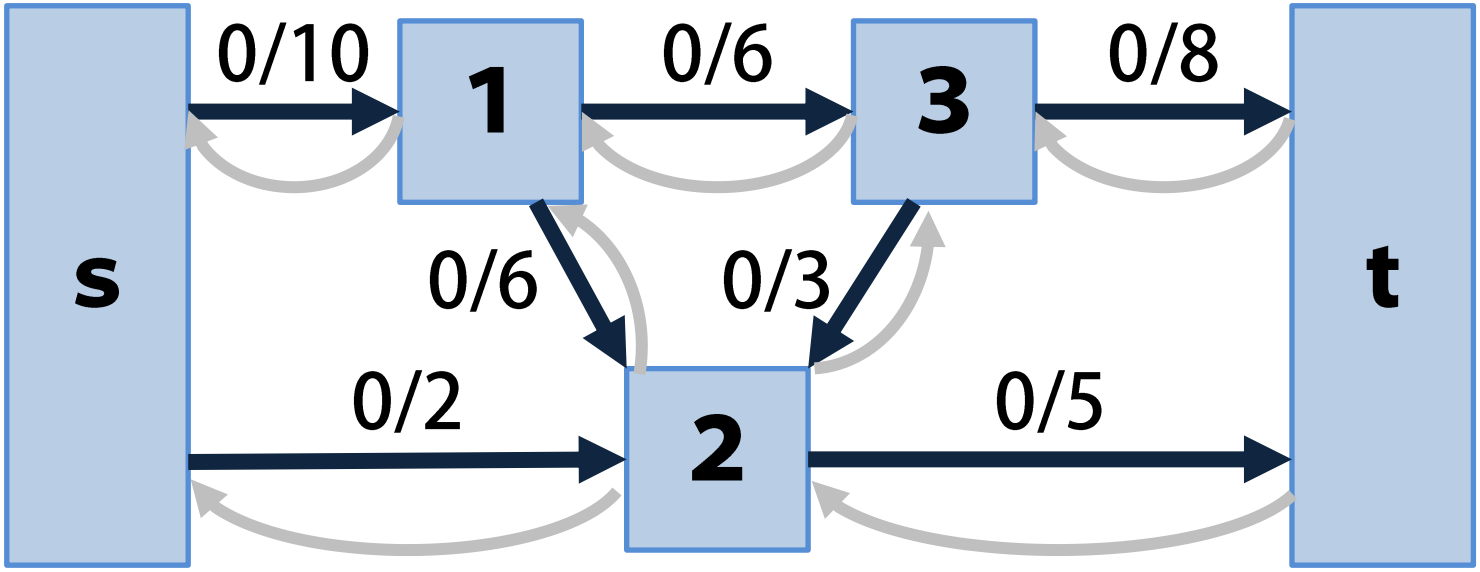
しかし

- s→1→2→tを1減、 s→1→3→tを1増
- s→2→tを1増　**最大流が11になった！**



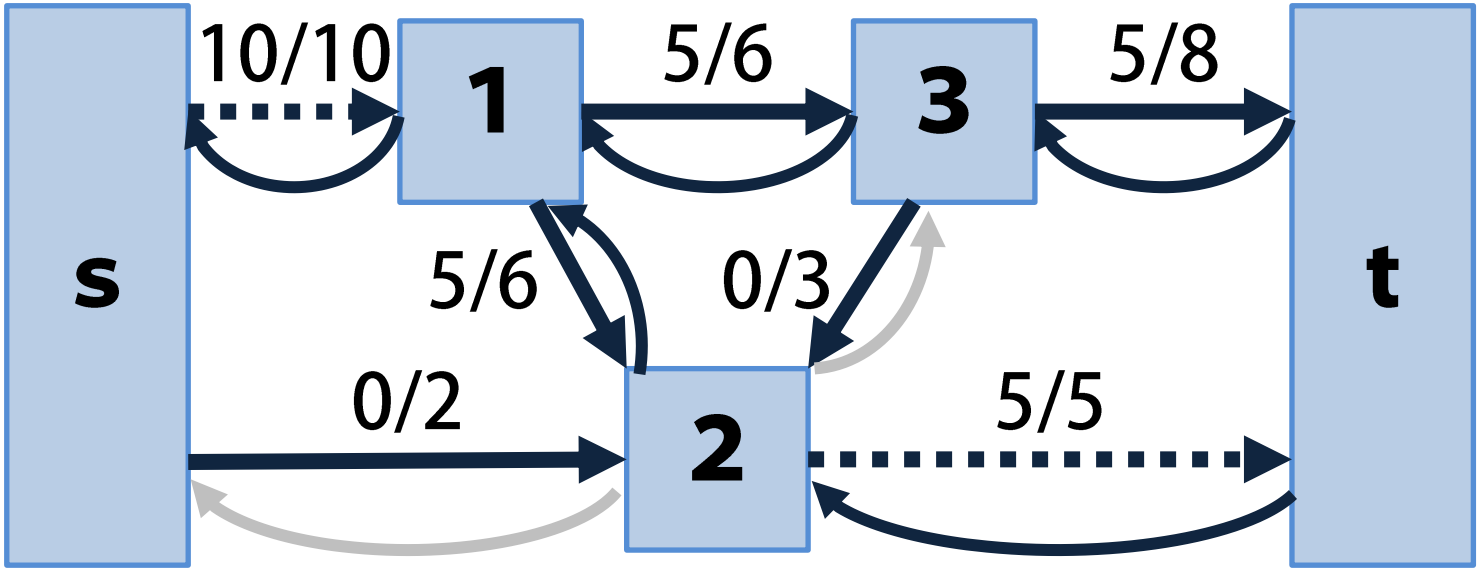
Ford-FulKerson algorithm

- 流れと逆方向の辺(逆辺)を考える
- まだ流せる辺と流れている辺の逆辺で貪欲法



Ford-FulKerson algorithm

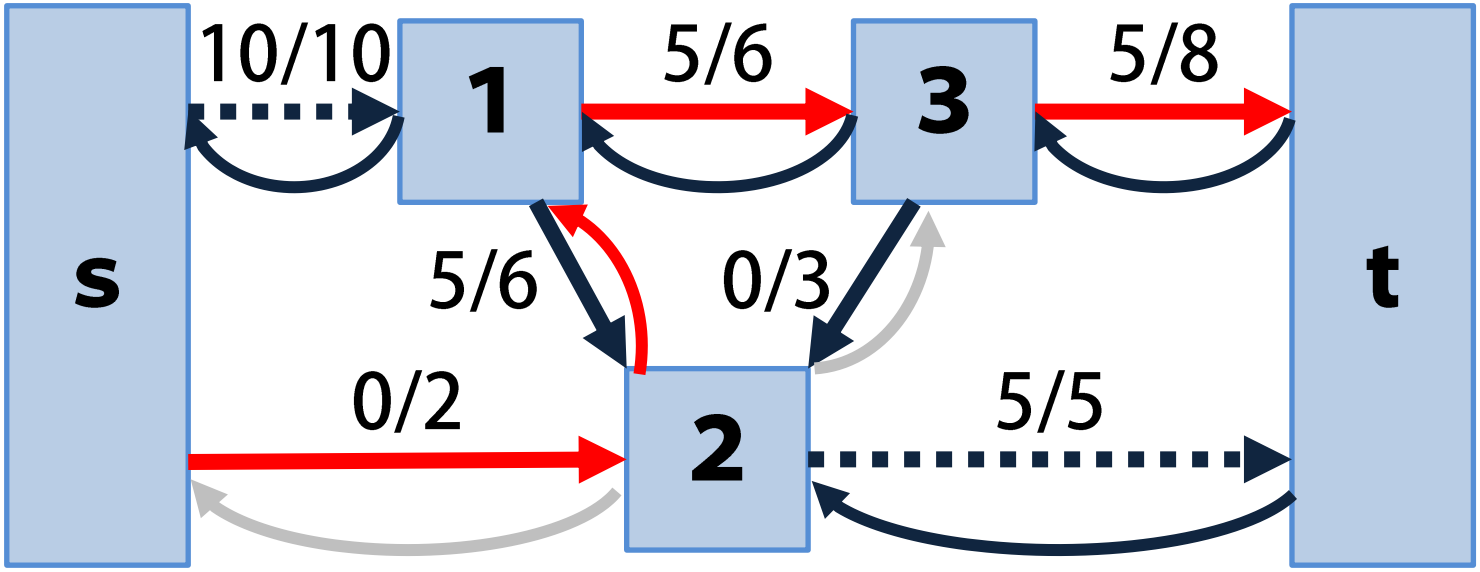
さきほど貪欲法で失敗したところを再現



Ford-FulKerson algorithm

さきほど貪欲法で失敗したところを再現

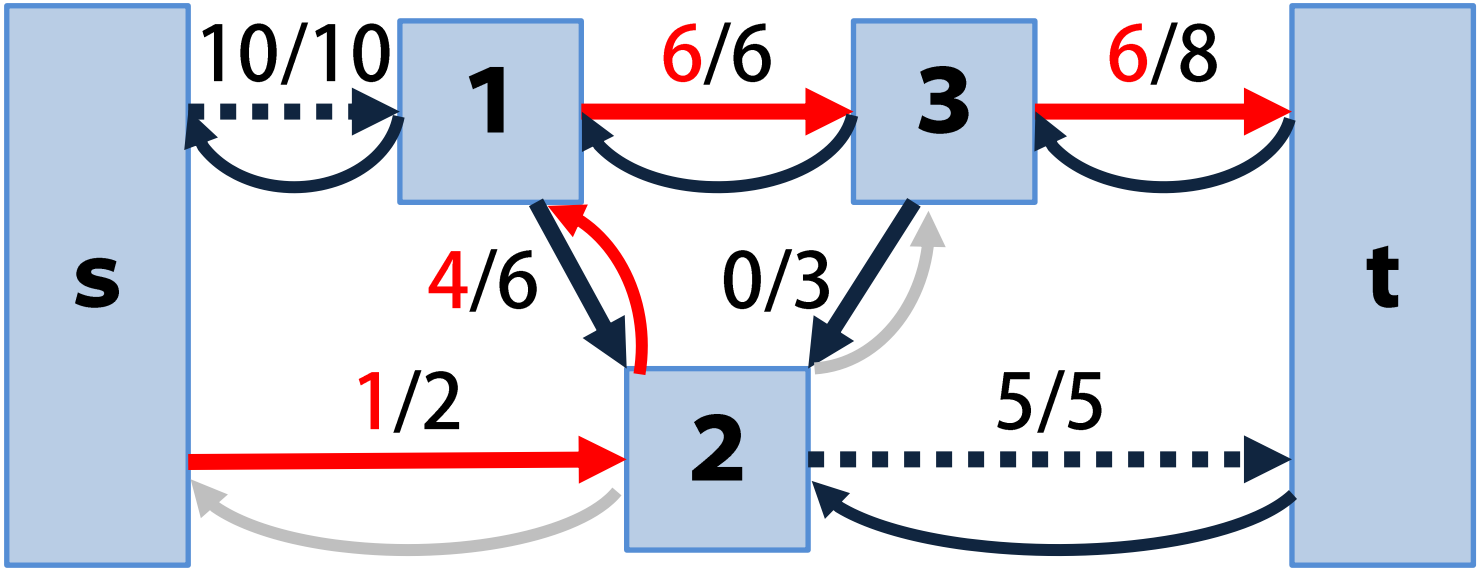
さらに1流せることがわかる！



Ford-FulKerson algorithm

さきほど貪欲法で失敗したところを再現

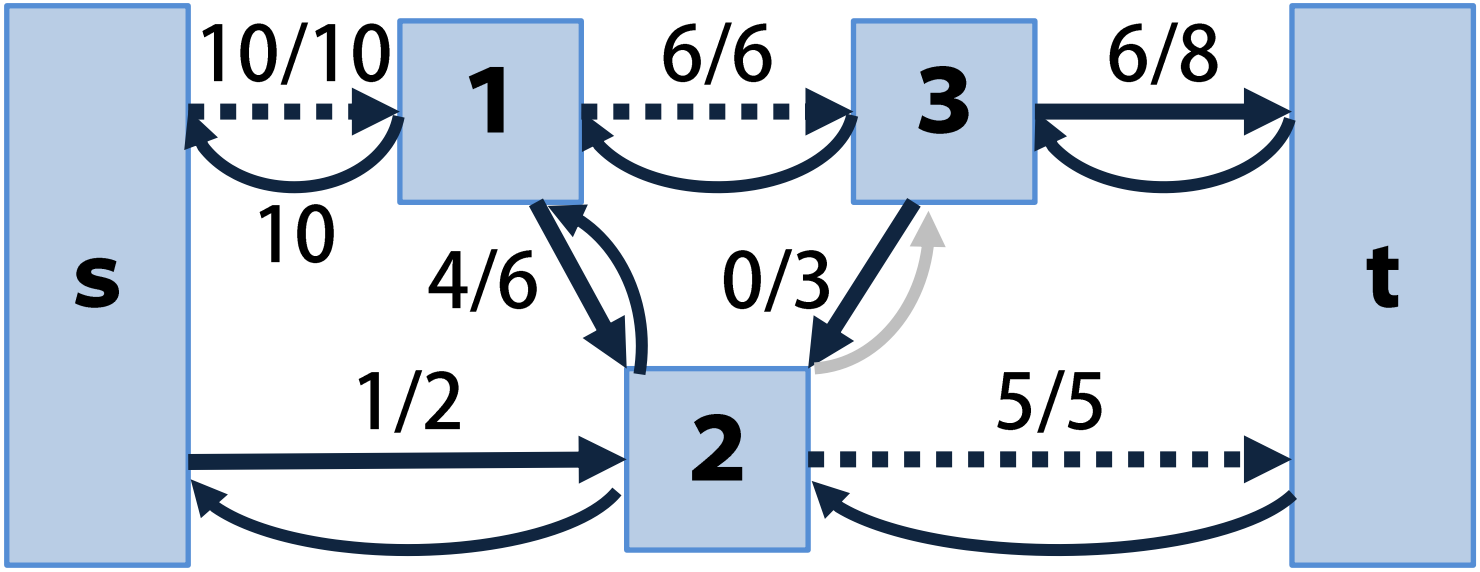
さらに1流せることがわかる！



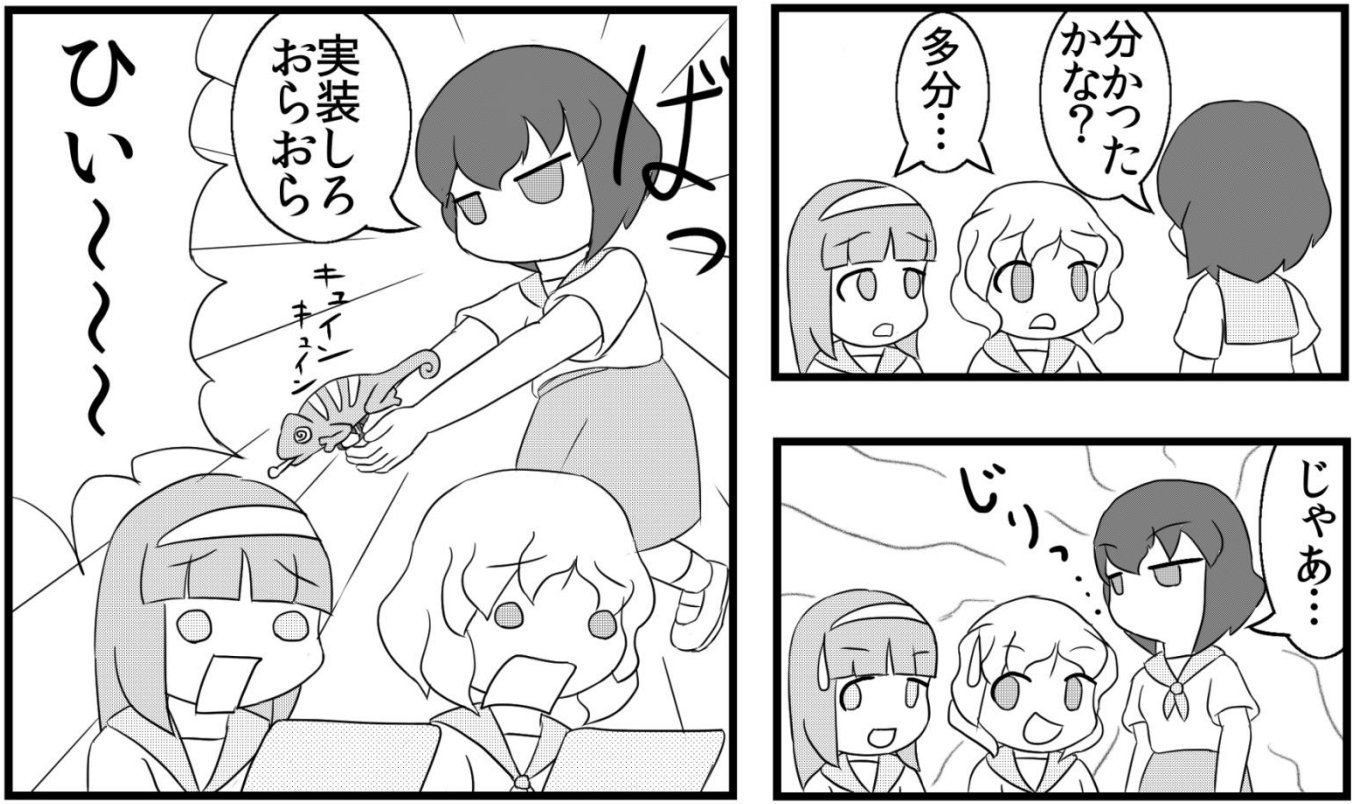
Ford-FulKerson algorithm

これでもう流せないとわかる

アルゴリズムの正当性の証明は略します



分かったかな？



実装(1/3)

```
//辺を表す構造体(行き先、容量、逆辺)
struct edge { int to, cap, rev; };
vector <edge> G[MAX_V]; //グラフの隣接リスト表現
bool used[MAX_V]; // DFSですでに調べたかのフラグ

// fromからtoへ向かう容量capの辺をグラフに追加する
void add_edge(int from, int to, int cap) {
   G[from].push_back( (edge) {to, cap, G[to].size()});
   G[to].push_back( (edge) {from, 0, G[from].size() -1});
}
```

実装(2/3)

```
int dfs(int v, int t, int f) {//増加パスをDFSで探す(vからtへfだけ流そうとする)
    if (v == t) return f;
    used[v] = true;
    for (int i=0; i < G[v].size(); i++) {
        edge &e = G[v][i];
        if (!used[e.to] && e.cap > 0) {
            int d = dfs(e.to, t, min(f, e.cap));
            if (d > 0) {
                e.cap -= d;
                G[e.to][e.rev].cap += d;
                return d;
            }
        }
    }
    return 0;
}
```

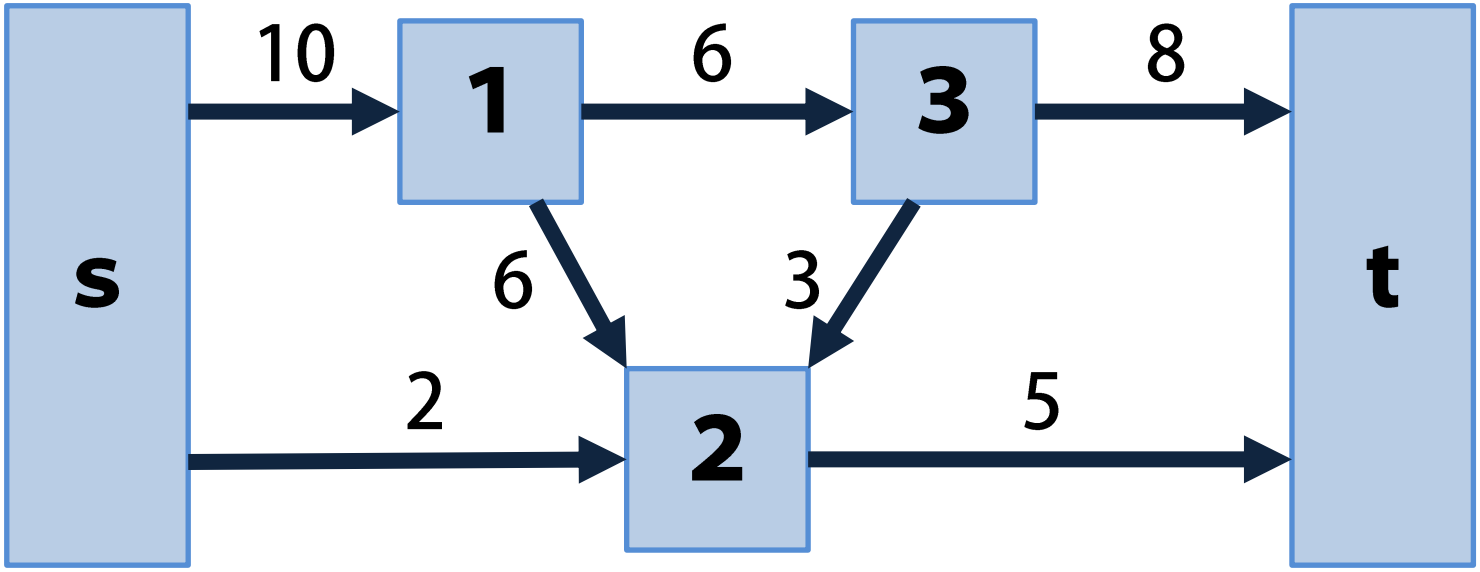
実装(3/3)

```
// sからtへの最大流を求める
int max_flow(int s, int t) {
    int flow = 0;
    for (;;) {
        memset(used, 0, sizeof(used));
        int f = dfs(s, t, INF);
        if (f == 0) return flow;
        flow += f;
    }
}
```

最悪計算量：最大流をFとすると$O(F|E|)$

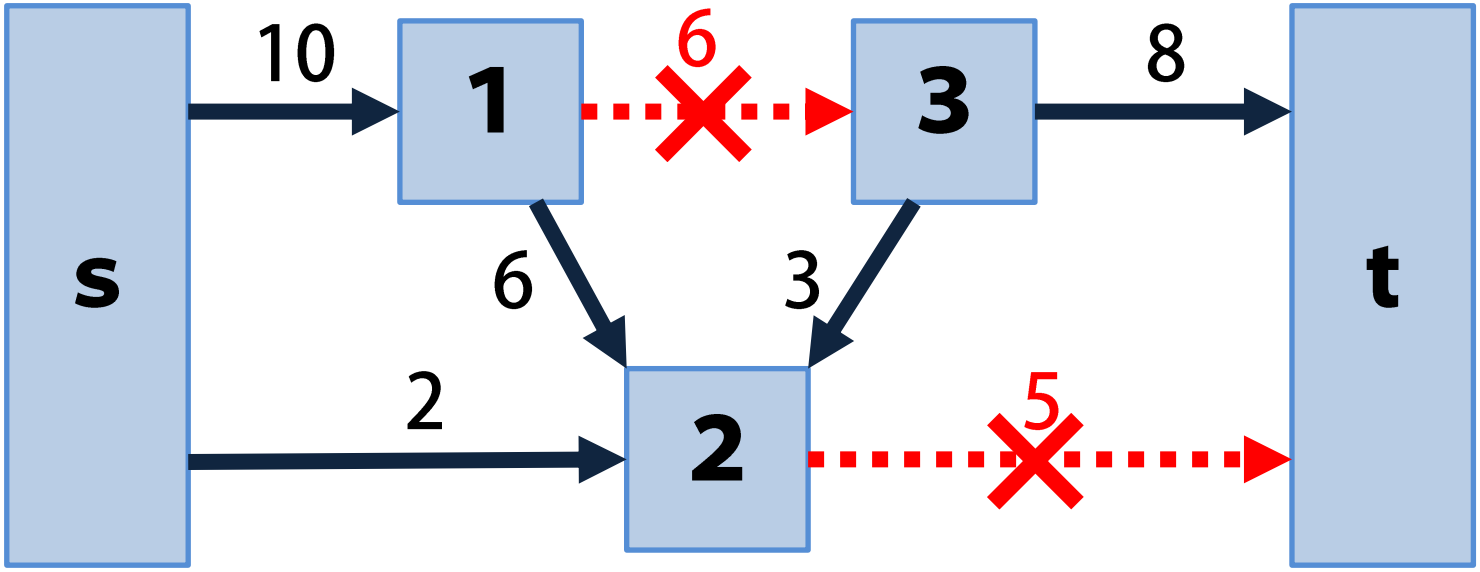
最小カット

- 点sから点tへ流れる水をせきとめる
- 除去すべき辺の和の最小値は？



最小カット

- 点sから点tへ流れる水をせきとめる
- 除去すべき辺の和の最小値は？



最小カット

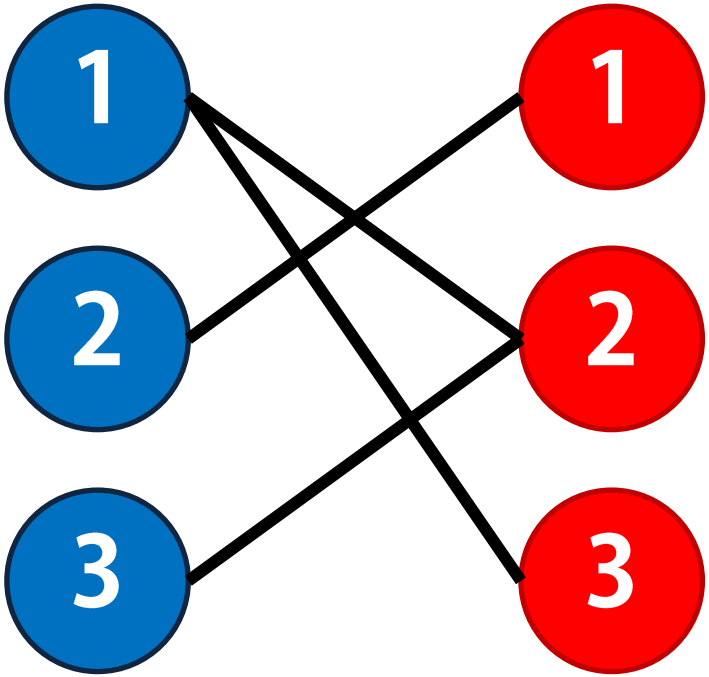
- 点sから点tへ流れる水をせきとめる
- 除去すべき辺の和の最小値は？

答えは最大流と同じ値

証明略
(情報学科2回後期・電気電子工学科3回後期で開講される「グラフ理論」を参照)

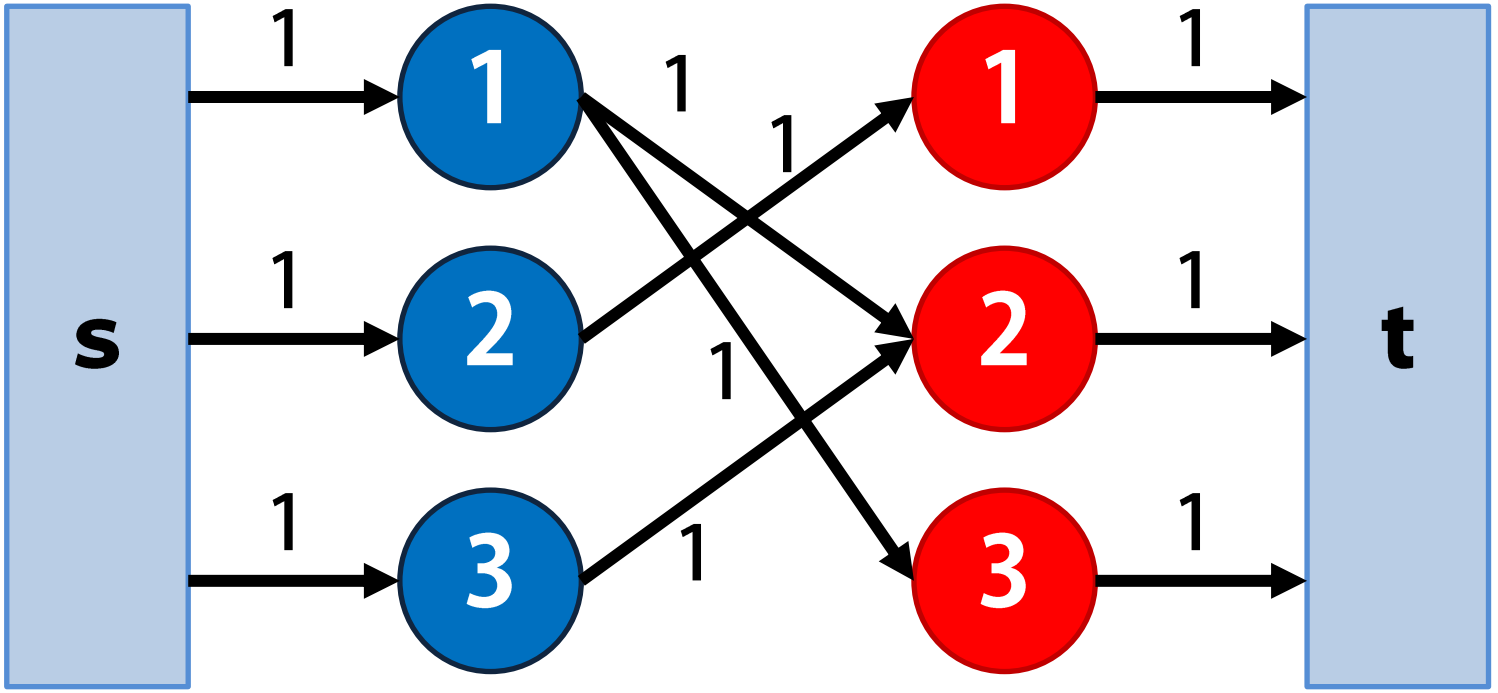
二部マッチング

男女でフォークダンスをします
友達同士のペアは最大で何組できるか？



二部マッチング

こうすれば最大流問題になる



一般マッチング

「二人組作ってー」

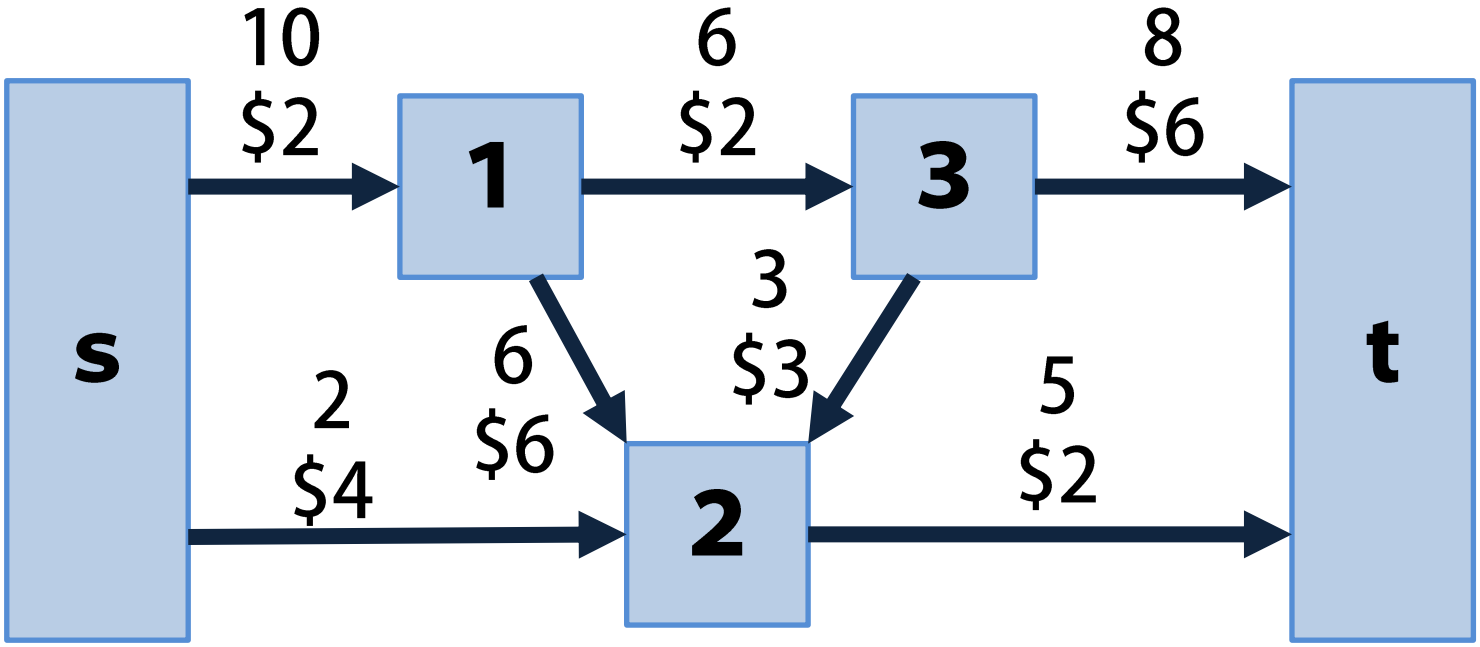
複雑すぎて競プロで扱うことはまずない

完全マッチング(全員友達と組めるか)は
tutteの行列というもので判断できる

これも行列式・恒等式が出てくるので激ムズ

最小費用流

- 川を開通してsからtにＦだけ水を流す
- 最も安くしたときにかかる費用は？



方針

- 最大流と同じように逆辺をはる
  (逆辺に流す費用)=(元の辺に流す費用)×(－1)
- tまで流すのに安い経路から順に採用
  最短経路問題
  負の費用があるのでBellman-Ford法
- Ｆ流せるようになったらそこで中断

問題を解いてみよう

競技プログラミング練習会 Advanced

おつかれさまでした

激ムズ！

ライブラリがないと太刀打ちできないね

われらが先輩cosさんのGithubとかもあるよ

日頃から蟻本写経などもしっかりやろう

ICPC参加しようぜ！

申込の時期だぜ！

次回はalphaくんによるbitDPです

次々回以降はNormalと合体するかも？