L21

MAX FLOW +

APPLICATIONS.

4102

4.07.2016

abhi shelat

Max flow

Min Cut

userid:

What are the 2 restrictions on a flow f: 1 capacity $f(e) < c(e)$

2 $\text{INFLOW}(v) = \text{OUTFLOW}(v)$ for all $v \in V - \{s, t\}$

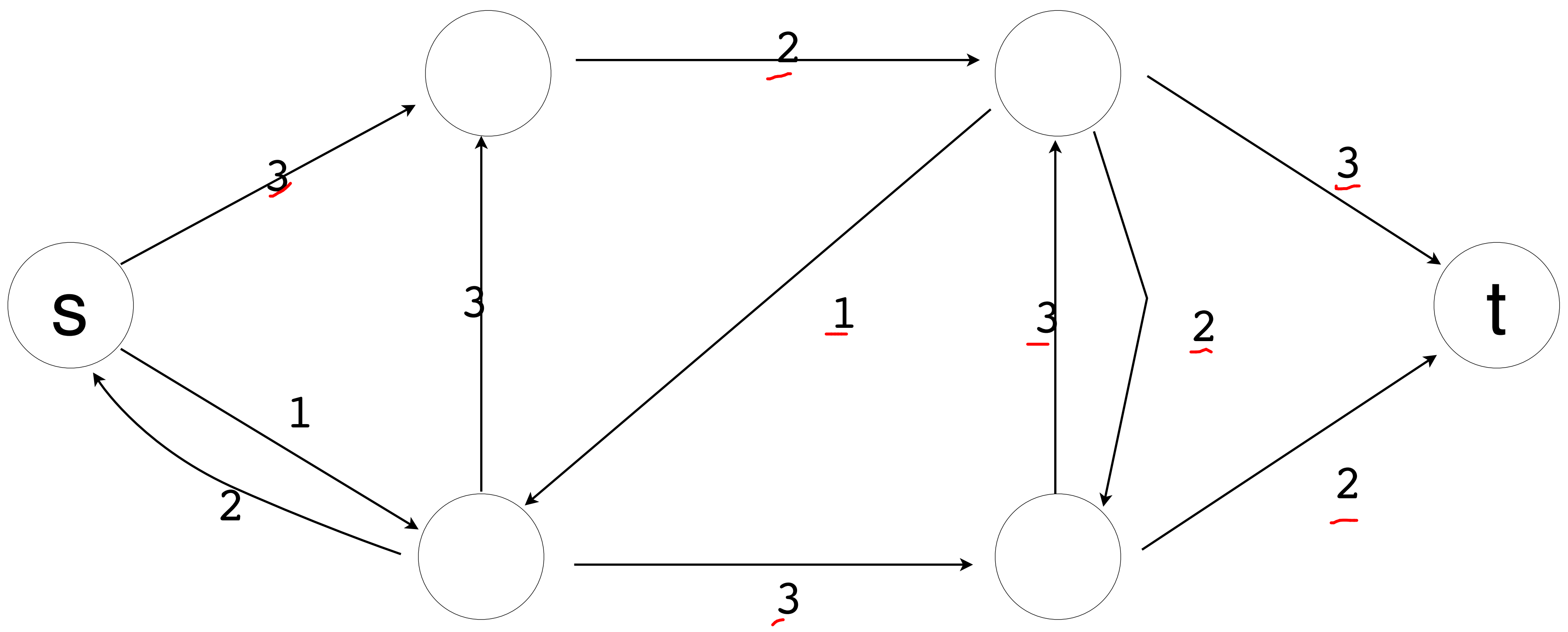
What is the value of a flow |f| :

$$|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s)$$

How does the Ford-Fulkerson algorithm work?

finds augmenting paths until it cannot.

example $G=(V,E), c$



flow

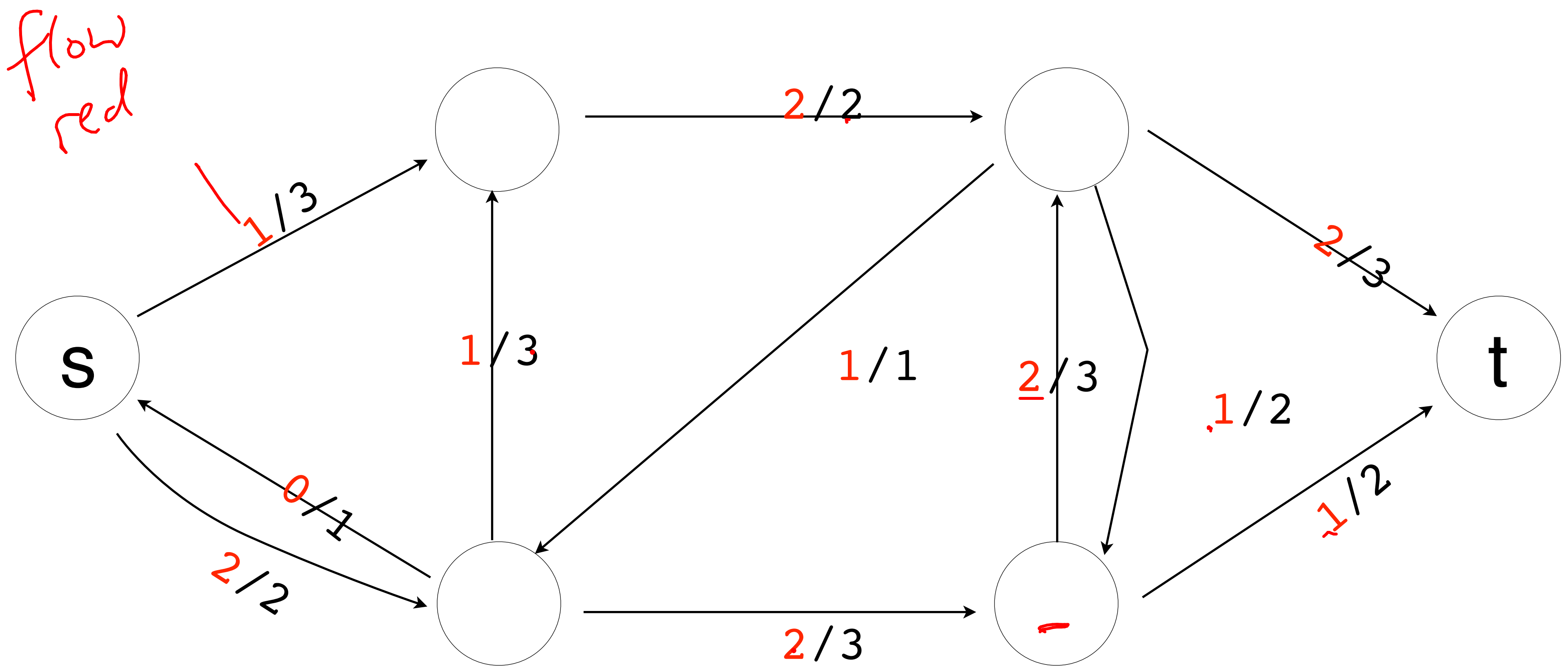
map from edges to numbers: $f(e) \to \mathbb{R}^+$

capacity constraint:

flow constraint:

$$|f| =$$

example



Residual graphs

$G_f = (V, E_f)$ given a flow f, define the

residual graph $G_f = (V, E_f)$, c_f

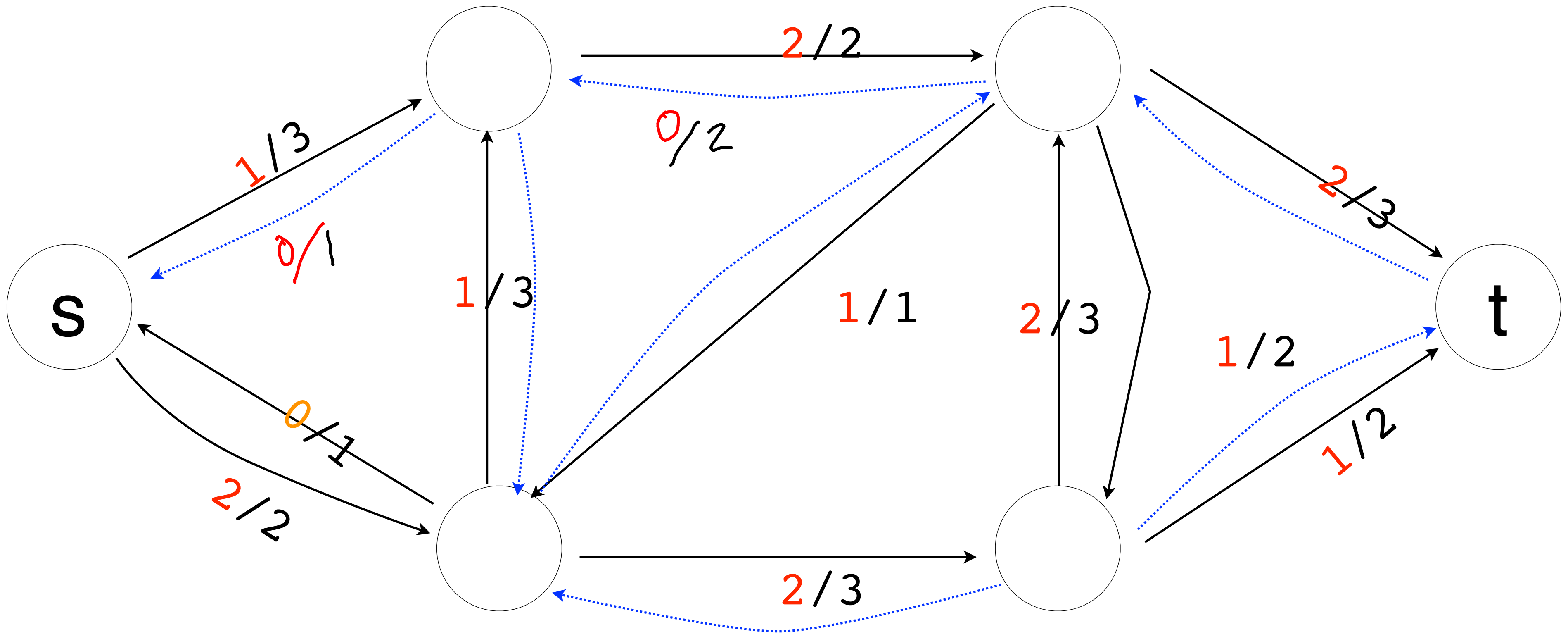
E_f: consists of all $e \in E$ such that

1. if $f(e) > 0$, then add e to E_f and $c_f = c(e) - f(e)$

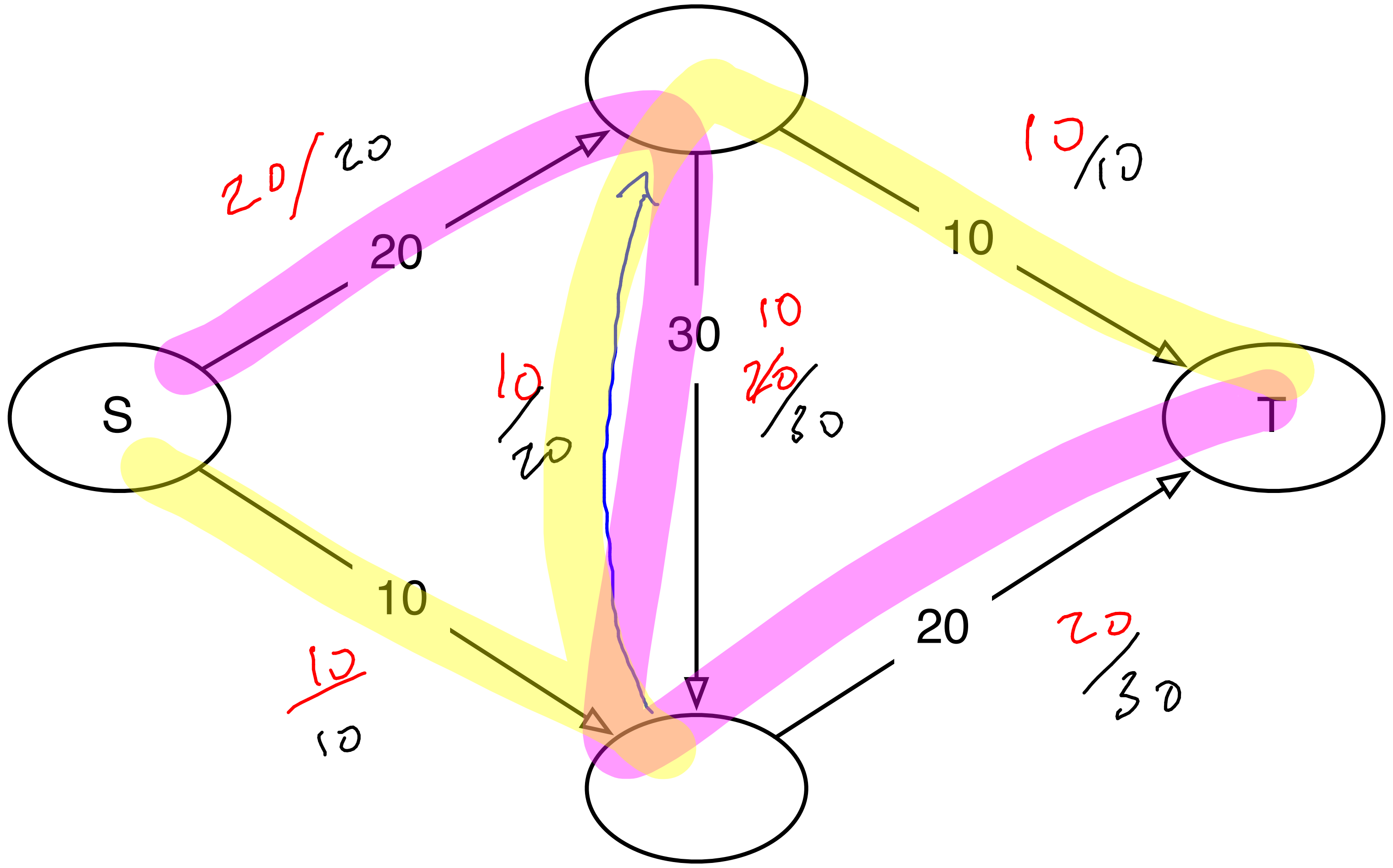
also, $e = (u, v)$ add the edge e'

$e' = (v, u)$ with $c_f(e') = f(e)$

example residual graph



why residual graphs ?



augmenting paths

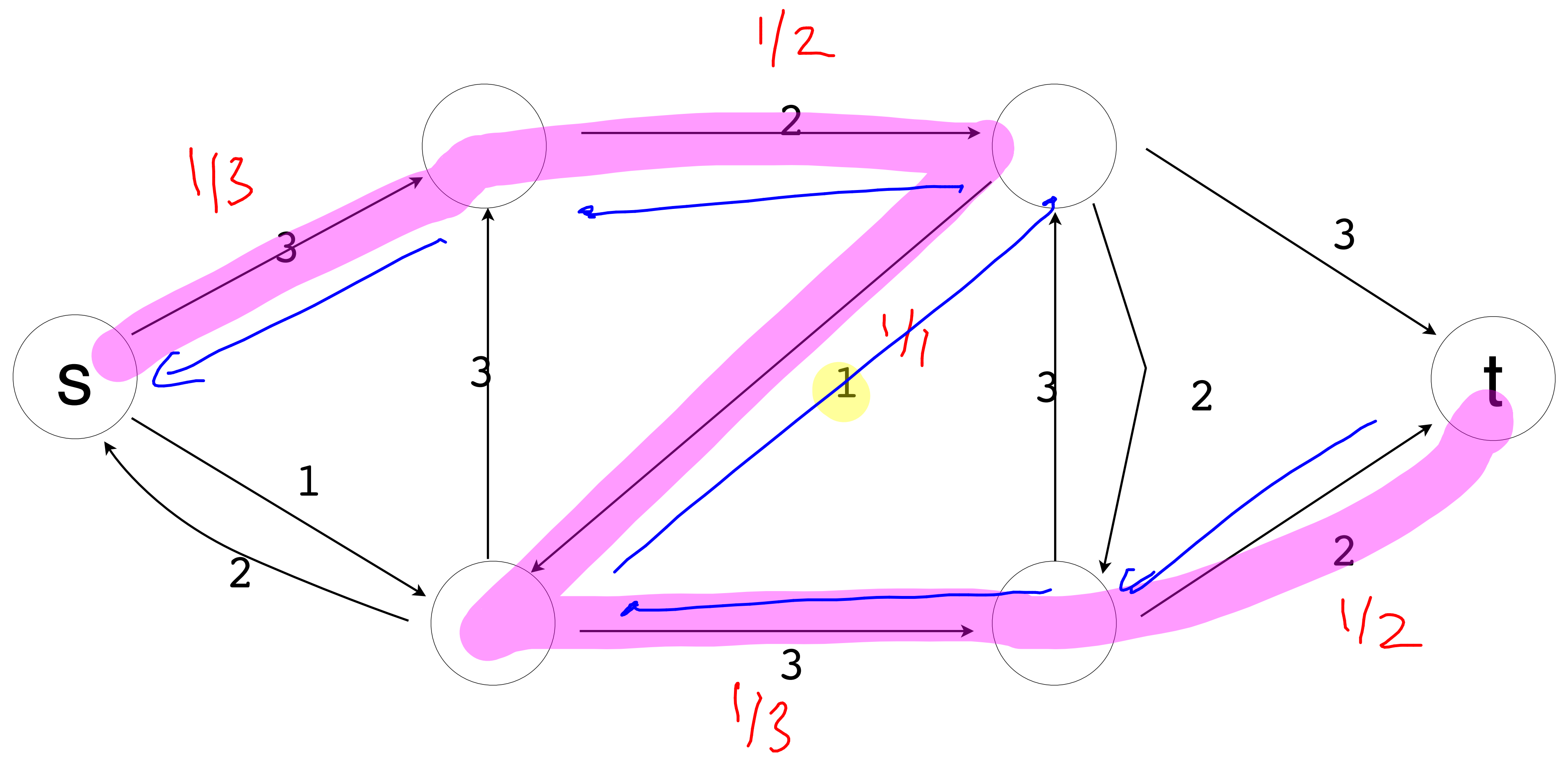
Def: A simple path in G_f from s to t.

Ford-Fulkerson

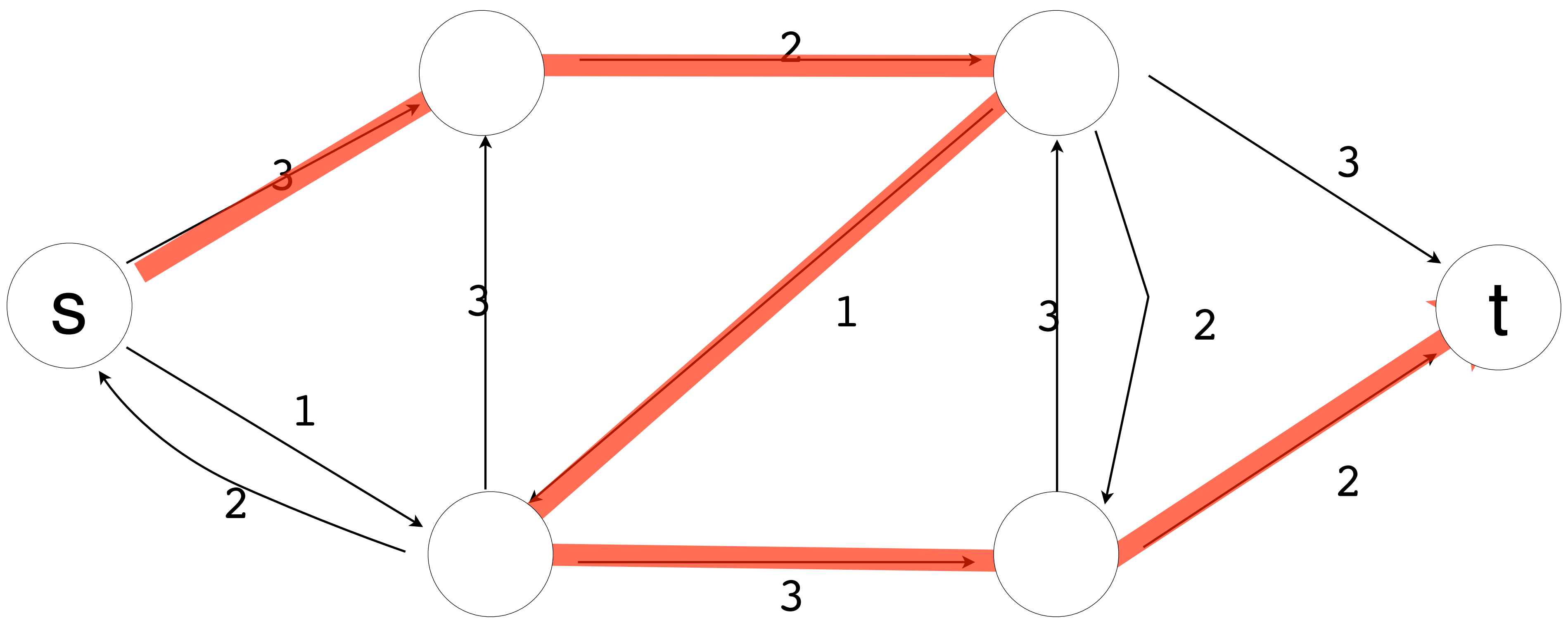
initialize $f(u,v) \leftarrow 0 \; \forall u,v$

while exists an augmenting path *p* in G_f

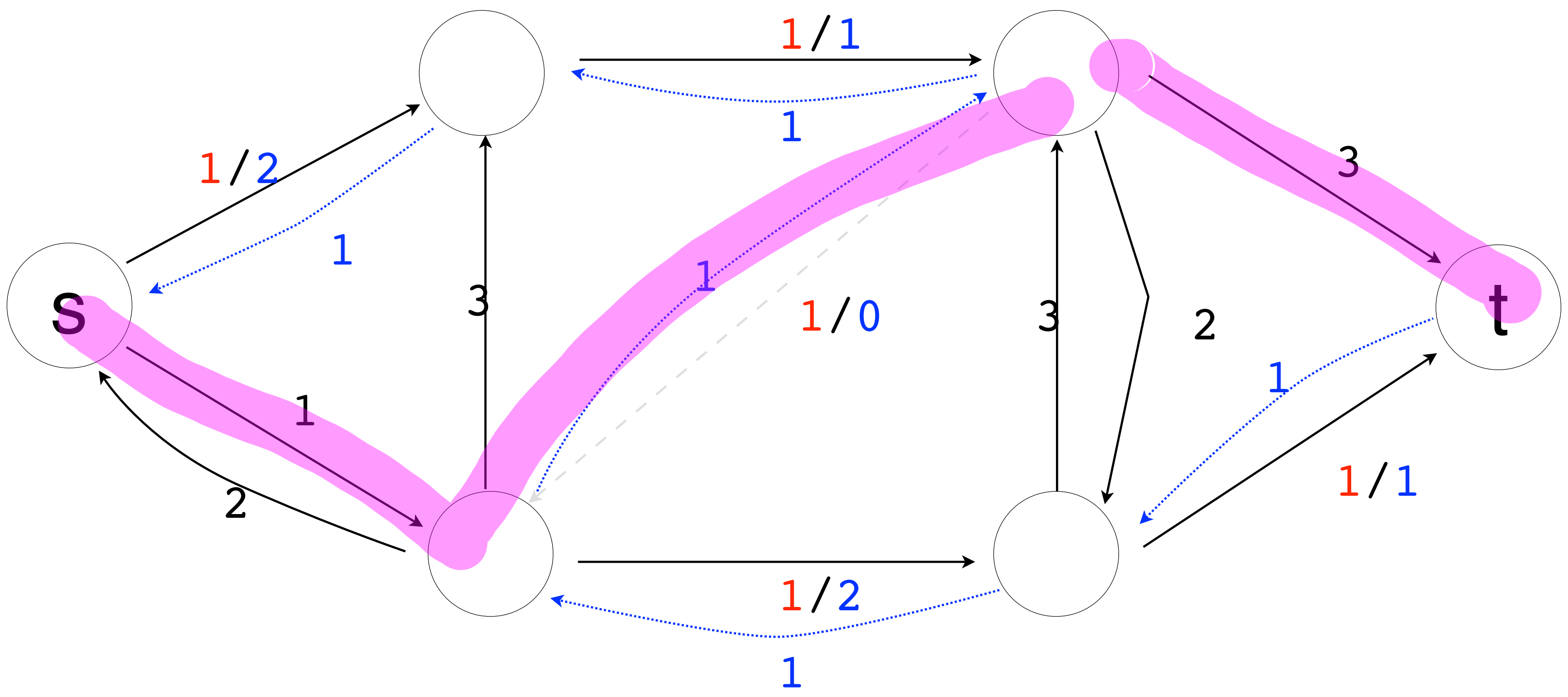
augment *f* with $c_f(p) = \min_{(u,v)\in p} c_f(u,v)$



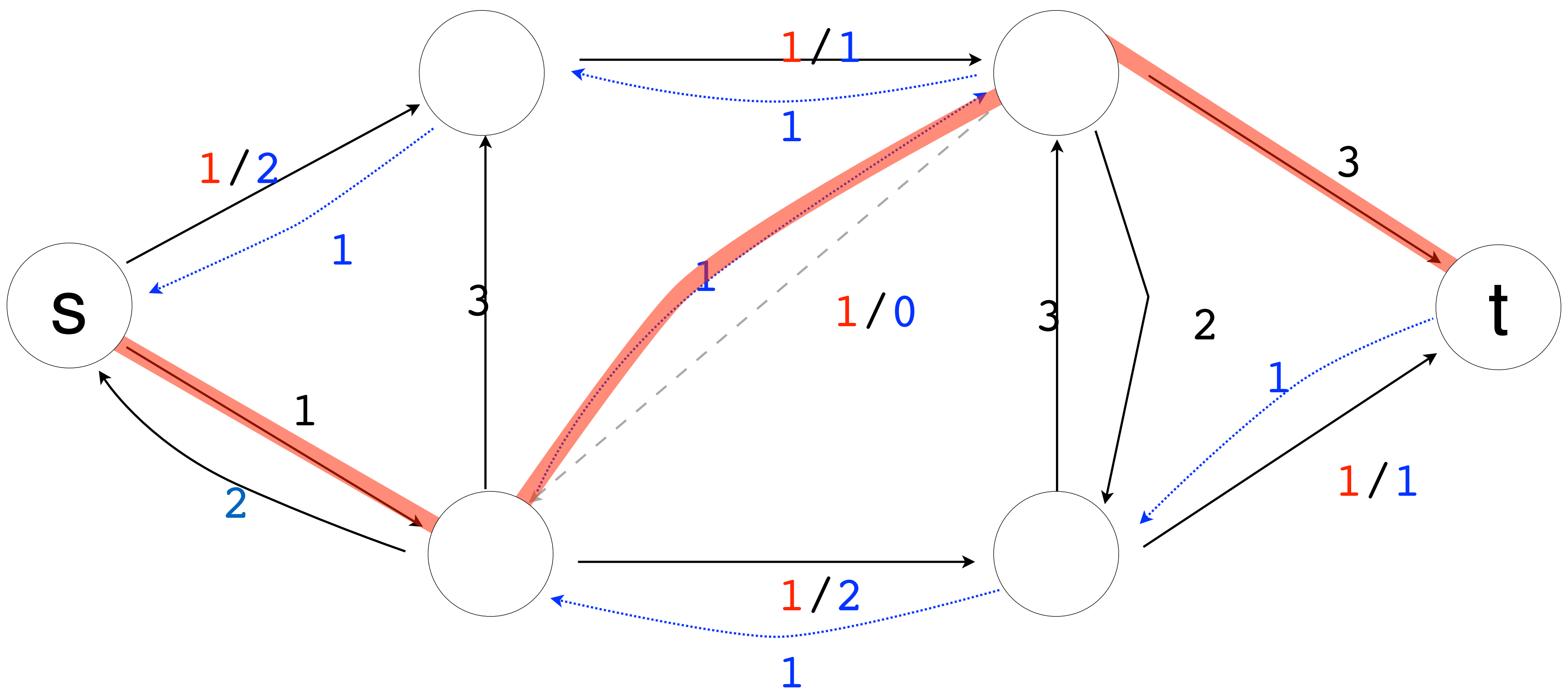
1/2
2
1/3
3
3
s
1
2
1/1
1
3
3
2
t
2
1/2
3
1/3


s
t
3
2
1
2
3
1
3
3
2
3
2

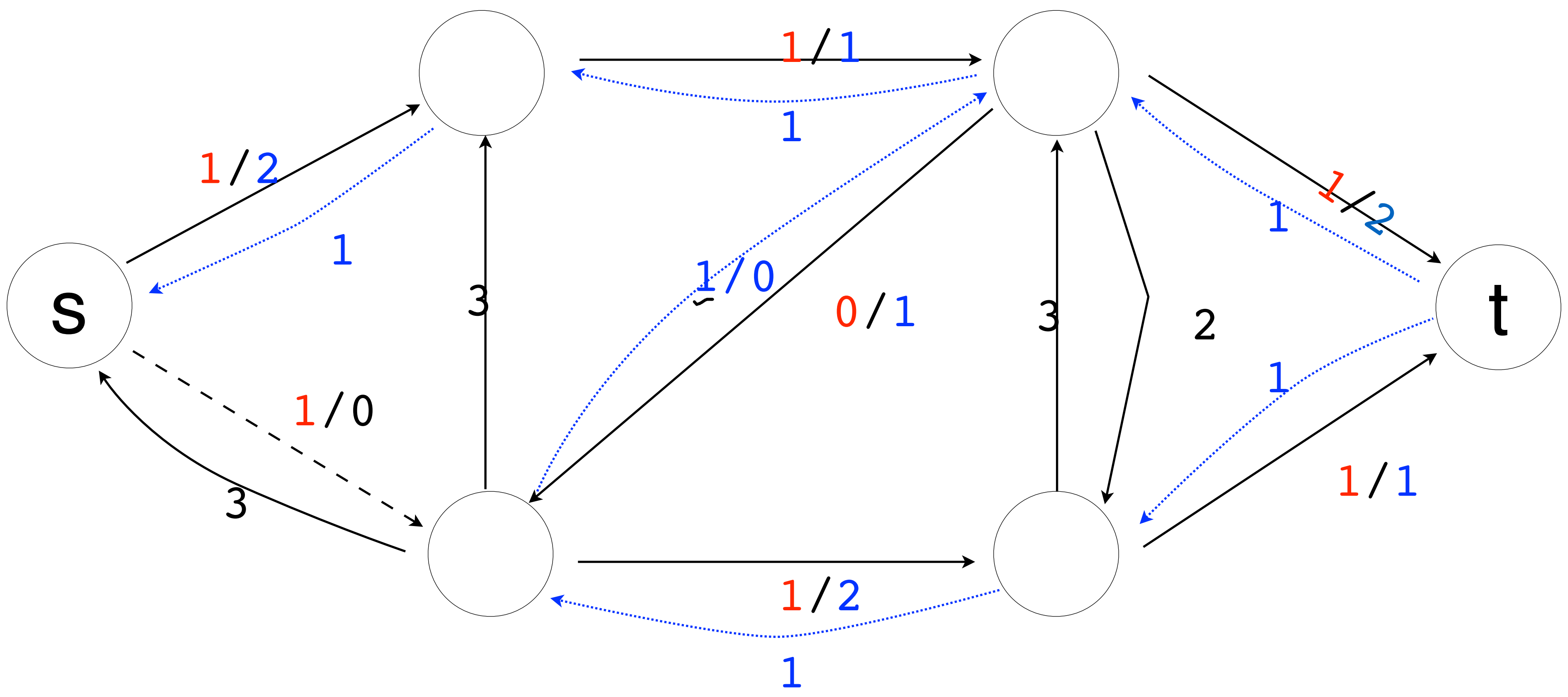
current flow/remaining capacity



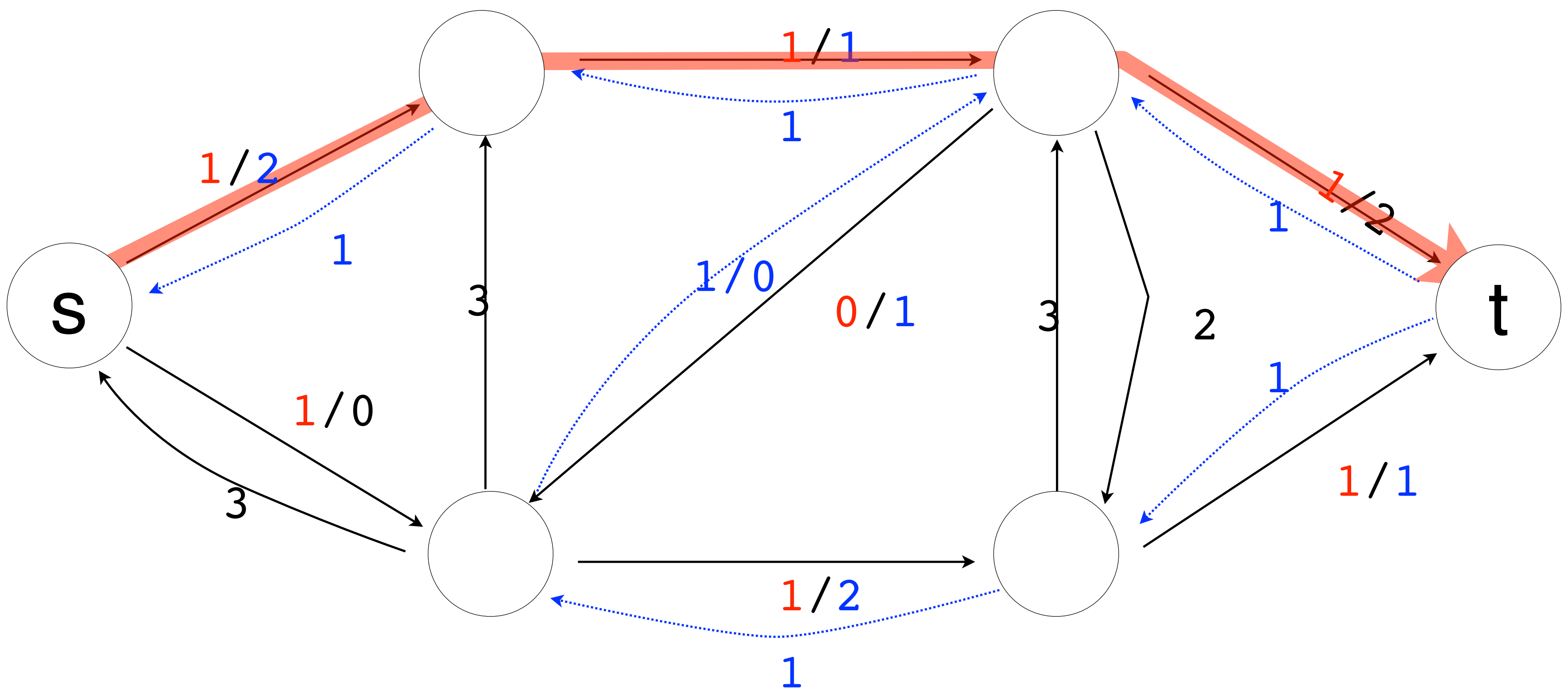
current flow/remaining capacity

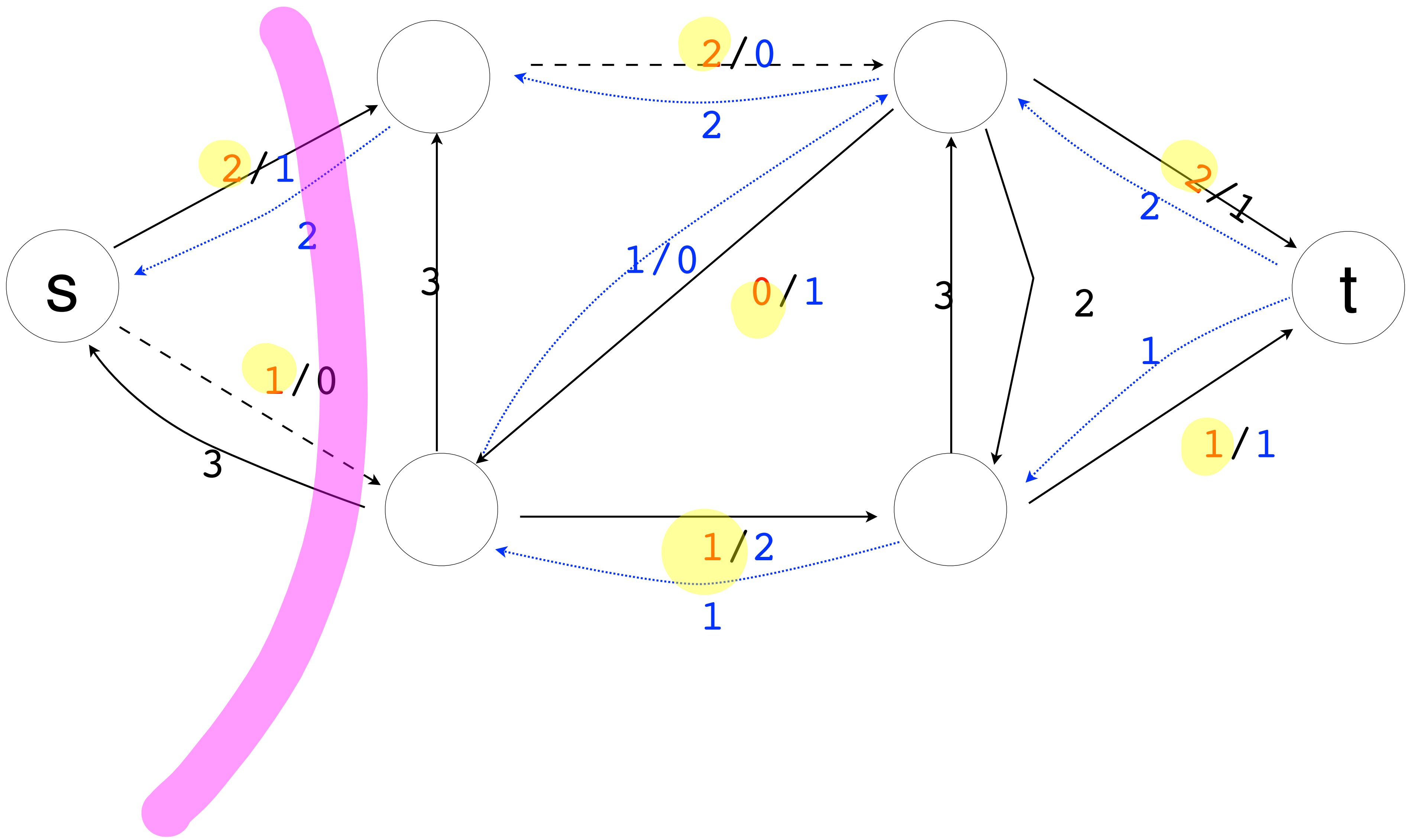


current flow/remaining capacity



current flow/remaining capacity





s
t
2 / 1
2
1 / 0
3
3
2 / 0
2
1 / 0
0 / 1
1 / 2
1
3
2
2 / 1
2
1
1 / 1

ford-fulkerson

initialize $f(u,v) \leftarrow 0 \; \forall u,v$
while exists an augmenting path *p* in G_f → #loops $\leq |f|$
augment *f* with $c_f(p) = \min_{(u,v)\in p} c_f(u,v)$

time to find an augmenting path: $\Theta(V+E)$ BFS.

number of iterations of while loop:

#loops $\leq |f|$ because each iteration adds at least 1 unit to the flow.

Running time $O(E|f|)$

Cuts

Def of a cut: Partition (S,T) such that $s \in S$ and $t \in T$ and $V = S \cup T$.

cost of a cut:

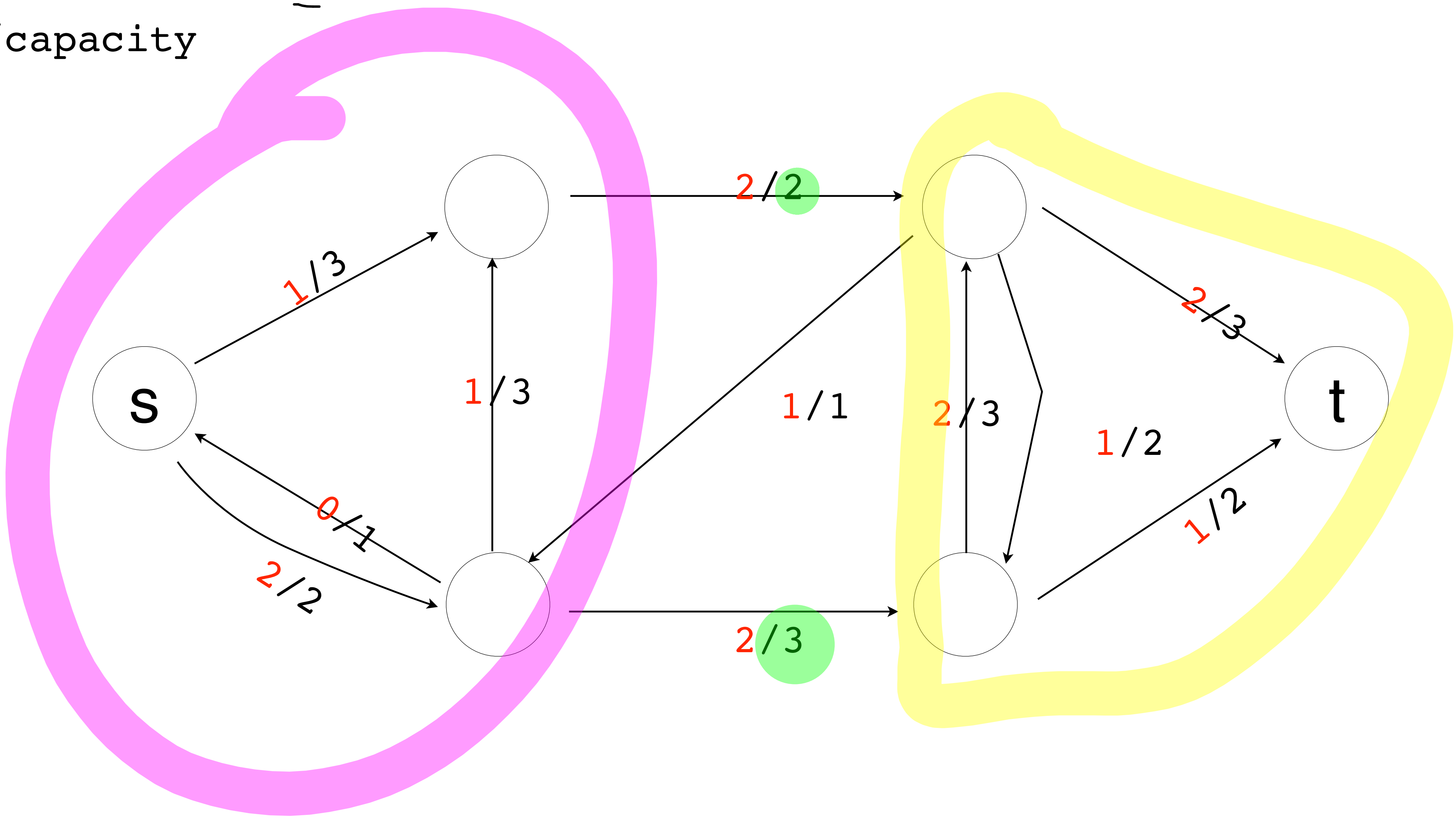
$$||S,T|| = \sum_{u \in S} \sum_{v \in T} c(u,v)$$

lemma: [MinCut] for any $f, (S, T)$ $|f| \leq \|S, T\|$

for any $f, (S,T)$ it holds that $|f| \leq ||S,T||$

$|f| \leq 5$

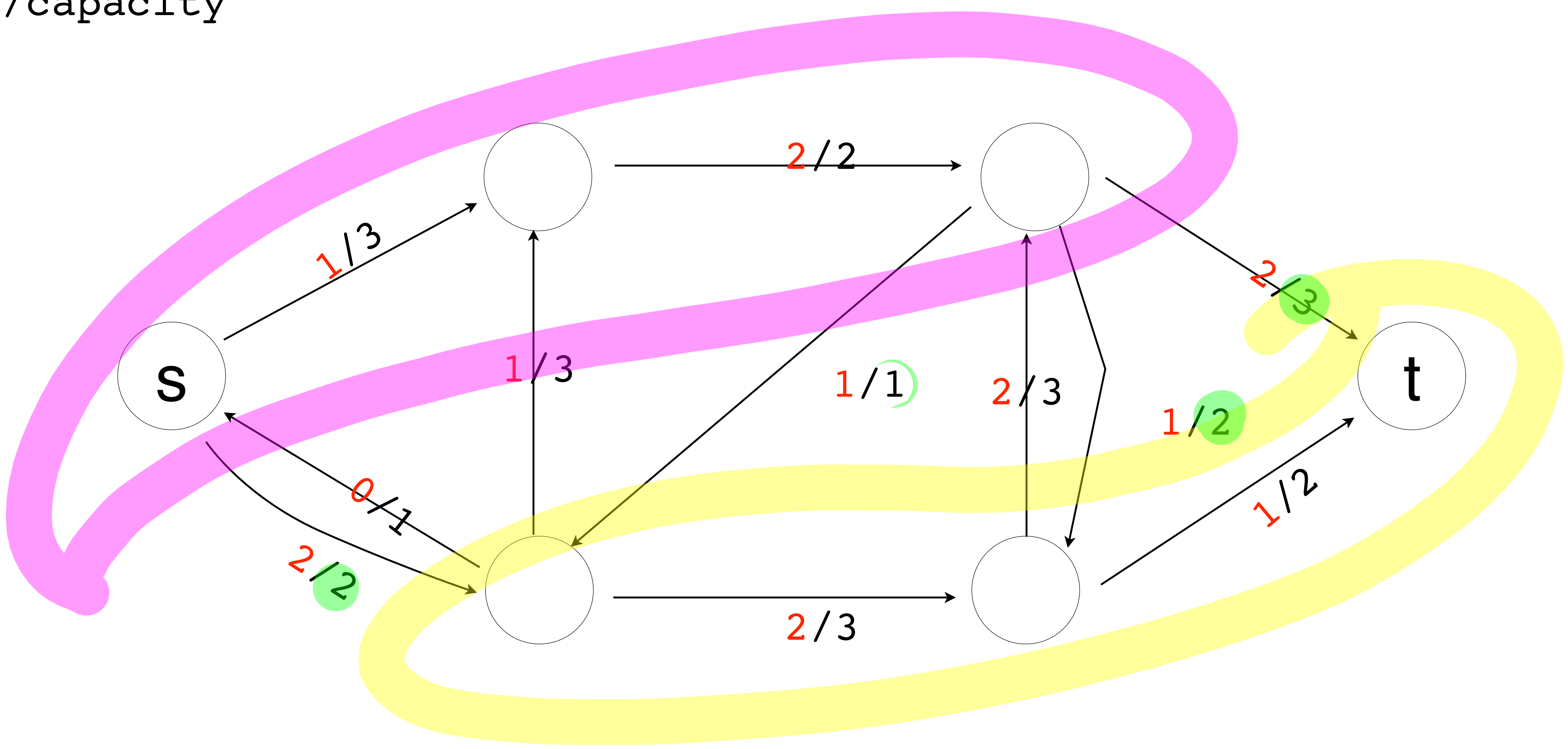
flow/capacity



for any $f, (S,T)$ it holds that $|f| \leq ||S,T||$

$|f| \leq 5$

flow/capacity



Main point

for any $f, (S,T)$ it holds that $|f| \leq ||S,T||$

Proof: Consider some flow f:

$$|f| = \underbrace{\sum_{v\in V} f(s,v) - \sum_{w\in V} f(w,s)}$$

for s

① for any $u \in S - \{s\}$ we know:

$$\sum_{v\in V} f(u,v) - \sum_{w\in V} f(w,u) = 0$$

by adding "0" in the form of ①

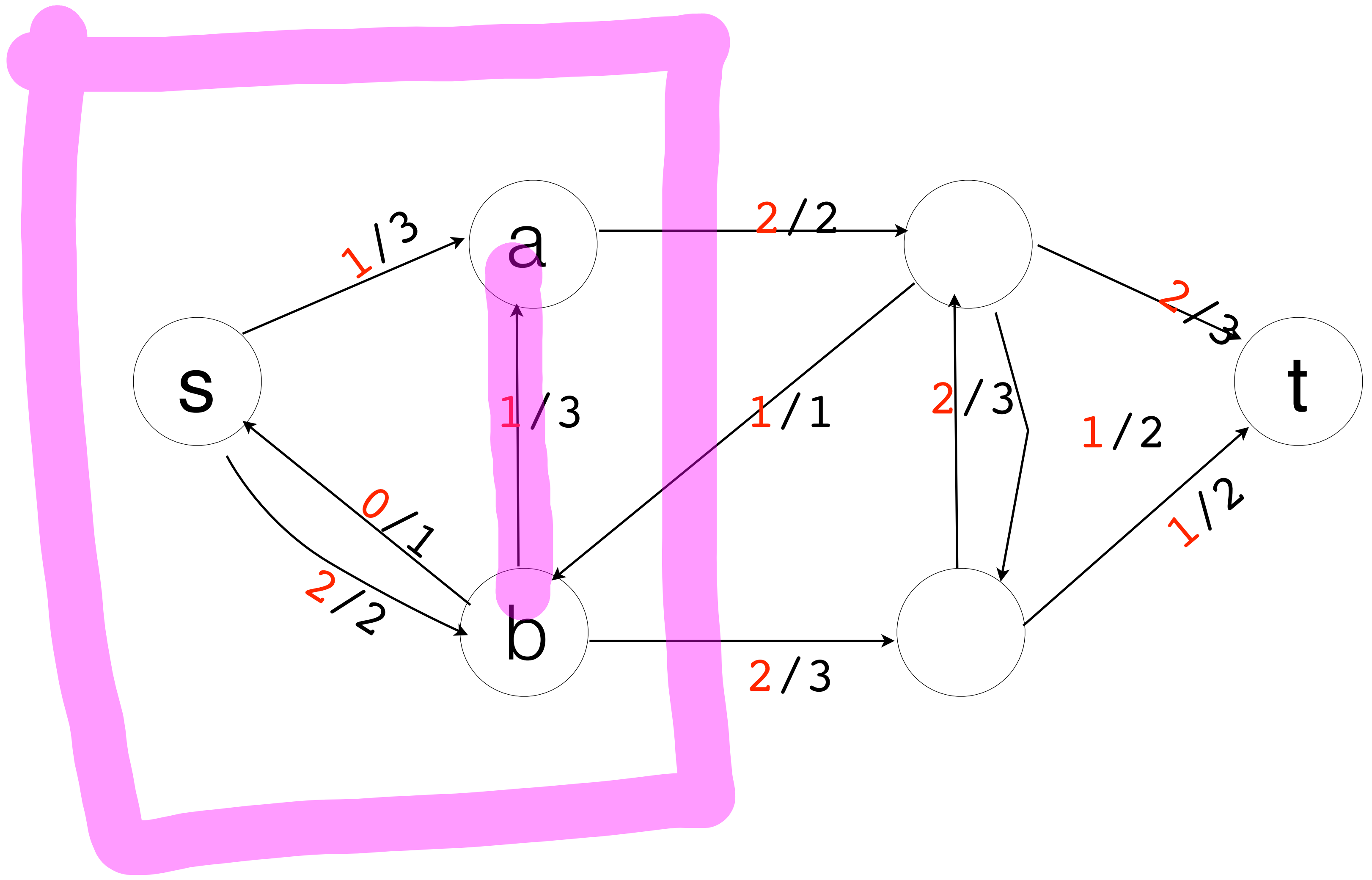
$$= \sum_{u\in S}\left[\sum_{v\in V} f(u,v) - \sum_{w\in V} f(w,u)\right]$$

$$= \sum_{u\in S}\left[\sum_{v\in T} f(u,v) + \underbrace{\sum_{v\in S} f(u,v) - \sum_{w\in S} f(w,u)} - \sum_{w\in T} f(w,u)\right] \qquad (1)$$

A property to remember

For any $f, (S,T)$ it holds that $|f| \leq ||S,T||$

proof:



Edge (b,a) is in S.

(2)

f(b,a) is added & then subtracted

⇒ contributes 0 to |f|

$$|f| = \sum_{u \in S} \left[\sum_{v} f(u,v) + \sum_{v \in S} f(u,v) - \sum_{w \in S} f(w,u) - \sum_{w} f(w,u) \right]$$

u = b f(b,a)

u = a − f(b,a)

for any $f, (S, T)$ it holds that $|f| \leq ||S, T||$

(finishing proof)

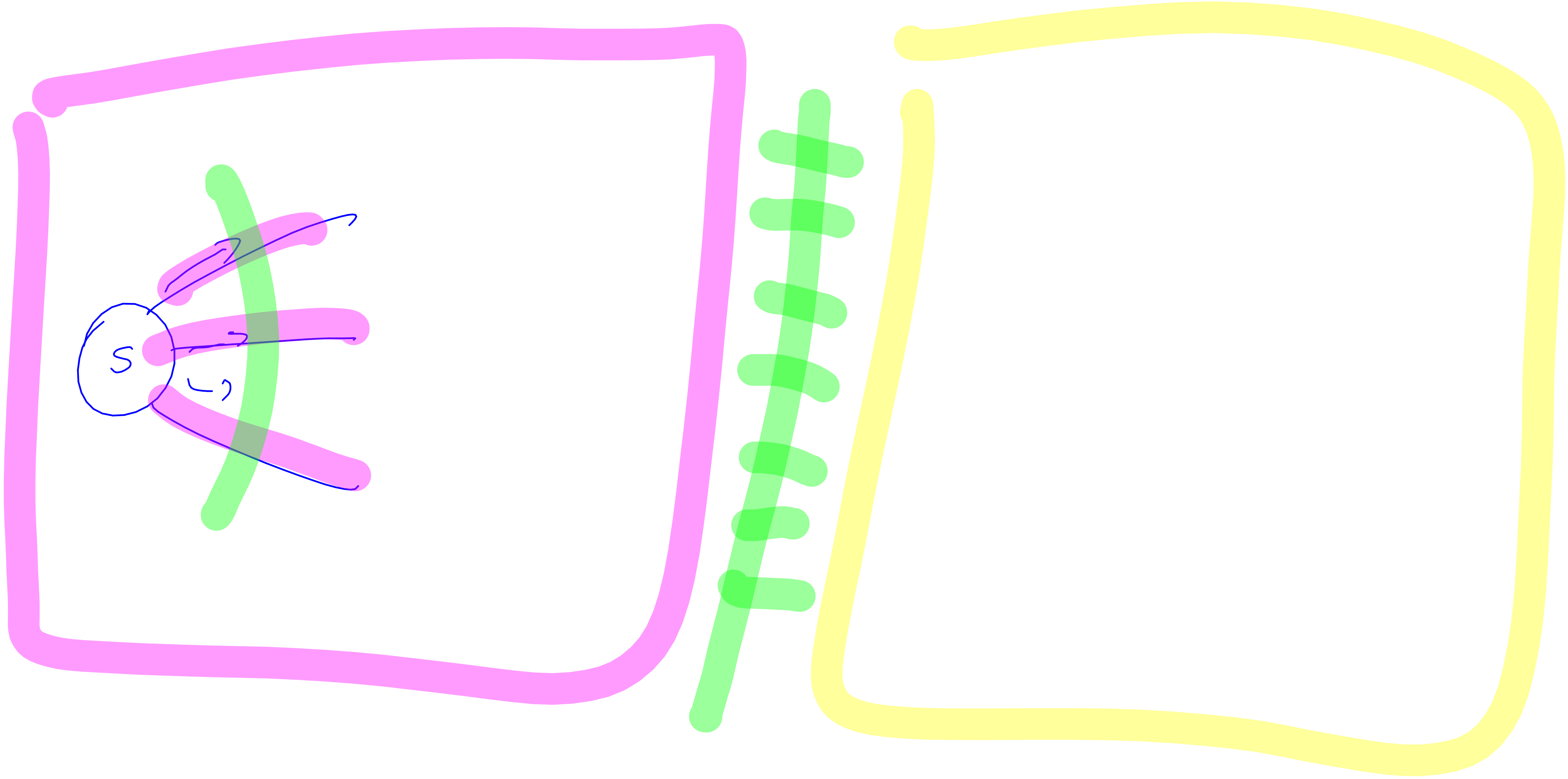
$$|f| = \sum_{u \in S} \left[\sum_{v \in T} f(u,v) - \sum_{w \in T} f(w,u) \right]$$

follows by (1) and (2)

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

for $u \in S$
for $v \in S$
sum $+= f(u,v)$

$$\leq \sum_{u \in S} \sum_{v \in T} c(u,v) = ||S, T||$$


S

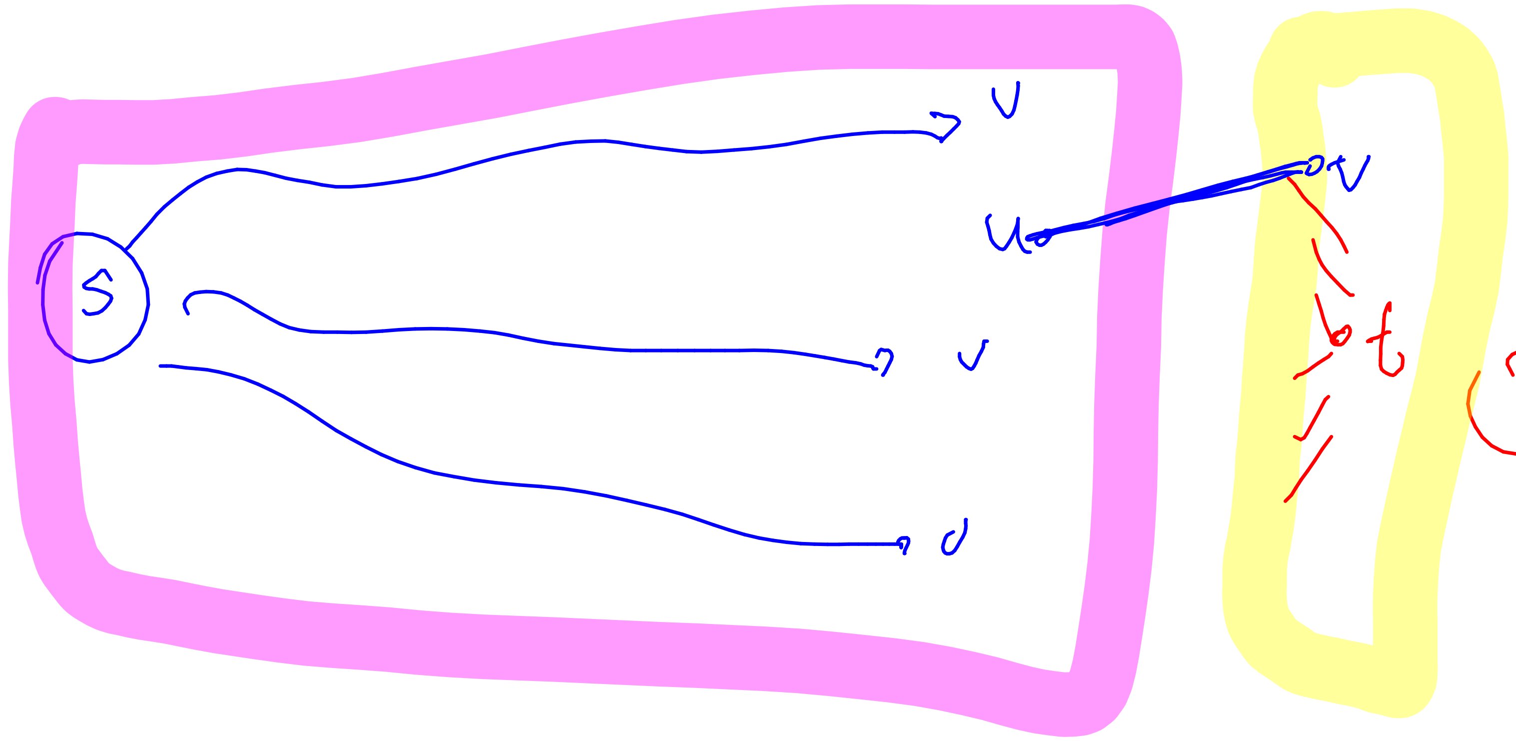
Thm: max flow = min cut

(duality) LP

$$\max_f |f| = \min_{S,T} ||S,T||$$

If f is a max flow, then G_f has no augmenting paths.

Define $S = \{ v \mid \exists$ a path p from s to v with $c_f(p) > 0$ in $G_f \}$



Define $T = V - S$.

① (S,T) is a cut.

$s \in S$ (Yes)

is $t \in S$?? (No) b/c of first line

there are no aug paths from s to t

Thm: max flow = min cut

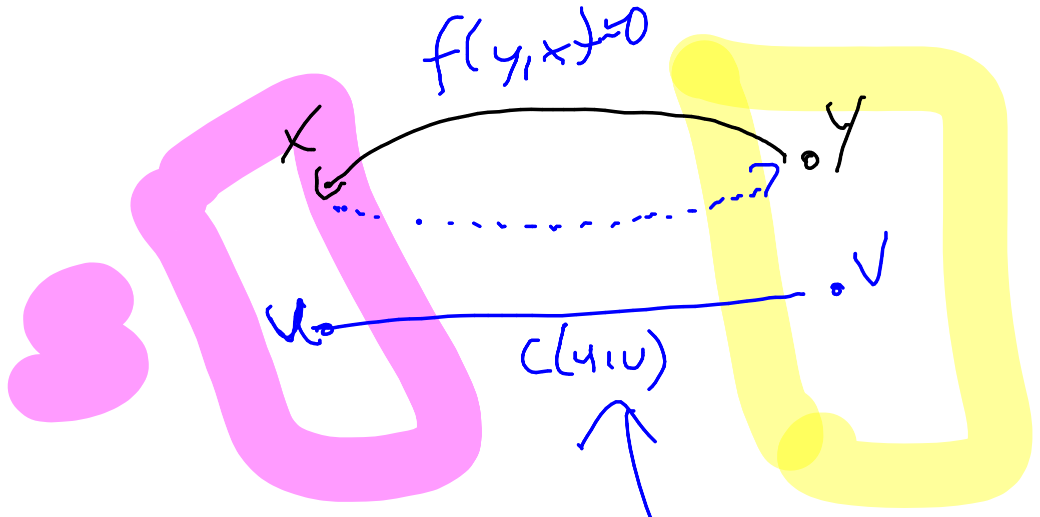
$$\max_f |f| = \min_{S,T} ||S,T||$$

① Consider (u,v) with $u \in S$ and $v \in T$.

$\underline{f(u,v)} = \underline{c(u,v)}$ why?? if $f(u,v) < c(u,v)$, then

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

which implies



must have zero residual capacity.

otherwise v would be in S.

② $f(y,x)=0$ for any $y \in T$ and $x \in S$.

why?? B/c o/w there would be a residual edge from $x \to y$ w/pos capacity $\Rightarrow y \in S$.

Why FF works

$G_f \Rightarrow (S,T)$

(continued)

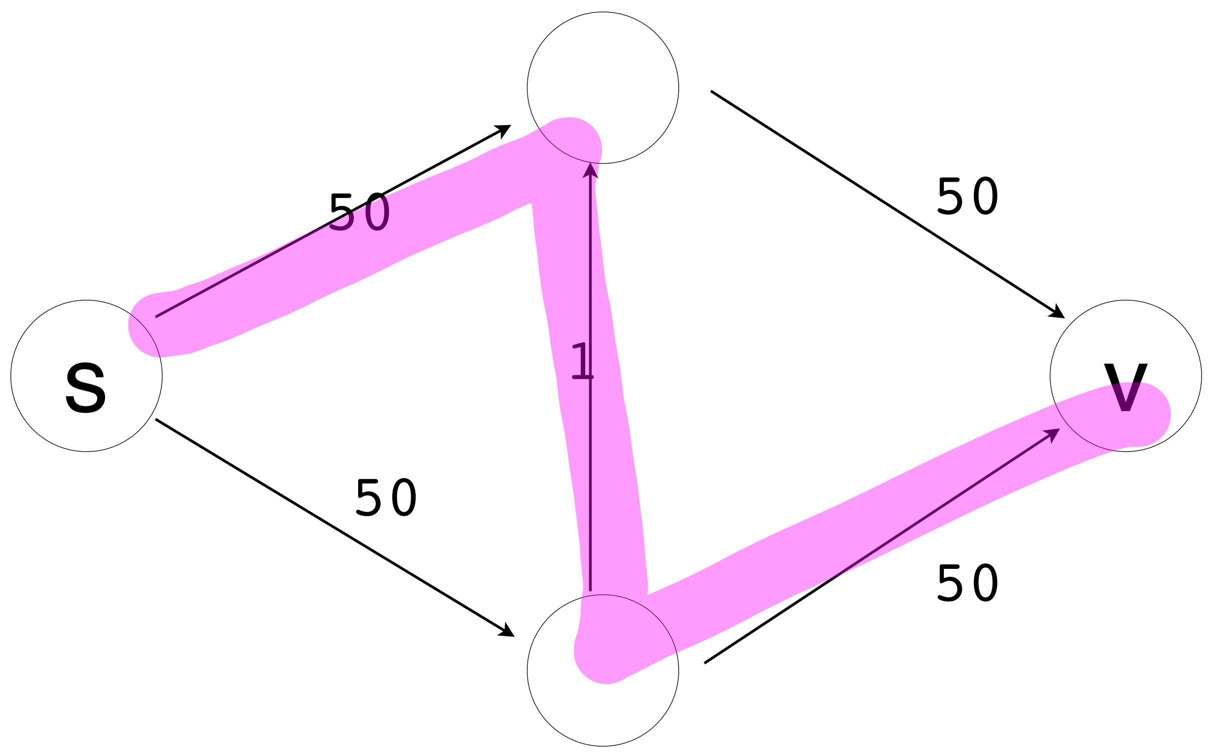
$$|f| = \sum_{u\in S}\left[\underbrace{\sum_{v\in T} f(u,v)} - \sum_{w\in T} f(w,u)\right]$$

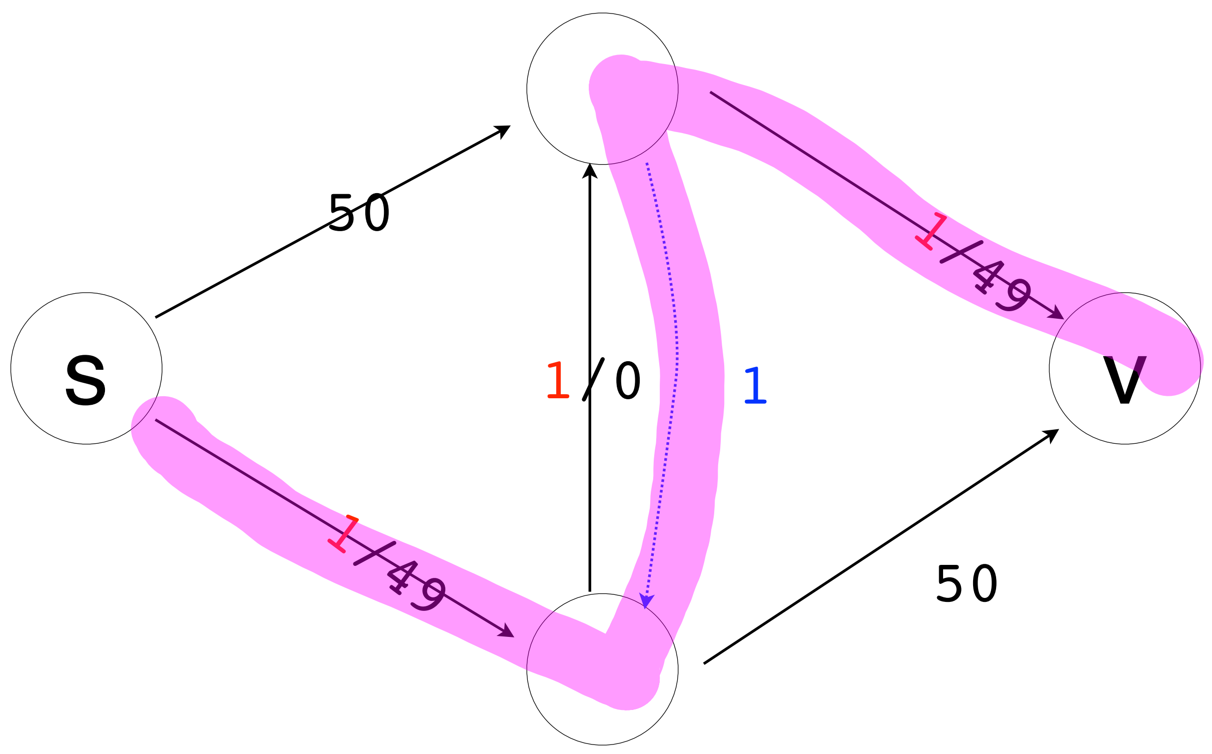
all 0.

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

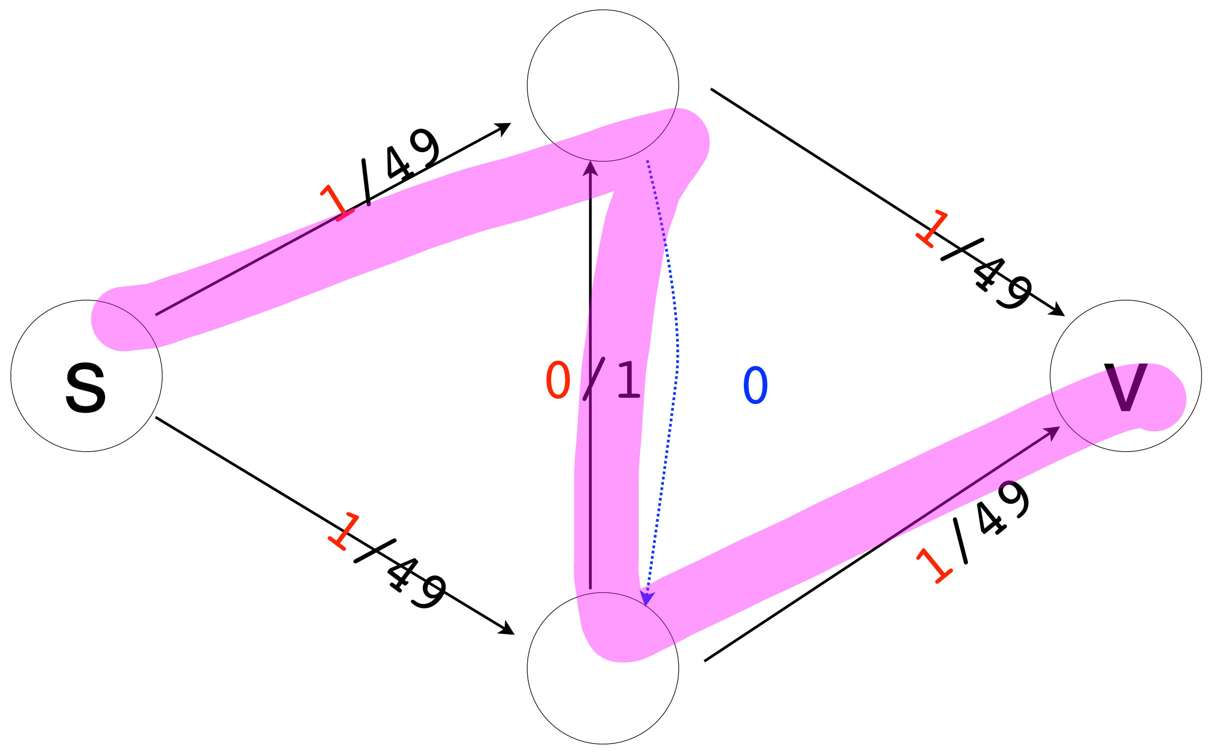
$$= \sum_{u\in S}\sum_{v\in T} c(u,v) = \|S,T\|$$

$\Theta(|E||f|)$



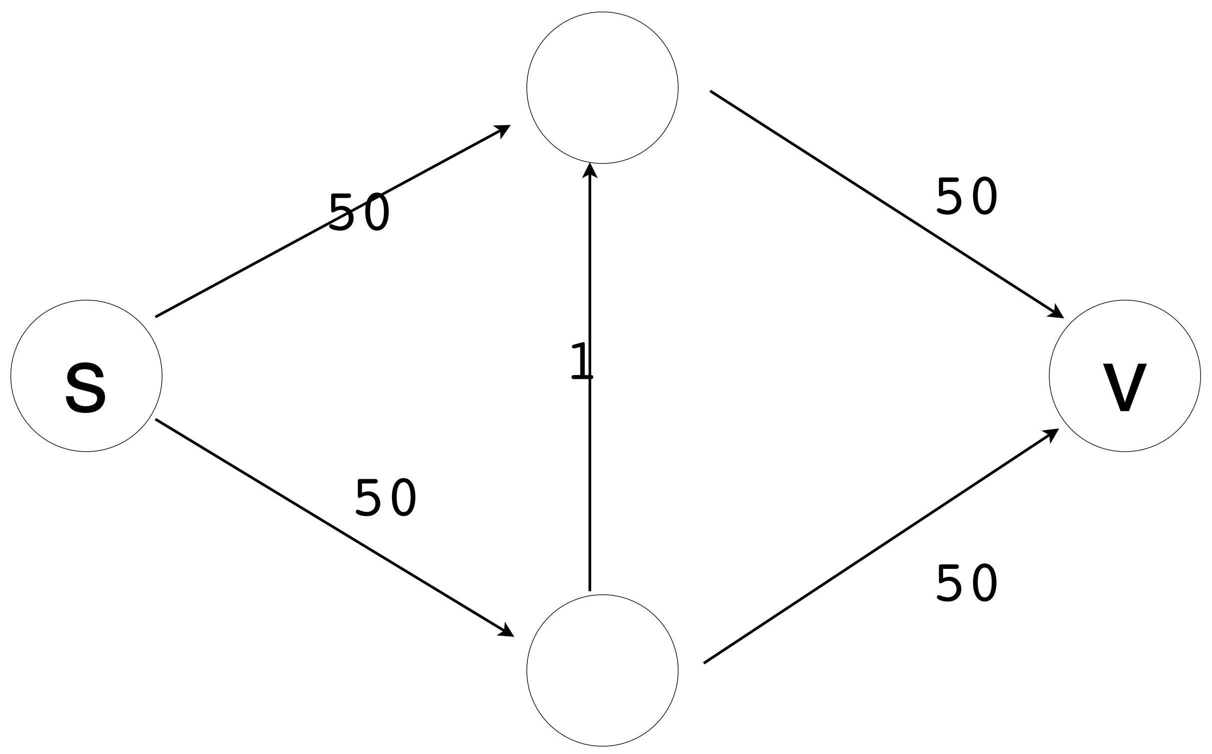


S
V
50
1/49
1/0
1
1/49
50



1/49
1/49
0/1
0
S
V
1/49
1/49

root of the problem



Edmonds-Karp 2

choose path with fewest edges first.

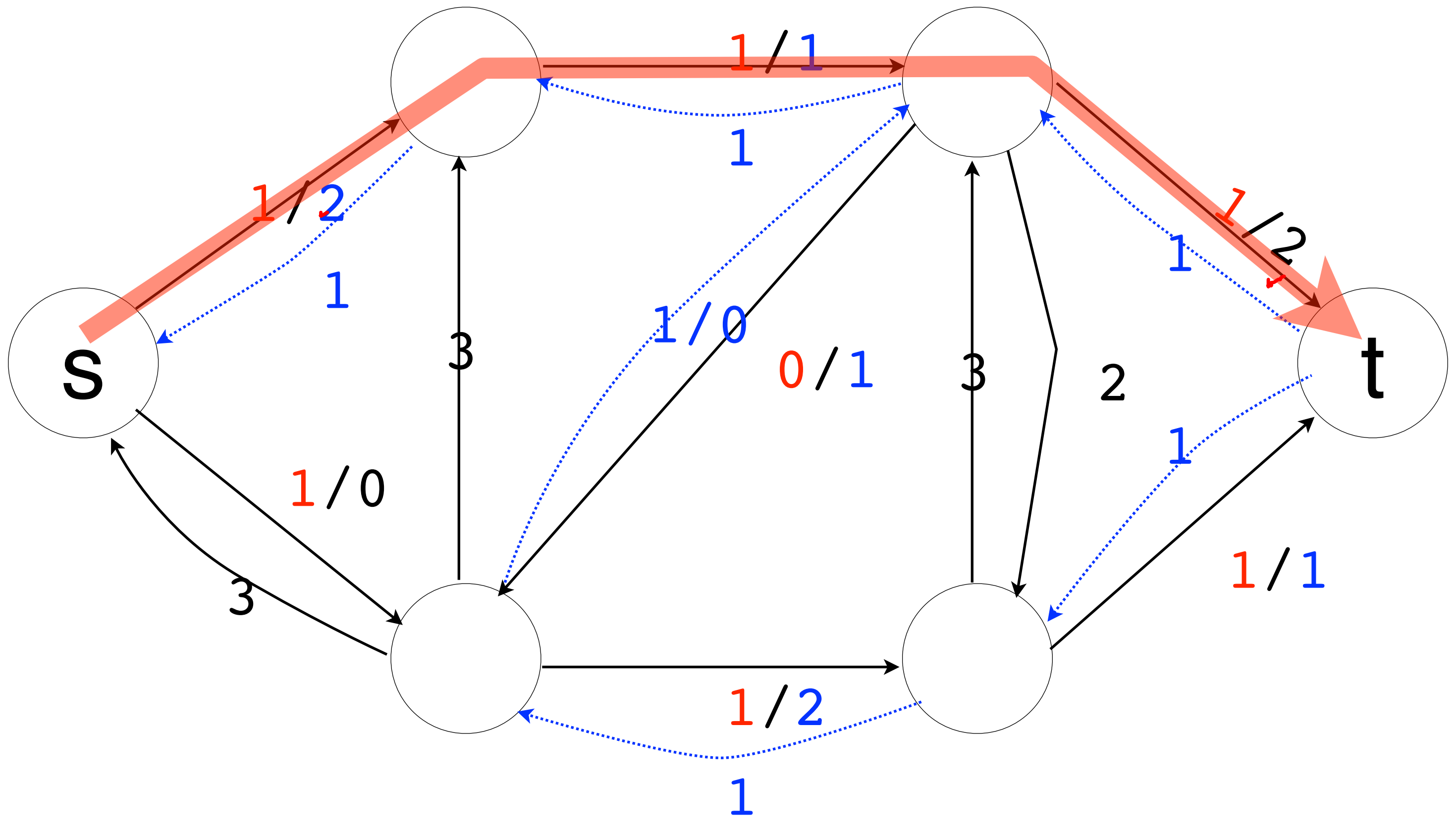
BFS

$\delta_f(s, v)$:

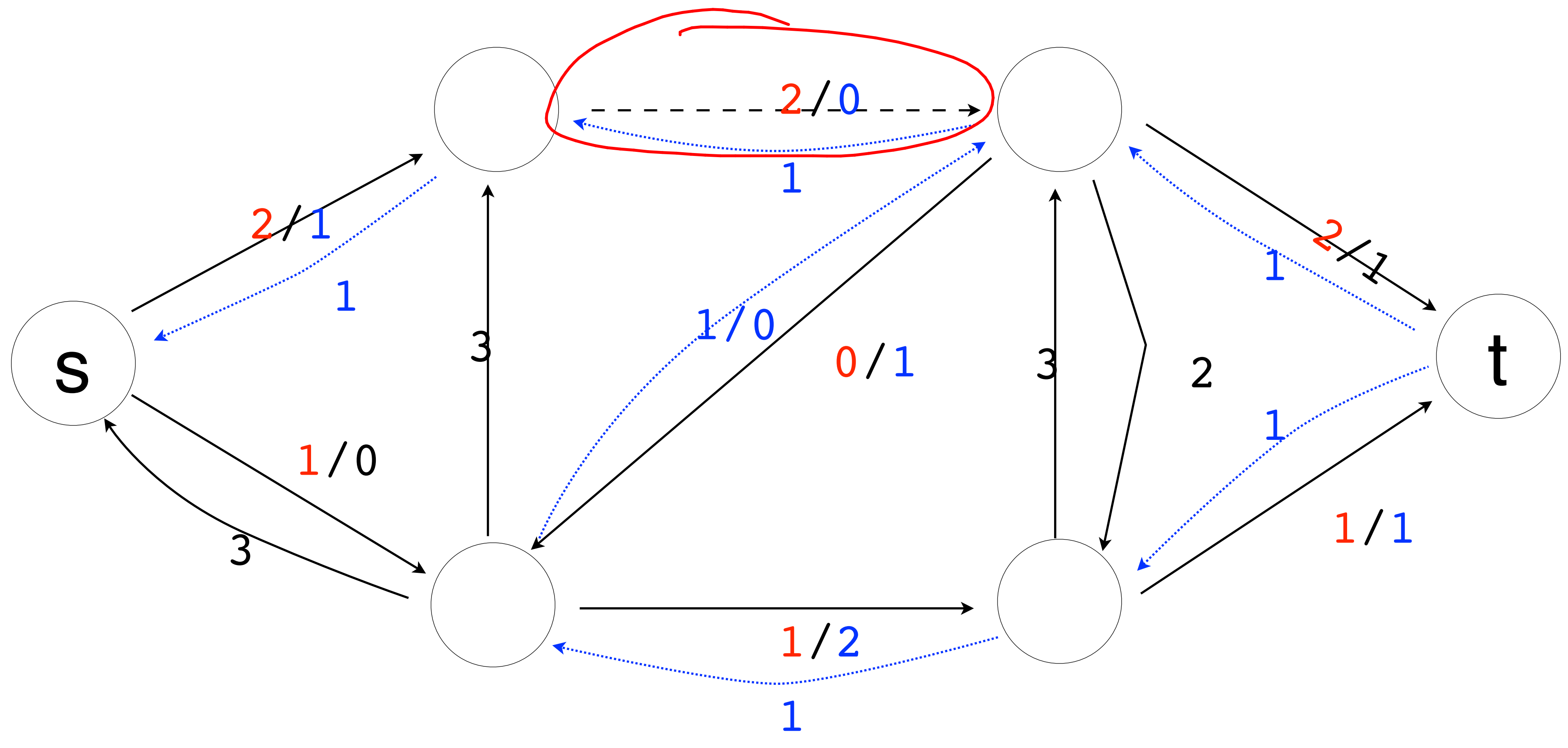
$\Theta(E^2V)$

$\delta_f(s, v)$ increases monotonically thru exec

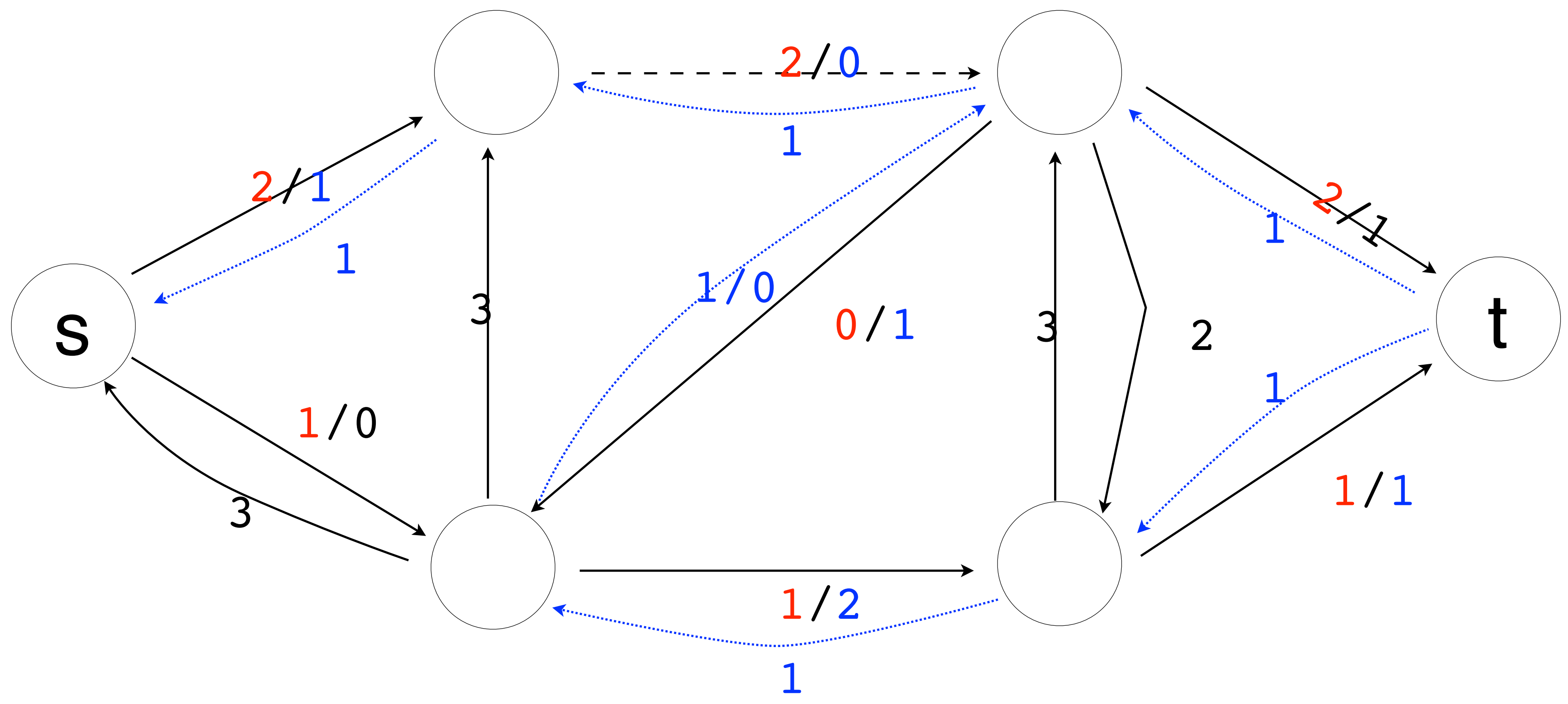
$\delta_{i+1}(v) \geq \delta_i(v)$



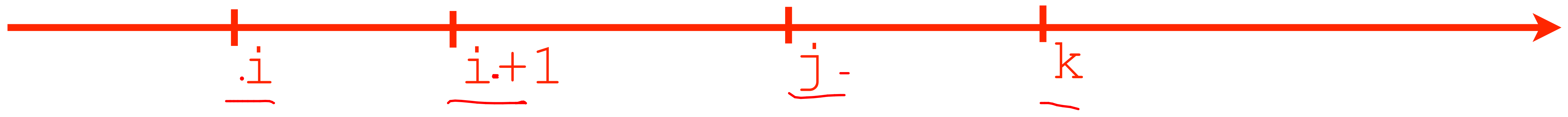
for every augmenting path, some edge is critical.



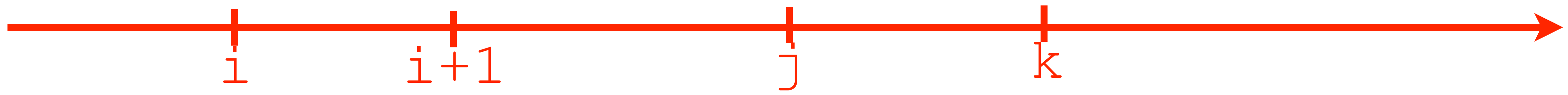
critical edges are removed in next residual graph.



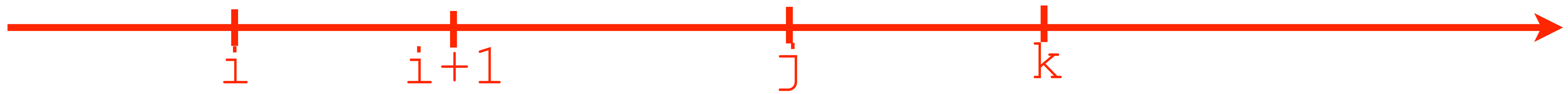
key idea: how many times can an edge be critical?



i
i+1
j
k

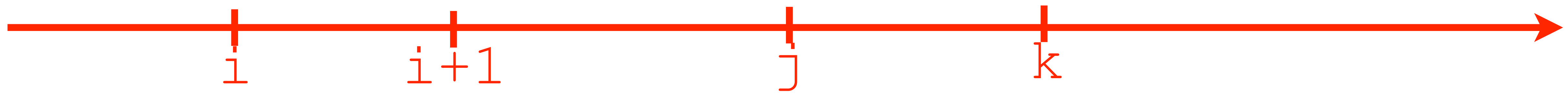


first time (u,v) is critical:

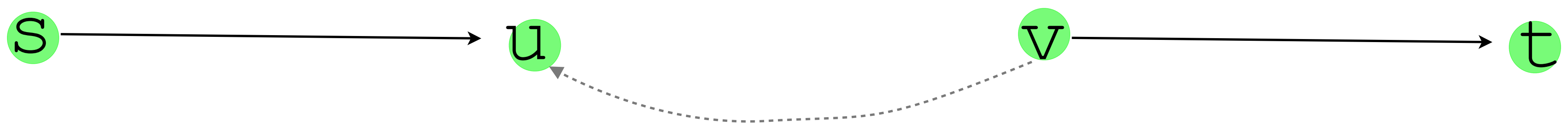


time i+1: (u,v) is critical: $\delta_{i+1}(s,v) \geq \delta_i(s,u) + 1$

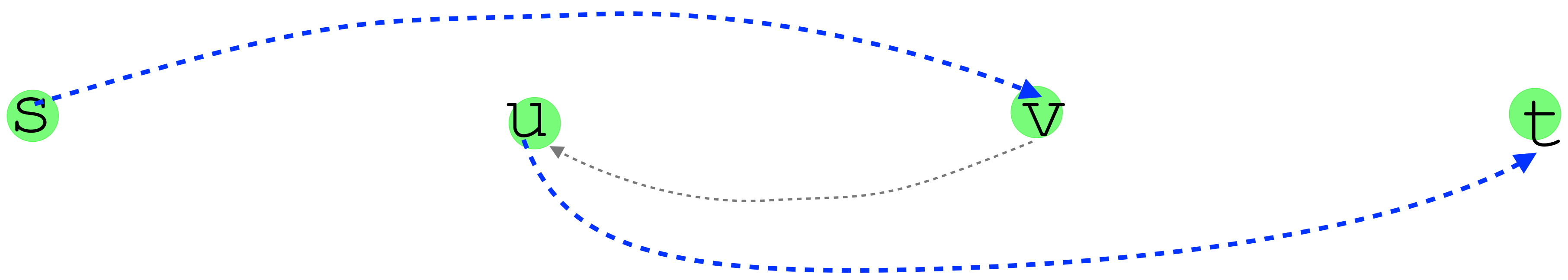
time j: Edge (u,v) STRIKES BACK



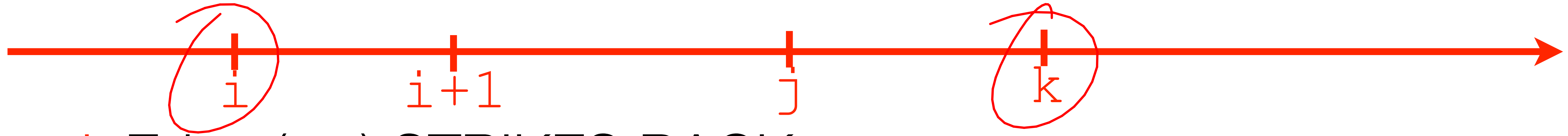
time i+1: (u,v) is critical: $\delta_{i+1}(s,v) \geq \delta_i(s,u) + 1$



time j: Edge (u,v) STRIKES BACK



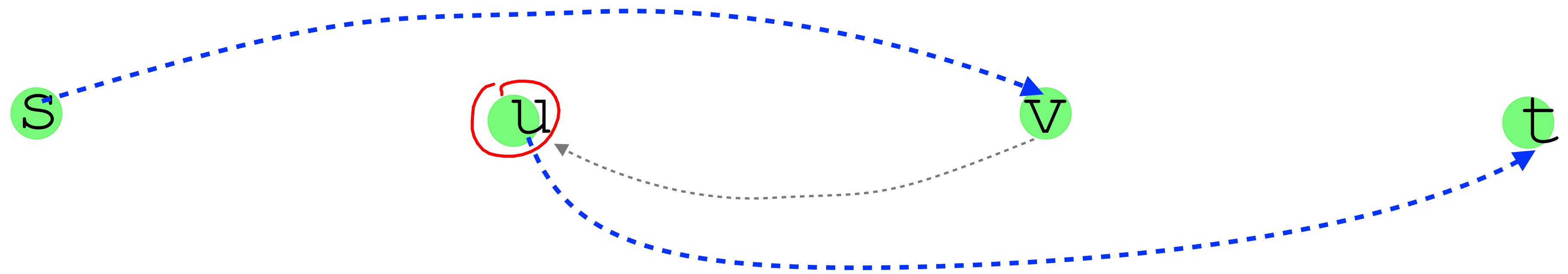
$$\delta_j(s,u) = \delta_j(s,v) + 1$$

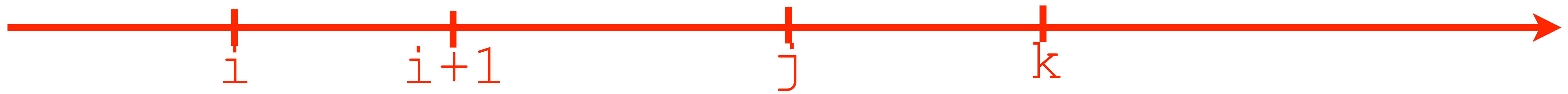


time j: Edge (u,v) STRIKES BACK

$$\delta_{i+1}(s,v) \geq \delta_i(s,u) + 1$$

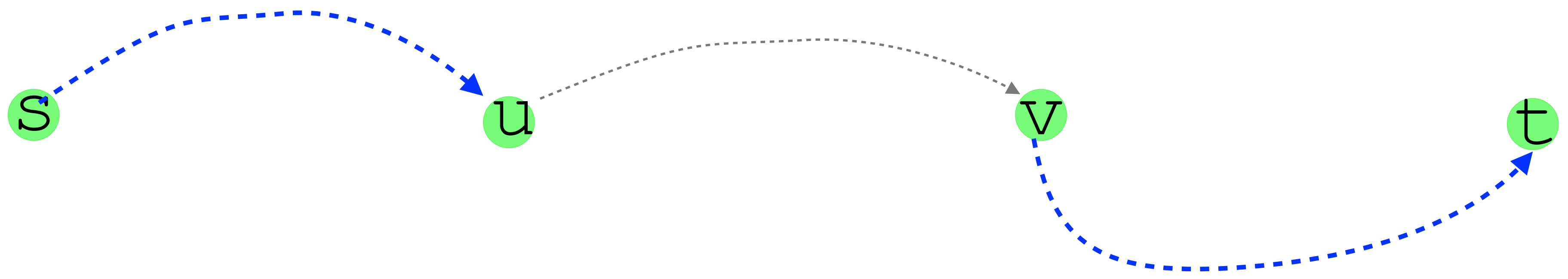
$$\delta_j(s,u) = \delta_j(s,v) + 1$$





time k: RETURN OF THE (u,v) critical

$$\delta_k(s,u) \geq \delta_i(s,u) + 2$$



QUESTION: How many times can (u,v) be critical?

edge critical only $\frac{V}{2}$ times.

there are only E edges.

ergo, total # of augmenting paths: $\Theta(EV)$

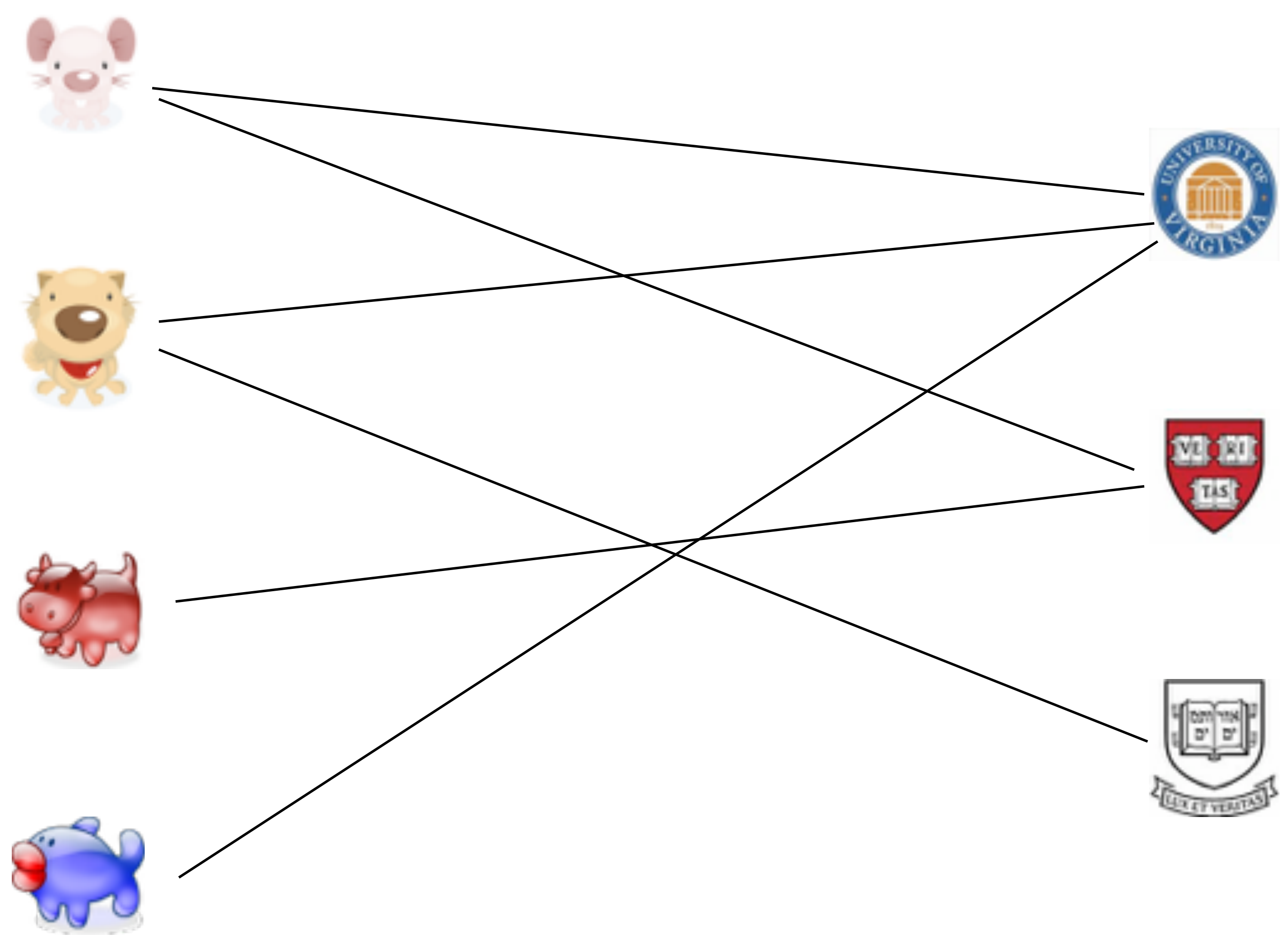
time to find an augmenting path: $\Theta(E+V)$

total running time of E-K algorithm: $\Theta(E^2V)$

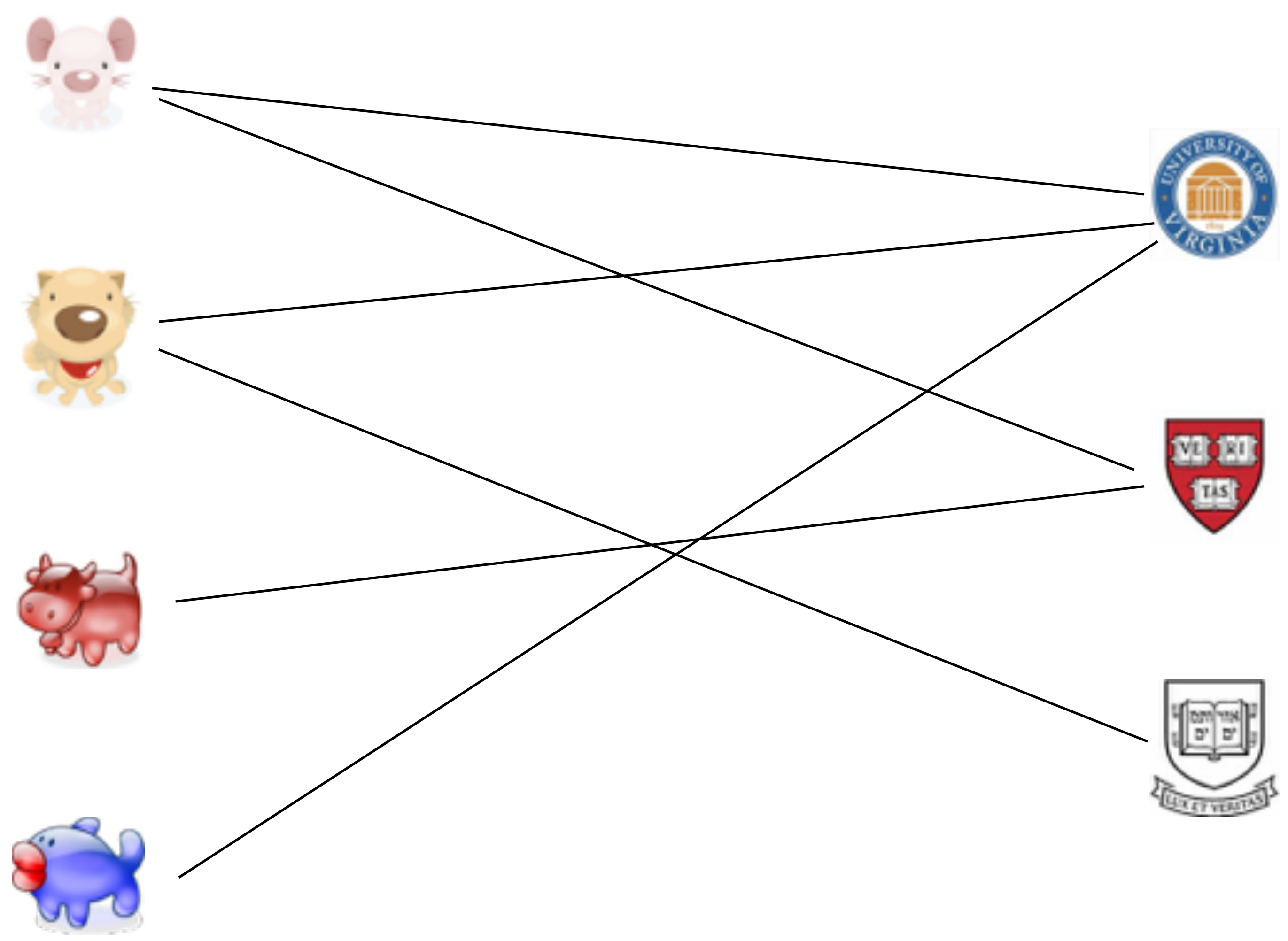
FF	$O(E|f^*|)$
EK2	
PUSH-RELABEL	
FASTER PUSH-RELABEL	

Bipartite

maximum bipartite matching



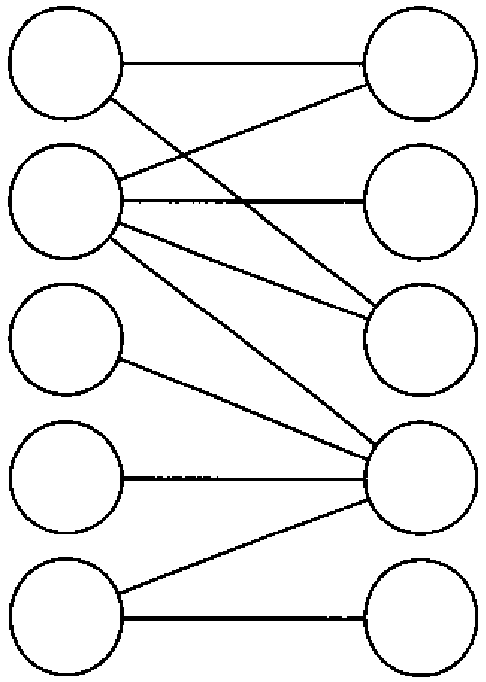
maximum bipartite matching



bipartite matching

problem:

algorithm

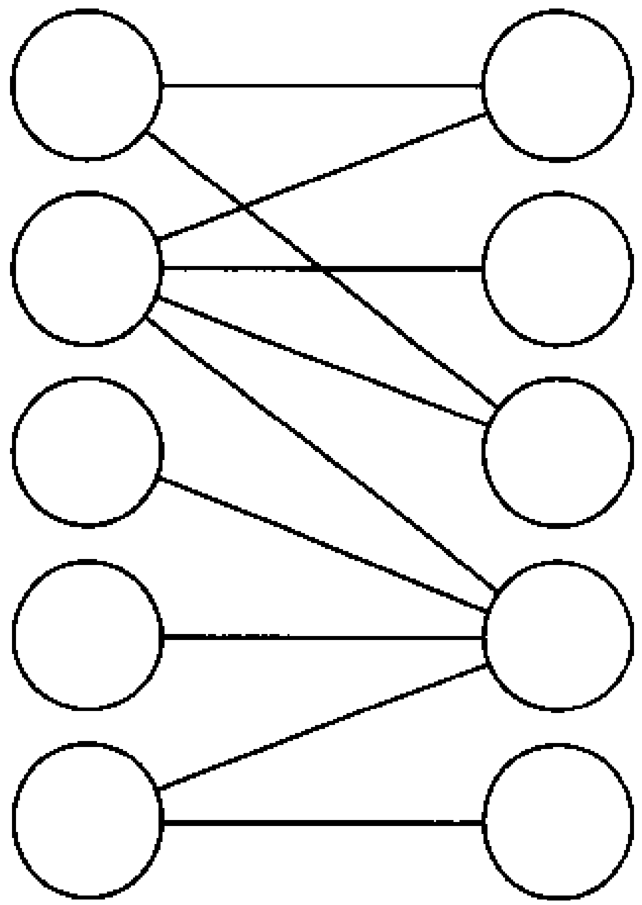


algorithm

1. MAKE NEW G'
FROM INPUT G.

2. RUN FF ON G'

3. OUTPUT ALL MIDDLE EDGES
WITH FLOW F(E)=1.



correctness

IF G HAS A MATCHING OF SIZE K, THEN

correctness

IF G' HAS A FLOW OF K, THEN

integrality theorem

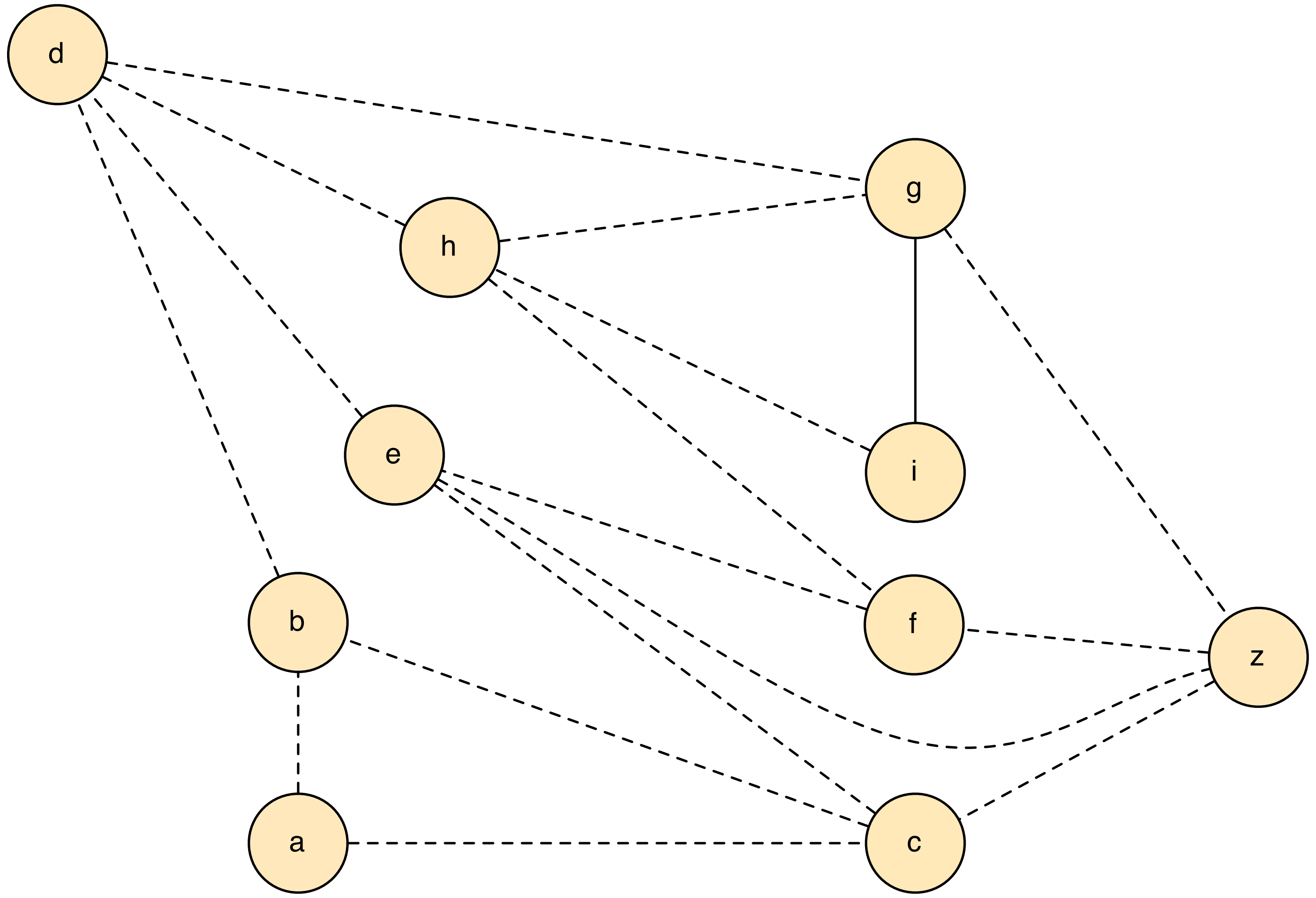
IF CAPACITIES ARE ALL INTEGRAL, THEN

correctness

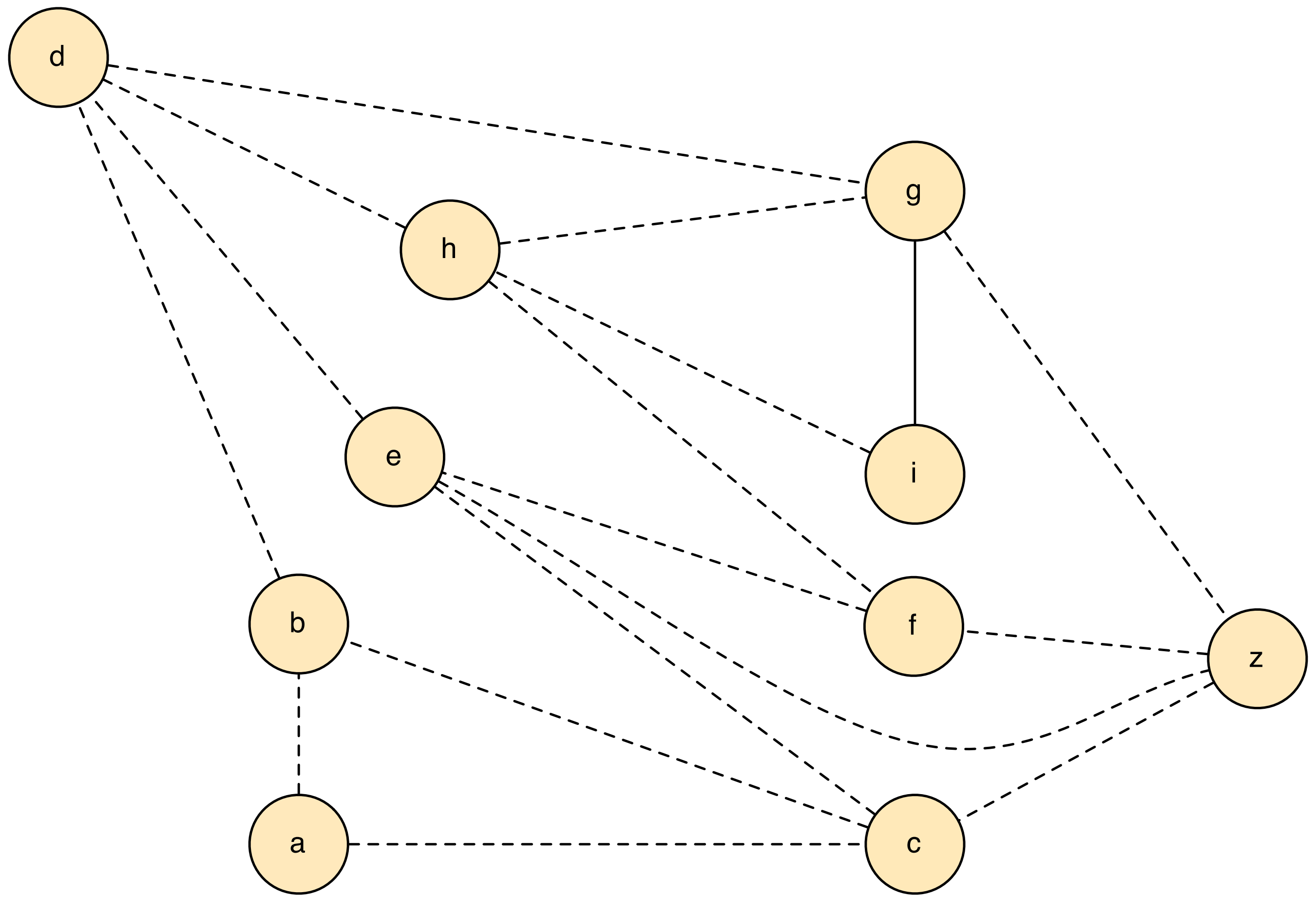
HAS A FLOW OF K, THEN G HAS K-MATCHING.

running time

edge-disjoint paths



algorithm



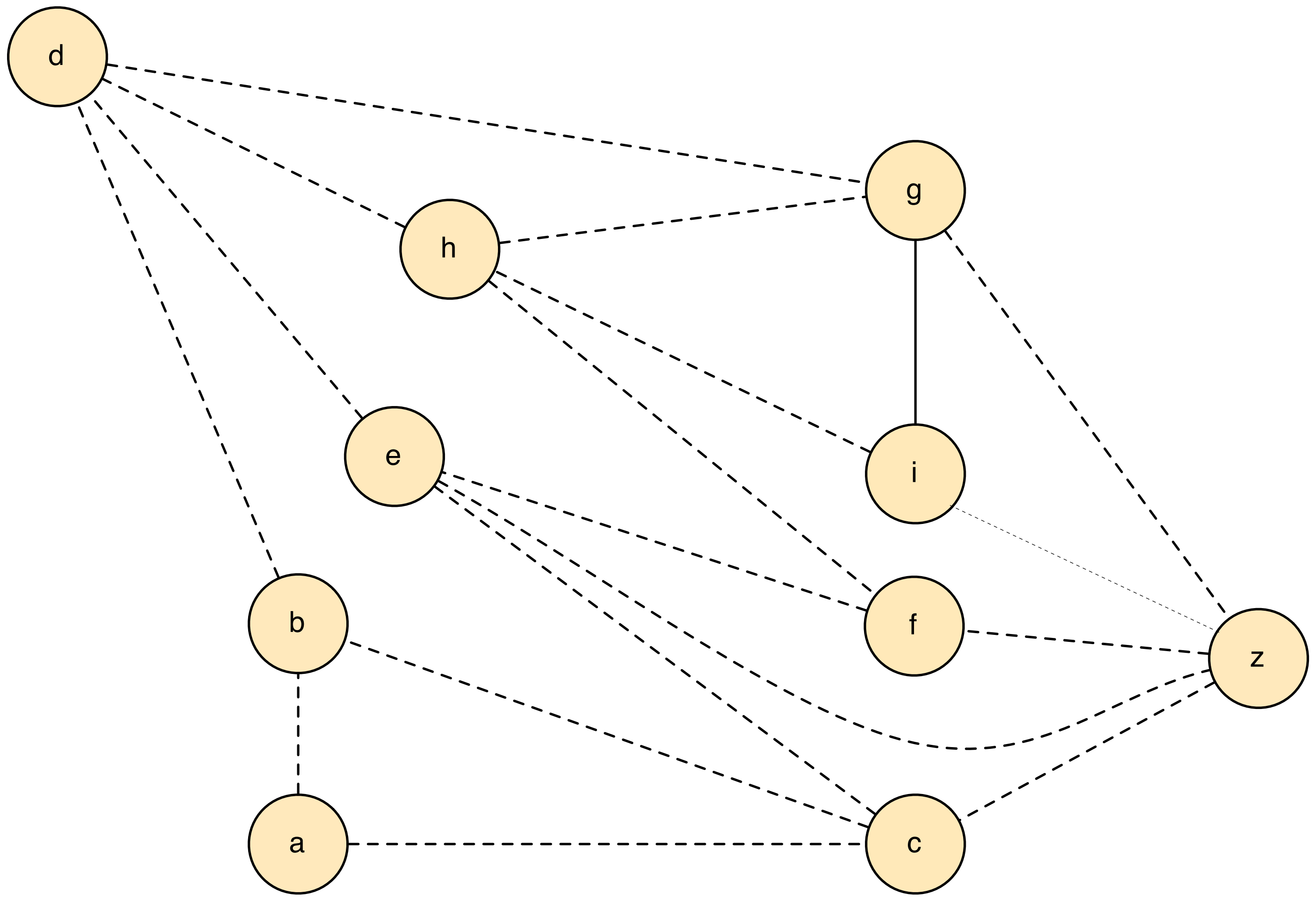
analysis

IF G HAS K DISJOINT PATHS, THEN

analysis

G’ HAS A FLOW OF K, THEN

vertex-disjoint paths

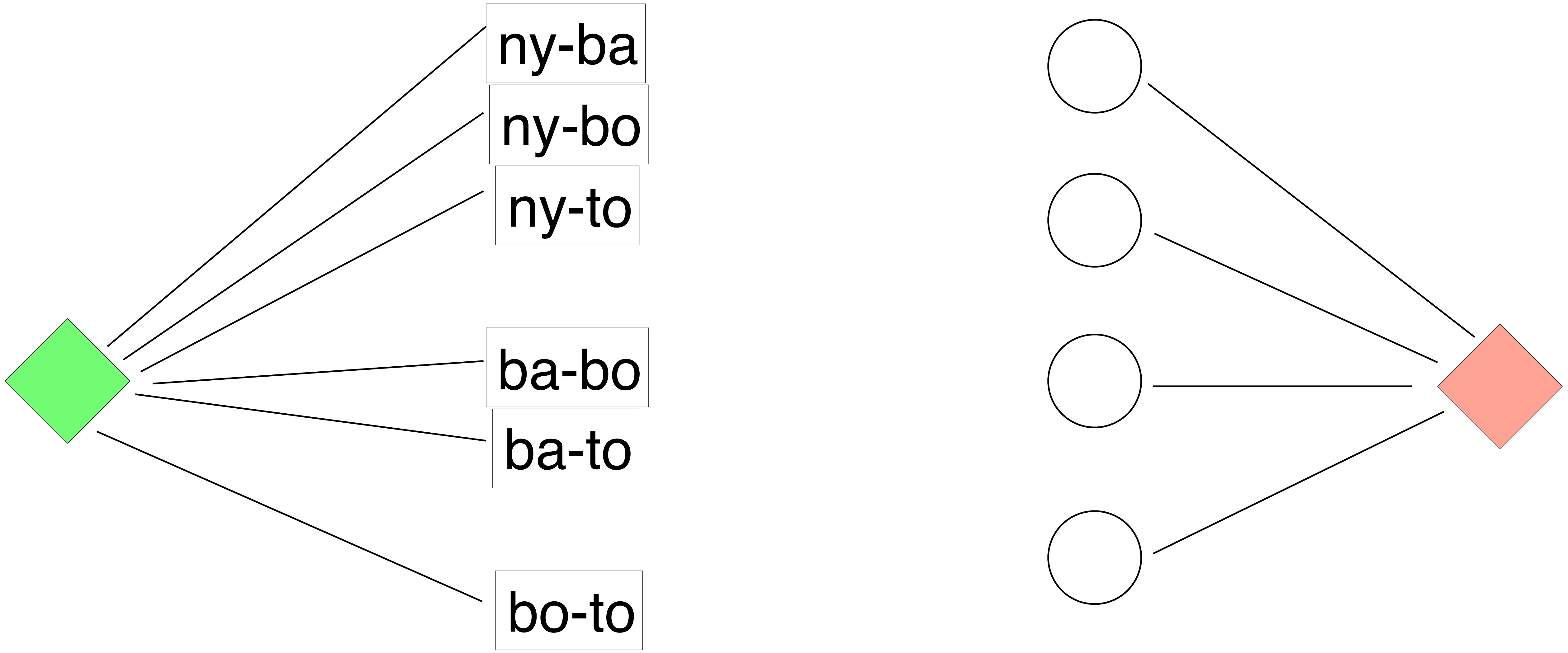


baseball elimination

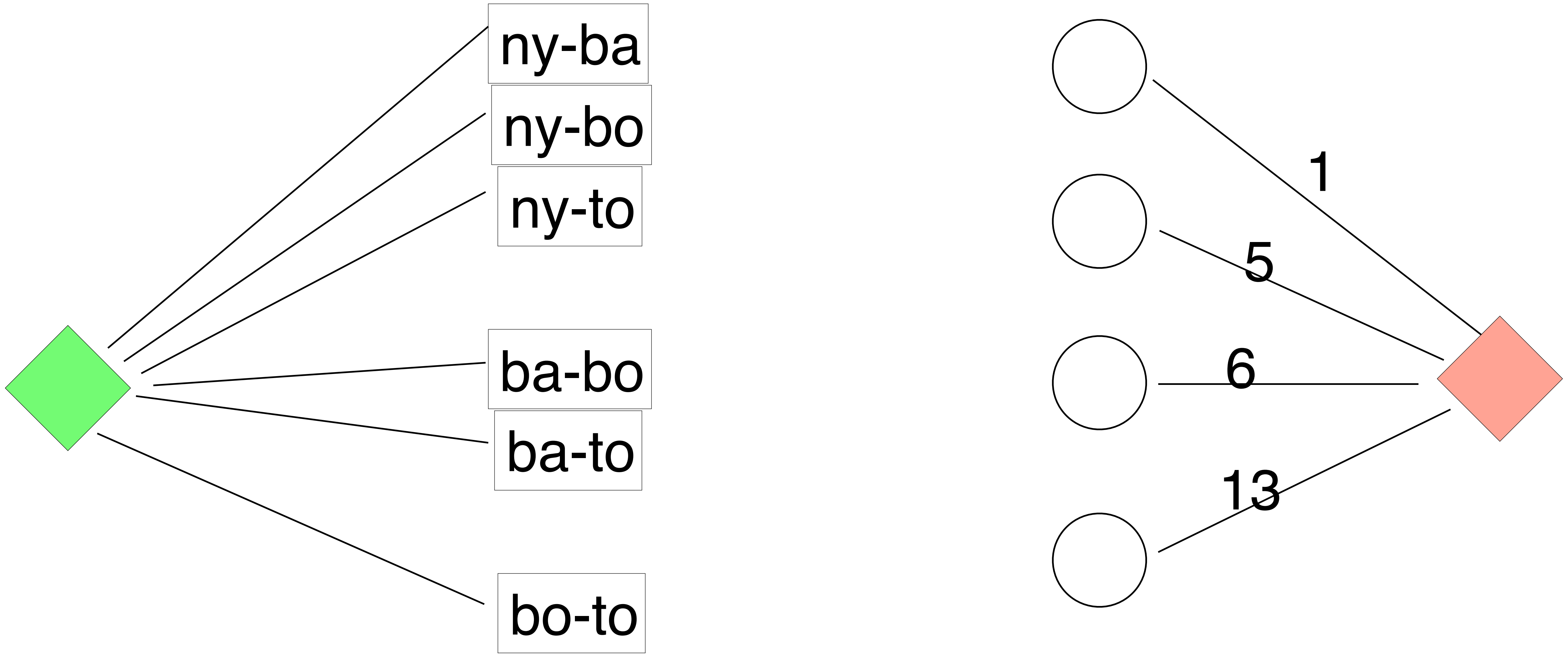
	W	L	Left	Against A	P	N	M
ATL	83	71	8	-	1	6	1
PHL	80	79	3	1	-	0	2
NY	78	78	6	6	0	-	0
MONT	77	82	3	1	2	0	-

baseball elimination

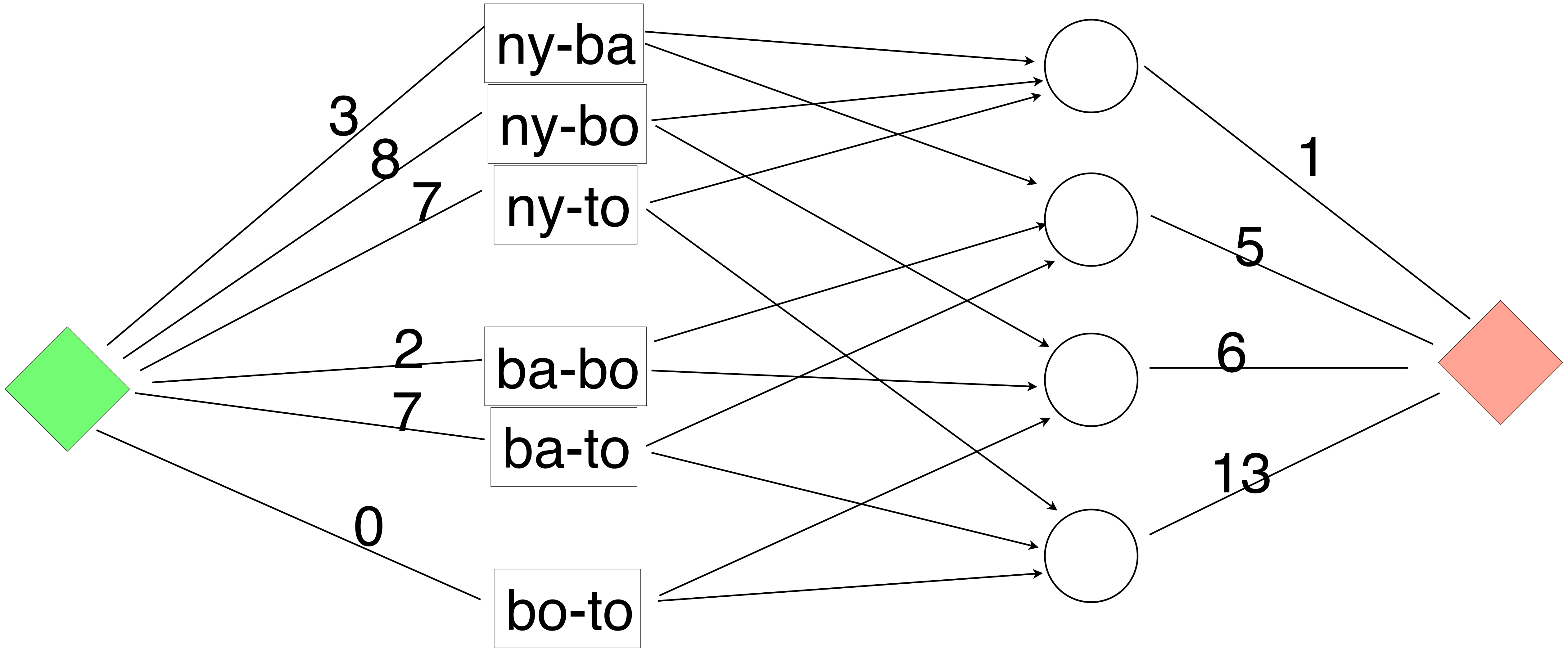
	W	L	Left	N	B	Bo	T	D
				Against				
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			



	W	L	Left	N	B	Bo	T	D
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			



	W	L	Left	N	B	Bo	T	D
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			



	W	L	Left	N	B	Bo	T	D
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			