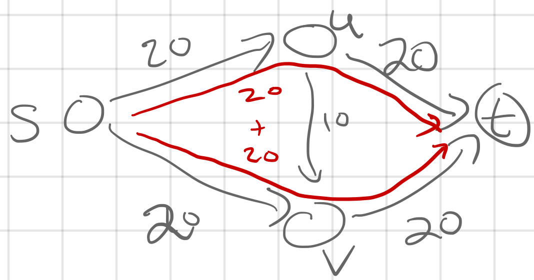
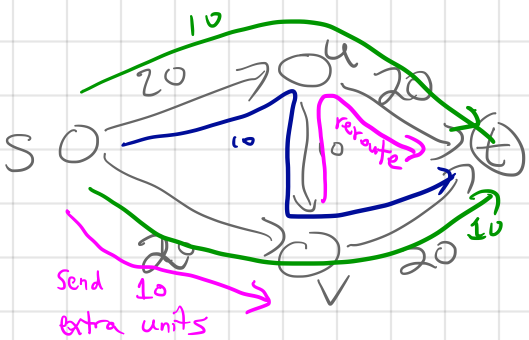
7 March 2018 Ford-Fulkerson Algorithm





If you had started solving the problem "greedily" and got into this state... how would you continue making progress toward the optimal solution?

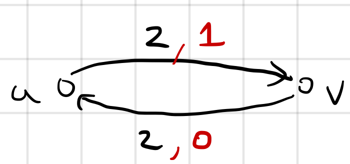
"Augmenting paths" will be the answer.

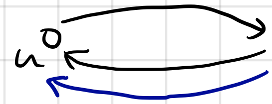
The net combination of these 2 "pink" operations continues to satisfy flow conservation, sends 10 extra units from s to t.

Residual graph: A diagram that encodes all the possibilities for modifying a flow f in a network G.

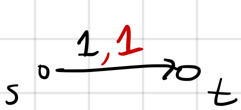
G_f = "residual graph of f"
= directed graph with vertex set $V(G)$
edge set {Forward edges} $\cup$ {Backward edges}
where $e = (u,v)$ belongs to {Forward edges} if $f(e) < c(e)$
and $\overleftarrow{e} = (v,u)$ belongs to {Backward edges} if $f(e) > 0$, where $e = (u,v)$

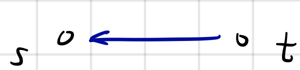
G_f could actually be a multigraph (2 vertices can have more than one edge between them) e.g.

G, f 

G_f 

Forward
Backward

G 

G_f 

Capacities in the residual graph:

$$e \in \{\text{Forward}\} \implies c_f(e) = c(e) - f(e)$$

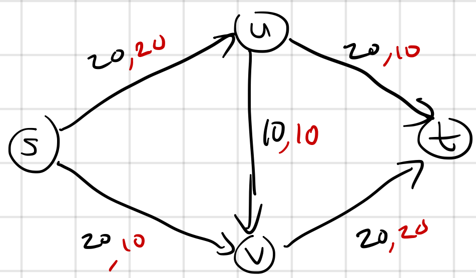
"the max amount of additional flow I could send on e"

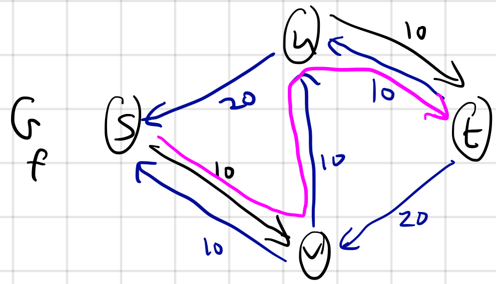
$$\overleftarrow{e} \in \{\text{Backward}\} \implies c_f(\overleftarrow{e}) = f(e)$$

$\overleftarrow{e} = (v,u)$ $\quad e = (u,v)$

"the max amount of flow I could shift away from e to some other edge"

E.g.





Augmenting path: A path from s to t in G_f.

Let P be an augmenting path.

$$\text{bottleneck}(f, P) \triangleq \min\{\, c_f(e) \mid \underbrace{e}_{\text{forward or backward}} \in P \,\}$$

"Augment f using P" means:

$$\forall\, e \in P \cap \{\text{Forward}\} \qquad f(e) := f(e) + \text{bottleneck}(f, P)$$

$$\forall\, e \in P \cap \{\text{Backward}\} \qquad f(e) := f(e) - \text{bottleneck}(f, P).$$

This operation never violates a capacity constraint.

On forward edge e, $f(e)$ increases by at most $c(e) - f(e)$, so new $f(e)$ doesn't exceed $c(e)$.

On backward edge $\overleftarrow{e}$, which is the reverse of $e \in E(G)$, $f(e)$ decreases by at most $f(e)$, so remains ≥ 0.

$n =$ # vertices
$m =$ # edges

FORD-FULKERSON ALGORITHM

Preprocess G to remove isolated vertices.
Initialize $f(e) = 0 \quad \forall e$

do

compute residual graph G_f — $O(m)$

if G_f contains an augmenting path P — e.g. BFS or DFS, $O(m)$

any choice of aug't path is considered a valid implementation of Ford-Fulkerson.

augment f using P

end if

until G_f has no augmenting path

output f

Does this algorithm terminate?

Not necessarily, if edges are allowed to have irrational capacities.

But yes, it always terminates, if capacities are rational numbers.

The case with rational capacities reduces to the case with integer capacities by scaling. (Multiply all rational capacities by a common denominator to make them integers.)

When capacities are integers, the residual capacities $c_f(e)$ are always integers. (Proof: induction on number of loop iterations. Augmenting f using P preserves integrality when bottleneck(f, P) is an integer.)

Augmenting f using P increases $v(f)$ by bottleneck(f, P), which is at least 1 assuming integer residual capacities.

Since $v(f)$ grows by at least 1 in each iteration and never exceeds $C \triangleq c(\{s\}, V \setminus \{s\})$ = combined capacity of edges leaving s,

the number of loop iterations is $\leq C$.

Running time $O(m)$ per iteration $\Longrightarrow$ overall running time $O(mC)$.