G:DAG-Trio-hard

原案：Yazaten

問題文:ixmel

解説:T.M

概要

- 問題設定はDと同じなので割愛
- DとGがDAGTrio

誤解法

- Dと同じように1本取って橋を列挙
- 橋の列挙にO(E)
- 各辺についてやるからO(E)回

- 全体でO(E^2)
- 間に合わない

解法

- まず与えられたグラフがDAGかそうでないか調べる

- 調べ方はトポロジカルソートや強連結成分分解など
- DFSでもできる

解法（DAGでない）

- DAGでない時はDとほぼ一緒

- 3つに分解した後はそれぞれがDAGでないといけない
- そのため1つでも閉路があると、
  その閉路に含まれる弧のどれかを取らないといけない

- 取らなかった時はその閉路がなくならない

解法(DAGでない)

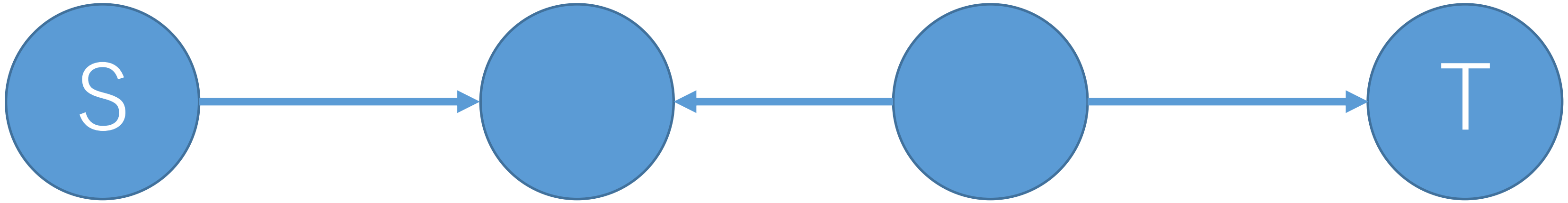
- DFSを行うと1つの閉路に含まれる弧を列挙できる
- 列挙した辺すべてに対して
  - 取り除く
  - DAGか調べる
  - 橋の列挙を行う

- 閉路に含まれる弧は高々V個
- よってO(V*E)

解法（DAGのとき）

- どこを切っても残った連結成分はかならずDAGになる
- そのため弧の向きは不要になる（むしろ邪魔）

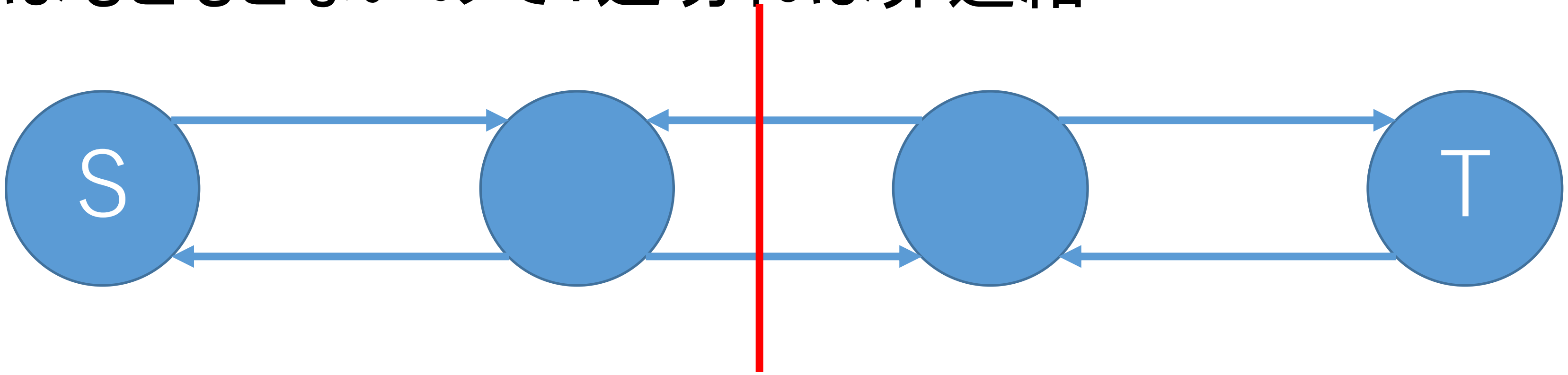
- 最小カットはSからTへの増大道はなくなるが、非連結になるとは限らない
- 増大道はないが連結のケース



解法(DAGのとき)

- そのため逆辺を張り無向グラフ化する
- 辺のコストは逆辺を含みすべて1

- こうしておけば増大道がなくなったとき非連結にすることができる
- 例えば赤線で切れば非連結になる
- 逆辺はもともとないので1辺切れば非連結



解法（DAGのとき）

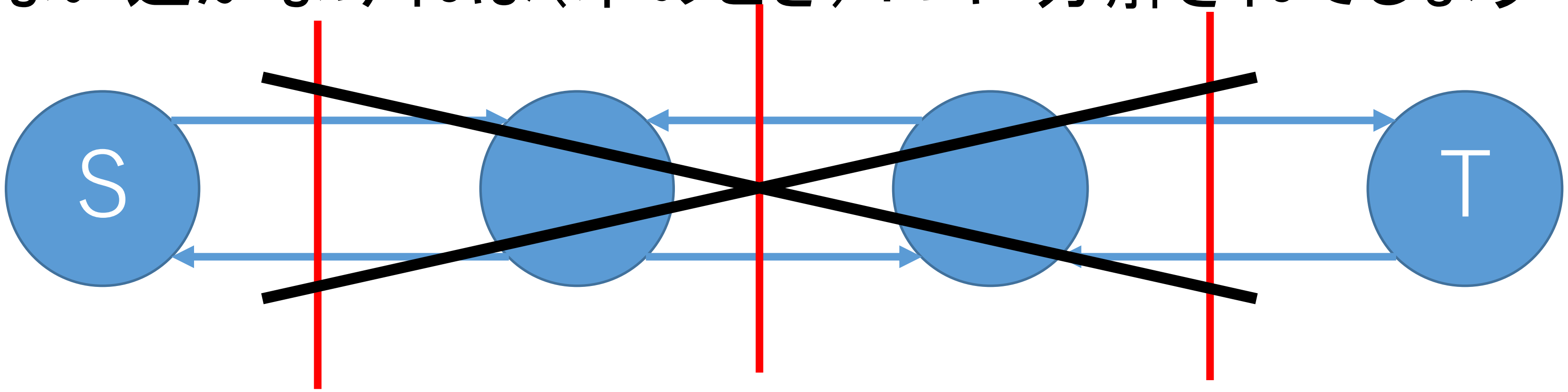
- 橋を列挙する
- 橋の数によってやることが変わってくる

解法(橋が2本以上)

- 橋が2本以上なら、そのうち2つの橋を切れば3つに分解できる

- しかしもう一辺切らなければならない

- 橋でない辺があればそれを切れば3つに分解できる
- 橋でない辺がなければ(木のとき)4つに分解されてしまう
- O(E)



解法（橋が2本未満）

- 適当に2点をを決める
- その間とどこかを切って3つに分解することを考える

- 2点の決め方
- 1点は適当に固定する
- もう一点はそれ以外の点を全部ためす
- 以下の処理をO(V)回行うことになる

解法（橋が2本未満）

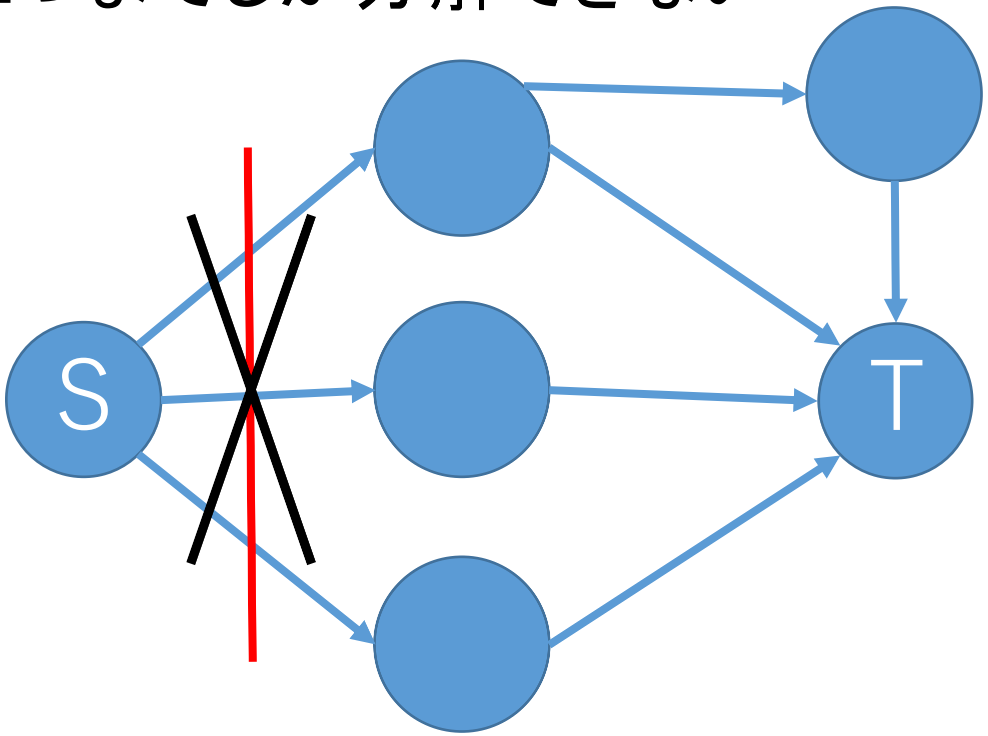
- 選んだ2点のうちどちらかをソース
- もう片方をシンクとする

- ソースからシンクへ最大流を流す
- その最大流の値と橋の数によってやることが変わってくる

解法（最大流が3以上）

- 最大流が3以上だとすると、最小カットは3以上になる
- つまりその2点間をカットするには3辺以上切る必要がある
- カットした後は何もできないので最大2つまでしか分解できない
- 次のペアを調べる

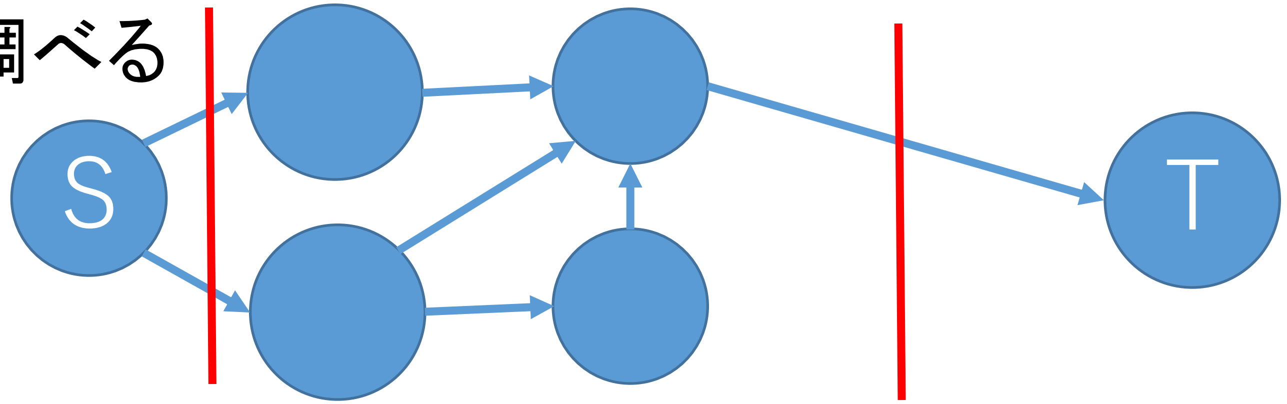
- 3以上になった瞬間に打ち切る
- フォードファルカーソンならO(FE)
- Fが3のため実質O(E)



解法(最大流が1、橋1のとき)

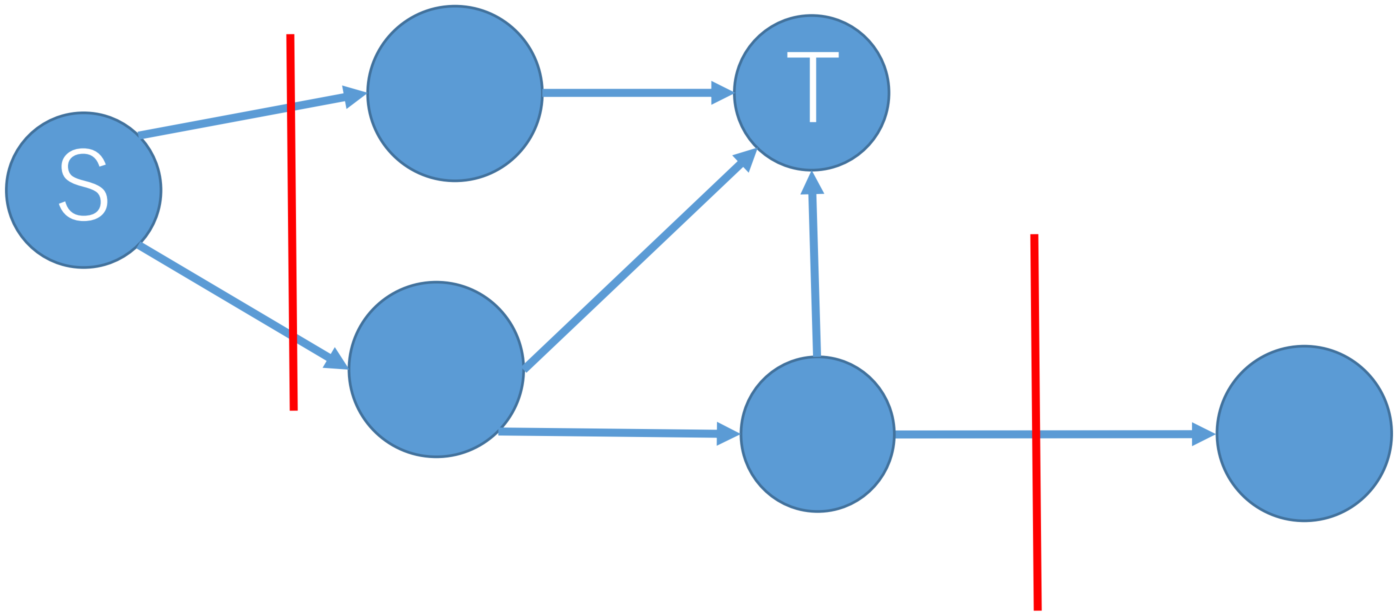
- 橋にフローが流れているときである
- 最小カットを行い2つに分解する

- 橋のソースに近い方のノードをシンクに
- 橋のシンクに近い方のノードをソースにしてそれぞれで最大流
- どちらか片方でも2ならば再び最小カットを行えば3つに分解できる
- 両方3以上なら次のペアを調べる



解法（最大流が2、橋1のとき）

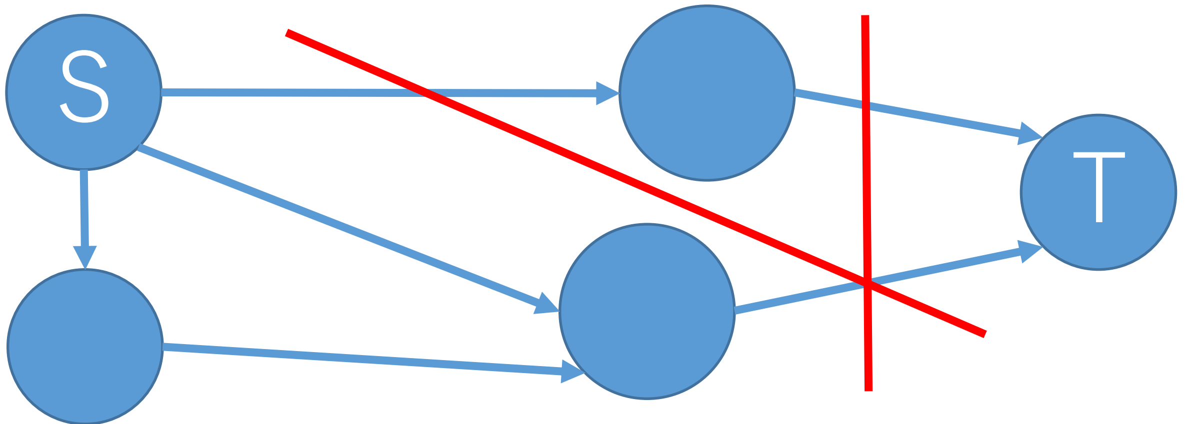
- 橋にフローが流れていない
- 最小カットとは無関係のところに橋がある
- 最小カットを行い、橋を切れば3つに分解できる



解法(最大流が2、橋0のとき)

- 辺を一本取って、橋を2つ以上作らなければならない
- 1本は作られる(最小カット)

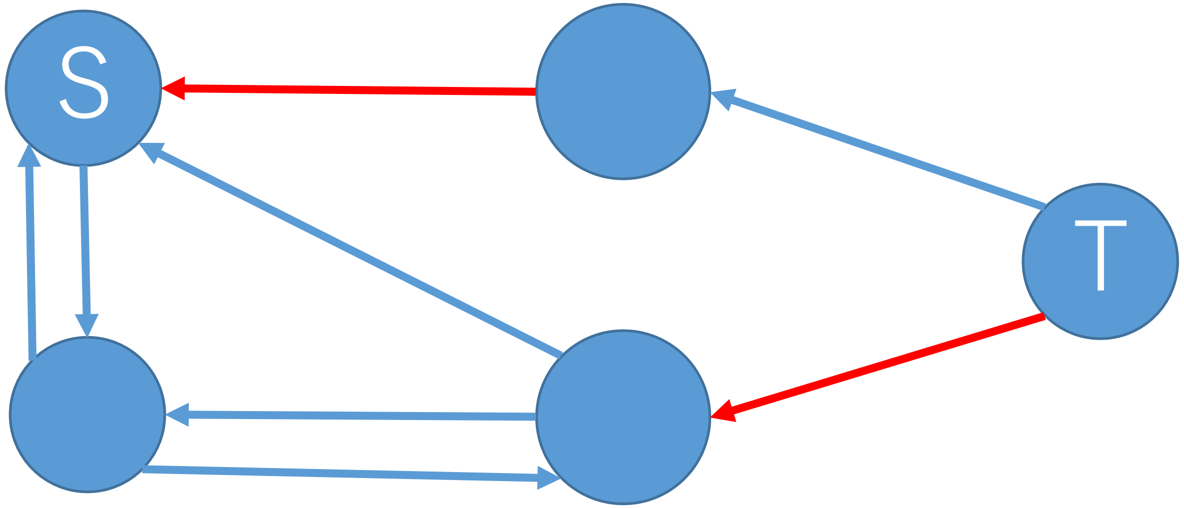
- 一本取って橋が2つ以上になったとする
- どちらの橋をとっても非連結になる(最小カット)
- つまり辺を共有する2つの最小カットのペアがあればよい



解法（最大流が2、橋0のとき）

- フローを流した後のグラフで考える
- シンクからフローが流れていない弧を使っていけるノード集合を探す
- 最小カットを調べるときと同じ

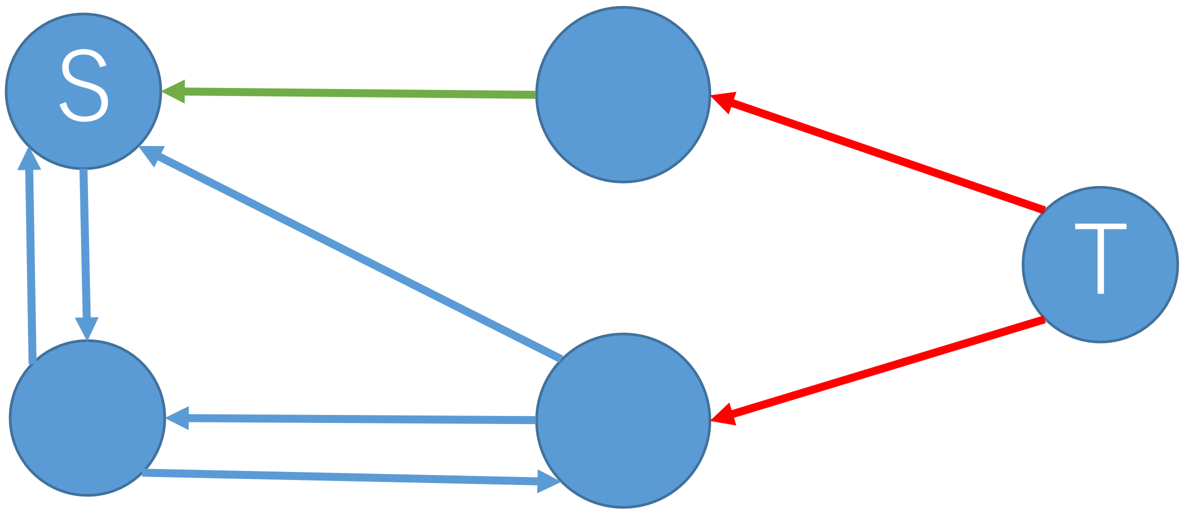
- 必ずTにはたどり着かず、ノード集合に入る弧（赤い弧）が2つある



解法（最大流が2、橋0のとき）

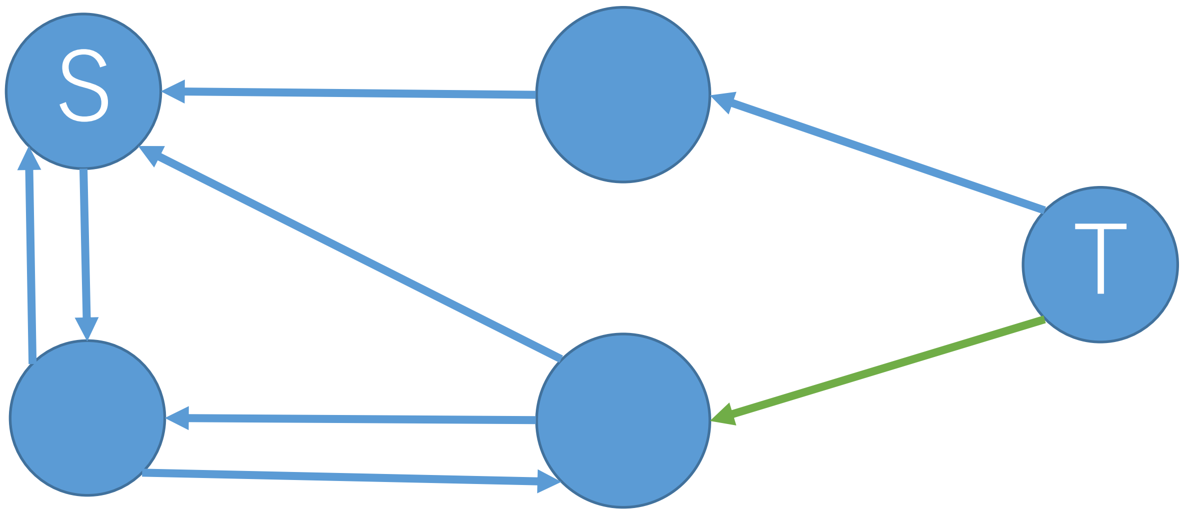
- 片方の弧を突破して、到達可能ノード集合を求め続ける
- このときはまた2つの弧が見つかるかもしれない
- 1つも弧が見つからないかもしれない（Tに到達）

- 緑の弧を突破するとまた赤い弧2つが見つかる



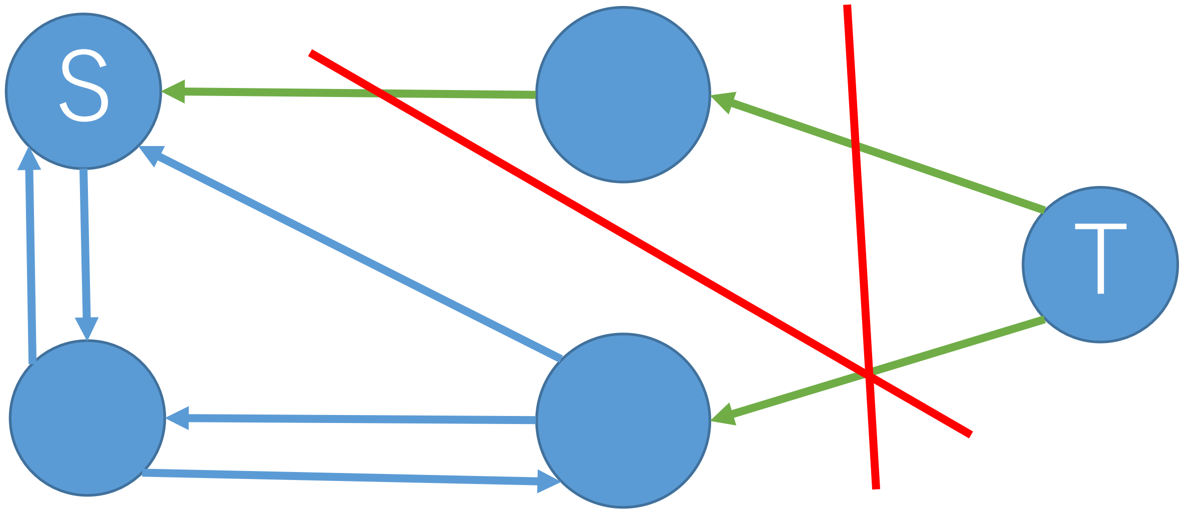
解法（最大流が2、橋0のとき）

- もう片方の弧も突破して同じようにやる
- このときはTに到達したため弧は見つからない



解法（最大流が2、橋0のとき）

- 新しく見つかった2つの弧のペアが一致する場合
  その2つの弧で同じように続ける
- 両方弧がなかった場合は次のシンクとソースのペアへ
- 弧が異なった場合
  辺を共有する2つの最小カットであるため3つに分解できる



解法（最大流が2、橋0のとき）

- 新しく探し出したノードしか見なければ速い
- 別の弧から到達した場合は2つ持っておく
- それでもO(E)に収まる
- 1つしか新しいノードがない場合は
  もう一つは突破しなかった方の弧の根元のノード（前と同じ）
  であるため必ず2つのペアは異なる
- よってO(E)
- 全体でO(V*E)

ちょっとした証明

- 突破後、突破しなかった方の弧が出ているノードに到達不可とする
- すると新しい弧の1つは突破しなかった方の弧になる
- 到達できないのだから別のところで探索が止まっている
- (Tに到達したら、突破しなかった方の弧が出ているノードに到達できる)
- 新しい2つの弧の内1つはもう見つけているので、見つける弧は1つ
- 突破しなかった弧と新しく見つけた弧も最小カットのペア
- よって突破する前と突破した後は、
  2つの辺を共有する2つの最小カットのペアである

ちょっとした証明

- 前述のとおり、共通する辺は突破しなかった方の弧である
- そしてもう片方の探索では突破する方の弧である
- 突破した方の弧は見つけることができない
- よって見つけた弧のペアは必ず異なる

ちょっとした証明

- 突破後、互いに突破しなかった方の弧が出ているノードに到達可とする
- つまり新しい2つの弧を見つけることになる
- 突破する前と、突破した後に共通の辺がない
- よって辺を共有する2つの最小カットのペアではない
- また合流地点があるはずである
- その合流地点からの探索は全く同じものになる
- よって同じ弧のペアになる

ちなみに

- snukeさん曰く、DAGのときはO(V+E)で解ける
- さらに聞いた話によると頑張るとO(ElogE)でも解けるらしい
- T.Mにはさっぱりわかりません

ジャッジ解

- ixmel　311行　(C++)
- T.M　203行　(C)

- 正答率　FA
- Online　2/6　Triangoop　(60min)
- Onsite　-/-　-　(-min)