5800
Max Flows 2

mar 27/28/29 2022

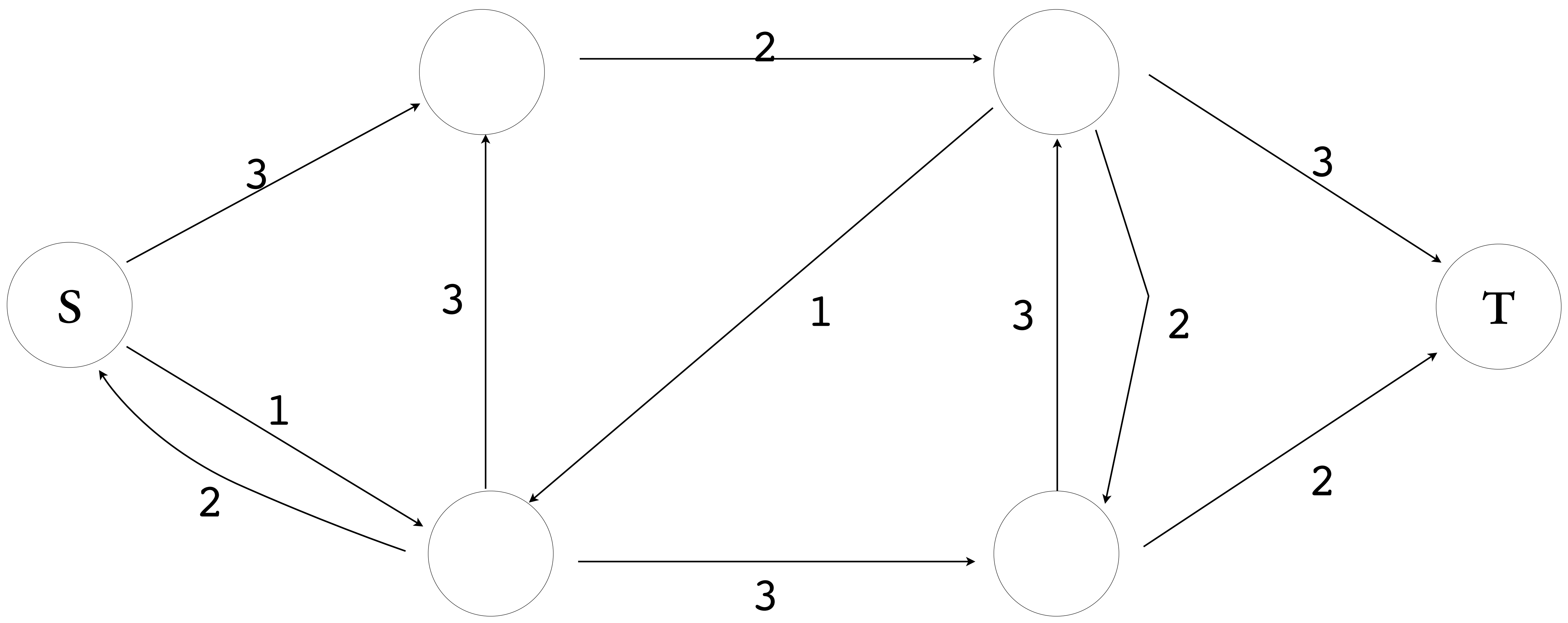
shelat

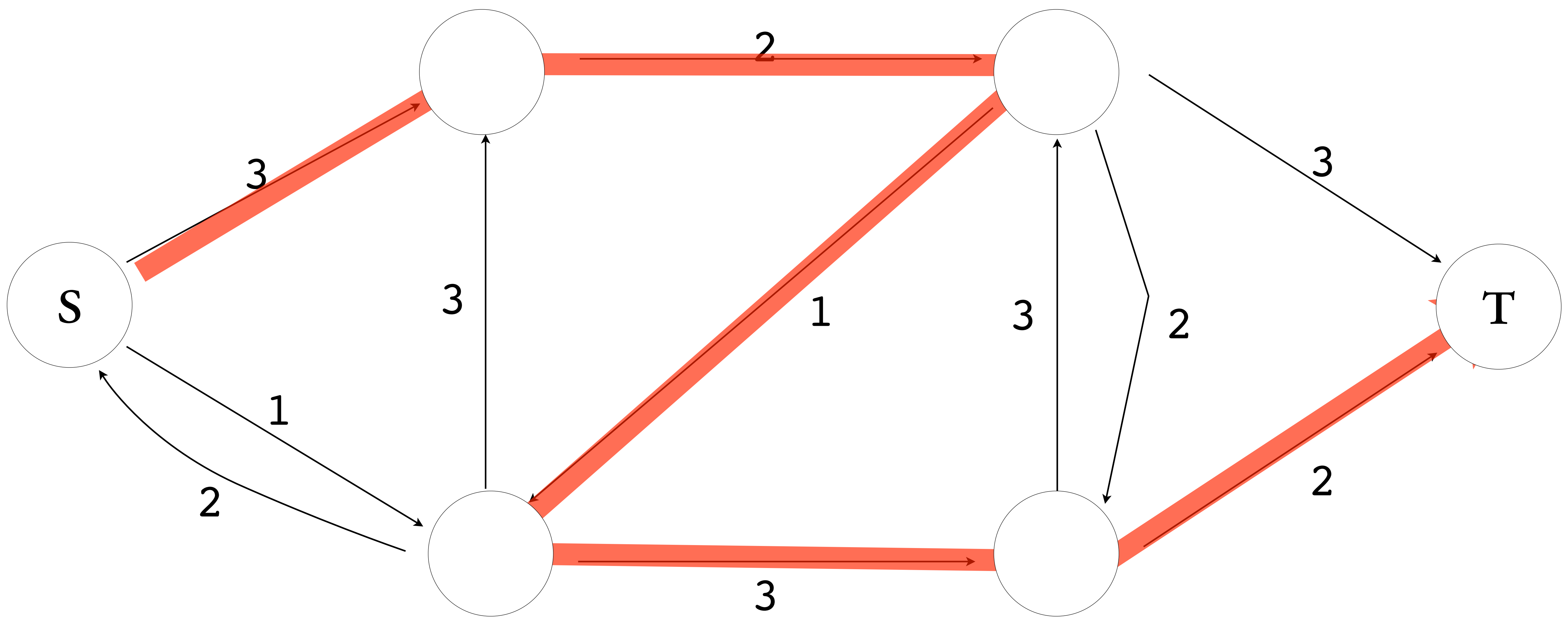
Ford-Fulkerson

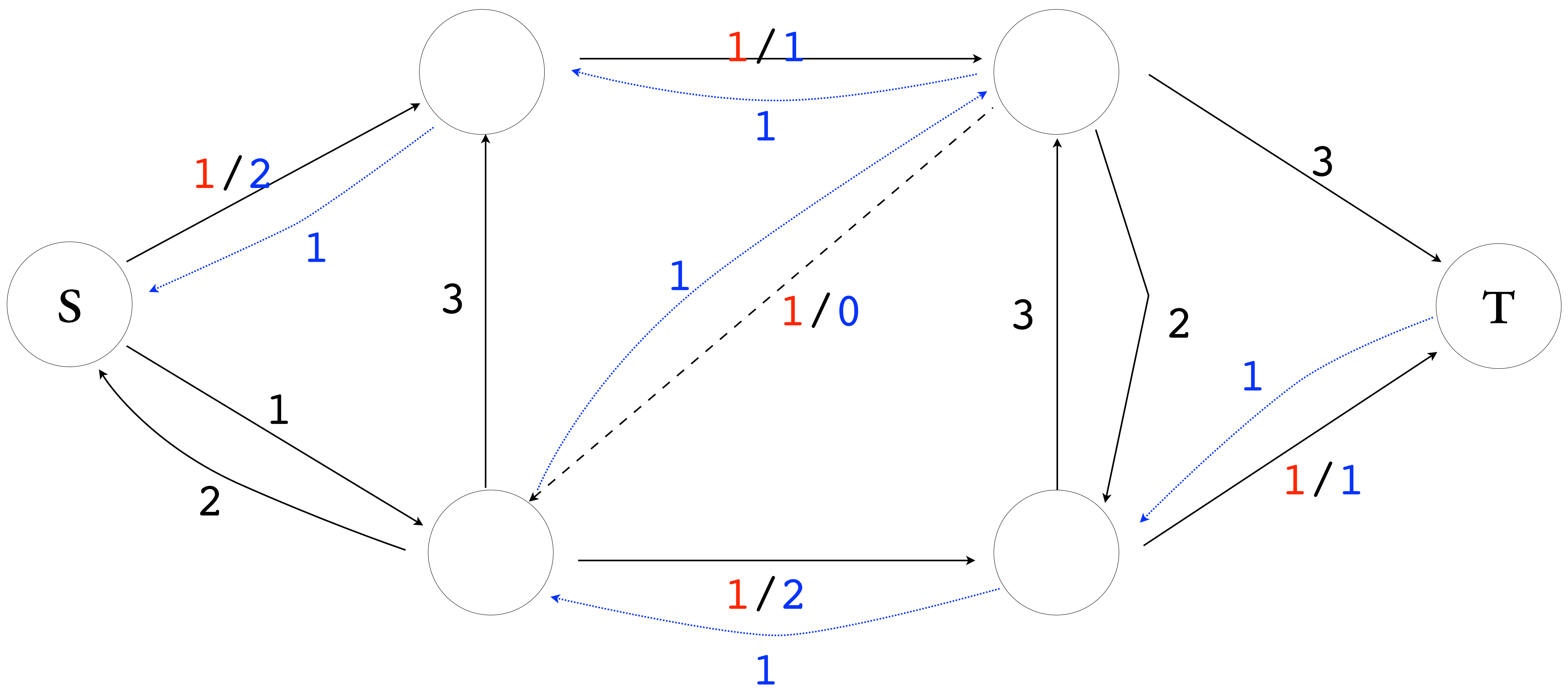
INITIALIZE $f(u,v) \leftarrow 0 \; \forall u, v$

WHILE EXISTS AN AUGMENTING PATH p IN G_f

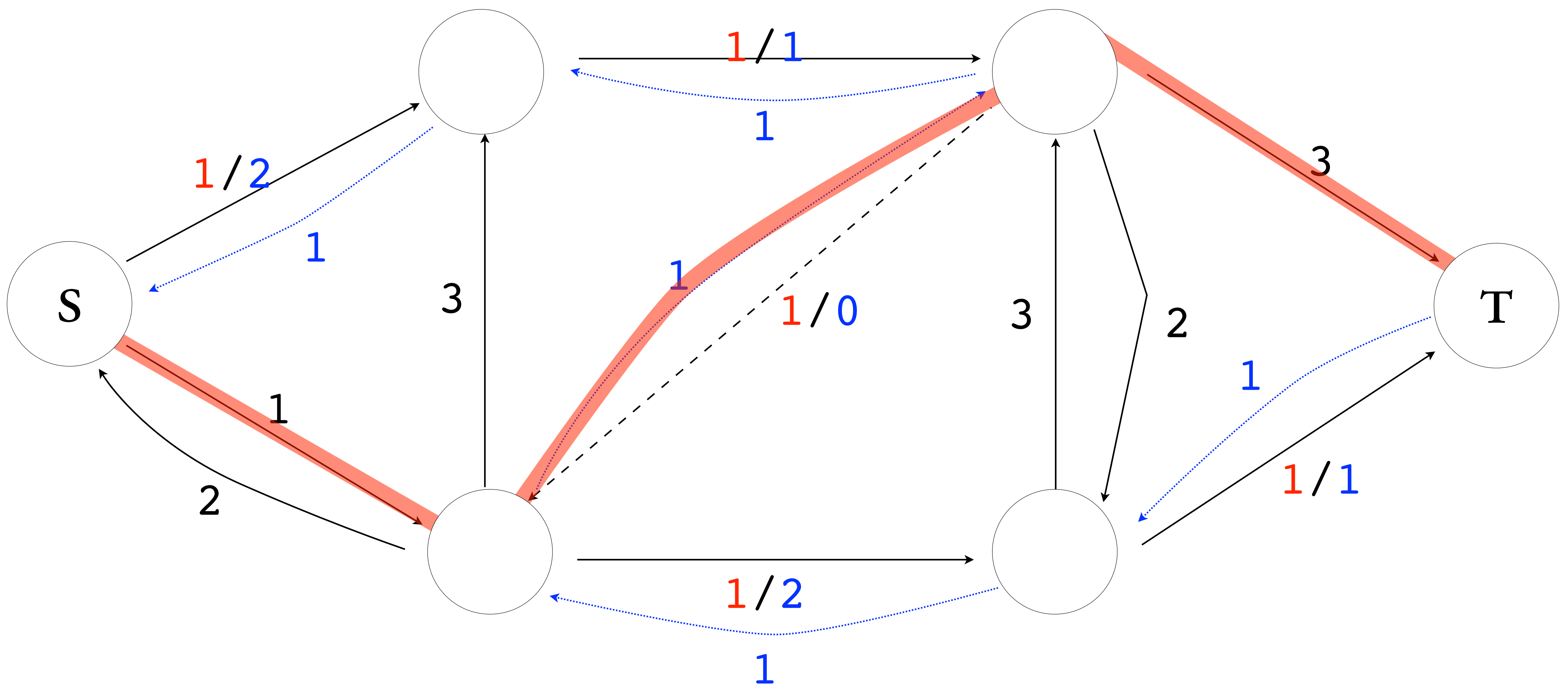
AUGMENT f WITH $c_f(p) = \min_{(u,v) \in p} c_f(u,v)$


S
T
3
1
2
3
2
1
3
3
2
3
2

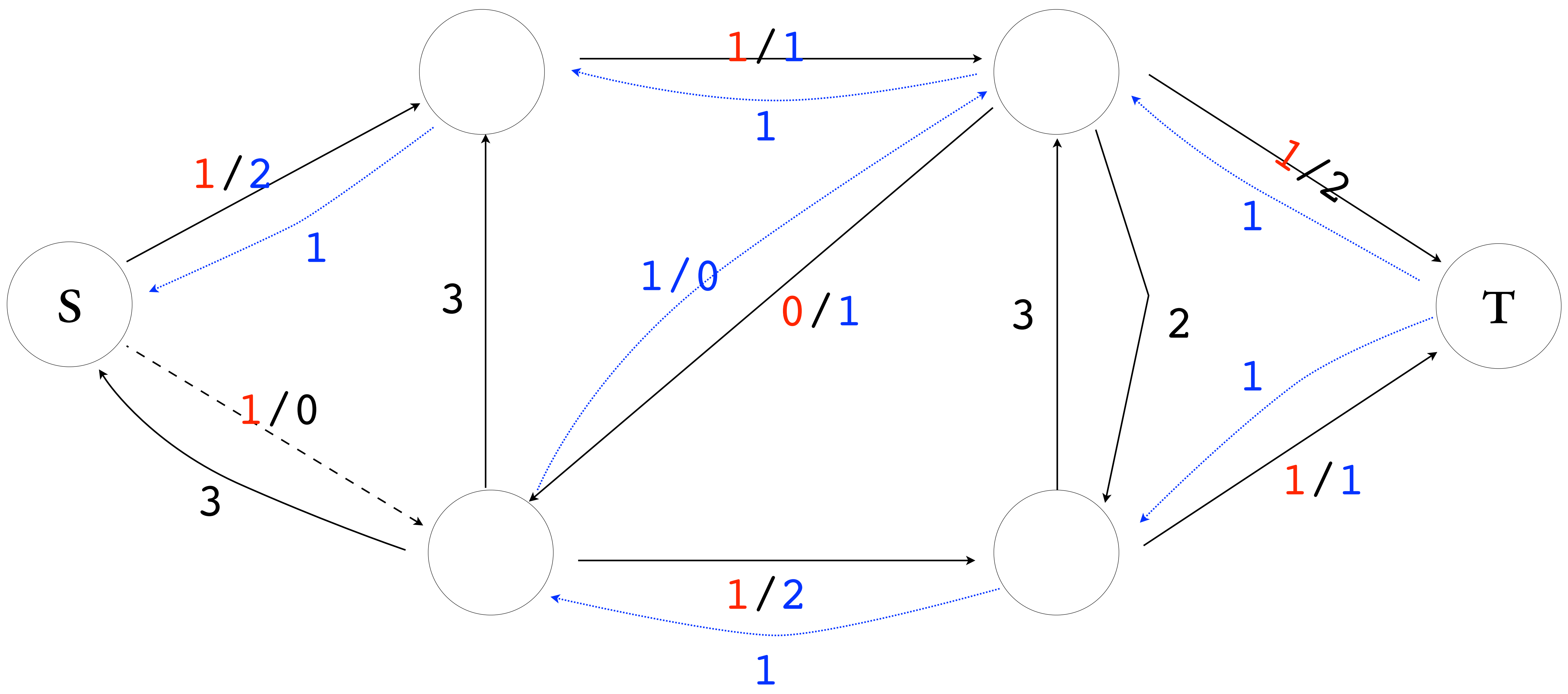

S
T
2
3
3
3
1
3
2
1
2
2
3
2



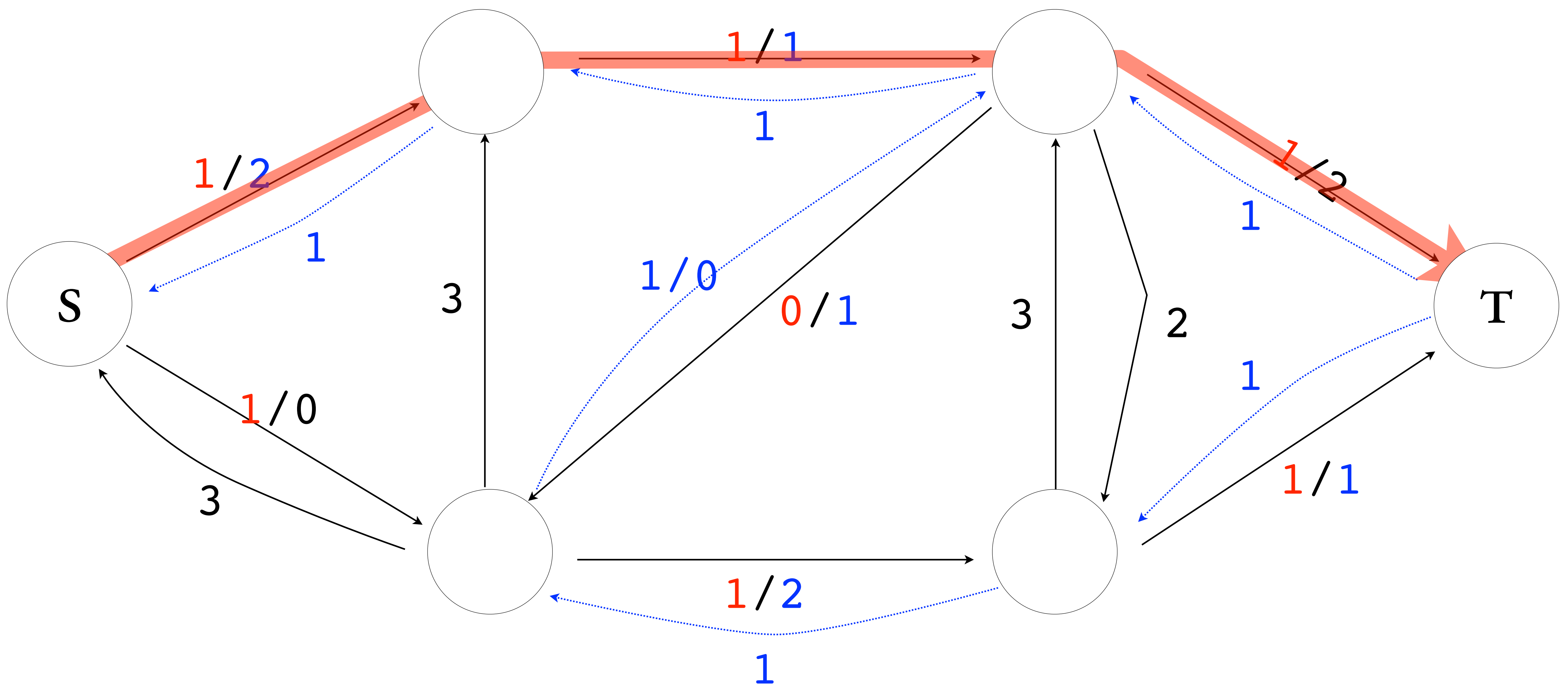
S
T
1 / 2
1
1 / 1
1
3
3
1
1 / 0
3
2
1
2
1
1 / 2
1
1 / 1

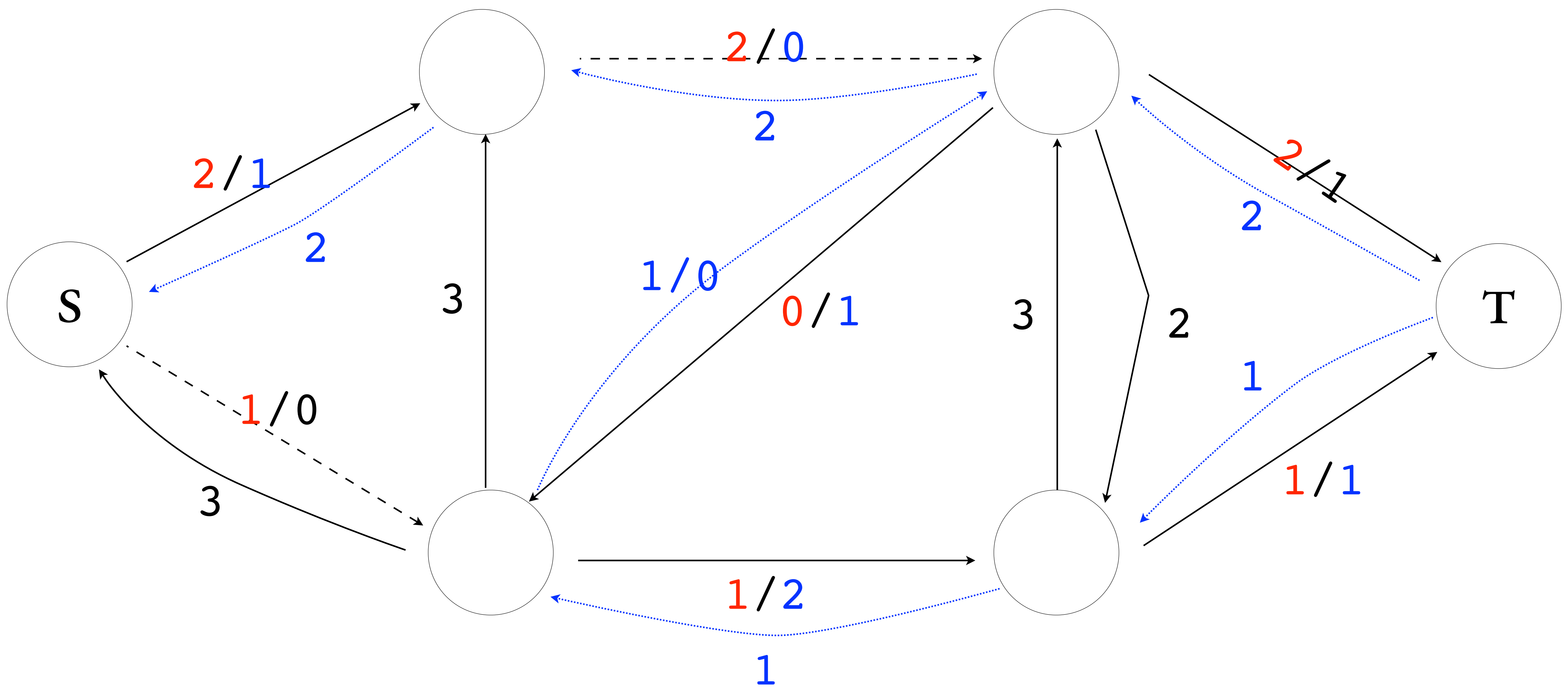


S
T
1 / 2
1
3
1
2
1 / 1
1
1
1 / 0
1 / 2
1
3
3
2
1
1 / 1



S
T
1/2
1
3
1/1
1
1/0
0/1
3
2
1/2
1
1
1/0
3
1/2
1
1/1


S
T
1/2
1
1/1
1
1/2
1
3
1/0
0/1
3
2
1
1
1/1
1/0
3
1/2
1


S
T
2 / 1
2
2 / 0
2
1 / 0
0 / 1
3
3
2
2 / 1
2
1 / 0
3
1
1 / 1
1 / 2
1
3

FORD-FULKERSON

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

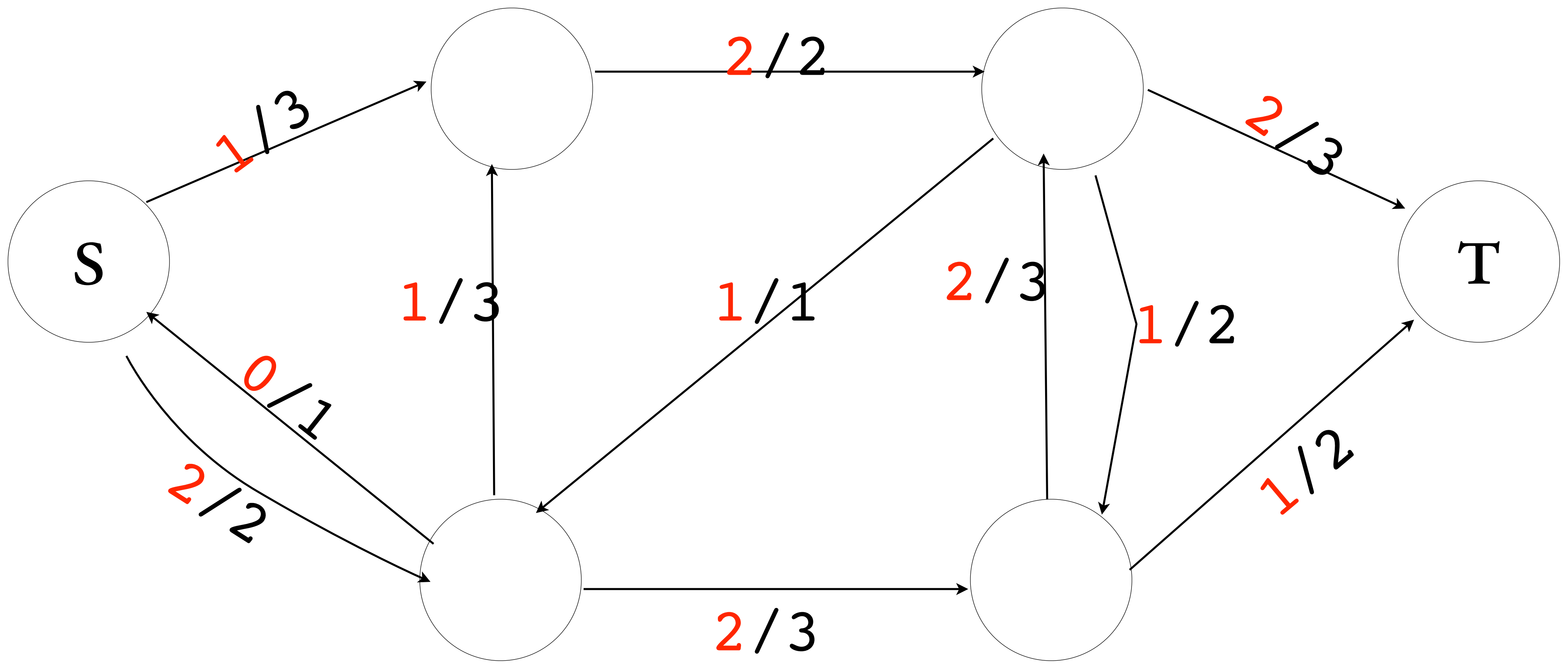
WHILE EXISTS AN AUGMENTING PATH p IN G_f

AUGMENT f WITH $c_f(p) = \min_{(u,v) \in p} c_f(u,v)$

TIME TO FIND AN AUGMENTING PATH:

NUMBER OF ITERATIONS OF WHILE LOOP:

FOR ANY $f, (S, T)$ IT HOLDS THAT $|f| \leq ||S, T||$



Thm: max flow = min cut

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

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

Thm: max flow = min cut (cont)

FORD-FULKERSON

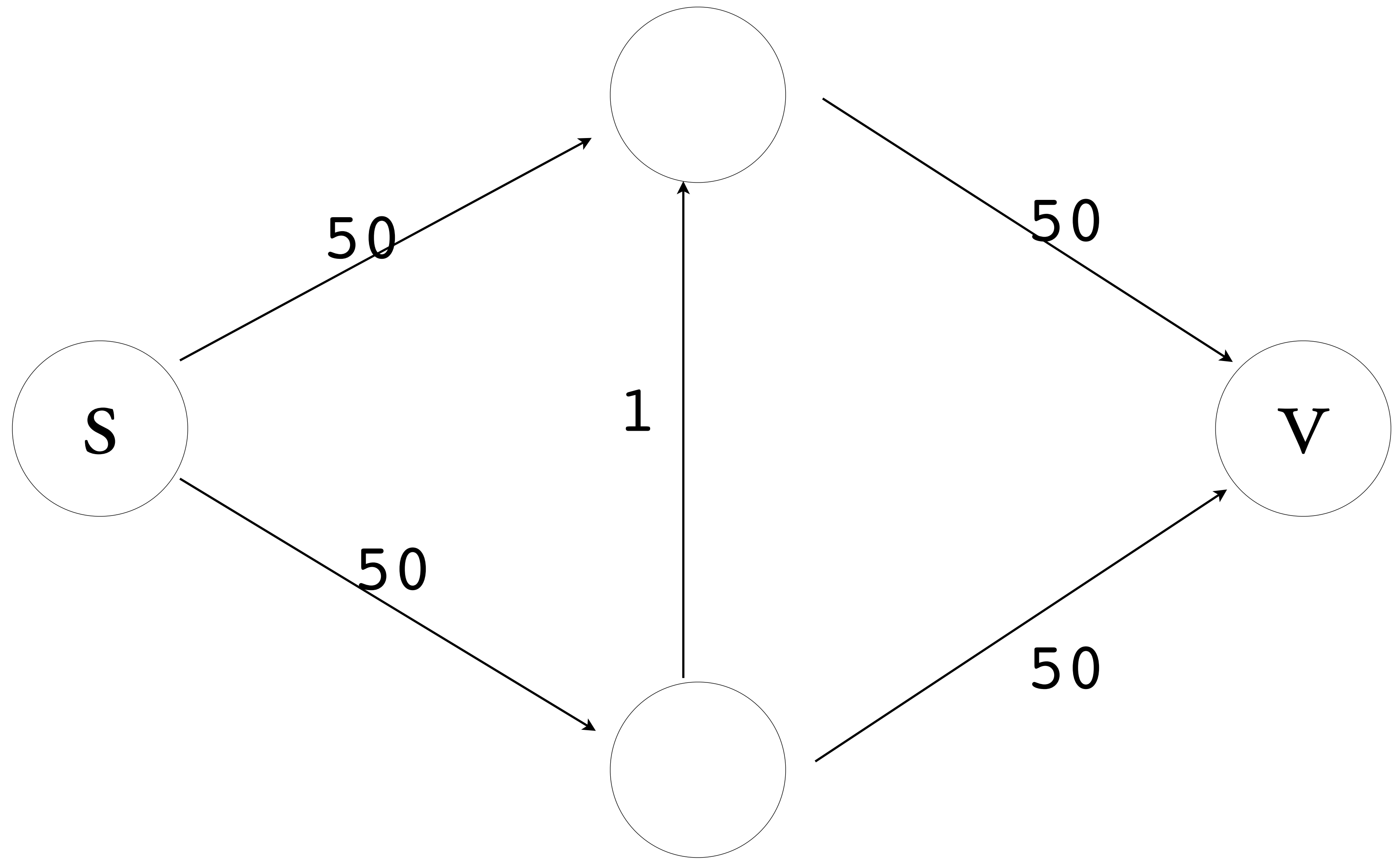
INITIALIZE $f(u,v) \leftarrow 0 \ \forall u,v$

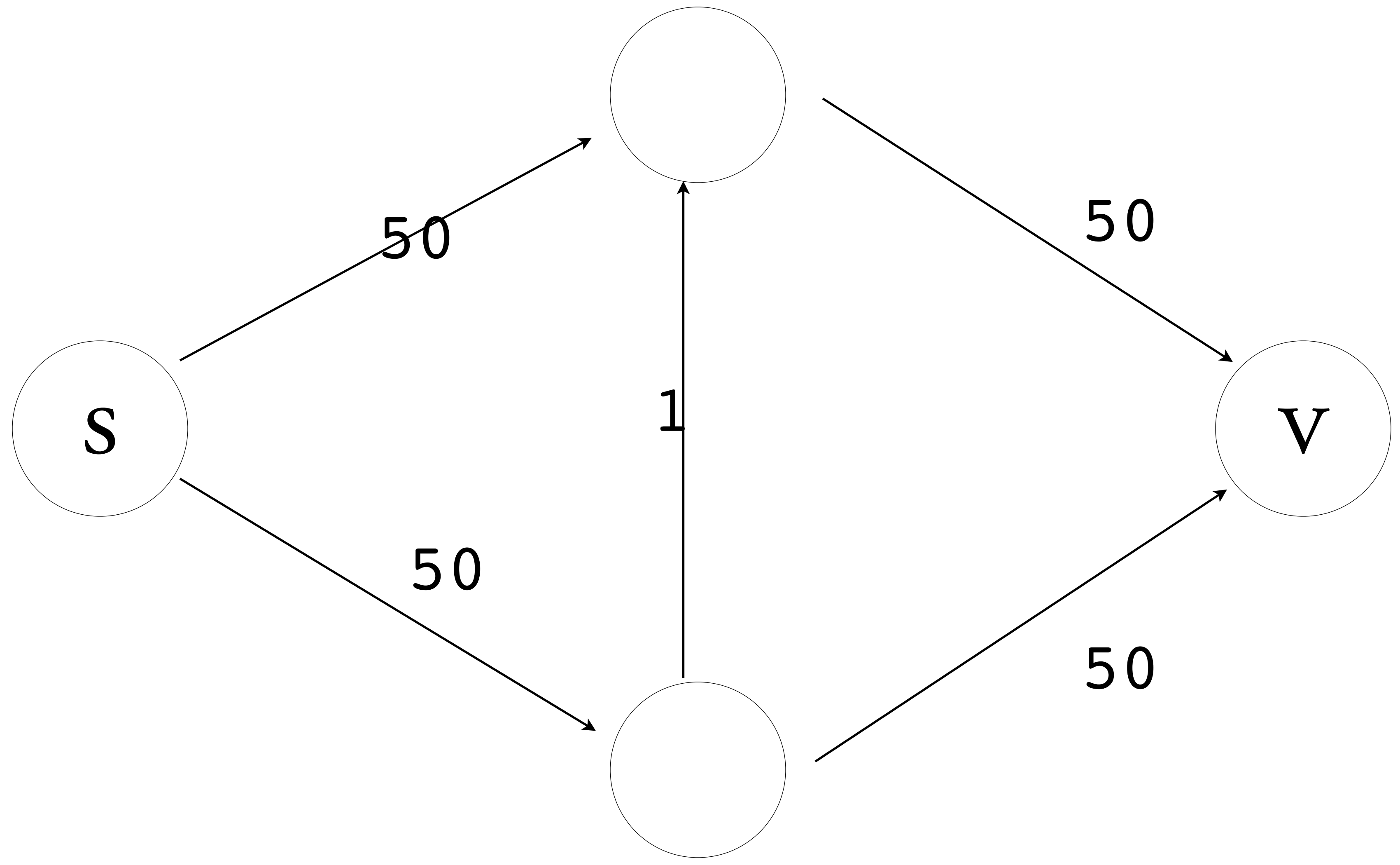
WHILE EXISTS AN AUGMENTING PATH p IN G_f

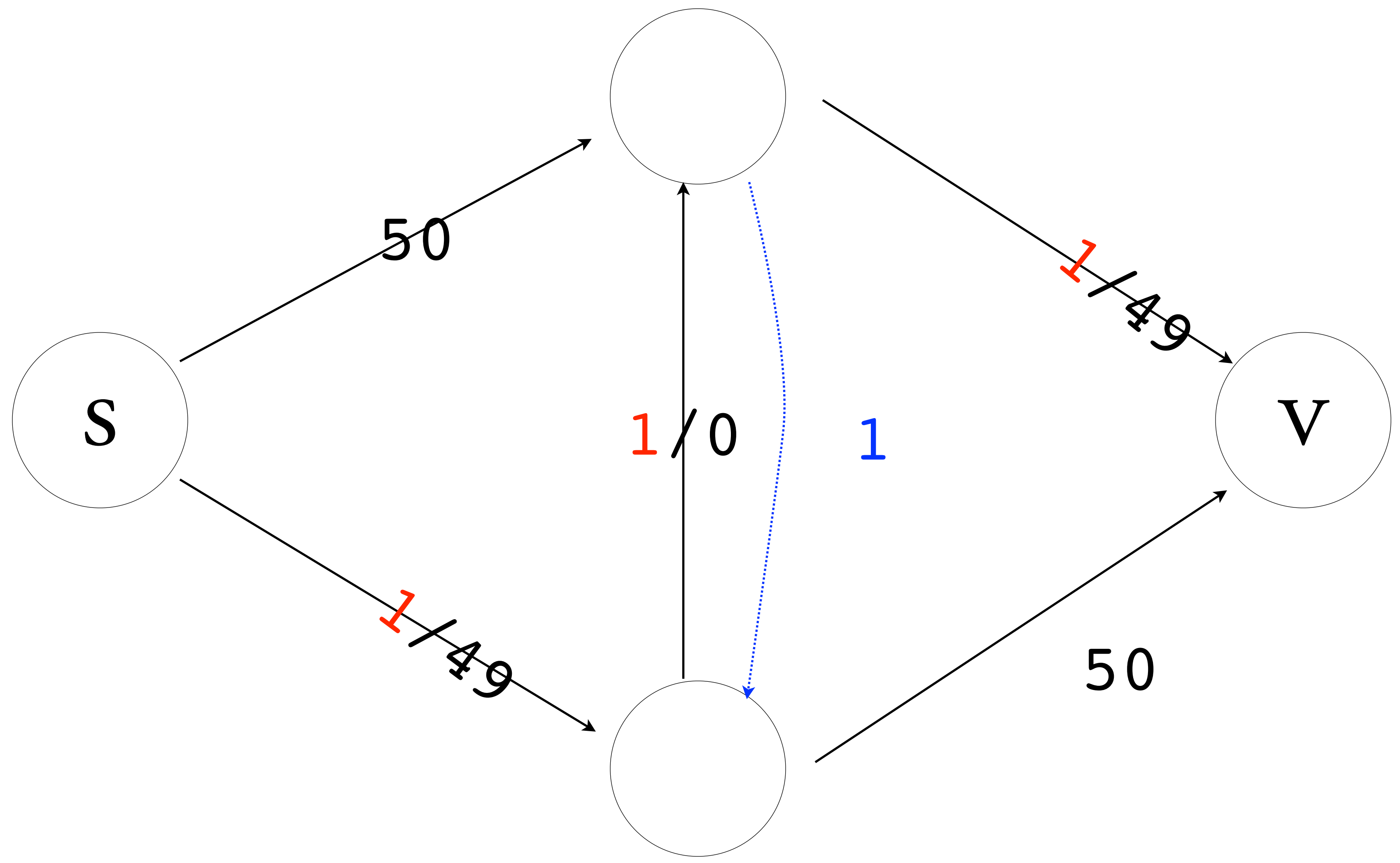
AUGMENT f WITH $c_f(p) = \min_{(u,v)\in p} c_f(u,v)$

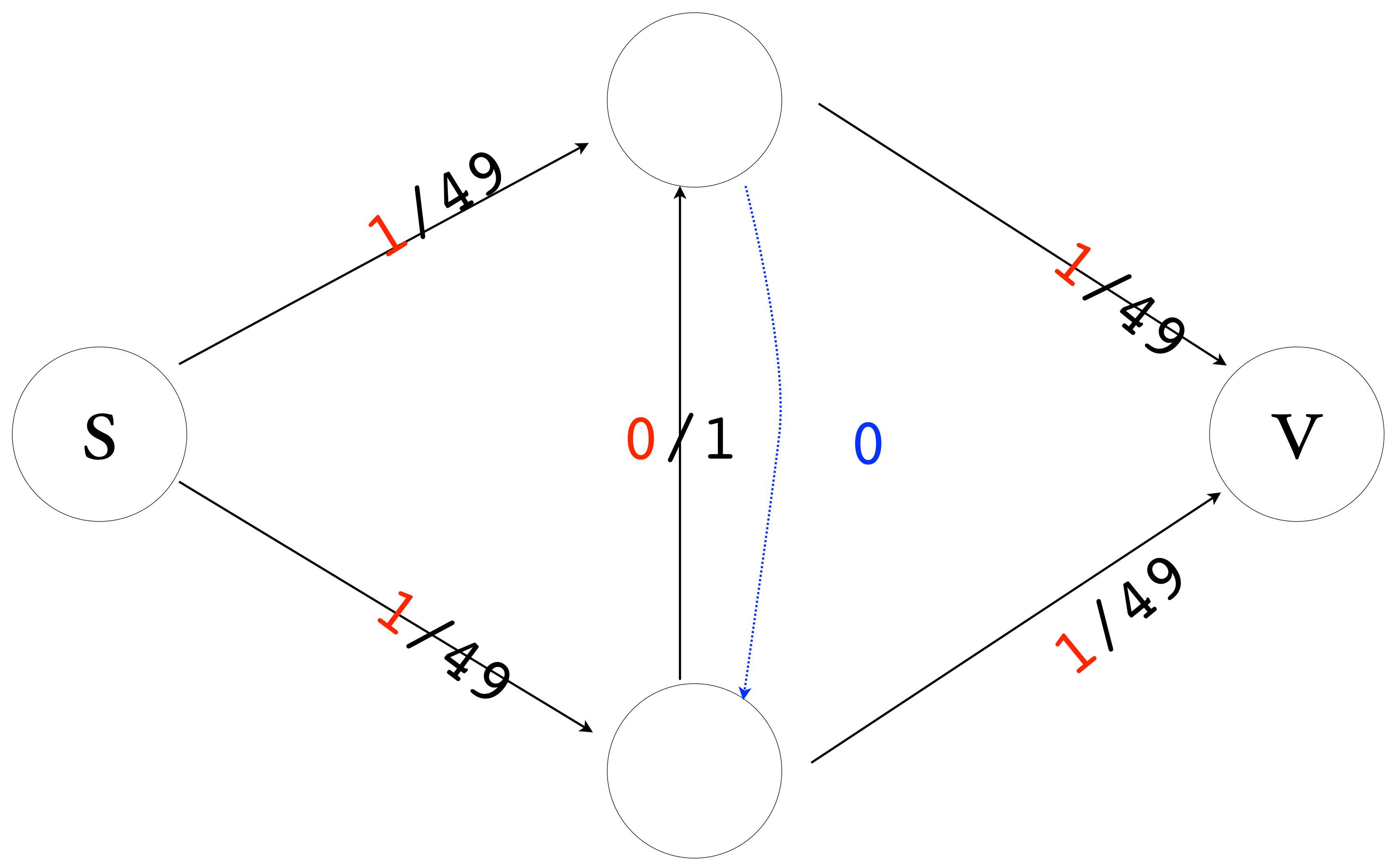
TIME TO FIND AN AUGMENTING PATH:

NUMBER OF ITERATIONS OF WHILE LOOP:


50
50
1
50
50
S
V

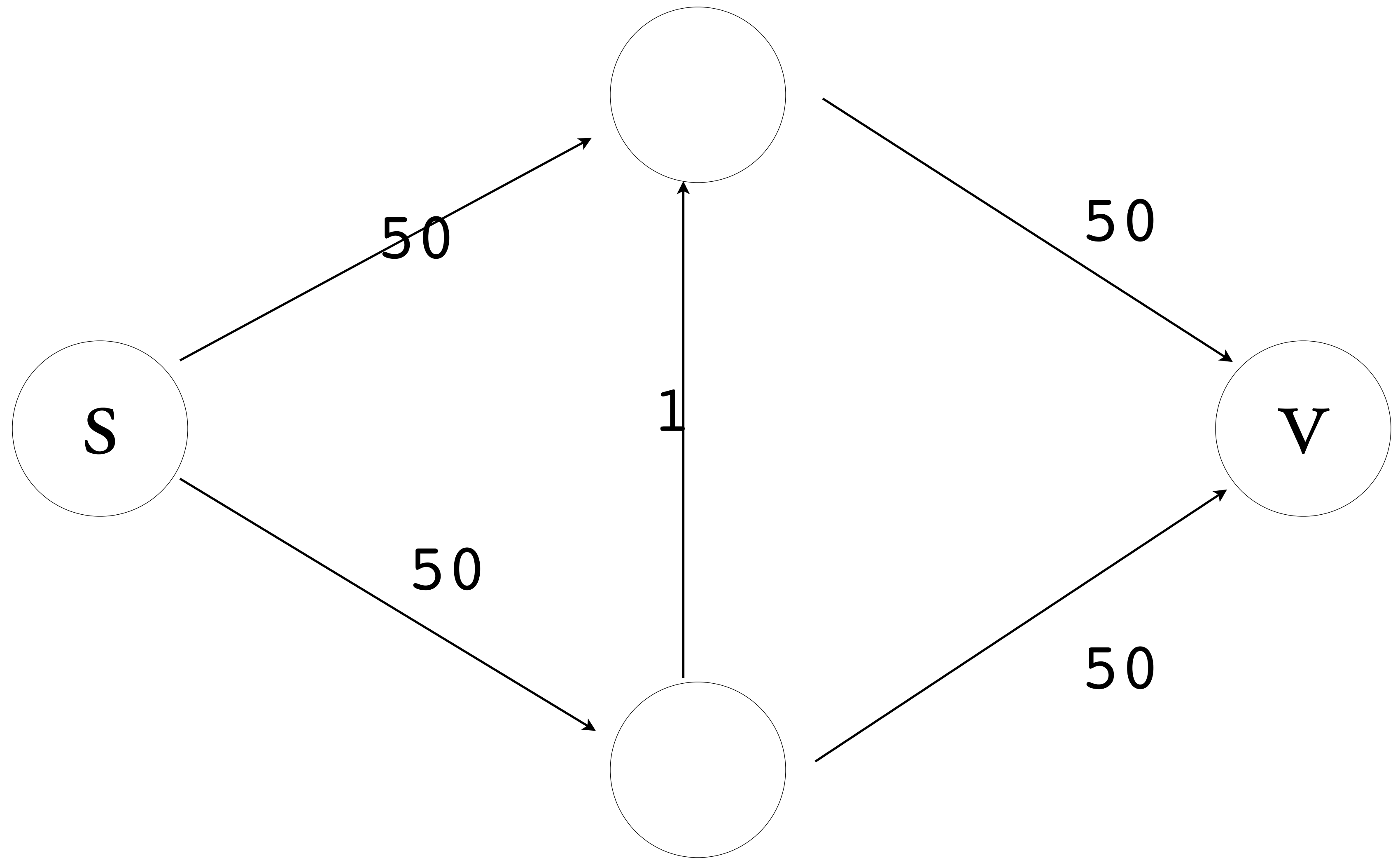

S
V
50
50
1
50
50


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



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

root of the problem



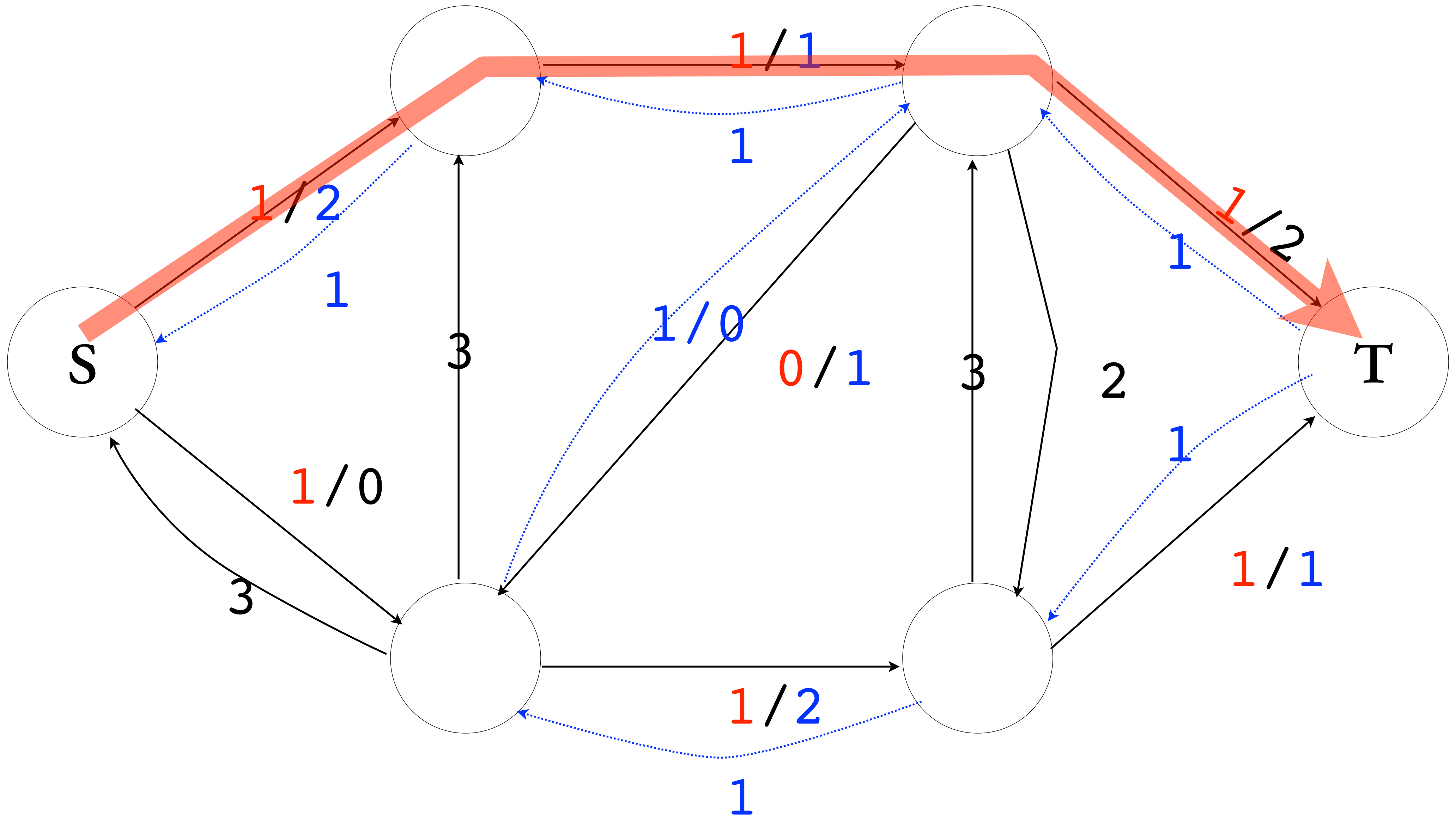
Edmonds-Karp 2

choose path with fewest edges first.

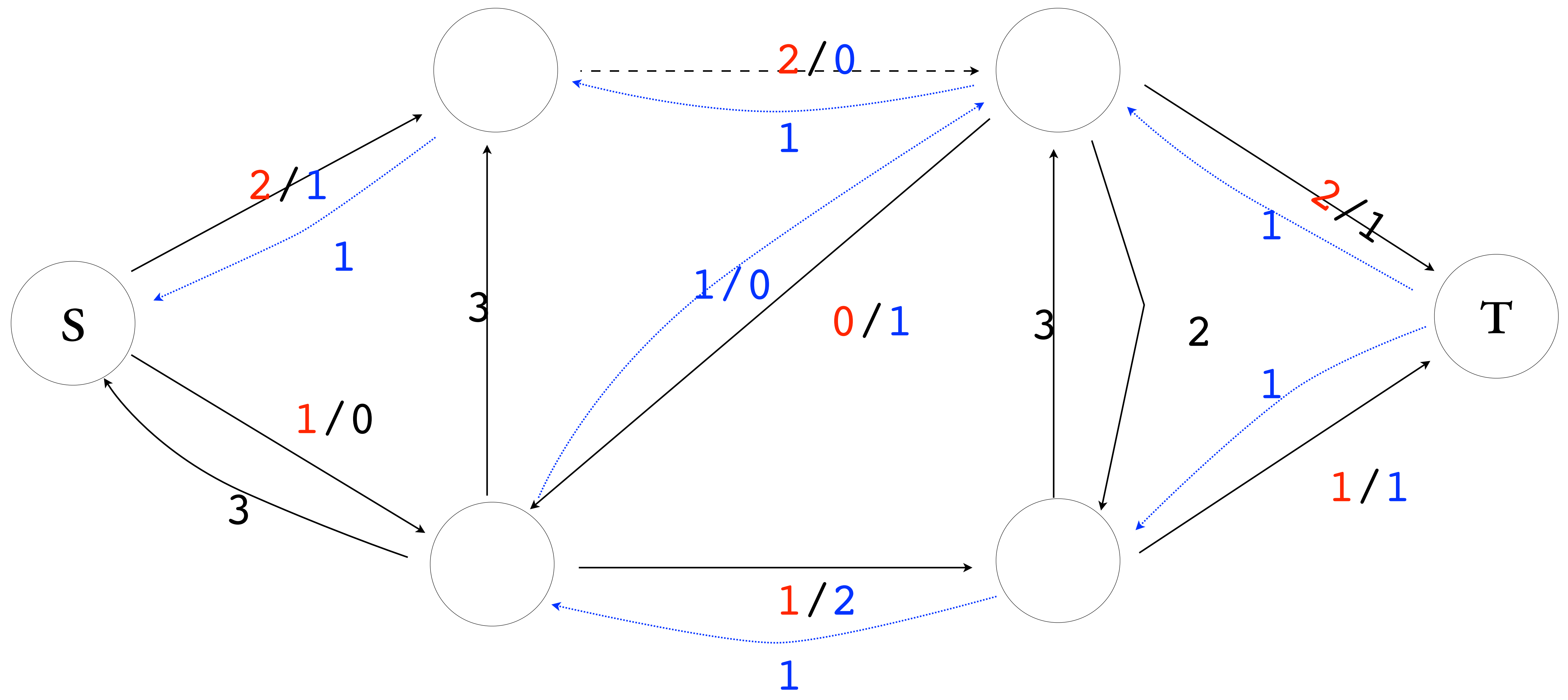
$\delta_f(s, v)$:

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

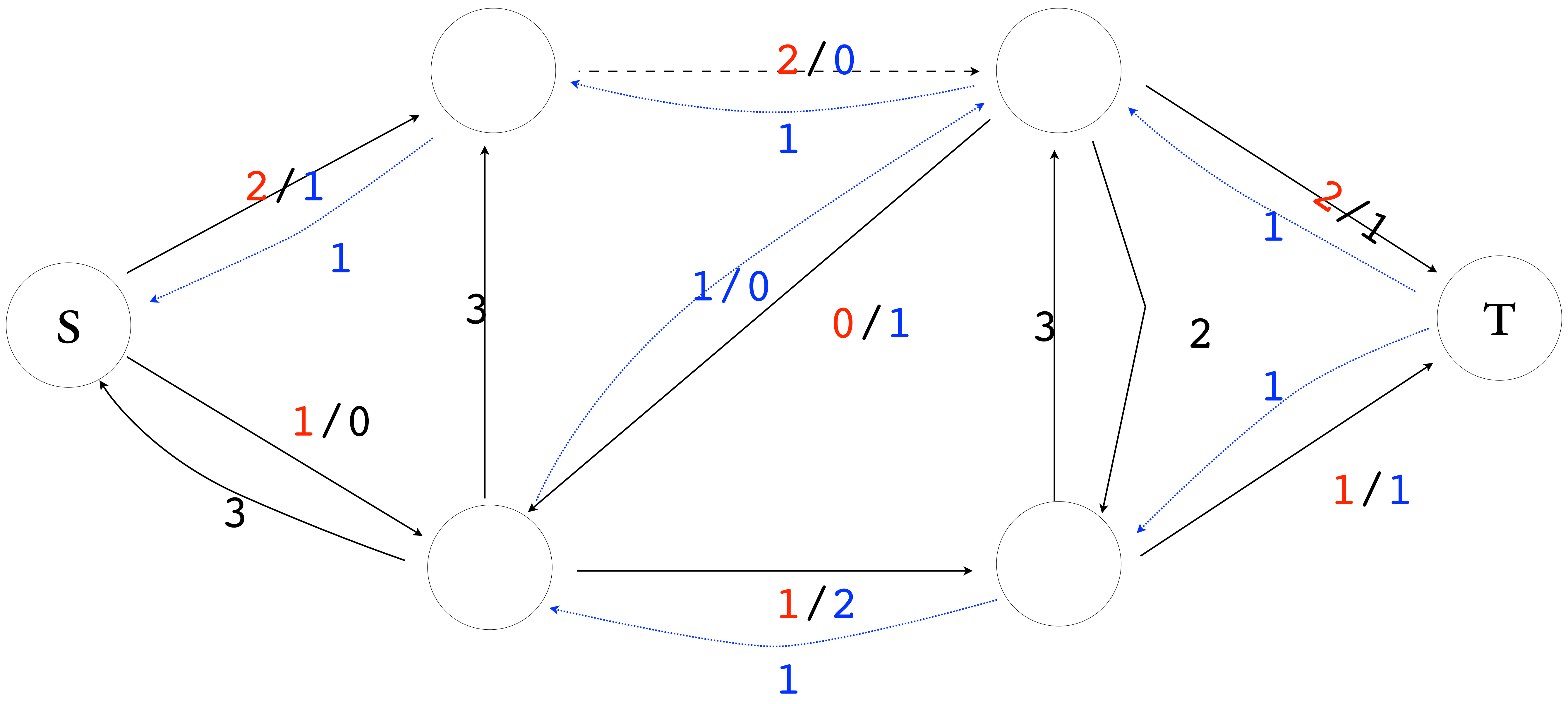
$$\delta_{i+1}(v) \geq \delta_i(v)$$



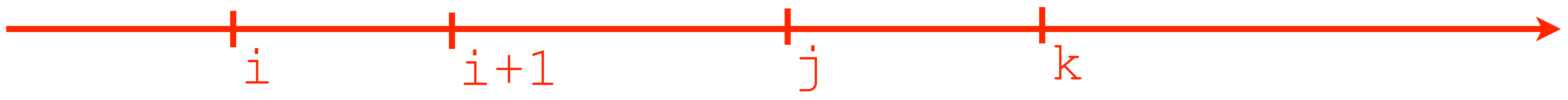
for every augmenting path, some edge is critical.

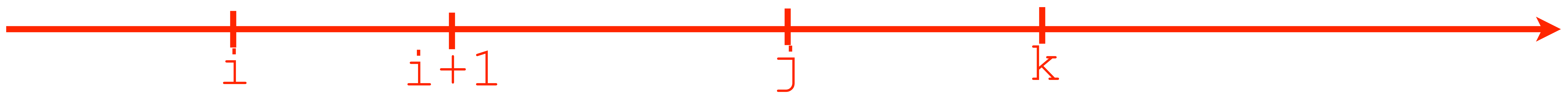


critical edges are removed in next residual graph.

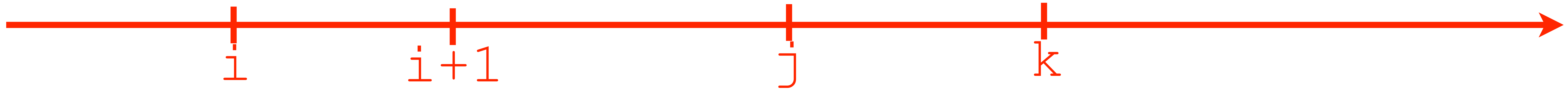


key idea: how many times can an edge be critical?

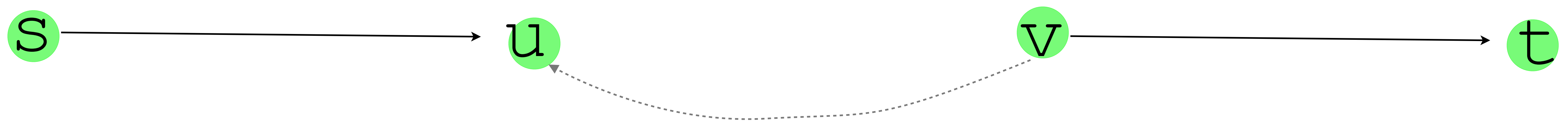

i
i+1
j
k



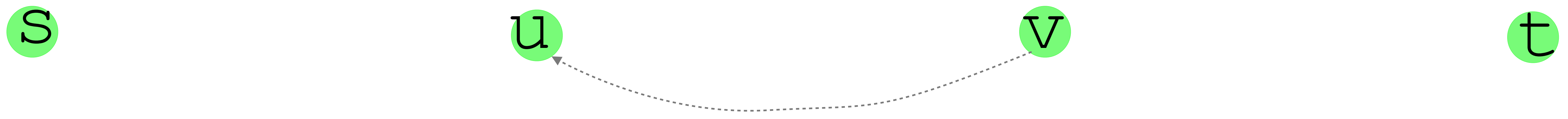
first time (u,v) is critical:

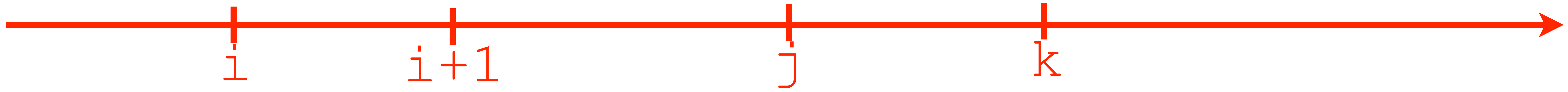


time i+1: (u,v) is critical: $\delta_{i+1}(s,v) \geq \delta_i(s,u) + 1$

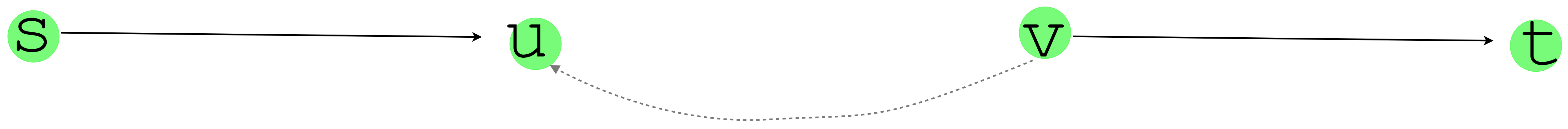


time j: Edge (u,v) STRIKES BACK

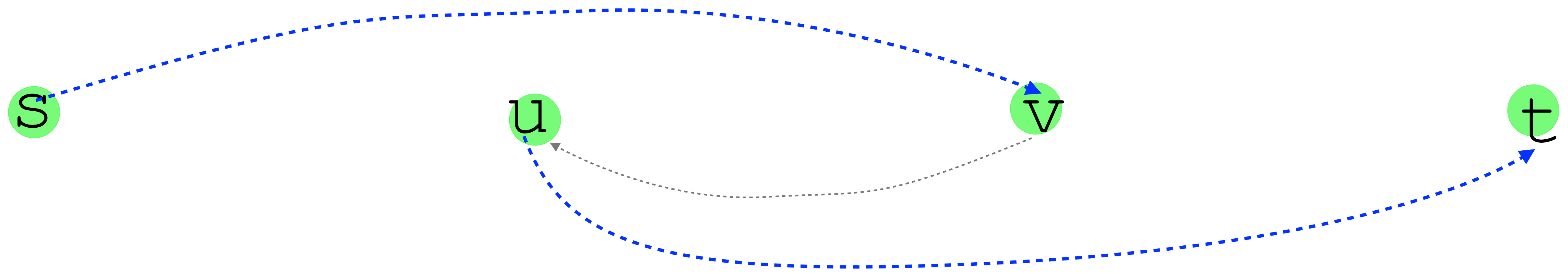




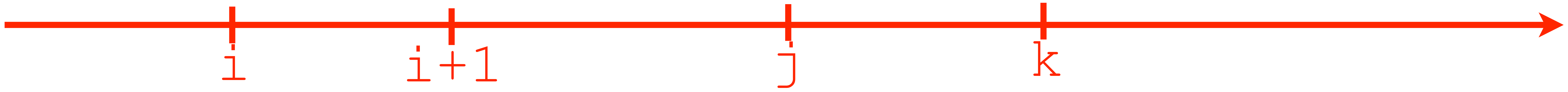
time i+1: (u,v) is critical: $\delta_{i+1}(s,v) \geq \delta_i(s,u) + 1$



time j: Edge (u,v) STRIKES BACK



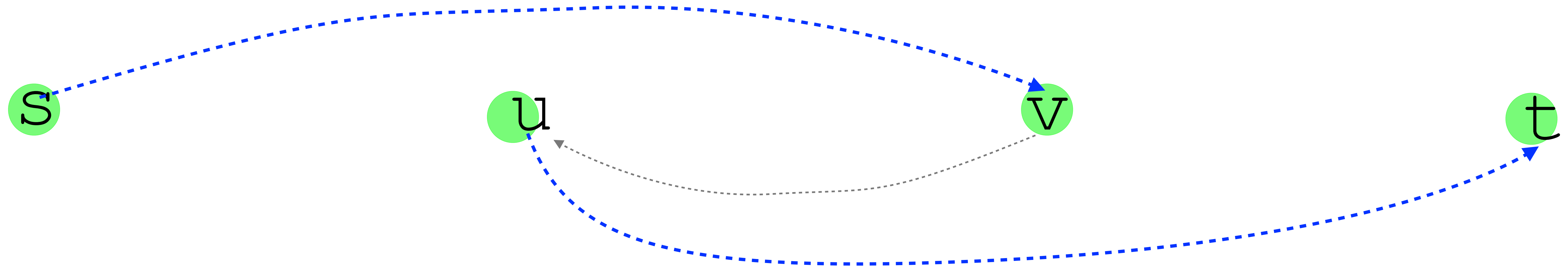
$\delta_j(s,u) = \delta_j(s,v) + 1$

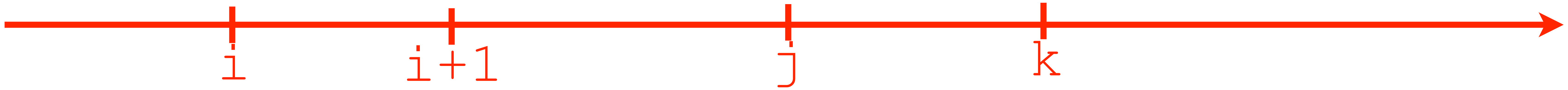


time j: Edge (u,v) STRIKES BACK

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

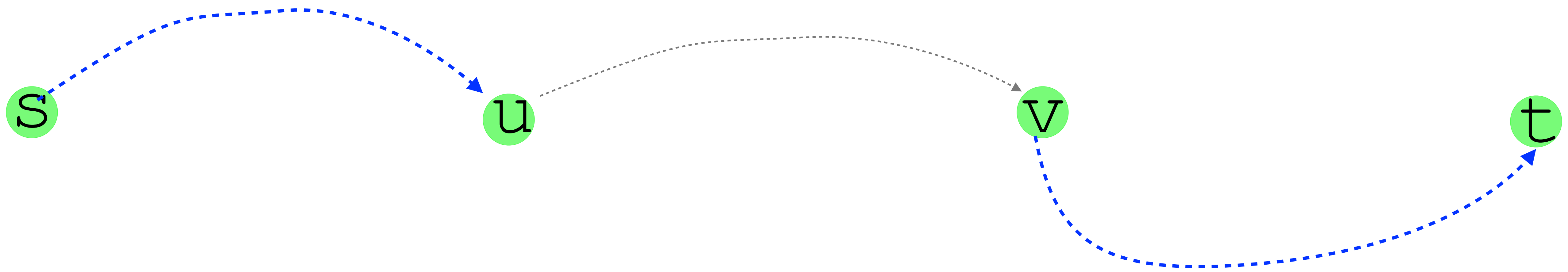
$$\delta_j(s,u) = \delta_j(s,v) + 1$$





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   times.

there are only   edges.

ergo, total # of augmenting paths:

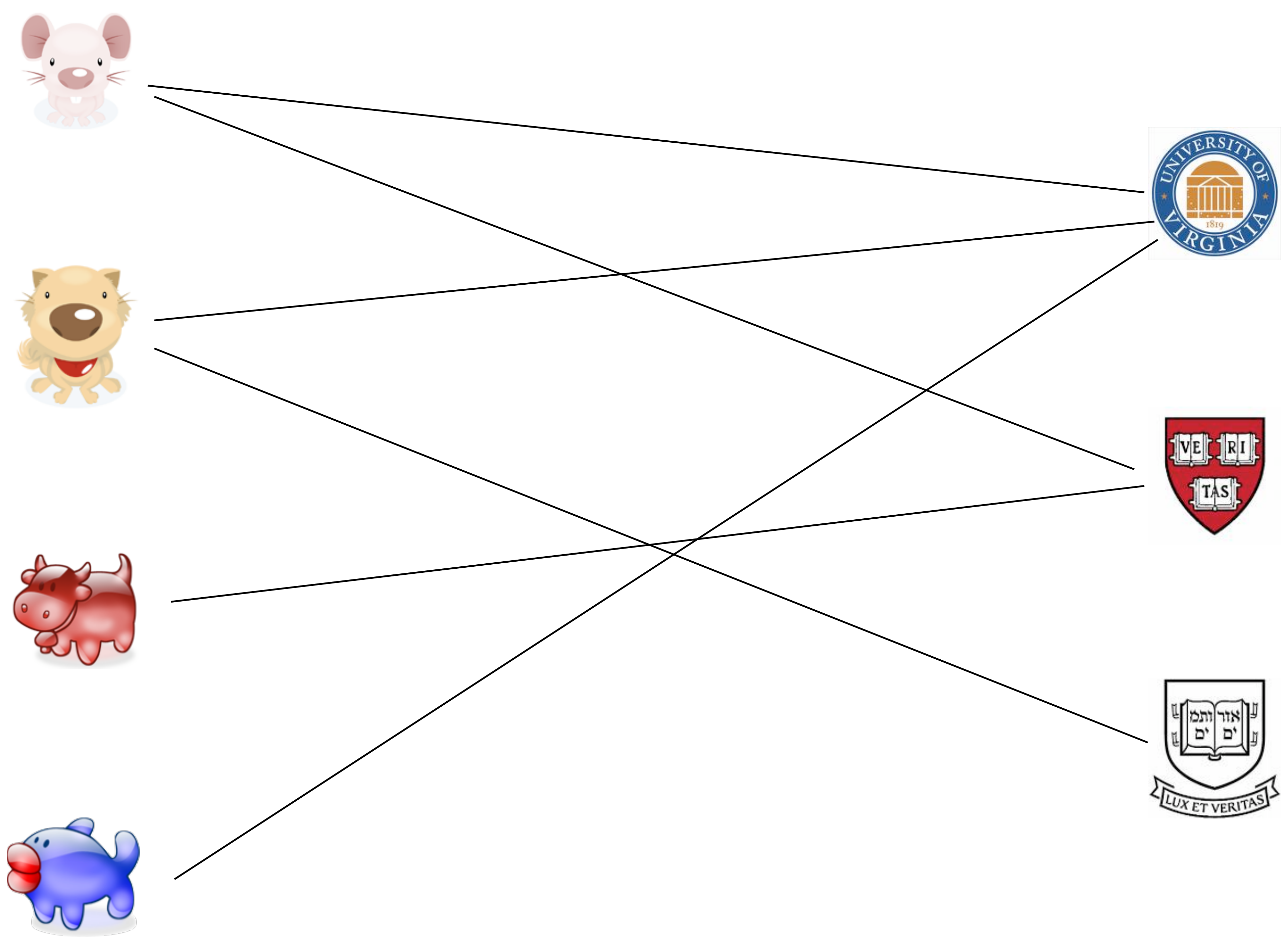
time to find an augmenting path:

total running time of E-K algorithm:

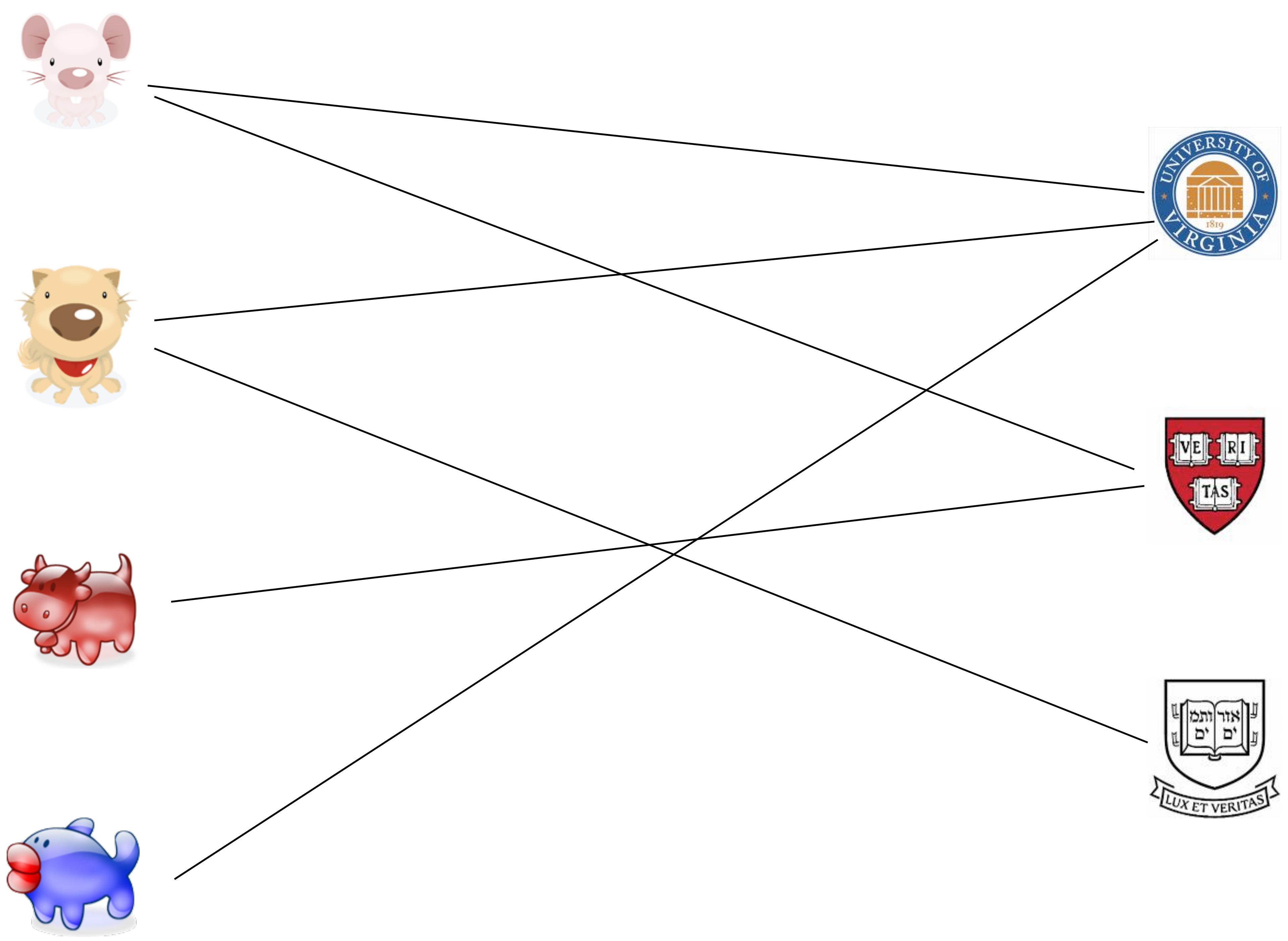
FF	$O(E|f^*|)$
EK2	
PUSH-RELABEL	
FASTER PUSH-RELABEL	

Bipartite Matchings

maximum bipartite matching



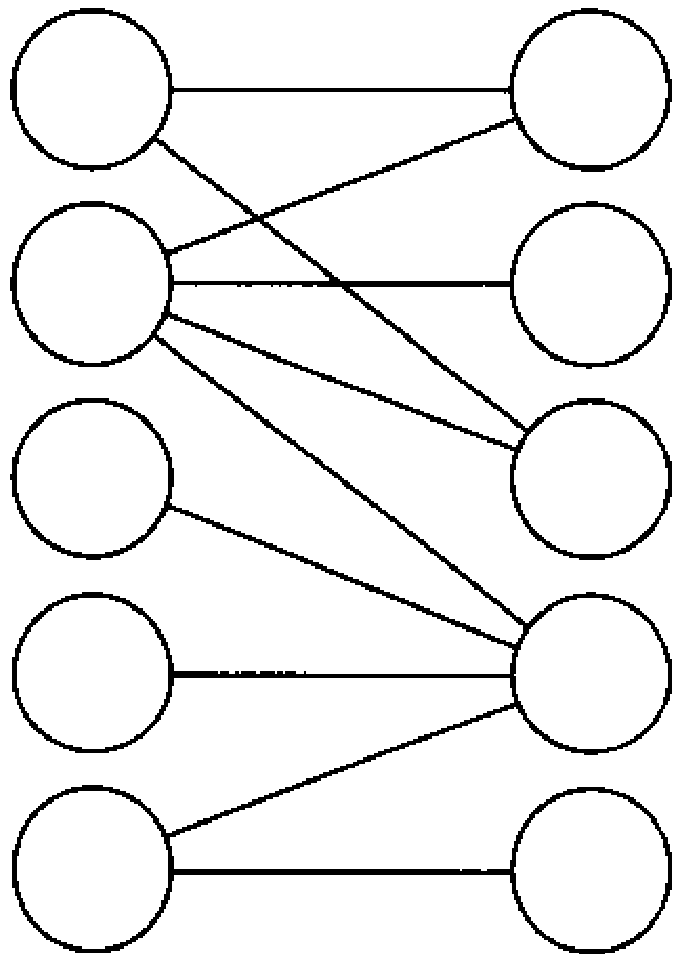
maximum bipartite matching



bipartite matching

PROBLEM:

algorithm

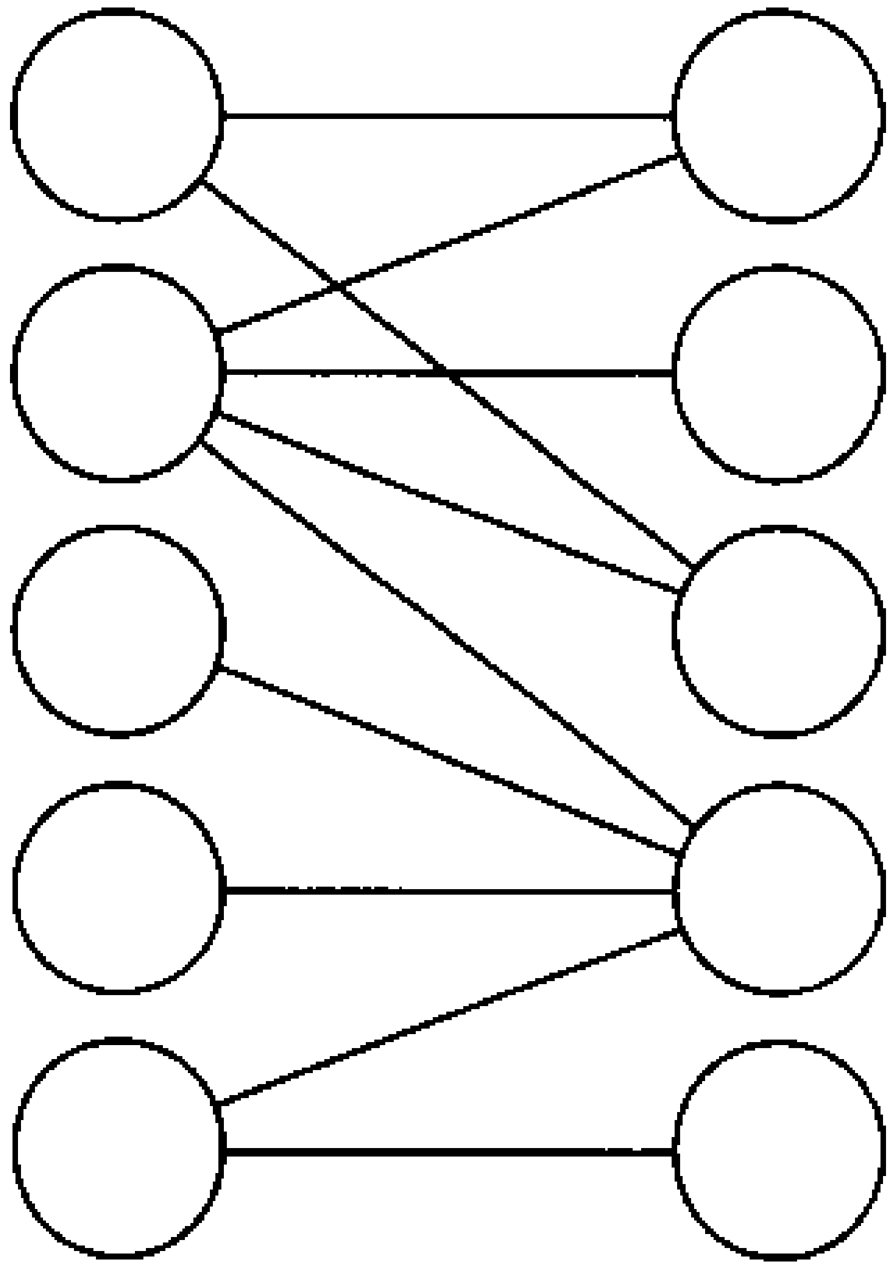


algorithm

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

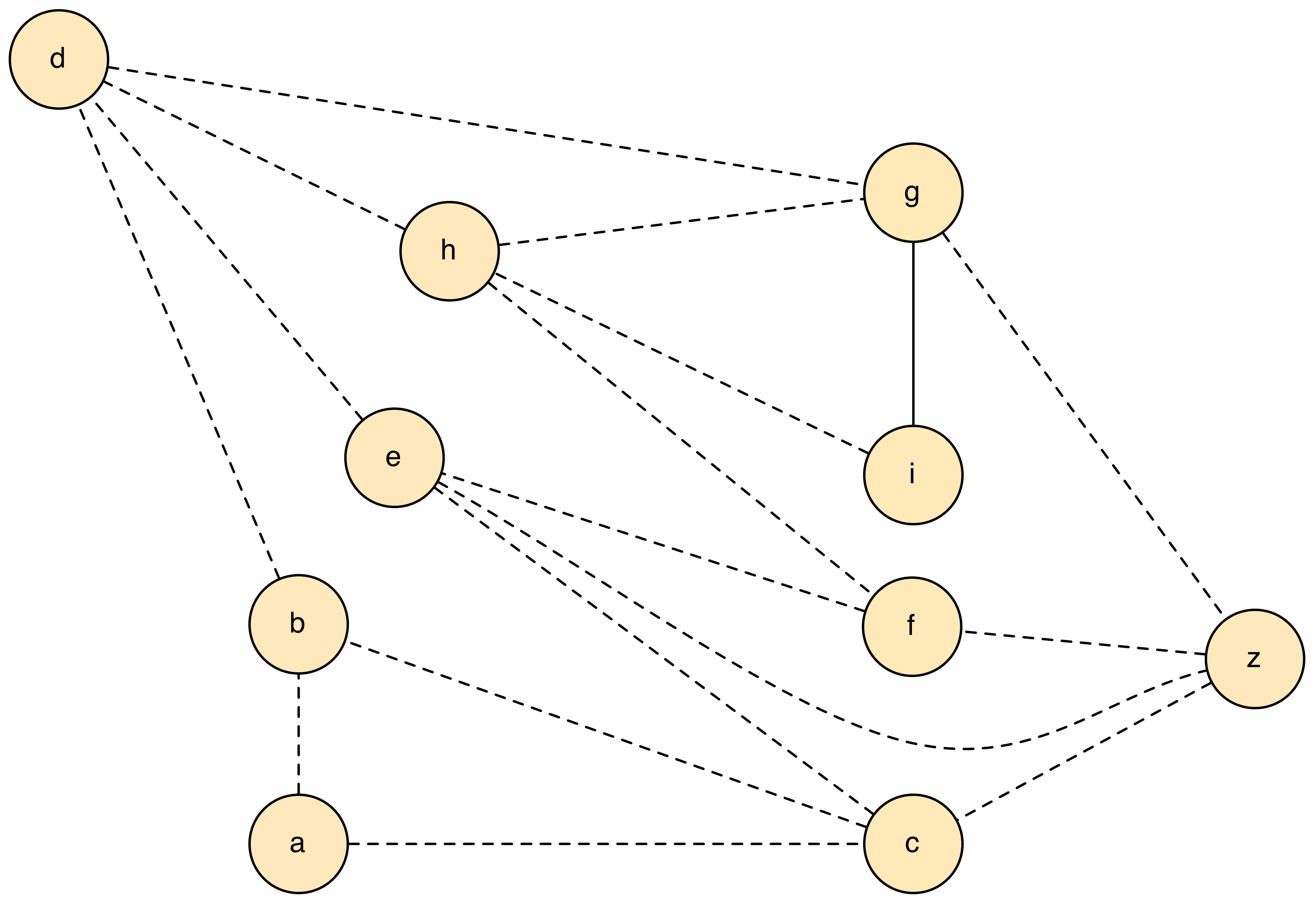
IF CAPACITIES ARE ALL INTEGRAL, THEN

correctness

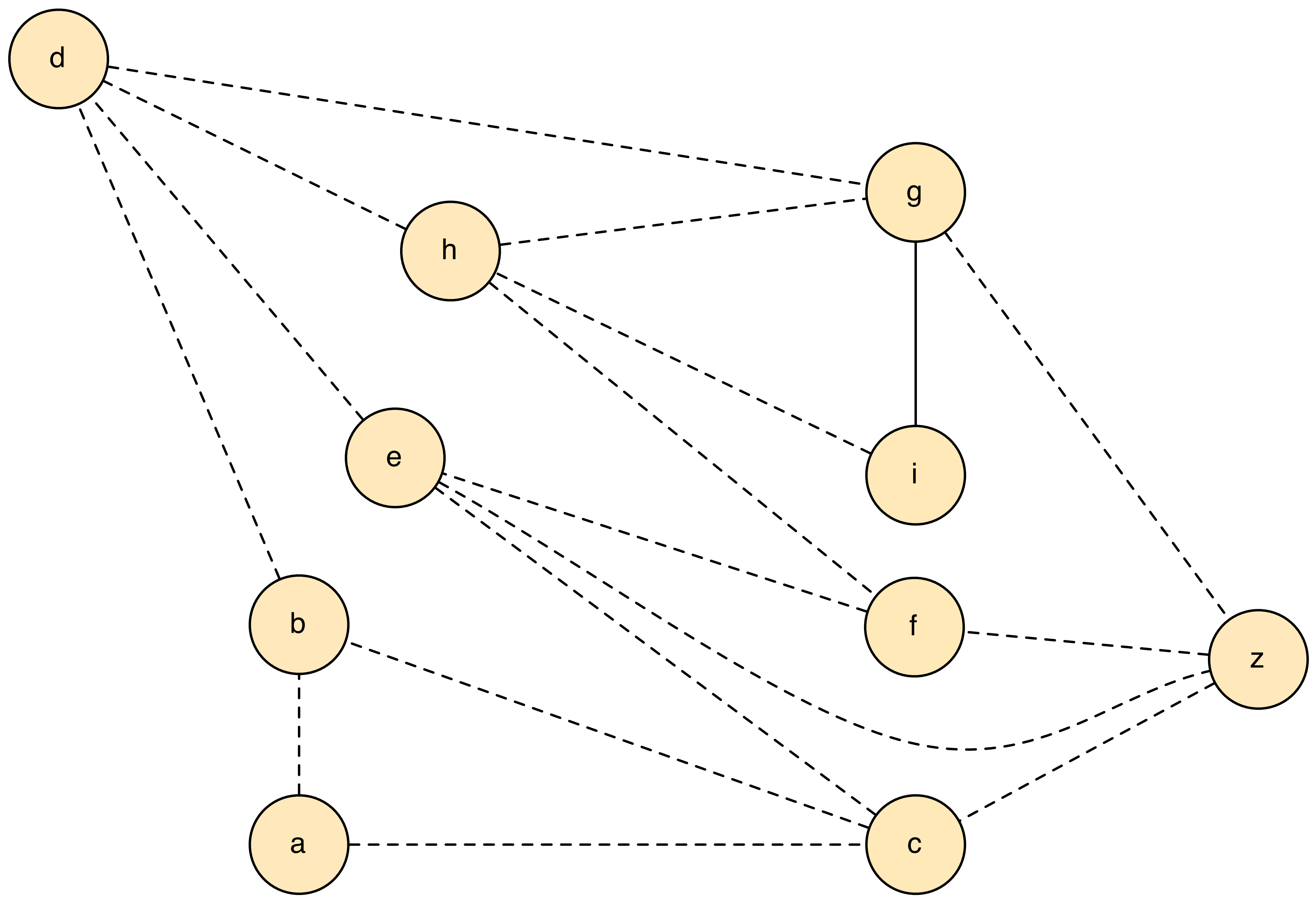
IF G' HAS A FLOW OF K, THEN G HAS K-MATCHING.

running time

edge-disjoint paths



algorithm



1. Compute max flow
2. Remove all edges with $f(e) = 0$.
3. Walk from s.
    1. If you reach a node you have visited before, erase flow along path
    2. If you reach t, add this path to your set, erase flow along path.

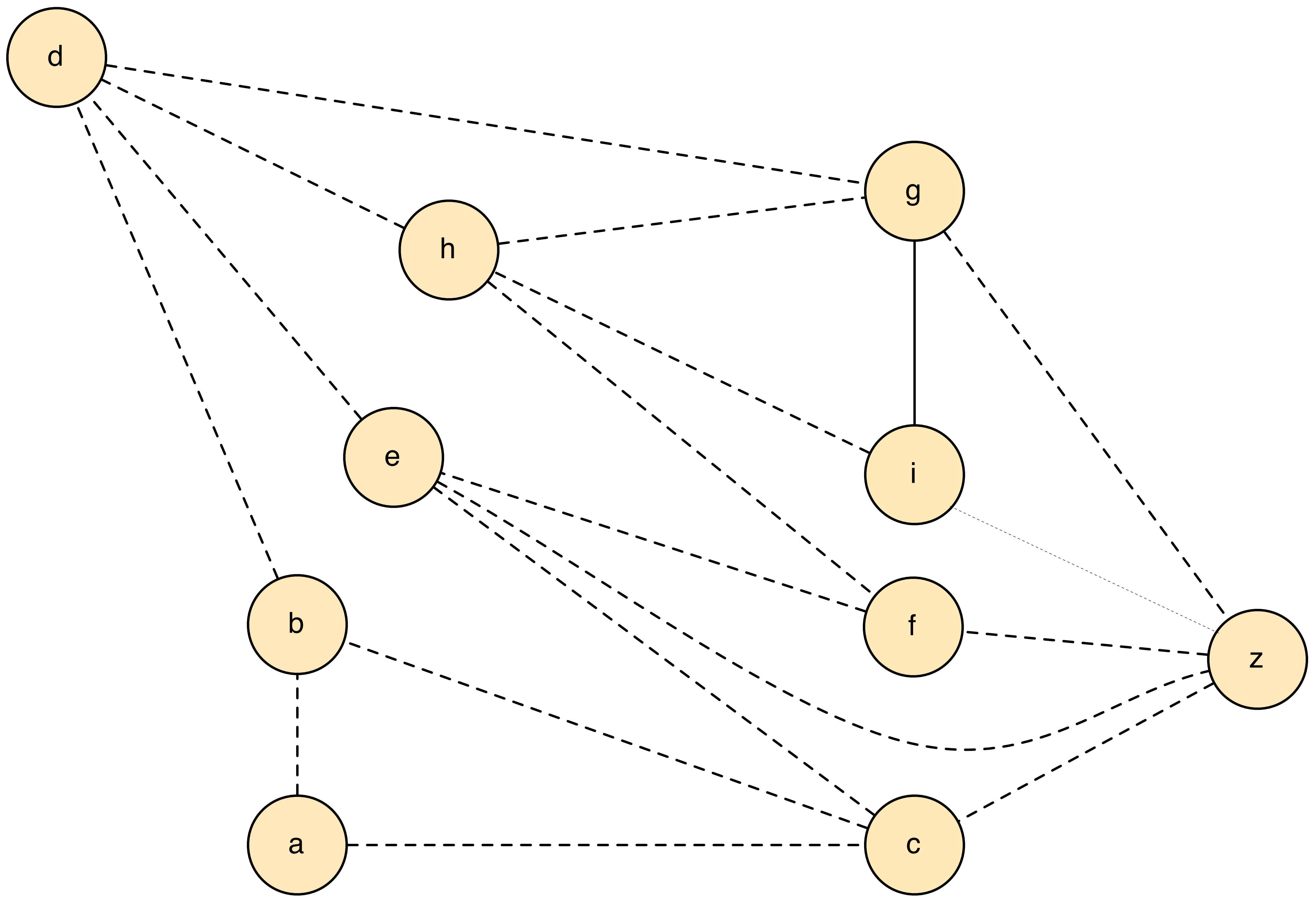
analysis

IF G HAS K DISJOINT PATHS, THEN

analysis

IF G' HAS A FLOW OF K, THEN

vertex-disjoint paths

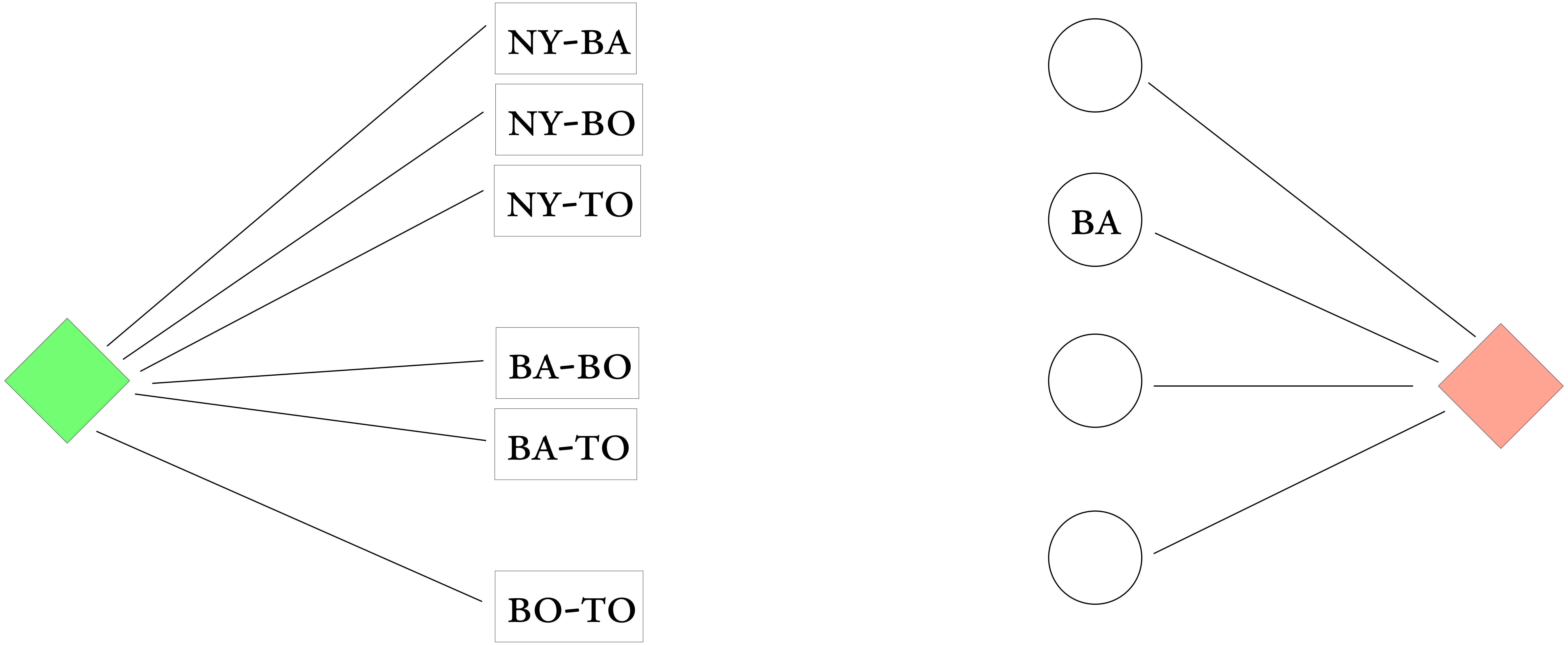


BASEBALL ELIMINATION

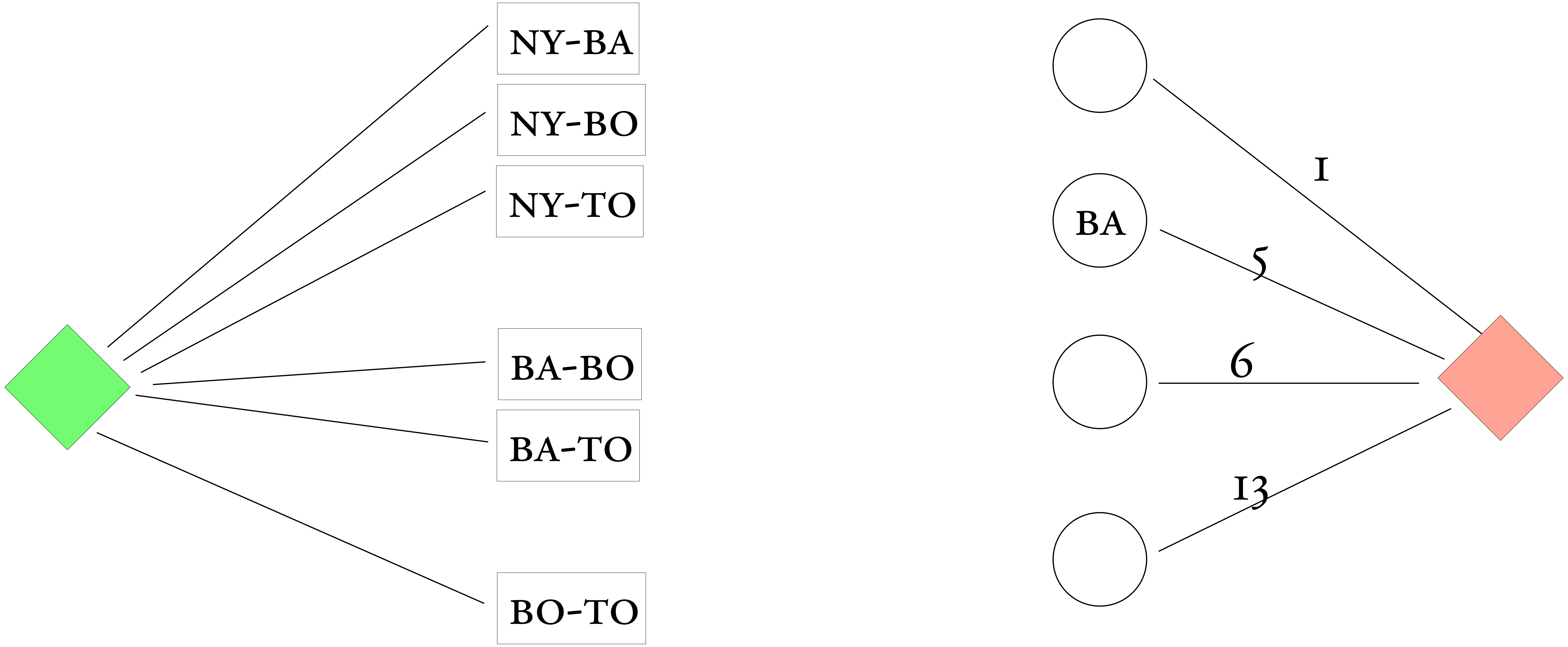
				Against			
	W	L	Left	A	P	N	M
ATL	83	71	8	-	1	6	1
PHL	80	79	3	1	-	0	2
NY	78	78	6	6	0	-	0
MONT	77	82	3	1	2	0	-

BASEBALL ELIMINATION

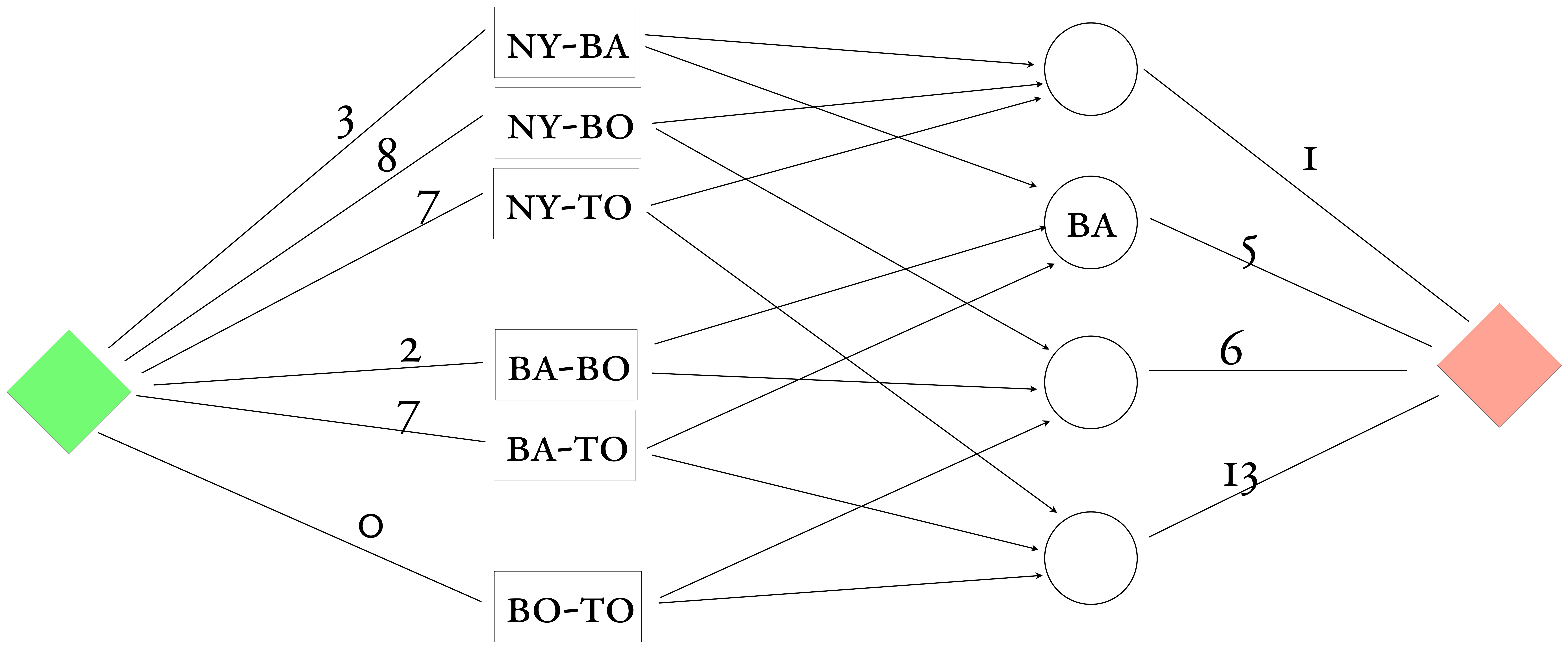
				Against				
	W	L	Left	N	B	Bo	T	D
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			



	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			



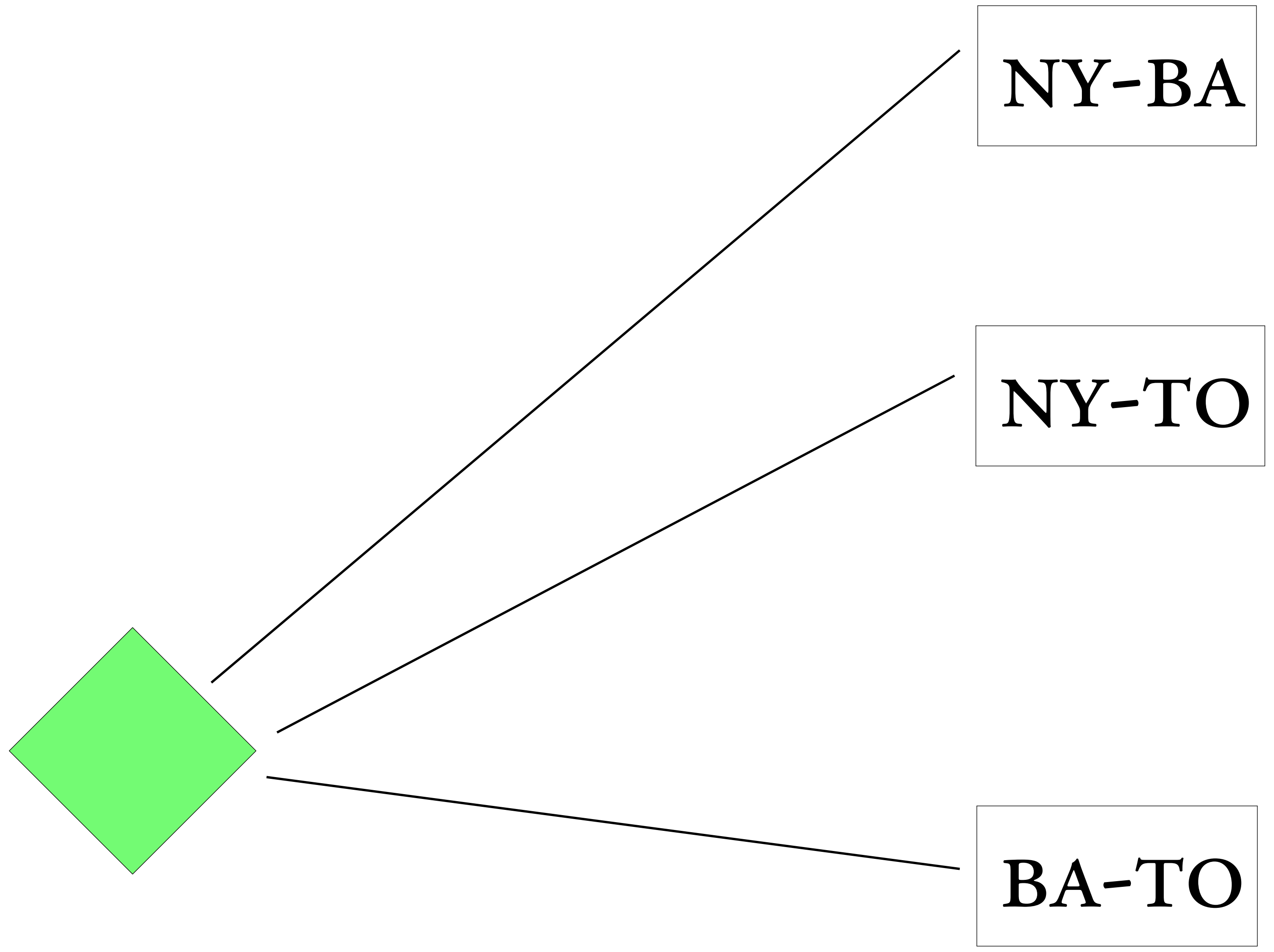
	W	L	Left	N	B	Bo	T	D
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			

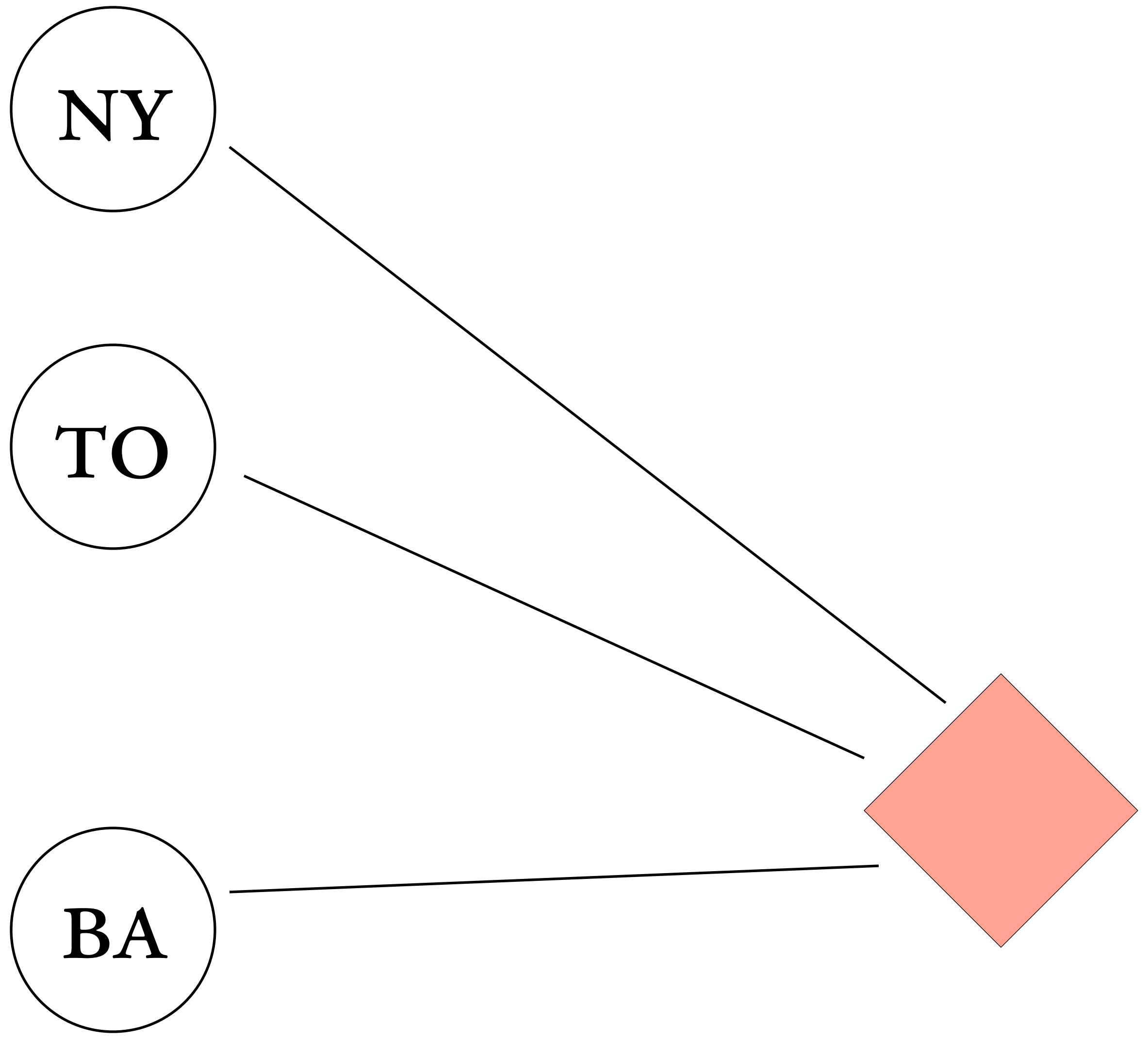


	W	L	Left	N	B	Bo	T	D
NY	75	59	28		3	8	7	3
BAL	71	63	28	3		2	7	4
BOS	69	66	27	8	2			
TOR	63	72	27	7	7			
DET	49	86	27	3	4			

BASEBALL ELIMINATION

	W			Against	
		N	B	Bo	T
NY	90		1	4	6
BAL	88	1		4	1
BOS	79				
TOR	87	6	1	4	





	W	L	Left	N	B	Bo	T
NY	90				1	4	6
BAL	88			1		4	1
BOS	79			4	4		4
TOR	87			6	1	4	

Why it works

Thm: A team T has been eliminated if the maxflow of graph G is less than the total number of games left between the other teams in the league.