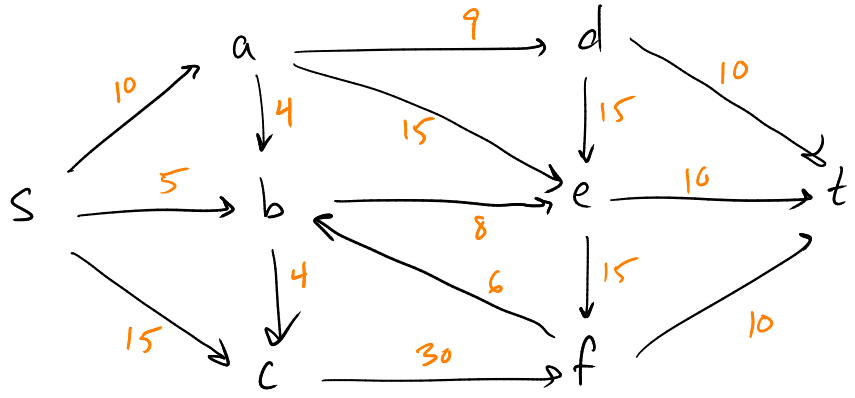


GreedyFlow(G, s, t):

repeat:

"augmenting path"

P = find a path from s → t with positive capacity remaining.

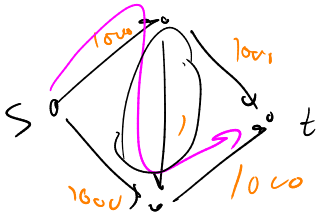
is shortest? max capacity? min capacity?

$\alpha \leftarrow$ bottleneck of P, ie. min remaining capacity of any edge in P.

for each edge $e \in P$:

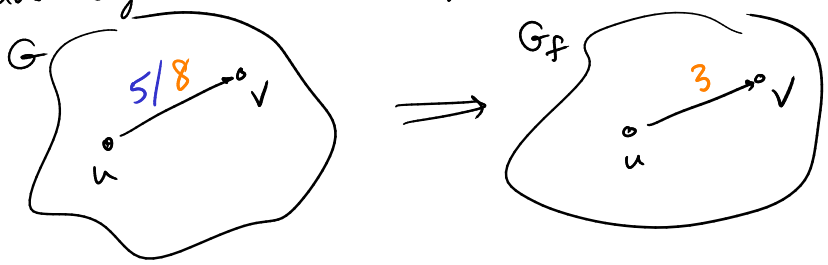
$f(e) \leftarrow f(e) + \alpha$.

while any augmenting path remains.

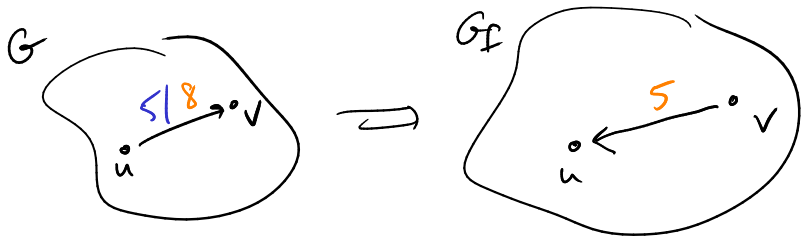


Defn Given a flow network G with flow f, define the residual network G_f as follows:

- G_f has the same vertices as G.
- For each edge $e = (u, v)$ in G where $f(e) < c(e)$, add an edge $u \to v$ in G_f with capacity $c(e) - f(e)$.



- For each edge $e = (u, v)$ in G where $f(e) > 0$, add an edge $v \to u$ in G_f with capacity $f(e)$.



Ford-Fulkerson(G, s, t):

f ← zero flow

while there are s→t paths in G_f:

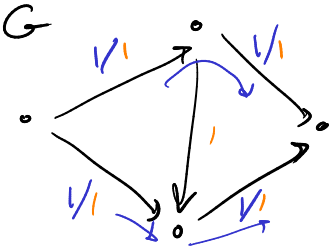
P ← pick some s→t path in G_f

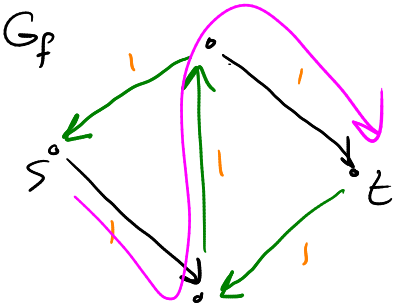
α ← min capacity of any edge in P

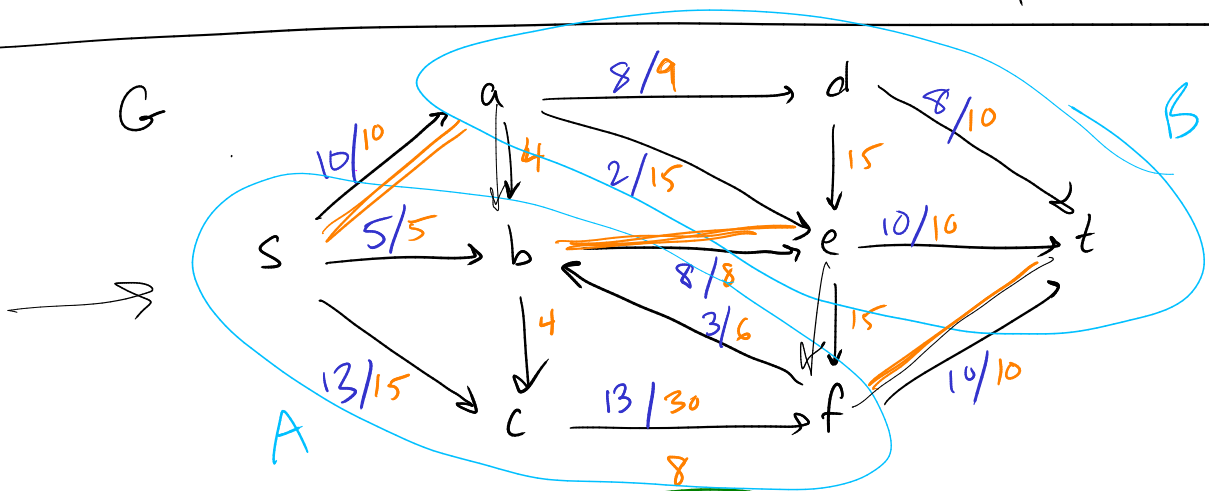
for each edge e ∈ P:

if e is a "forwards edge", add α to f(e).

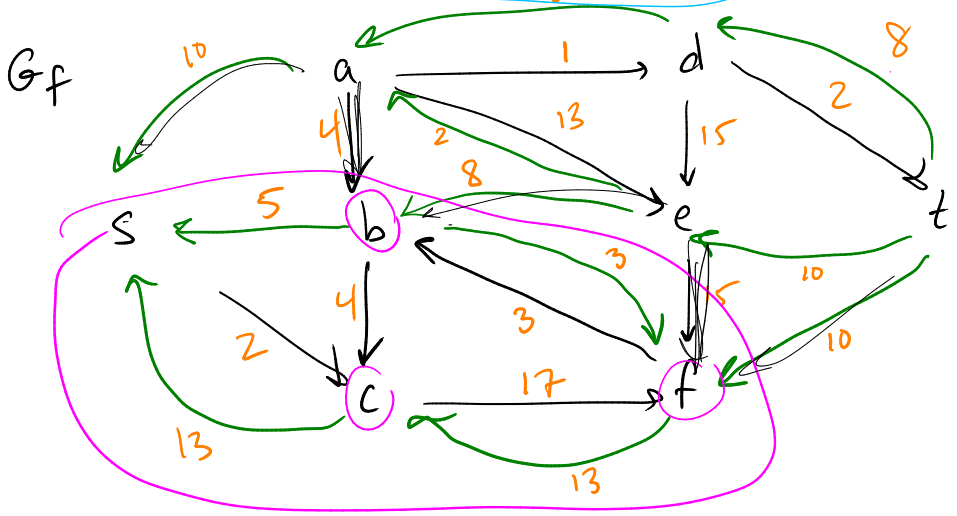
if e is a "backwards edge", subtract α from f(e).







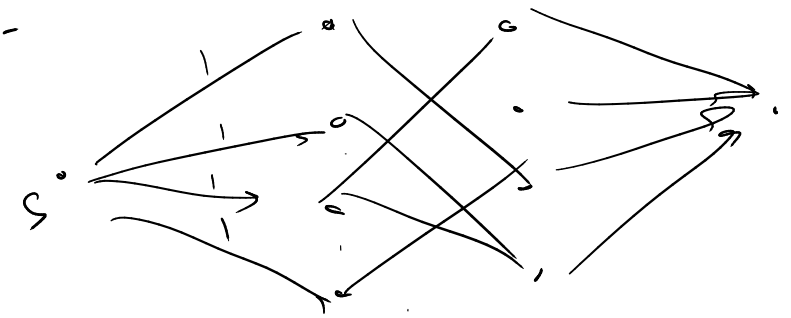
C(A,B)
= 10+8+10
= 28.



Observation: Flow always increases by at least 1 every iteration, (hence the algorithm will terminate).

→ Can run in $O(C \cdot E)$ time where C is the total capacity of edges out of s, E is # of edges, assuming the stuff in the loop can take $O(E)$ time.

eg.



Bipartite graph w/ n vertices on each side

$$C = n.$$

$$E = n + n^2 + n = O(n^2).$$

F-F would take $O(n^3)$.

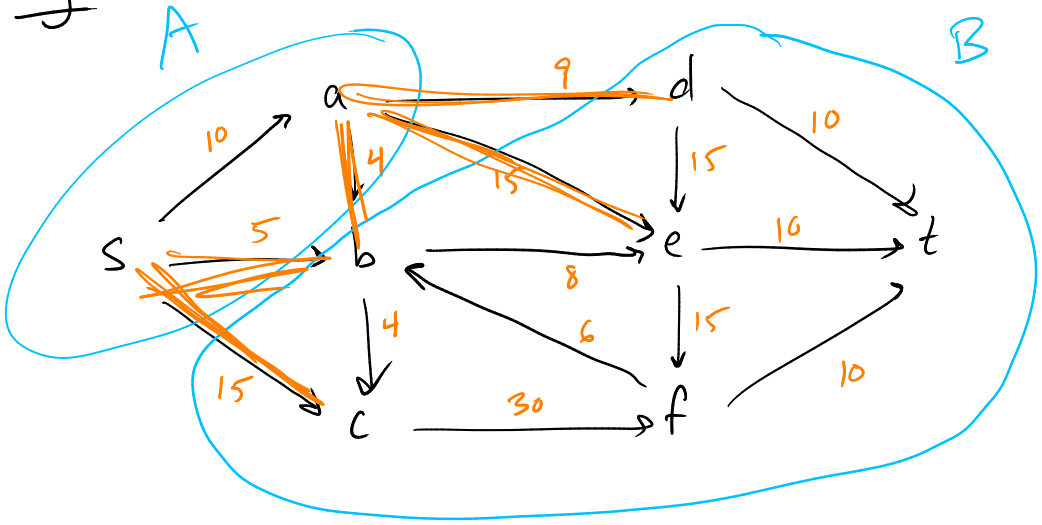
Defn An **s-t cut** in a flow network G is a partition of the vertices into 2 sets A, B such that $s \in A$, $t \in B$.

Defn The **capacity** of an s-t cut (A, B) is

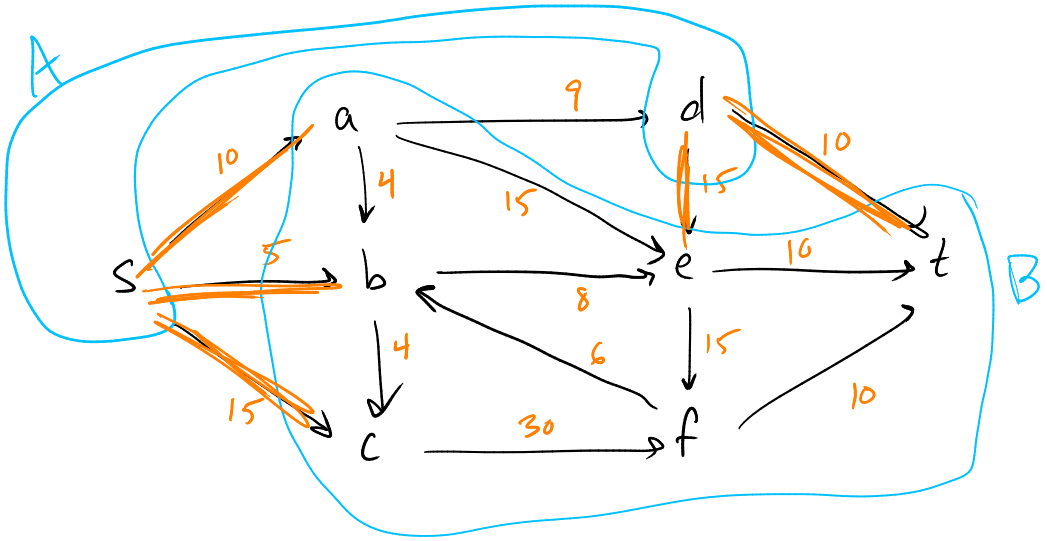
$$c(A, B) = \sum_{e : A \to B} c(e).$$

edges from A to B

eg.

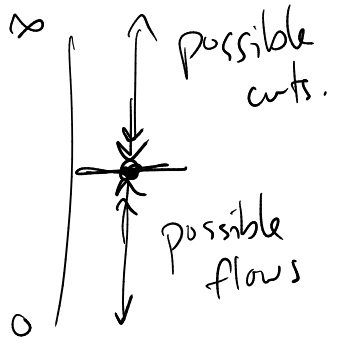


$c(A,B) =$
$9 + 15 + 4$
$+ 5 + 15$
$= 48.$



$c(A,B) =$
$10 + 5 + 15 + 15$
$+ 10$
$= 55.$

Lemma (Bottleneck Lemma). For any flow f and any s-t cut (A,B), $v(f) \leq c(A,B)$.



Corollary.

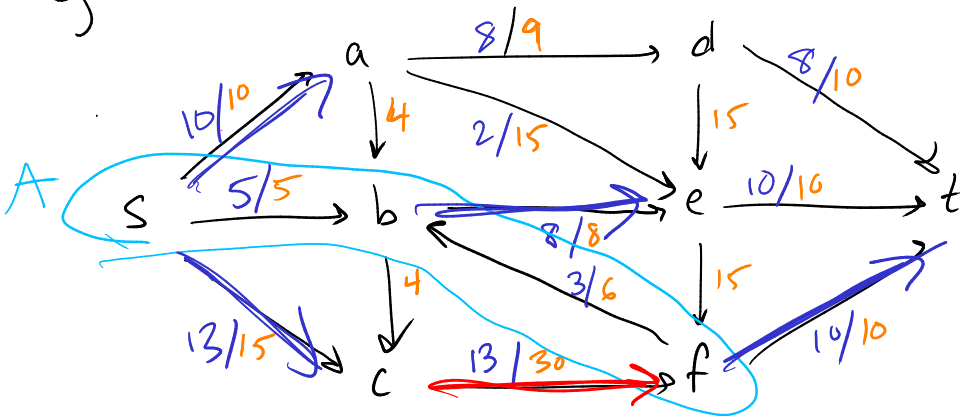
If $v(f) = c(A,B)$ for some flow f and s-t cut (A,B), then f is a max flow and (A,B) is a min cut.

{ ie. cut w/ smallest possible capacity.

Defn The net flow of (A,B) with respect to a flow f is

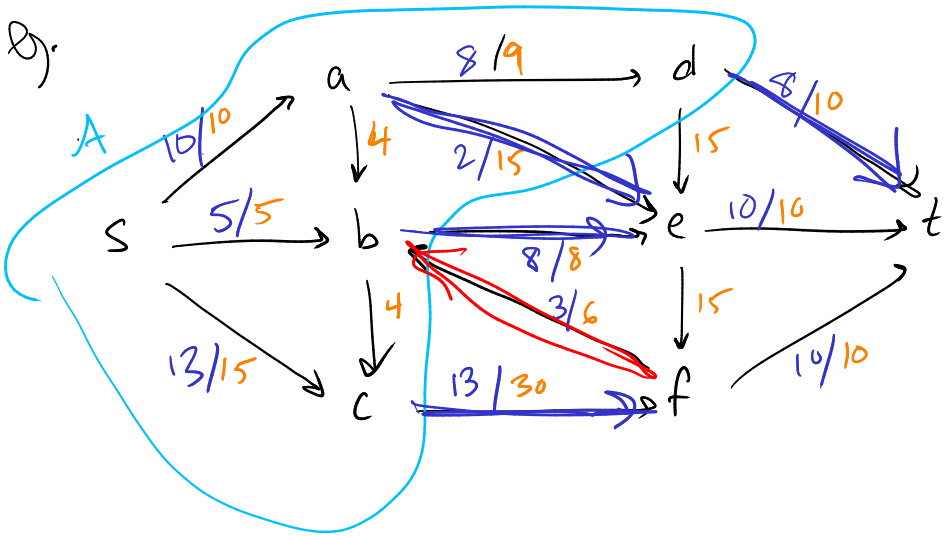
$$f(A,B) = \sum_{e:A\to B} f(e) - \sum_{e:B\to A} f(e).$$

eg.



$f(A,B) =$
$10 + 13 + 10 + 8$
$- 13 = 28.$

8).



$f(A,B) =$

$8 + 2 + 8 + 13$

$- 3 = 28.$

Lemma (Flow value Lemma): $f(A,B) = v(f)$ for any s-t cut (A,B) and any flow f.

Theorem. (Max flow / min cut). Given a flow network G and a flow f, the following are equivalent:

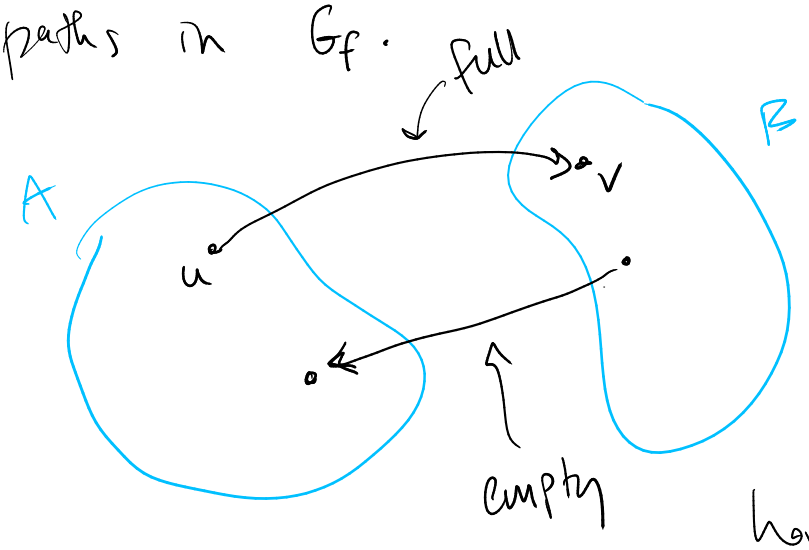
1. $v(f) = c(A,B)$ for some s-t cut (A,B).
2. f is a max flow.
3. There are no paths from $s \to t$ in the residual network G_f.

Proof. We will prove $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (1)$.

- $(1) \Rightarrow (2)$: this is corollary from above.
- $(2) \Rightarrow (3)$: Contrapositive is $\neg(3) \Rightarrow \neg(2)$. If there is an $s \to t$ path in G_f, we can increase f along this path, so f wasn't a max flow.

• (3) ⇒ (1): Let A be the set of vertices reachable from s in G_f, and let $B = V - A$.

$s \in A$ since s reachable from itself and $t \in B$ since we assumed (3), i.e. there are no $s \to t$ paths in G_f.



Edges $A \to B$ must be full, o'wise there would be an edge w/ excess capacity in G_f and we would have been able to reach B.

Edges $B \to A$ must be empty, o'wise there would be a backwards edge $A \to B$ in G_f.

Hence:

$$v(f) = f(A, B) \quad \text{(Flow Value Lemma)}$$

$$= \sum_{e: A \to B} f(e) - \sum_{e: B \to A} f(e) \quad \text{(defn of net flow)}$$

($\sum_{e: A \to B}$: full; $\sum_{e: B \to A}$: empty)

$$= \sum_{e: A \to B} c(e) - 0$$

$$= c(A, B) \quad \text{(defn of cut capacity)}$$

∎

Corollary. Ford-Fulkerson stops when (3) is true. Hence (2) is also true, i.e. F-F correctly finds max flow. (+ also min cuts.)