Maximum Flows and Minimum Cuts

In 1950's, US defence researchers wrote a classified report about the train network in then Soviet Union. The network was modelled as a graph.

The graph had 44 vertices and 105 edges. The report determined the maximum amount of material that could be moved from Russia to Europe and the cheapest way to cause a disruption.

This is one of the first recorded instances of the max flow - min cut problem.

Max Flow Problem.

We are given a directed graph $G = (V, E)$ with two designated vertices s, t.

We are given capacity function $c : E \to \mathbb{R}^+$

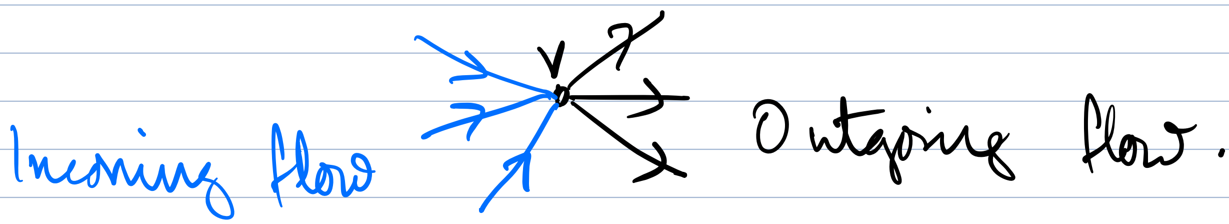
Goal: Want to assign flows to each edge, so that the following are satisfied.

Flow is $f : E \to \mathbb{R}$.

① For each edge $e \in E$, $0 \leq f(e) \leq c(e)$

② For each vertex $v \in V \setminus \{s, t\}$, we have

$$\sum_u f(u,v) = \sum_w f(v,w) \to \text{Flow conservation}$$



Subject to above, we want to maximize

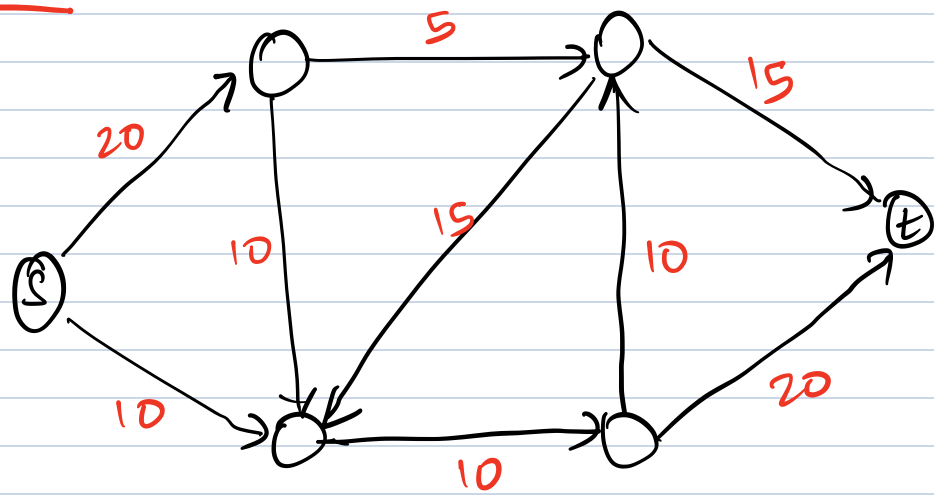
$$\text{Value of the flow } f \left\{ \; |f| = \underbrace{\sum_{w} f(s,w)}_{\text{Outgoing flow from } s} - \underbrace{\sum_{u} f(u,s)}_{\text{Flow incoming to } s.} \right.$$

It can be shown that $|f|$ is the net flow incoming to t as well. That is,

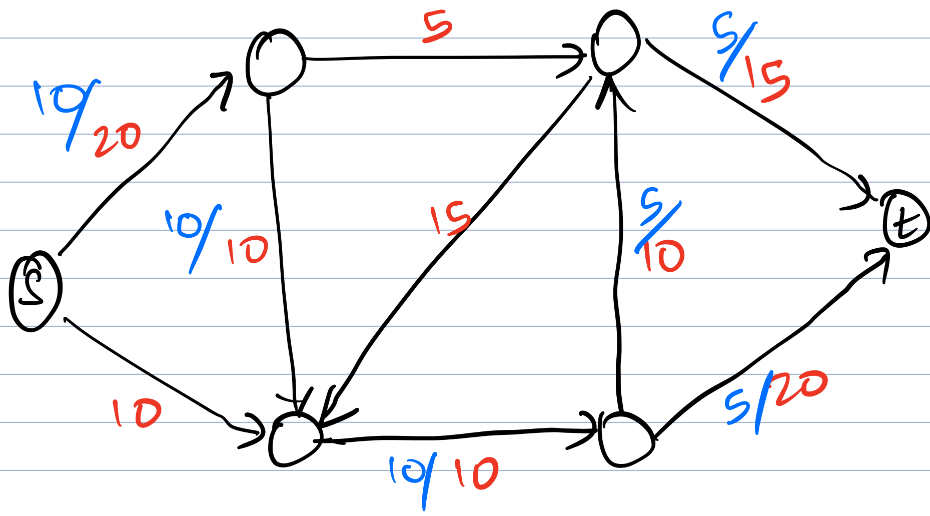
$$|f| = \sum_{u} f(u,t) - \sum_{w} f(t,w)$$

Why? For each $v \in V \setminus \{s,t\}$, we have flow conservation. So the net outgoing flow from s must equal the net incoming flow at t.

Example



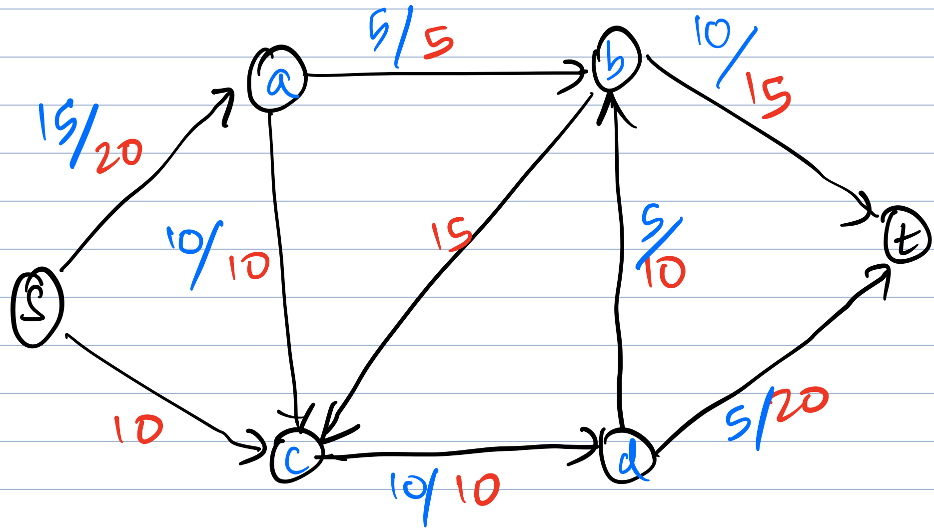
Capacities are indicated in red.



Flow in blue. (Flow = 0 in edges where no flow is indicated)

Total flow value here is 10.

The maximum flow problem is to compute the largest flow that can be pushed in this network, subject to constraints.



Here flow value = 15.

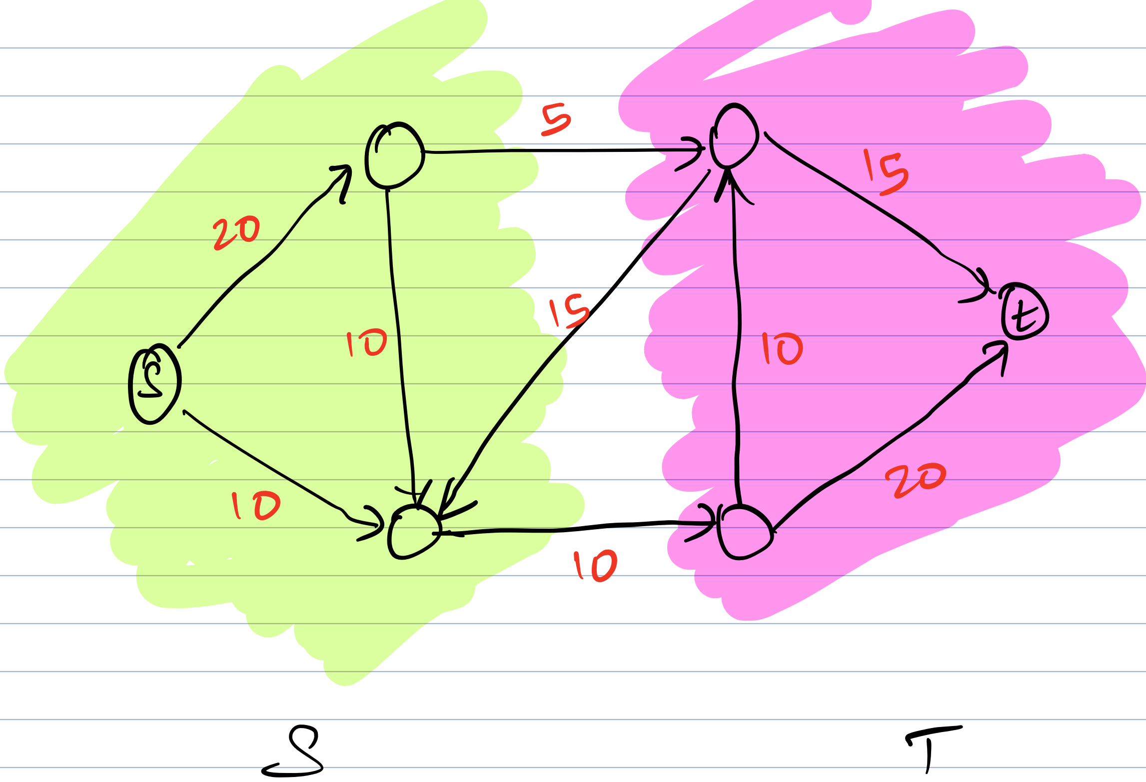
Minimum Cut Problem.

An (s, t) cut is a partition of the vertices into two disjoint subsets S and T such that $s \in S$, $t \in T$, $S \cap T = \phi$ and $S \cup T = V$.

The capacity of the cut (S, T) is given by the sum of all the edge capacities that are crossing from S to T.

$$\text{Capacity of } (S,T) = \|S, T\| = \sum_{v \in S} \sum_{w \in T} c(v, w)$$

Note: If $(v, w) \notin E$, then we take that $c(v, w) = 0$.



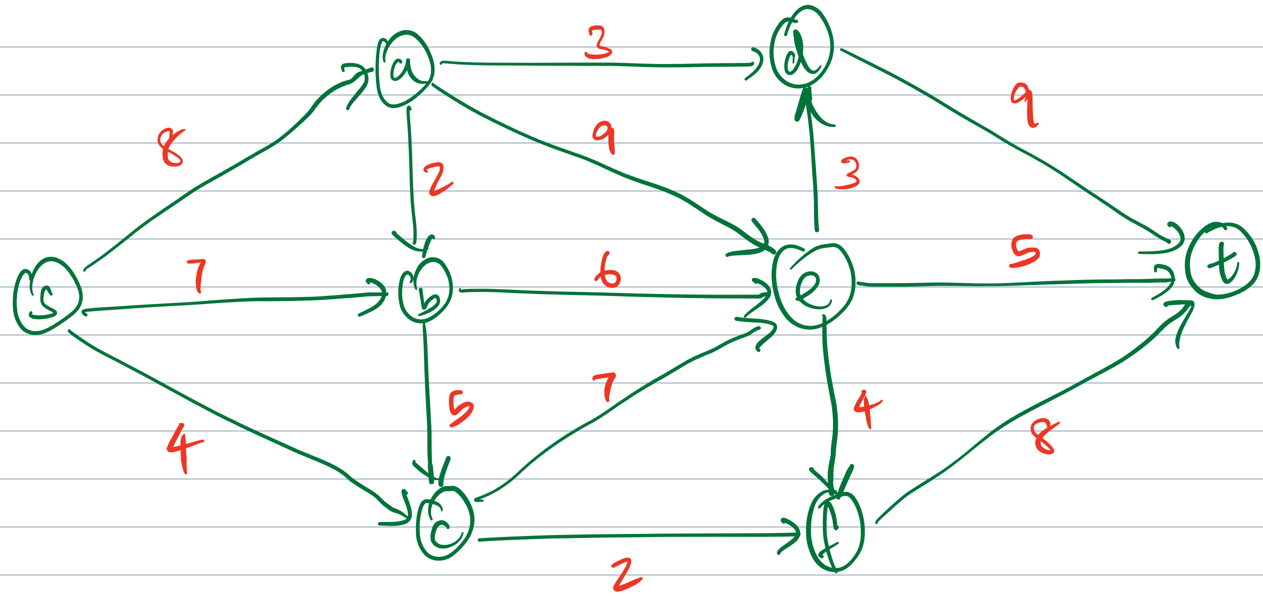
The capacity of the above cut is

$$\|S,T\| = 5 + 10 = 15.$$

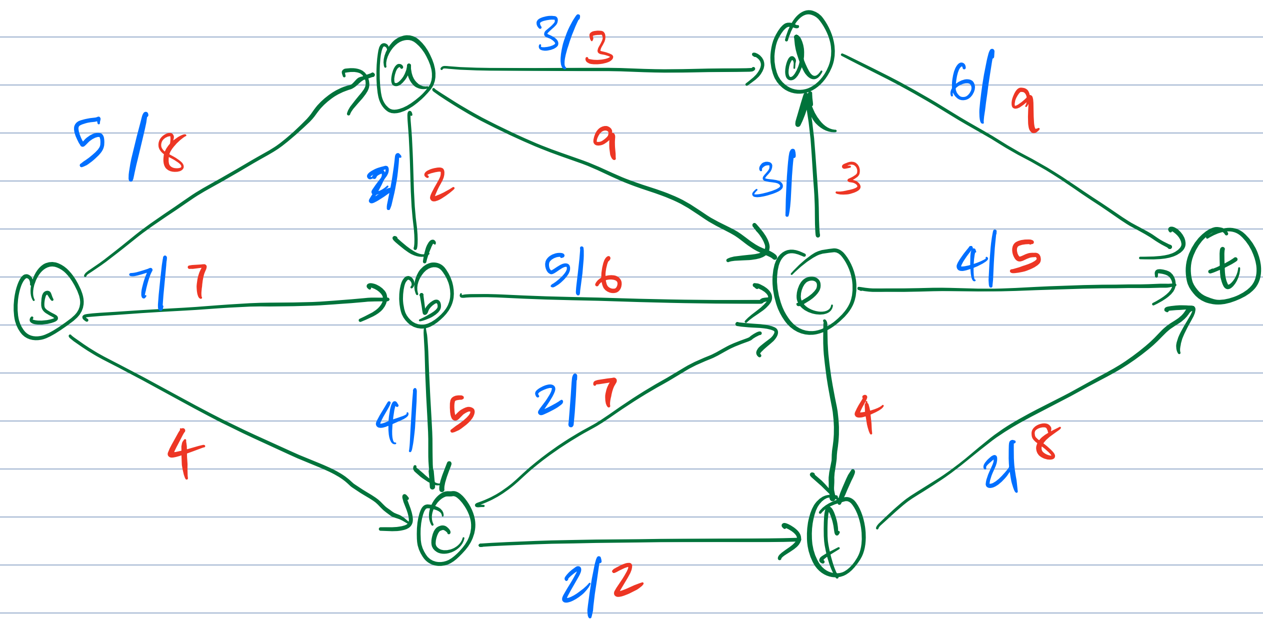
The minimum cut problem is to compute an (S,T)-cut of the least possible capacity.

Another example

Red: Capacities



Blue: Flow value at each edge.

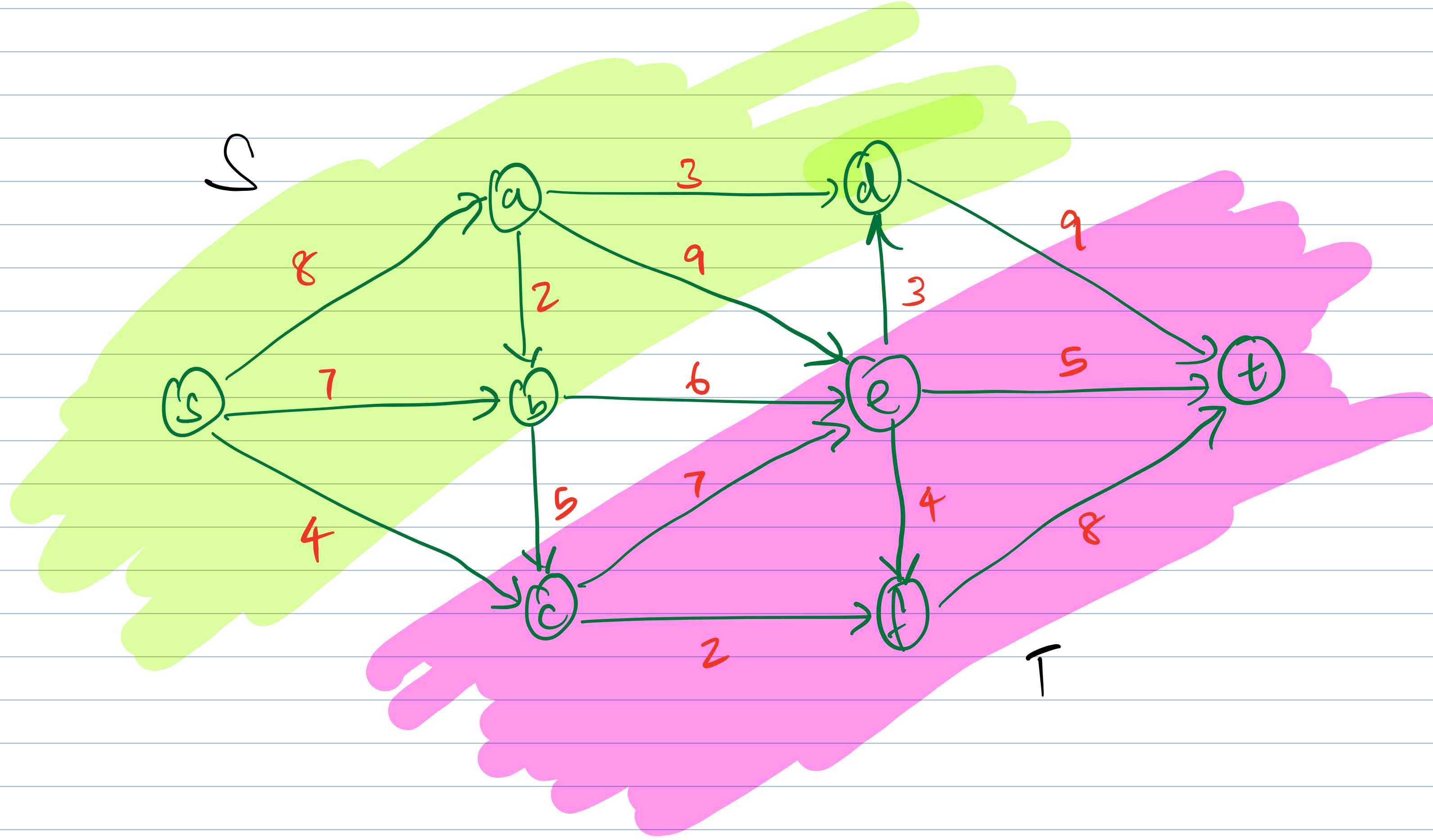


Red: Capacity

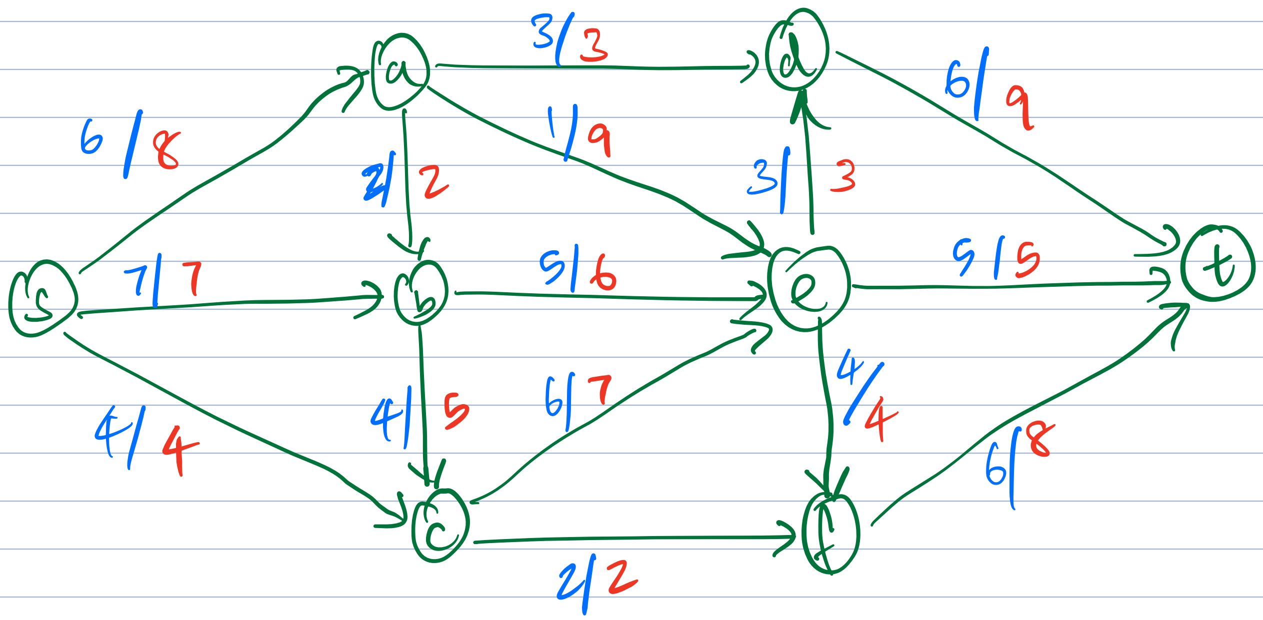
Blue: f.

Note: f is 0 when not indicated.

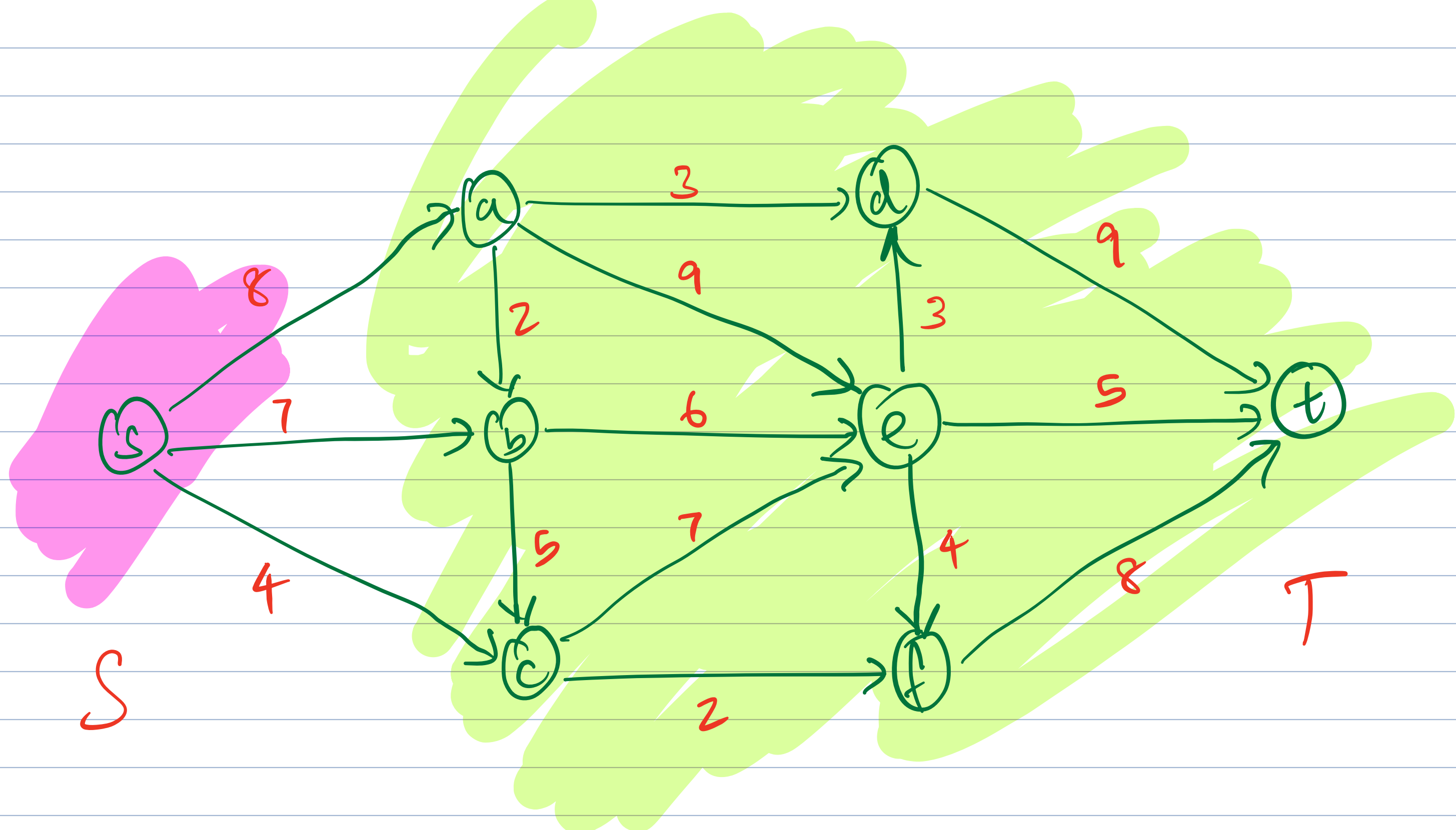
Value of flow $f = |f| = 12$



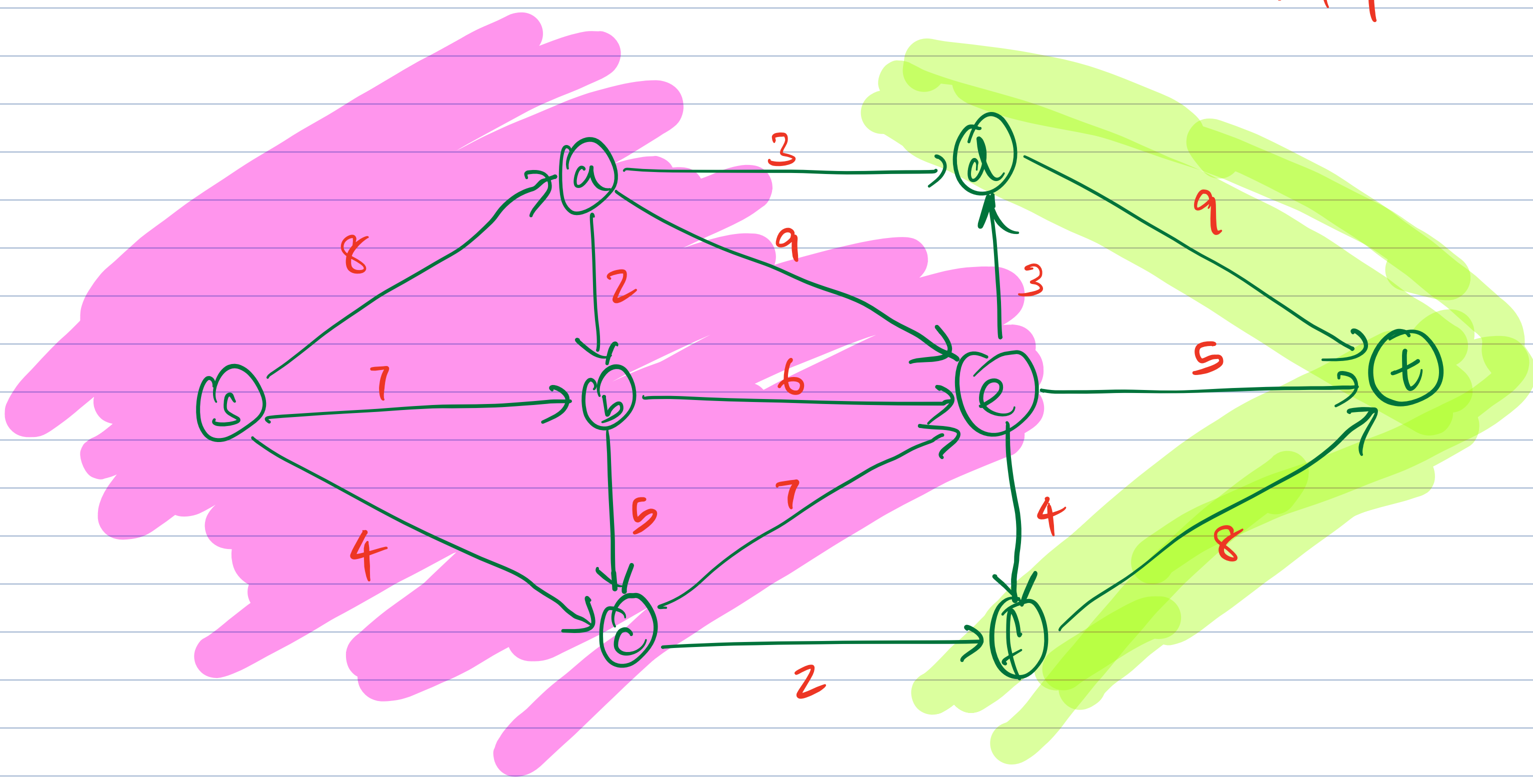
$\|S, T\| = 4 + 5 + 6 + 9 + 9 = 33.$



Total flow value = 17.



This cut has capacity $\|S,T\| = 8+7+4$
$= 19$



Capacity of this cut $= 3+3+5+4+2$
$= 17$

Theorem: Suppose G is a flow network. Let f be any feasible (s,t)-flow and (S,T) be any (s,t)-cut. Then value of the flow f, is at most the capacity of (S,T). That is $|f| \le \|S,T\|$

In particular **Max flow $\le$ Min cut.**

Proof: $$|f| = \sum_w f(s,w) - \sum_u f(u,s)$$

(Flow conservation at other v)
$$= \sum_{v \in S} \left[\sum_w f(v,w) - \sum_u f(u,v) \right]$$

$$= \sum_{v \in S} \sum_w f(v,w) - \sum_{v \in S} \sum_u f(u,v)$$

(Remaining flows from S to S)
$$= \sum_{v \in S} \sum_{w \notin S} f(v,w) - \sum_{v \in S} \sum_{u \notin S} f(u,v)$$

$$= \sum_{v \in S} \sum_{w \in T} f(v,w) - \sum_{v \in S} \sum_{u \in T} f(u,v)$$

Inequality 1 (ignoring negative term)

$$\leq \sum_{v \in S} \sum_{w \in T} f(v,w)$$

Inequality 2 (Flow (e) $\leq$ cap(e))

$$\leq \sum_{v \in S} \sum_{w \in T} c(v,w)$$

$$= \| S, T \|$$

Question: When do we get $|f| = \|S,T\|$?

This happens when both the inequalities are tight. Ineq 1 is tight when $\sum_{v \in S} \sum_{u \in T} f(u,v)$ is zero. That means there is no incoming flow from any edge that goes from T to S.

Inequality 2 is tight when

for each $v \in S$, $w \in T$, we have $f(v,w) = c(v,w)$

That is, each edge that goes from S to T is "saturated".

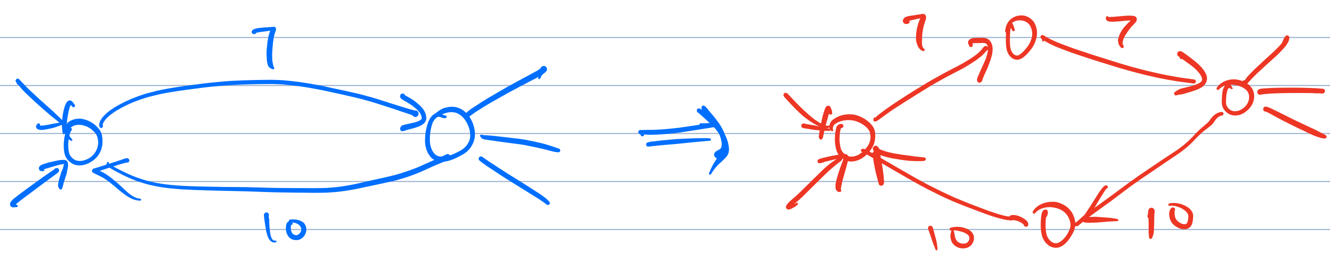
Theorem: Suppose G is a flow network. Let f be any feasible (s,t)-flow and (S,T) be any (s,t)-cut. Then $|f| = \|S,T\|$ if and only if f saturates all the edges from S to T and avoids all the edges from T to S.

Max flow - Mincut theorem: In every flow network with source s and target t, the value of the maximum flow is equal to the capacity of the minimum cut.

Proof: Let G be a graph, with s, t identified and $c : E \to \mathbb{R}_{\geq 0}$.

We further assume that G is reduced. That is, for any $u, v \in V$, at most one of the edges (u,v) or (v,u) exist.

We may assume the above by using the below transformation.



The resulting graph has the same max flow and min cut as the original graph.

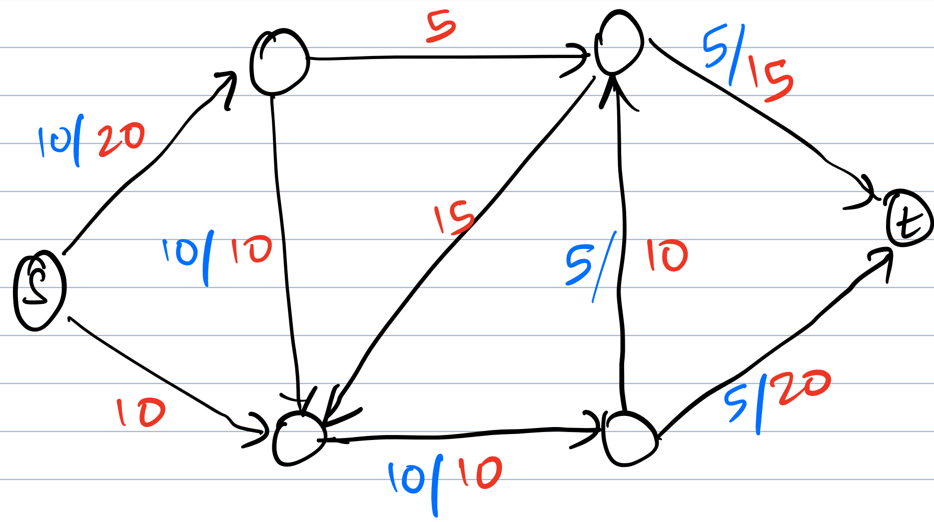
Suppose f is an (s,t)-flow in G. We define a new capacity function $c_f : V \times V \to \mathbb{R}$ called residual capacity.

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \\ f(v,u) & \text{if } (v,u) \in E \\ 0 & \text{otherwise} \end{cases}$$

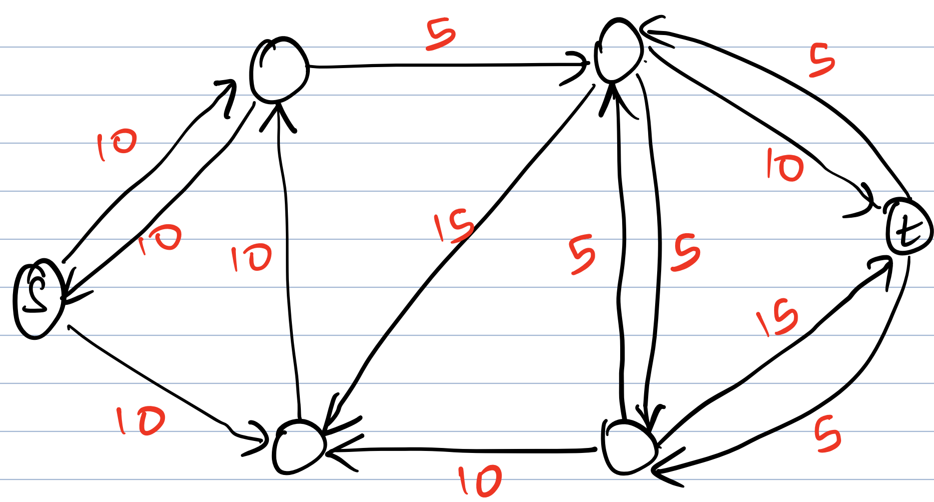
Intuitively, c_f denotes how much "more" flow each edge can carry. Note that

$c_f(u,v) \geq 0 \ \forall \ u,v$.

We define residual graph G_f containing all the edges for which $c_f(\cdot) > 0$.



G with flow f.



Residual graph G_f.

<u>Note</u>: If flow at edge u,v such that $0 < f(u,v) < c(u,v)$, then G_f contains both the edges (u,v) and (v,u).

Two cases: Either G_f contains an (s,t) path (with positive capacity) or it does not.

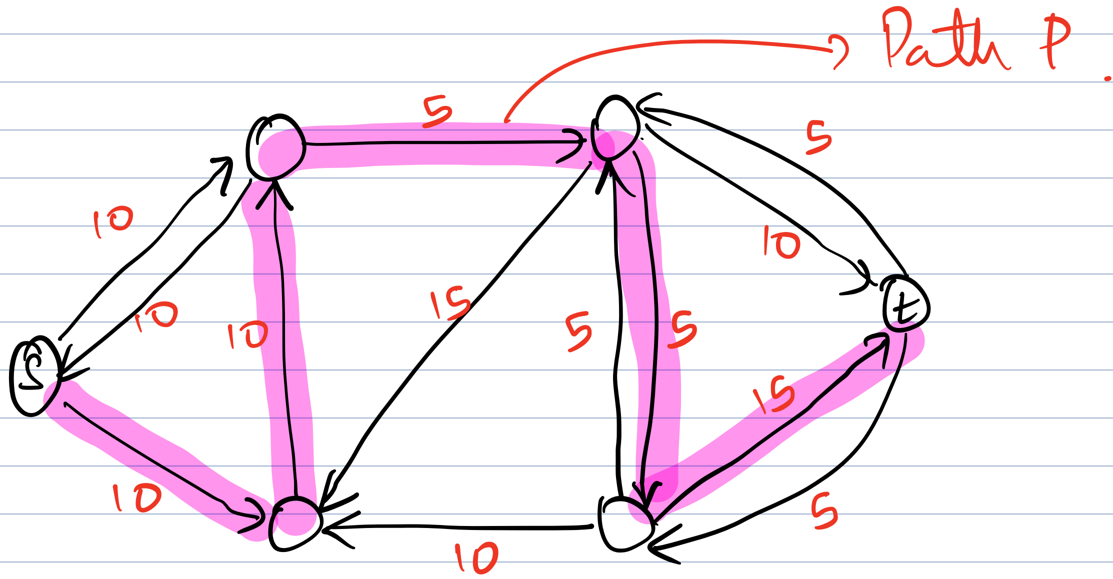
Case 1: Suppose G_f contains an (s,t) path P.

Let $F = \min_{(u,v)\in P} c_f(u,v)$ be the maximum flow we can send from s to t through P.

We define a new flow in G (original graph) as follows.

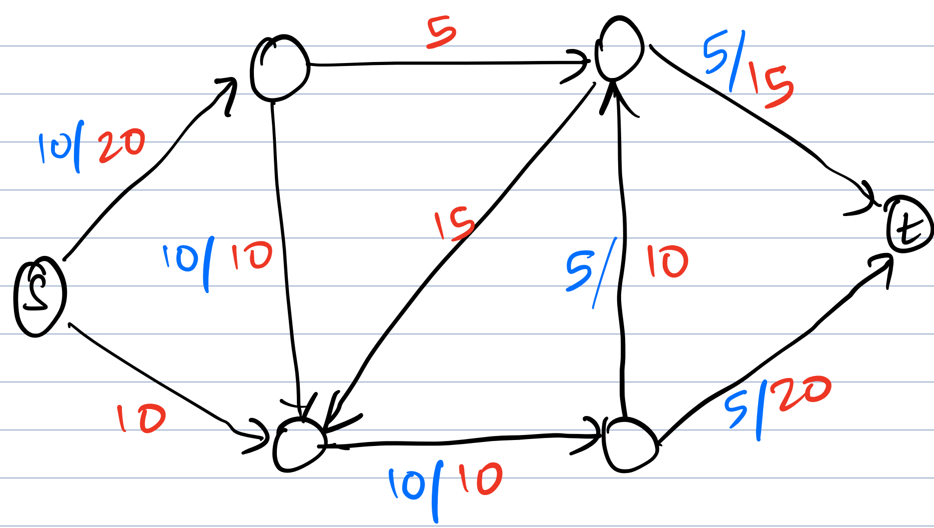
$$f'(u,v) = \begin{cases} f(u,v) + F & \text{if } (u,v) \in P \\ f(u,v) - F & \text{if } (v,u) \in P \\ f(u,v) & \text{otherwise} \end{cases}$$

Such a path P is called augmenting path.



Residual graph G_f.

Here $F = 5$.



G with flow f.

(need to update with f').

Claim: The new flow f' is feasible with respect to the original capacities c. That is, for all $e \in E$, $0 \le f'(e) \le c(e)$.

Proof: There are three cases.

① (u,v) is an original edge in G and P contains (u,v).

$$f'(u,v) = f(u,v) + F > f(u,v) \geq 0.$$

f is feasible

$$\begin{aligned} f'(u,v) &= f(u,v) + F \\ &\le f(u,v) + \underbrace{c_f(u,v)}_{} \\ &= f(u,v) + c(u,v) - f(u,v) \\ &= c(u,v). \end{aligned}$$

Hence f' is feasible for edge (u,v) in this case.

② (u,v) is an original edge G and P contains (v,u).

$$f'(u,v) = f(u,v) - F < f(u,v) \leq c(u,v).$$

since f is feasible

$$\begin{aligned} f'(u,v) &= f(u,v) - F \\ &\geq f(u,v) - c_f(v,u) \\ &\quad \left(\text{since } F = \min_{e \in P} c_f(e)\right) \\ &= f(u,v) - f(u,v) = 0 \end{aligned}$$

Hence f' is feasible for (u,v).

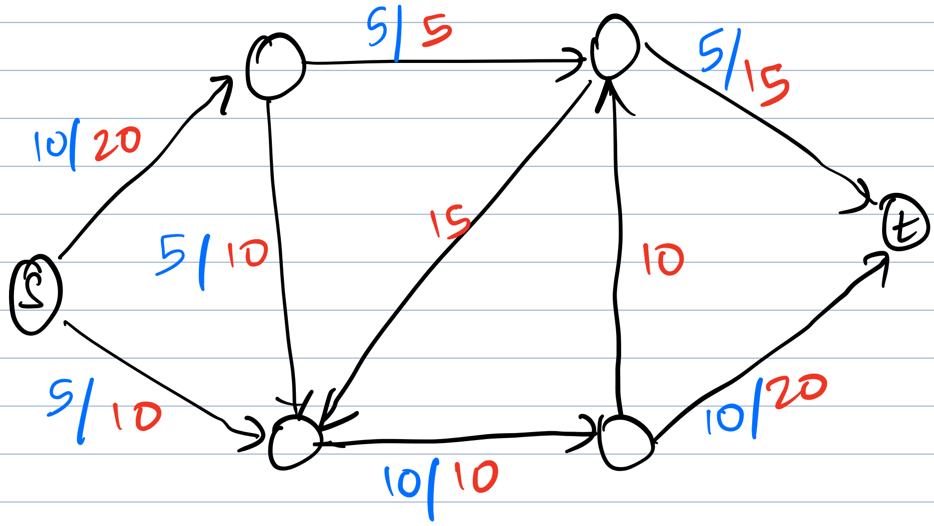
③ (u,v) is an original edge of G. Neither (u,v) nor (v,u) is in P. In this case $f'(u,v) = f(u,v)$ and f' remains feasible w.r.t. (u,v).

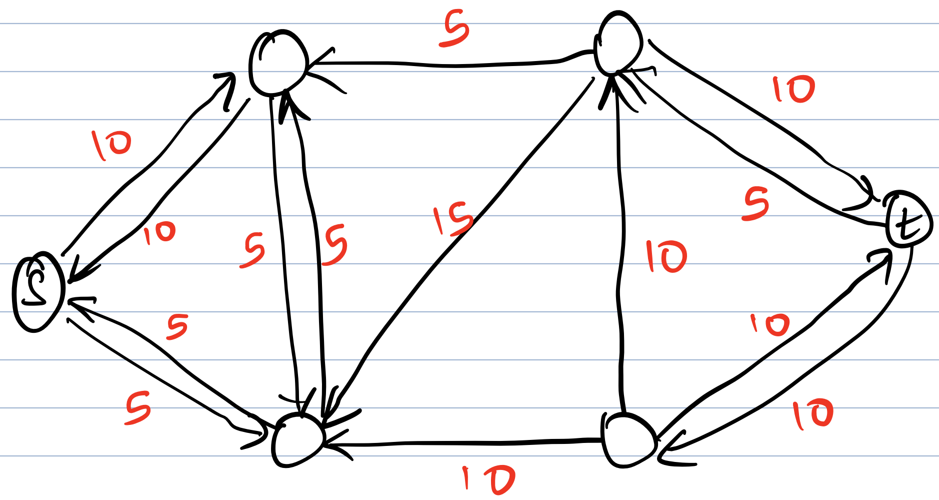
Value of f' : $|f'| = |f| + F > |f|$.

Hence f' is a feasible flow with value larger than f.

So we conclude that if there is an augmenting path P in G_f, then f is not a max flow.

Case 2: Suppose there is no augmenting path in G_f.





Consider the set of vertices reachable from s in G_f. Let us call this set S. Note that $t \notin S$ since the case assumes so. Let $T = V \setminus S$.

Hence (S,T) is an (s,t) cut.

Consider the pair (u,v) where $u \in S$, $v \in T$.

If (u,v) was in $E(G)$, then

$$c_f(u,v) = c(u,v) - f(u,v) = 0.$$

Hence $f(u,v) = c(u,v)$.

The edge (u,v) is saturated by f.

If (v,u) was in $E(G)$, then

$$c_f(u,v) = f(v,u) = 0.$$

The edge (v,u) is avoided by f.

So for all $u \in S$, $v \in T$, the edge (u,v) is saturated by f or edge (v,u) is avoided by f. Hence $|f| = \|S,T\|$. Which means f is a max flow and S,T is a min (s,t) cut.

We showed that f is maximized if and only if G_f has no augmenting path.

We also saw that G_f has no augmenting path $\Longrightarrow$ There is an S,T which is a min (s,t)-cut.

That is $\|S,T\| = |f|$.

If there is an (S,T) with $\|S,T\| = |f|$, we saw that f is maximum flow.

Hence maximum flow = minimum cut.

This yields an algorithm. (Ford Fulkerson algorithm)

→ Start with $G = G_f$

→ Repeat till no augmenting path in G_f

- → Find an augmenting path in G_f.
- → Send maximum possible flow through P.
- → Recalculate f and G_f.

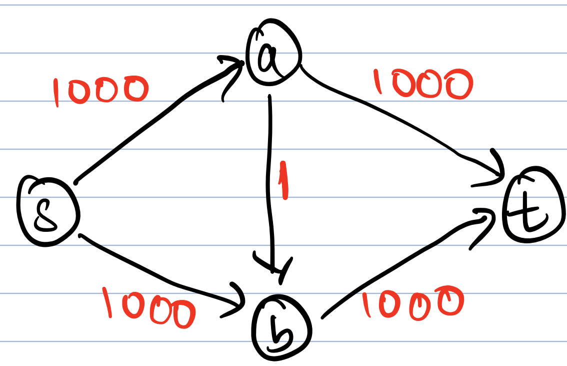
What is the time complexity?

If all capacities are integers, then each augmenting flow will be integral and hence we need $\leq |f^*|$ augmentations where $|f^*|$ is max flow value.

So running time $= O(|E|\,|f^*|)$

$O(|E|)$ time required per iteration.

Bad example :



Two rules by Edmonds and Karp.

① Choose the augmenting path with the largest bottleneck value.

Runs in $O(|E|^2 \log |E| \log |f^*|)$ time.

② Choose the augmenting path with the smallest no. of edges.

Runs in $O(|V||E|^2)$ time