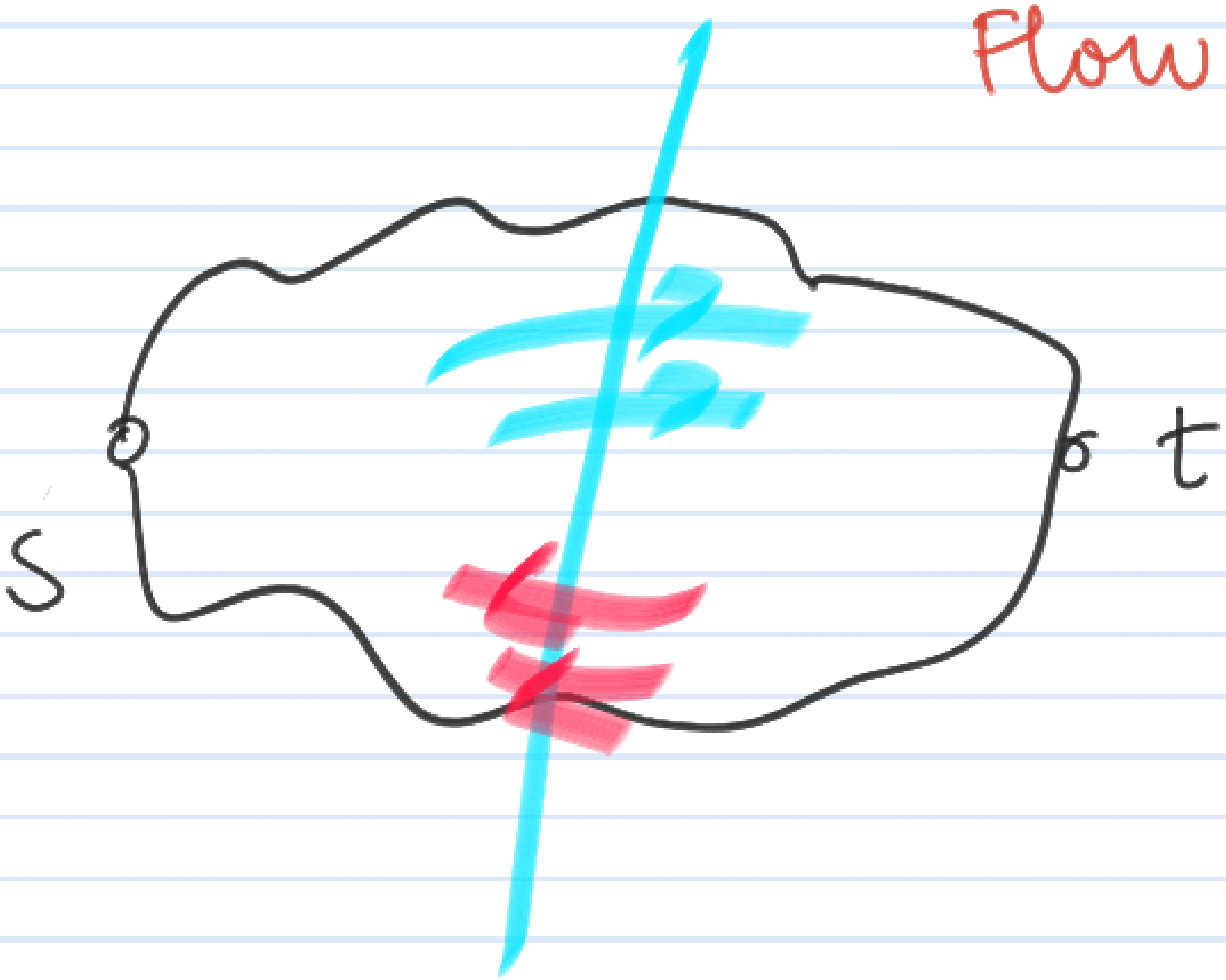
CS6130 : Advanced Graph Algorithms

Maximum Flow Problem

- Flows and cuts
- Max Flow, Min cut Theorem
- Optimality certificates
- Edmonds karp Algo.

Flows and cuts

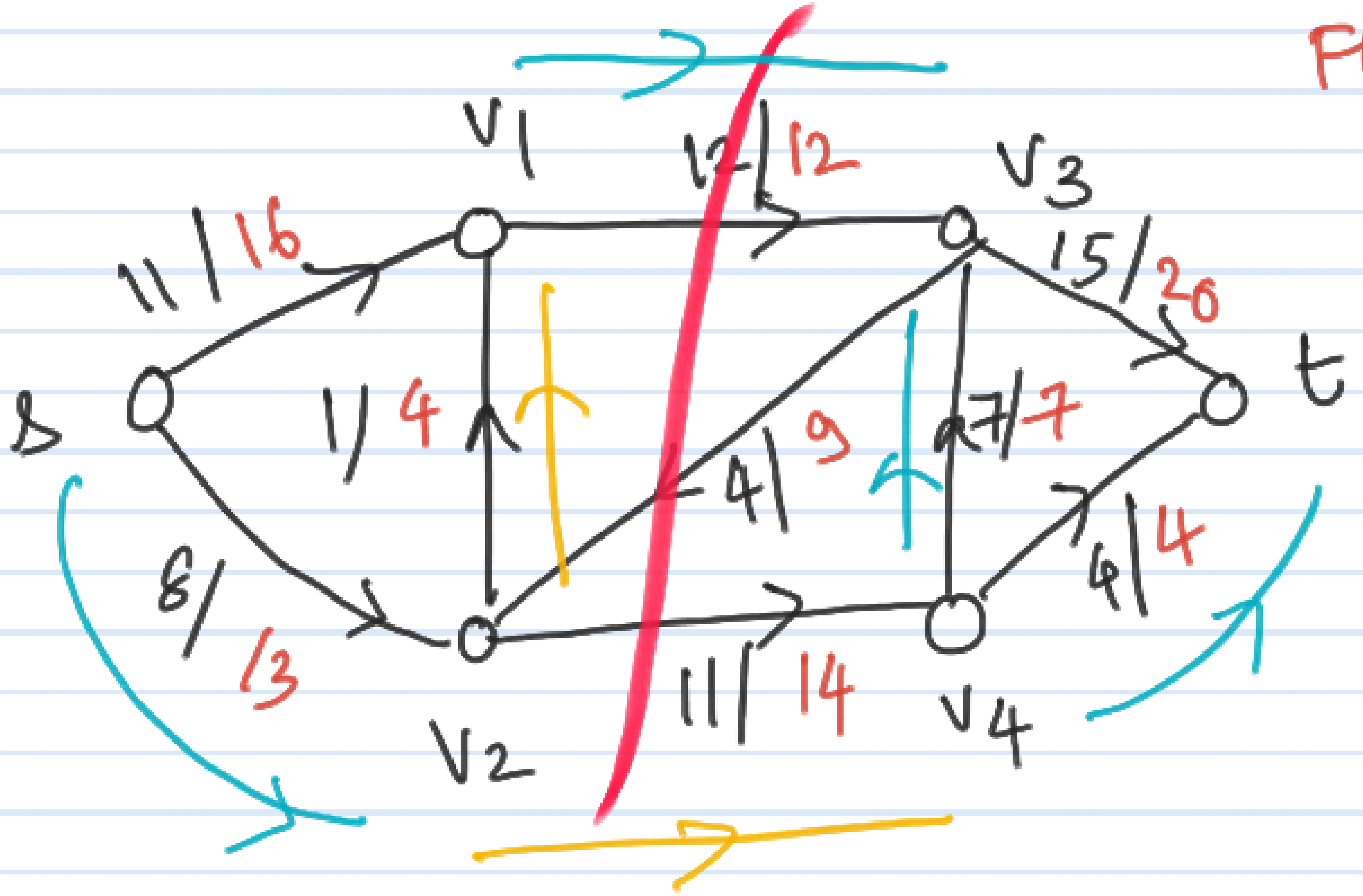


Cut in a flow n/w is a partition (S,T) of V : $s \in S$, $t \in T$

Net flow across a cut : $\sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$

Capacity of a cut : $\sum_{u \in S} \sum_{v \in T} c(u,v)$

Flows and cuts



write down capacity and net Flow

$(S_1, T_1) = (\{s, v_1, v_2\} \ \{v_3, v_4, t\}) \ ; f(S_1 T_1) := 19$

$(S_2, T_2) = (\{s, v_1, v_4\} \ \{v_2 \ v_3 \ t\}) \ \ f(S_2 T_2) = 19$

$f(S_1 T_1) = f(S_2 T_2) = |f|$

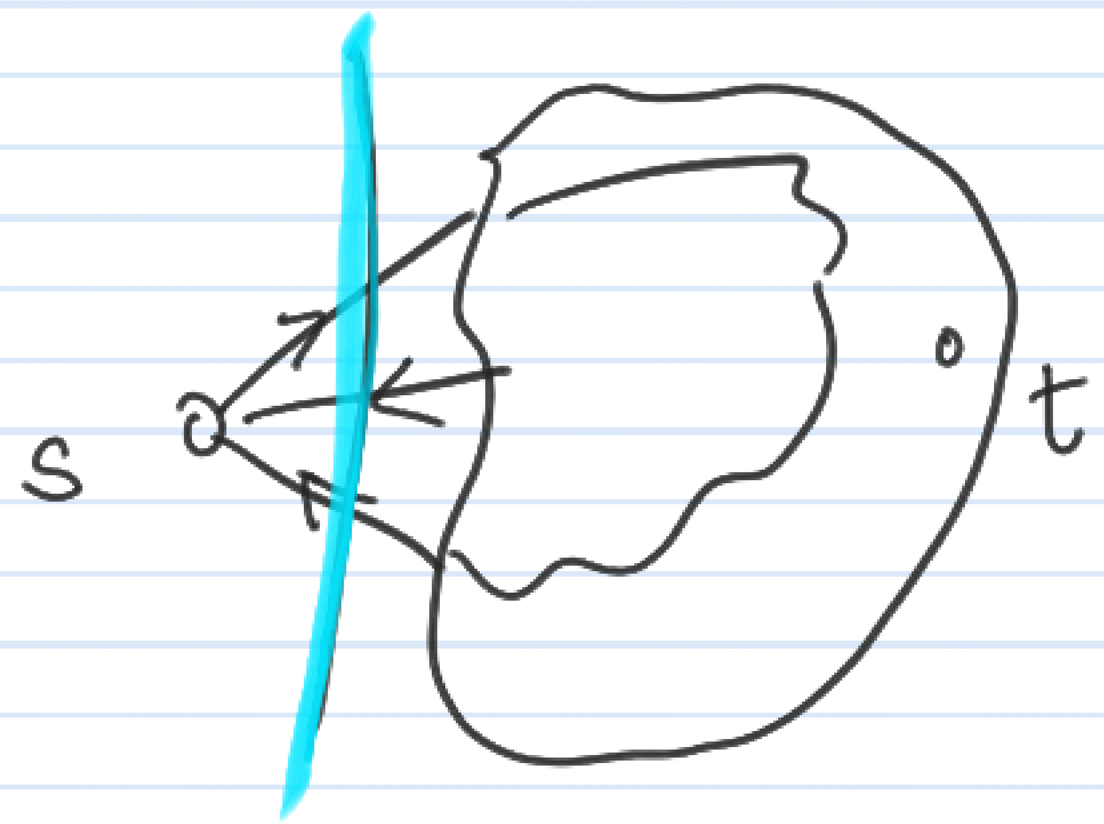
Minimum cut

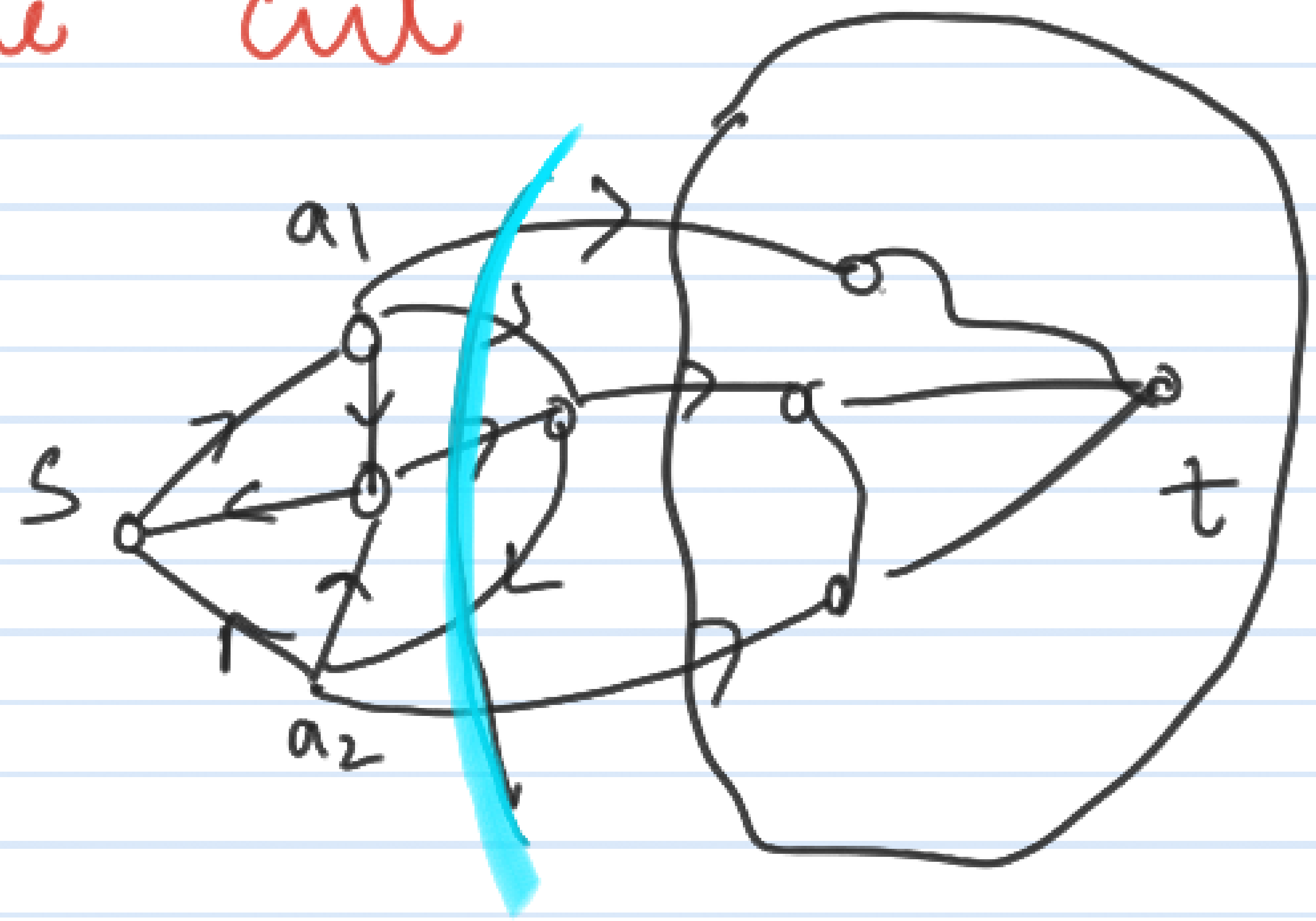
The cut that has minimum capacity is called as a min (s,t) cut.

- all cuts are s,t cuts in this context
- definition of capacity of cut and net flow are not symmetric

– make a note of this!

Net flow across a cut is invariant of the cut





$S_1 \; T_1$

$f(S_1 T_1) \le c(S_1 T_1)$

$S_2 \; T_2$

$f(S_2 T_2) \le c(S_2 T_2)$

Netflow across a cut is invariant of the cut

↳ assumming this statement what is the implication?

$$\text{netflow across } (s, V \setminus \{s\}) \le c(S_1, T_1)$$

$$\parallel \quad |f| \qquad\qquad \le c(S_2, T_2)$$

$$\vdots$$

$$\le c(S_k, T_k)$$

Net flow across a cut is invariant of the cut

↳ assumming this statement what is the implication?

value of any flow $\leq$ capacity of any cut

$\Downarrow$

value of max flow $\leq$ capacity of minimum cut.

Optimality of Ford Fulkerson algo.

Max flow Min cut Theorem

1) f is a max flow in G

2) The residual n/w G_f has no aug path

3) $|f| = c(S,T)$ for some cut (S,T) in G.

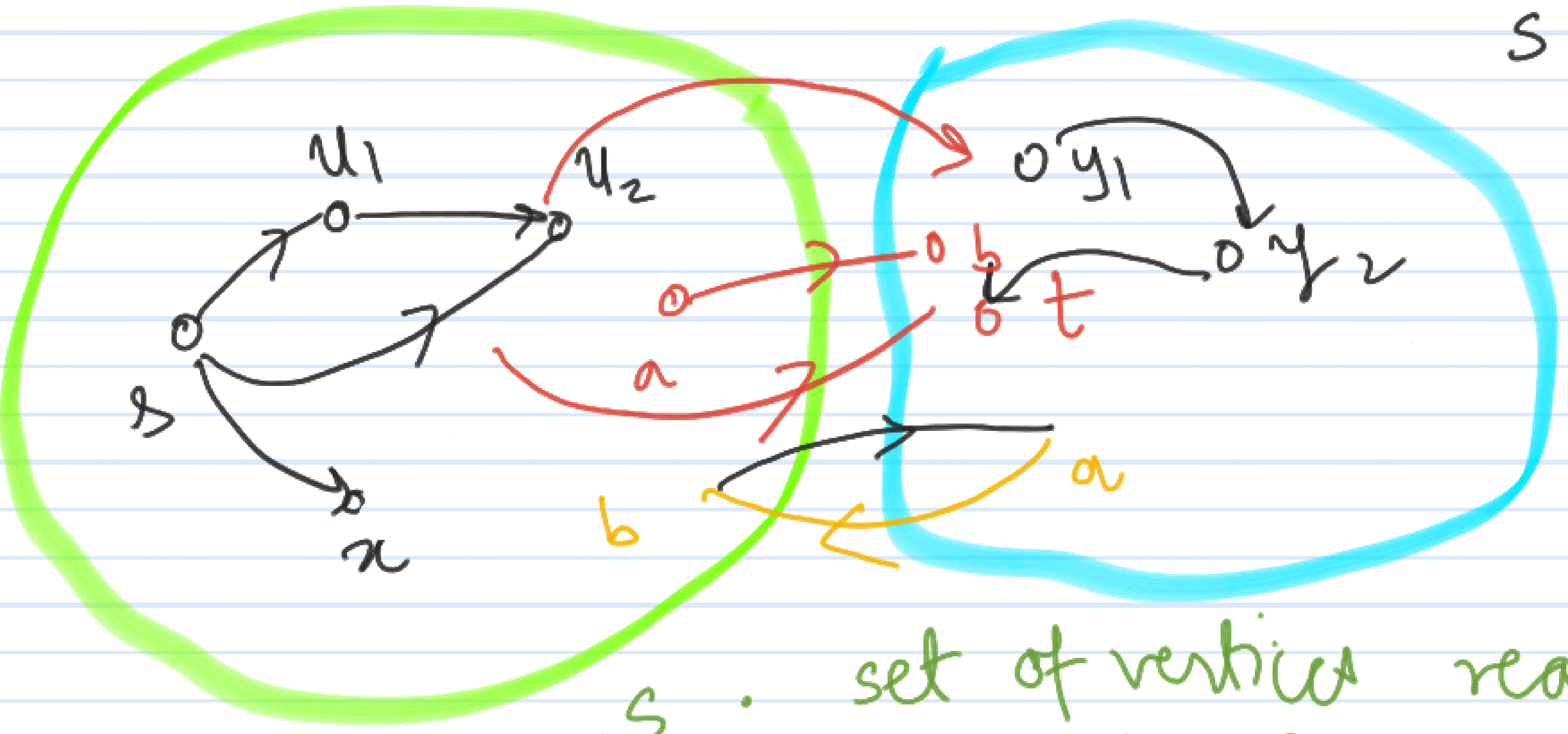
$1 \Rightarrow 2$, $2 \Rightarrow 3$, $3 \Rightarrow 1$.

Optimality of Ford Fulkerson algo.

$\boxed{\text{Max Flow Min cut Theorem :}}$

Exhibiting a matching cut

f : G_f : G_f does not admit any s,t, path.



$a \rightarrow b$ forward edges

$f(a,b) = c(a,b)$

$b \leftarrow a$ $f(a,b) = 0$

S : set of vertices reachable from s in G_f

Homework and discussion

- We obtained (S,T) cut in a particular way
- Try another method : collect in T_2 all vertices that can reach t in G_f $\quad S_2 = V \setminus T_2$

  — is (S_2, T_2) a min cut? Prove / Disprove

- Are there vertices which cannot reach t and cannot be reached from s in G_f → call them U

yes! , is there a relation of $|U|$ and # of min cuts?

We have assumed:

Net flow across a cut is invariant of the cut

Formally show

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

$$= |f| \rightsquigarrow \text{(value of flow)}$$

$\rightsquigarrow$ net flow across $(s, V \setminus \{s\})$

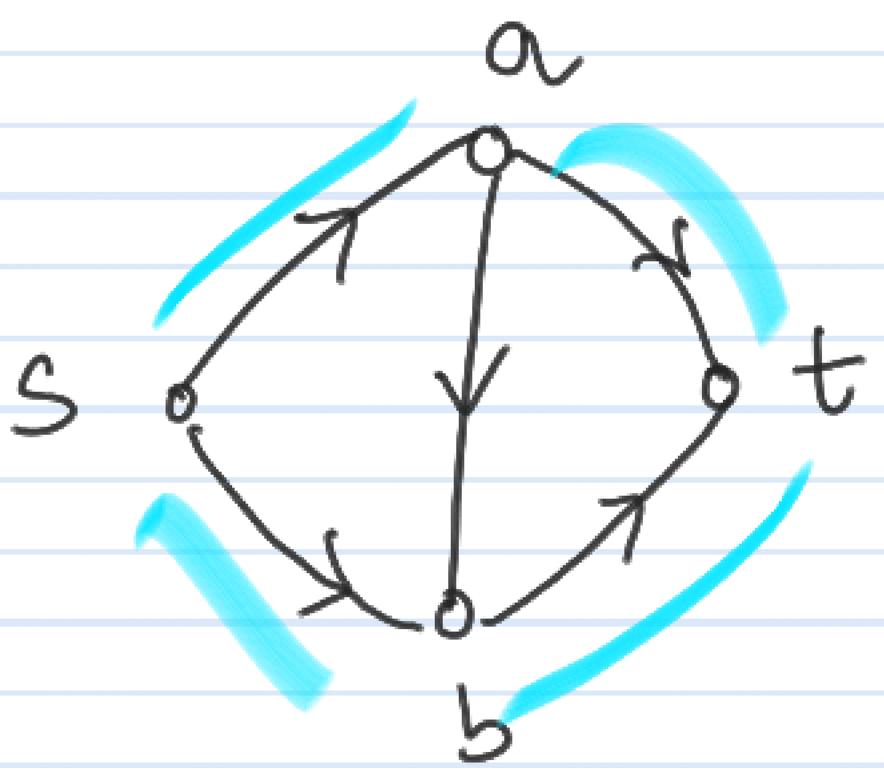
Uses conservation at all nodes except s, t

Summary of Ford Fulkerson Method.

- Idea of Residual n/w
- Finding aug path in Residual n/w
- At termination flow obtained is max flow (by max flow min cut theorem)
- Relation between flows and cuts.

Running time of basic algo : $\underbrace{|f^*|}\, O(m+n)$

Can we improve the running time?



$c(a,b) = 1$

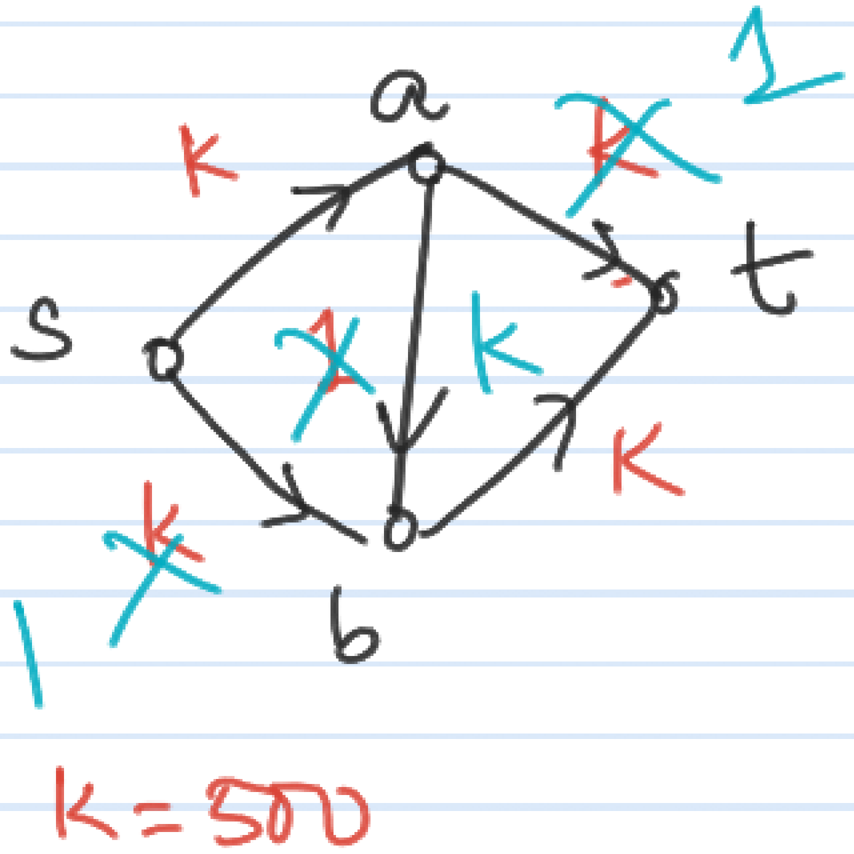
rest of the edges have large capacity

$\tilde{S}$ $\quad (\tilde{S} \cup \tilde{U}), \tilde{T}$ $\quad (\tilde{S} \cup X, \partial em$

$\tilde{T}$ $\quad \tilde{S}, (\tilde{U} \cup \tilde{T})$

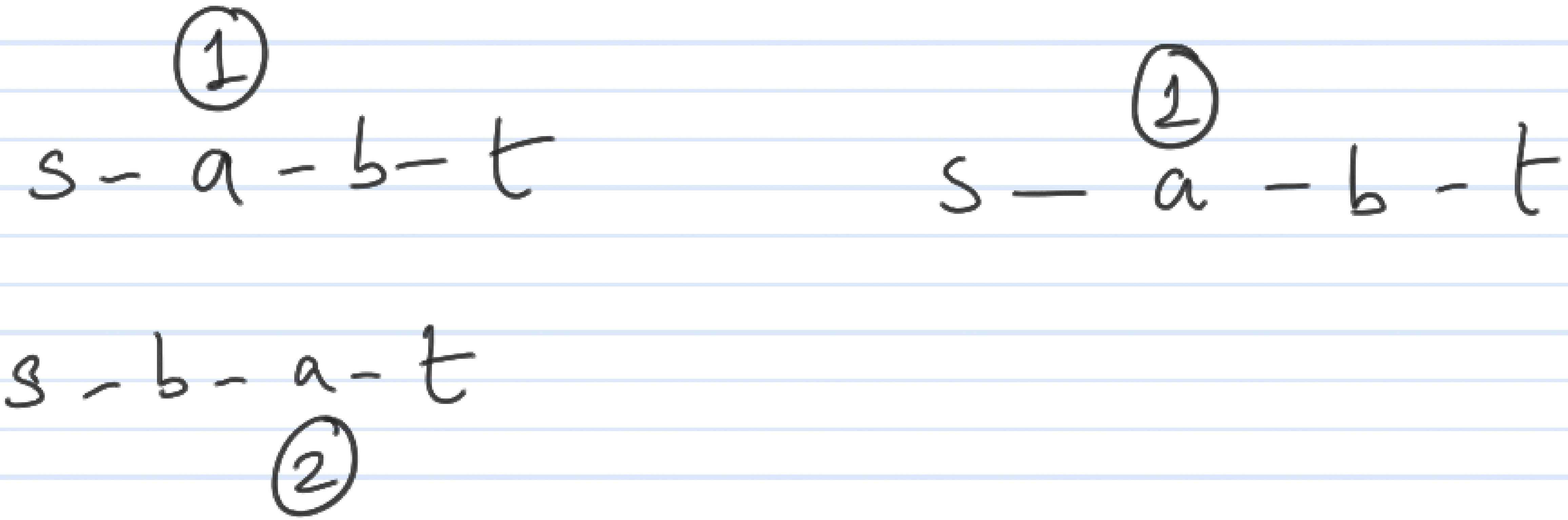
$V \setminus (\tilde{S} \cup \tilde{T}) = \tilde{U}$ $\quad X \subset \tilde{U}$

What happens in basic FF algo?



- Edge (a, b) becomes "critical" multiple times

can we bound this?



Ford Fulkerson Algorithm

↳ Edmonds Karp

Input : G, s, t, $c(e)$

start with $f(e) = 0$ for all e

while $\exists$ an aug. path ~~p~~ in G_f :

- find shortest aug. path, p in G_f

• aug. f along p to obtain f''

• set $f = f''$

end while

Shortest paths
unweighted directed
n/w

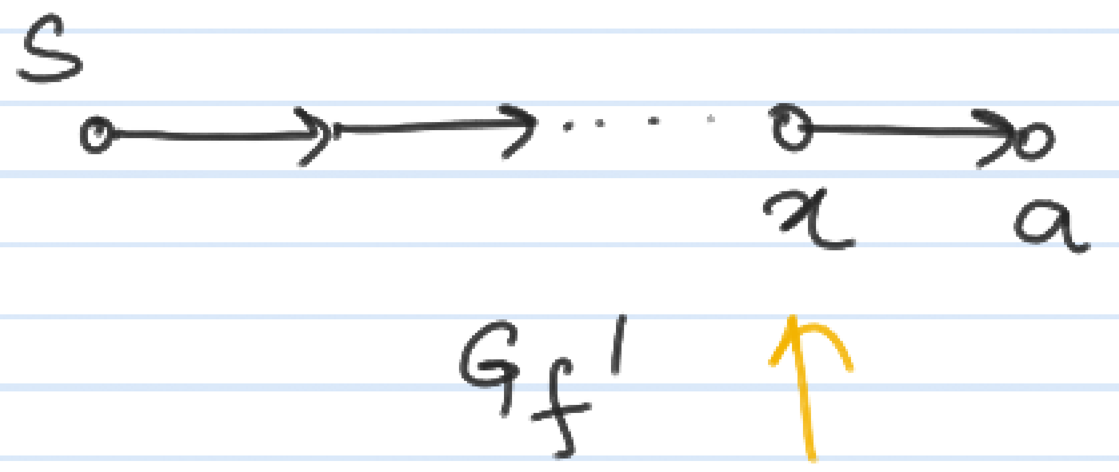
Edmonds Karp Algo.

- Use shortest path in residual n/w instead of any path.
- Note that while computing shortest path directed unweighted n/w is considered.
- How does using shortest path help?

How does shortest path help?

f and f' be 2 consecutive flows s-t.

$\exists\, a$ $\delta_f(s,a) > \delta_{f'}(s,a)$ and "a" is the $\boxed{\text{first}}$ such vertex on the path.



prev vertex on the s..a shortest path in $G_{f'}$

$\delta_{f'}(s,a) = \delta_f(s,x)+1$

Obs 1 : $\delta_{f'}(s,x) \geq \delta_f(s,x)$

by choice of first vertex to violate

Obs 2 : edge $(x,a) \notin G_f$ else

$\delta_f(s,a) \leq \delta_f(s,x)+1$

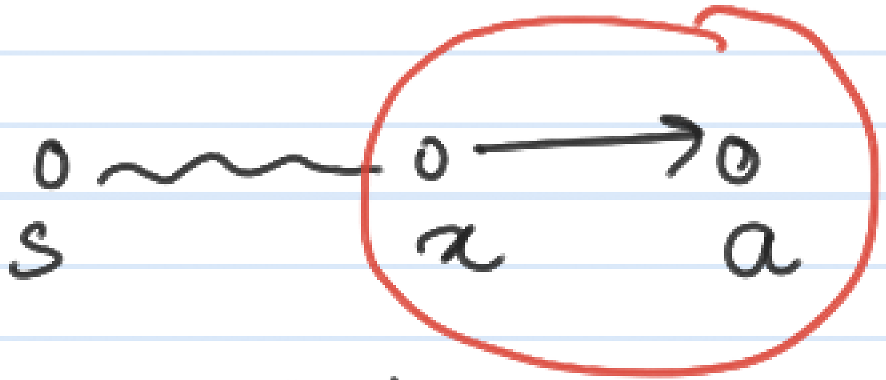
$\leq \delta_{f'}(s,x)+1 = \delta_{f'}(s,a)$

$\times$ contradiction

How does shortest path help?

f and f' be 2 consecutive flows s-t.

$\exists\, a \quad \delta_f(s,a) > \delta_{f'}(s,a)$ and "a" is the $\boxed{\text{first}}$ such vertex on the path.



$G_{f'}$

edge is newly added in $G_{f'}$ hence in f there was a flow along

$s \dashrightarrow a \overrightarrow{\longrightarrow x} \cdots t$

Thus in G_f

$$\delta_f(s,x) = \delta_f(s,a) + 1$$

$$\Rightarrow \delta_f(s,a) = \delta_f(s,x) - 1$$

$$\leq \delta_{f'}(s,x) - 1$$

$$= \delta_{f'}(s,a) - 1 - 1$$

a contradiction.

Bounding # of times an edge becomes "critical"

$$s \rightsquigarrow x \to a \rightsquigarrow t$$

$\boxed{G_f}$

$$\delta_f(s,a) \le \delta_{f'}(s,a)$$

Before $(x \to a)$ can become critical again we must send flow via $a \to x$

$$\delta_{f'}(s,x) = \delta_{f'}(s,a) + 1$$
$$\geq \delta_f(s,a) + 1$$
$$= \delta_f(s,x) + 2$$

$\Rightarrow$ shortest path distance of x has increased by at least 2

$$s \rightsquigarrow a \to x \rightsquigarrow t$$

$\boxed{G_{f'}}$

$\Rightarrow$ an edge can become critical at most $|V|/2$ times

Bound on # of iterations of Edmond's Karp algo

- an edge can get critical at most $O(n)$ times
- in each iteration at least 1 edge becomes critical

$\Rightarrow$ # of iterations $= O(m \cdot n)$

$\Rightarrow$ running time of Edmond's Karp $= O(m^2 n)$

↳ strongly polynomial time algo.