L20

max flow

4102

11.05.2013

Nathan Brunelle

Max flow

Min Cut

“Consider a rail network connecting two cities by way of a number of intermediate cities, where each link of the network has a number assigned to it representing its capacity. Assuming a steady state condition, find a maximal flow from one given city to the other.”

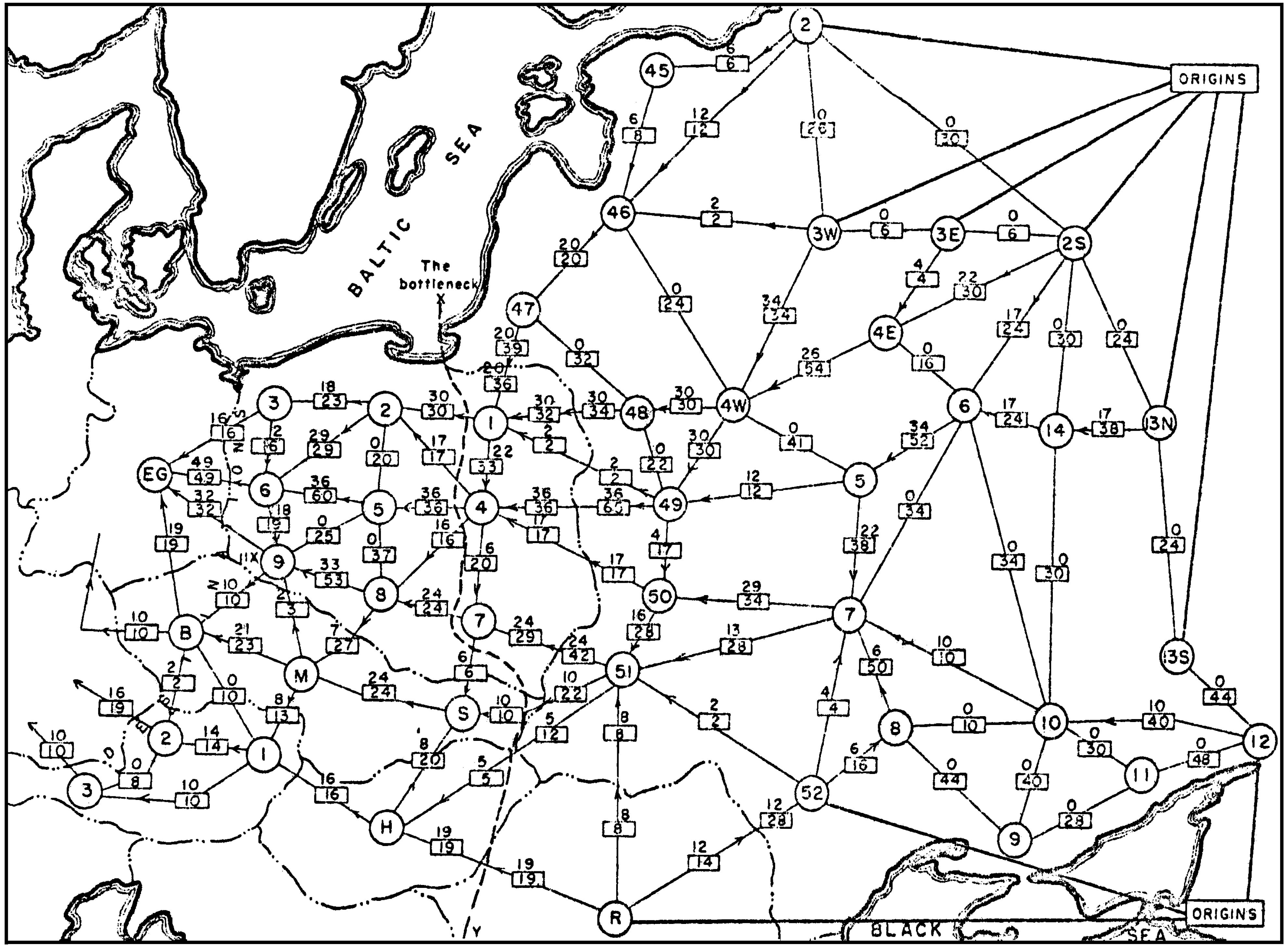


Figure 4 From Harris and Ross [3]: Schematic diagram of the railway network of the Western Soviet Union and East European countries, with a maximum flow of value 163,000 tons from Russia to Eastern Europe and a cut of capacity 163,000 tons indicated as 'The bottleneck'

courtesy Alexander Schrijver

FLOW NETWORKS

$$G = (V, E)$$

SOURCE + SINK:

CAPACITIES:

FLOW NETWORKS

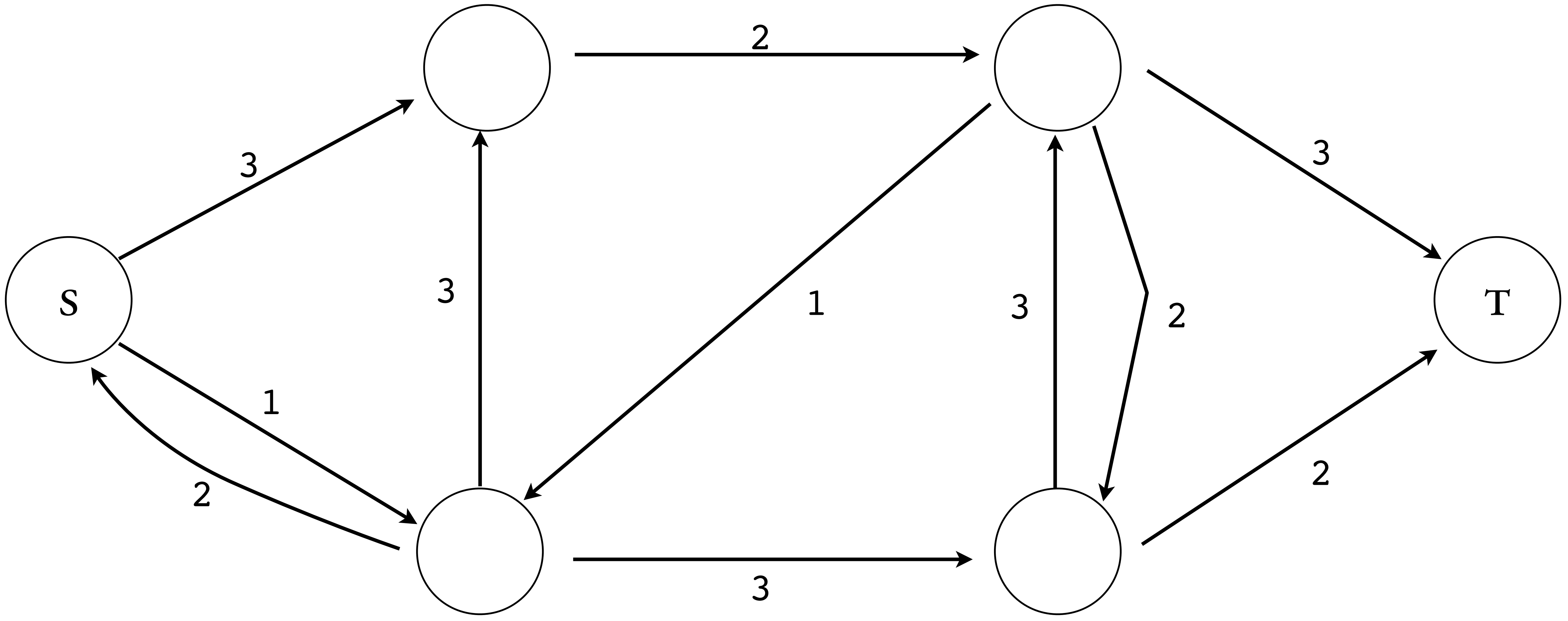
$$G = (V, E)$$

SOURCE + SINK: NODE S, AND T

CAPACITIES: $c(u, v)$

ASSUMED TO BE 0 IF NO (U,V) EDGE

EXAMPLE



FLOW

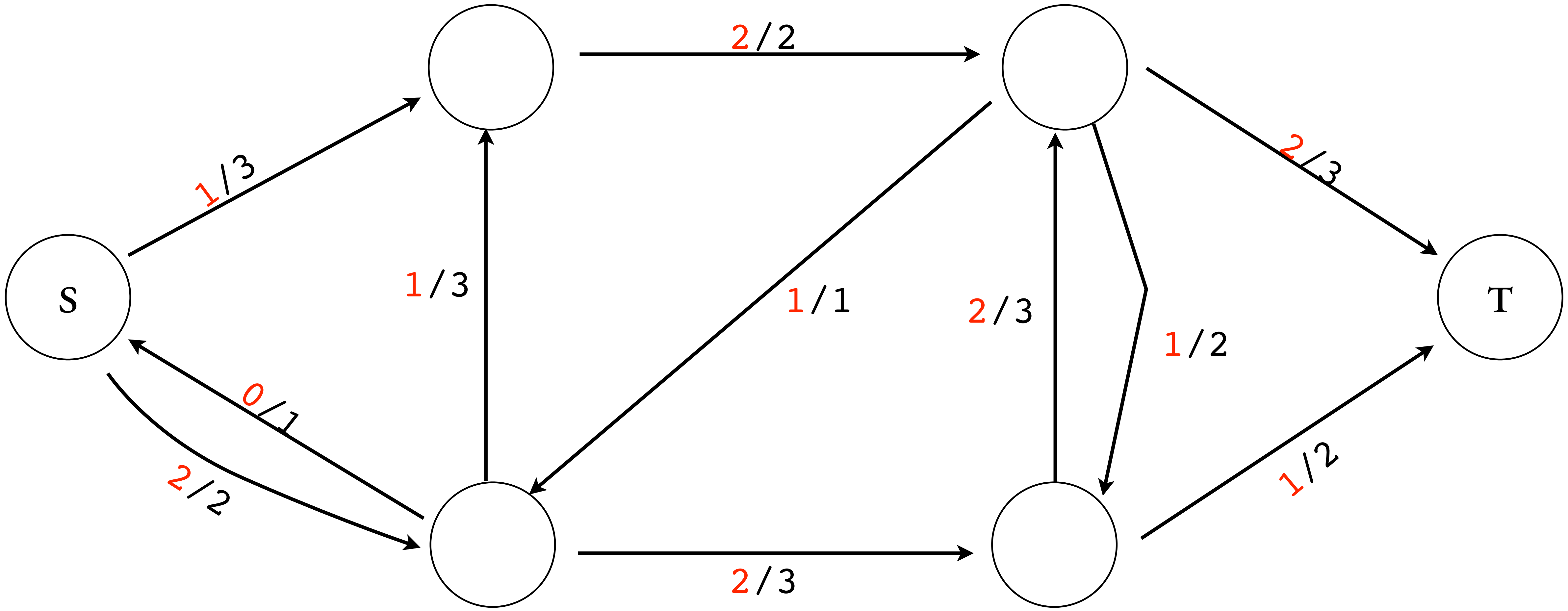
MAP FROM EDGES TO NUMBERS:

CAPACITY CONSTRAINT:

FLOW CONSTRAINT:

$$|f| =$$

EXAMPLE



MAX FLOW PROBLEM

given a graph G, compute

GREEDY SOLUTION?

Hundreds of applications

- bipartite matching
- edge-disjoint paths
- node-disjoint paths
- scheduling
- baseball elimination
- resource allocations

will discuss many of these applications in L22.

ALGORITHMS FOR MAX FLOW

CUTS

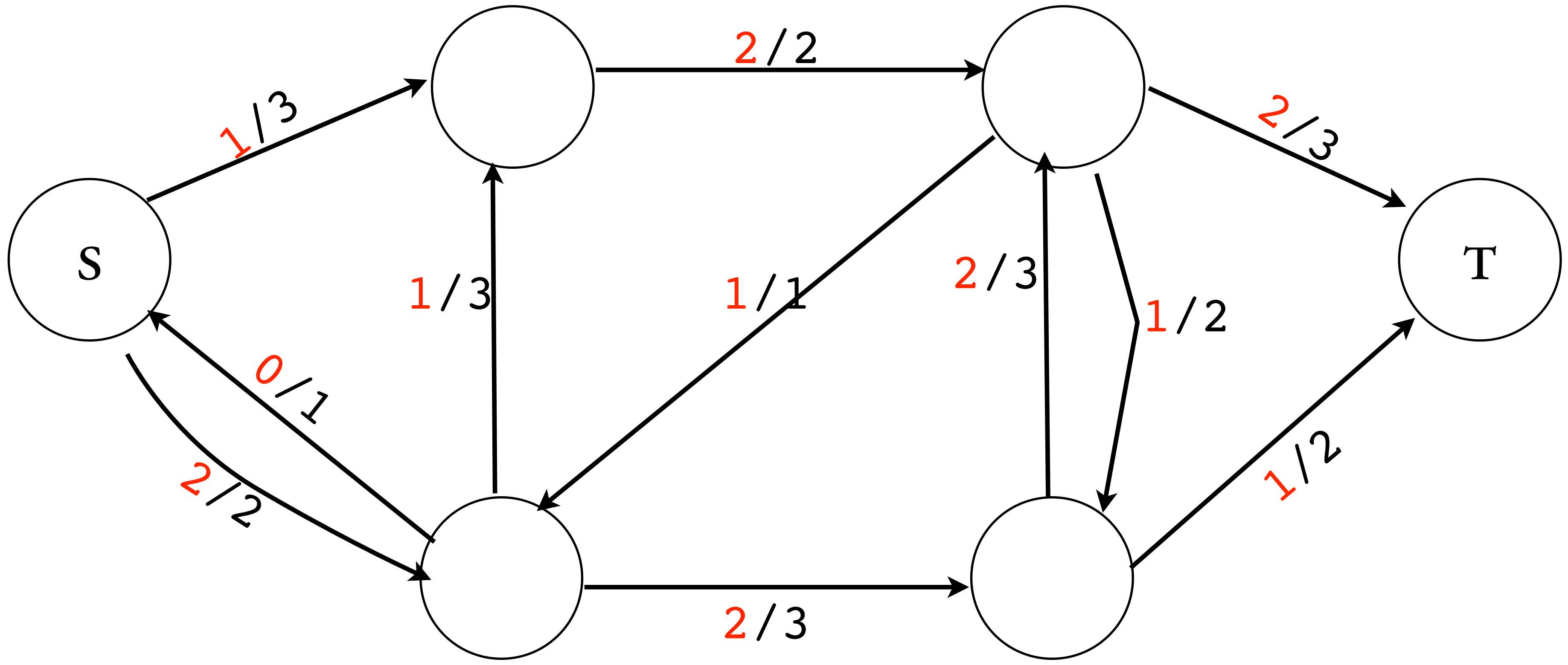
DEF OF A CUT:

COST OF A CUT:

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

LEMMA: [MIN CUT] FOR ANY $f, (S, T)$

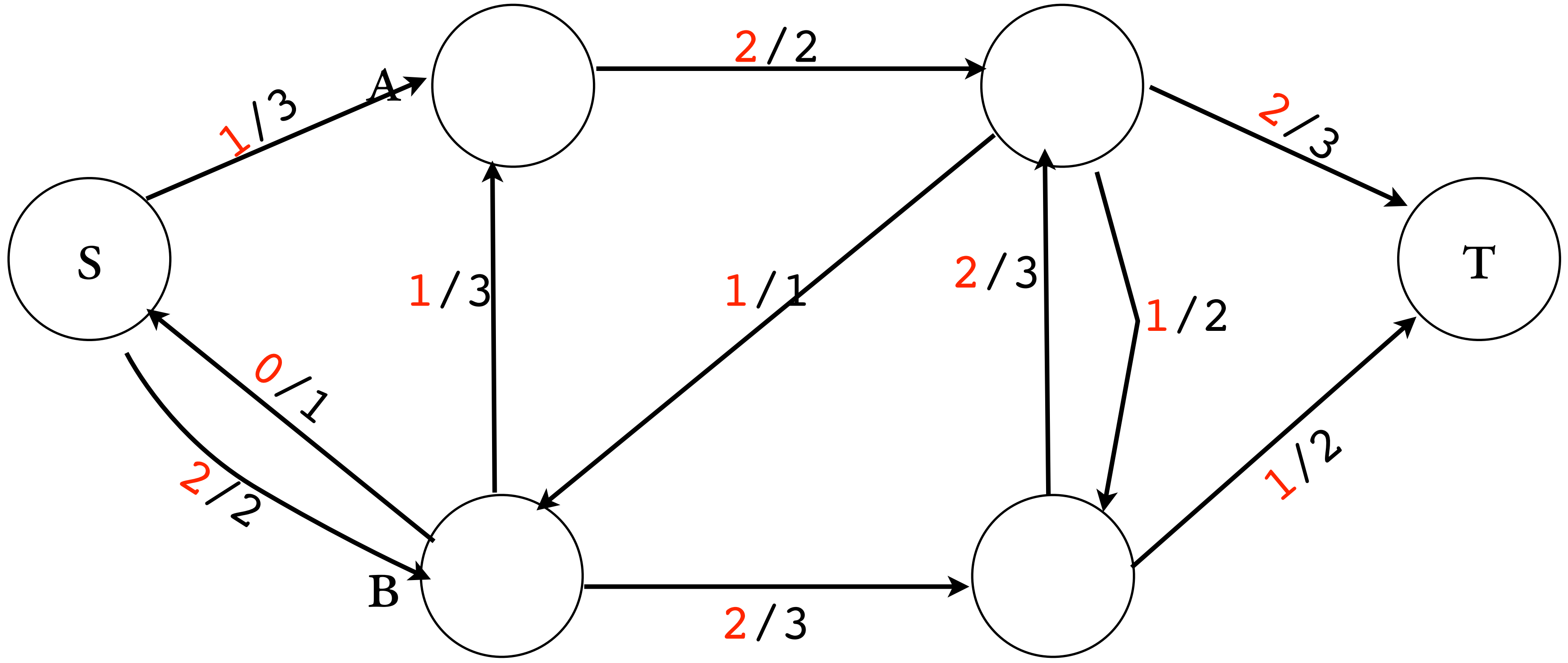
FOR ANY $f, (S, T)$ IT HOLDS THAT $|f| \leq ||S, T||$



EXAMPLE:

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

PROOF:


S
A
B
T
1/3
2/2
2/3
1/3
1/1
2/3
1/2
0/1
2/2
2/3
1/2

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

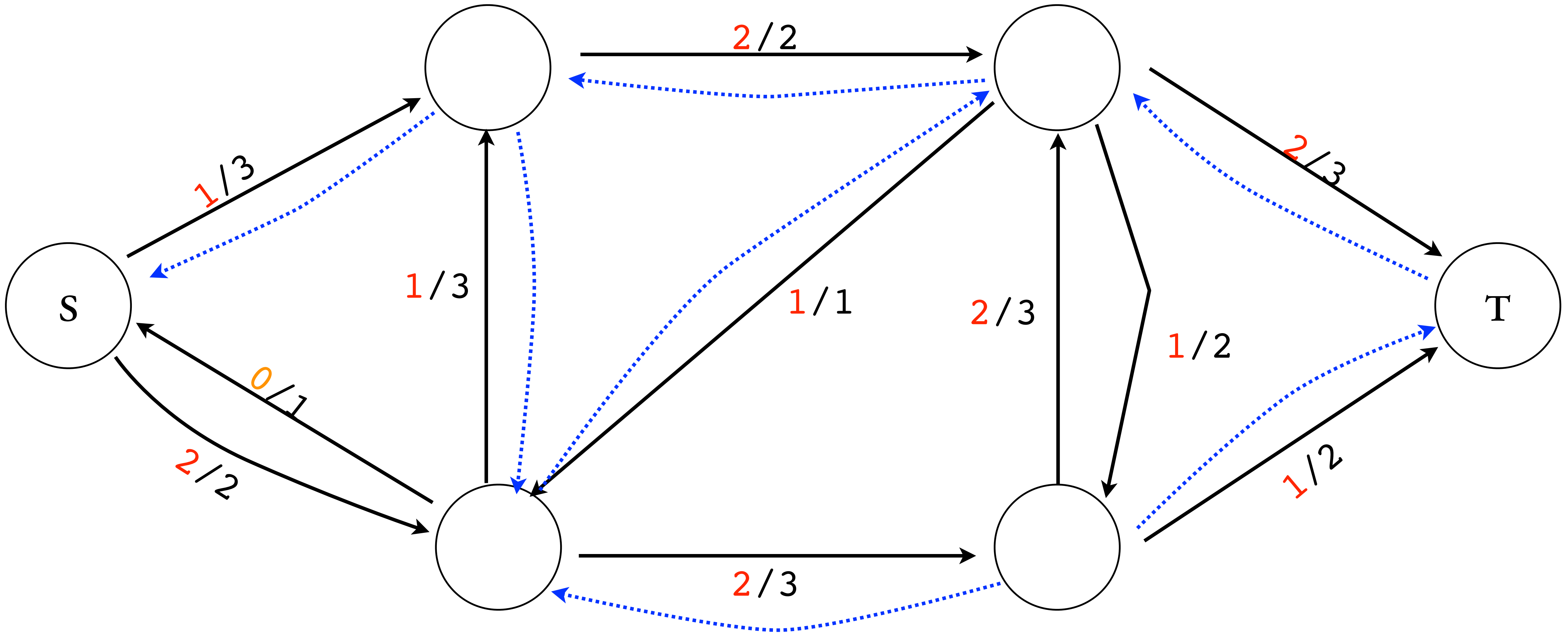
(FINISHING PROOF)

RESIDUAL GRAPHS

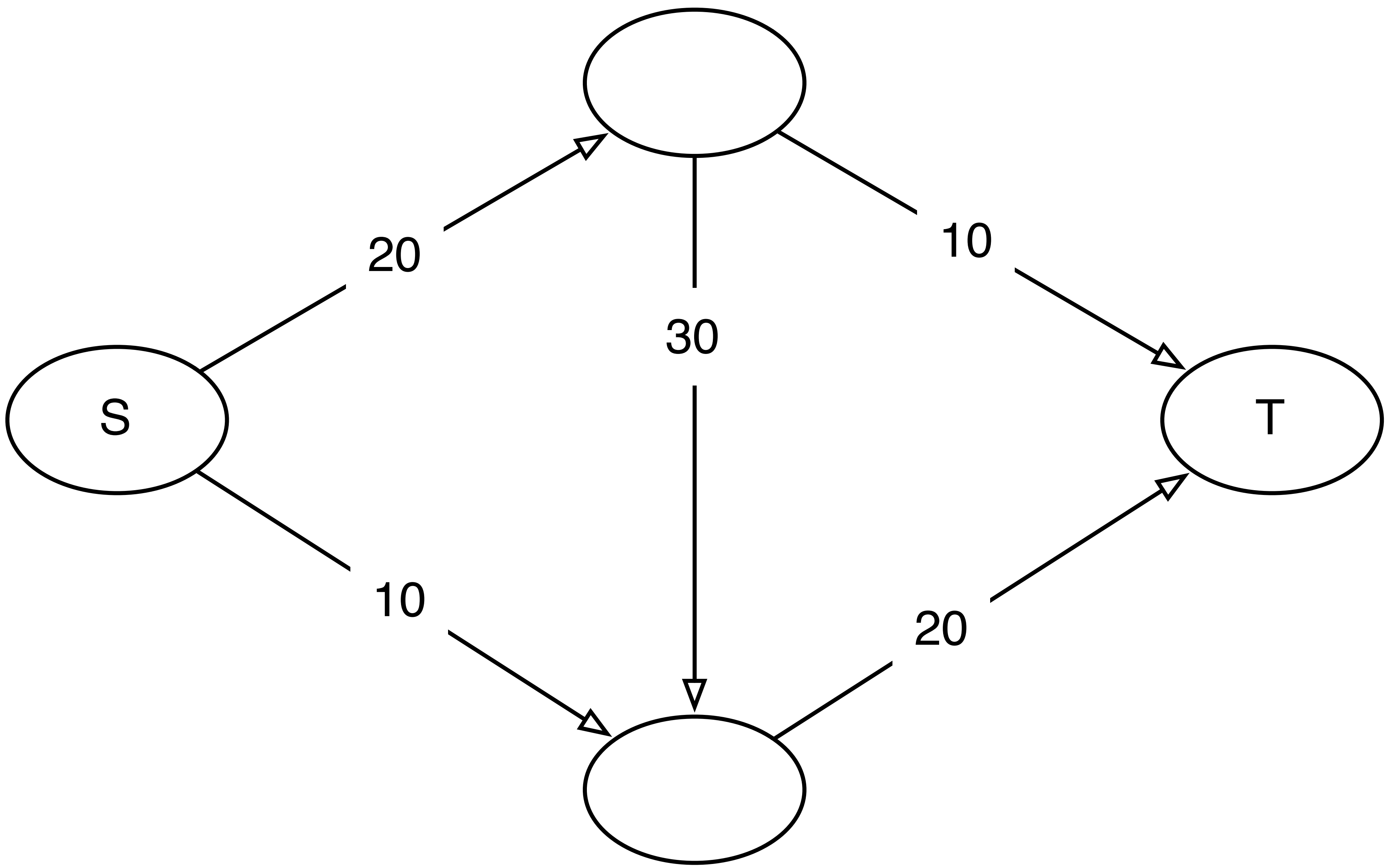
$G_f = (V, E_f)$

$$c_f(u, v) =$$

EXAMPLE RESIDUAL GRAPH



WHY RESIDUAL GRAPHS ?



AUGMENTING PATHS

DEF:

THM: MAX FLOW = MIN CUT

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

IF F IS A MAX FLOW, THEN G_F HAS NO AUGMENTING PATHS.

THM: MAX FLOW = MIN CUT

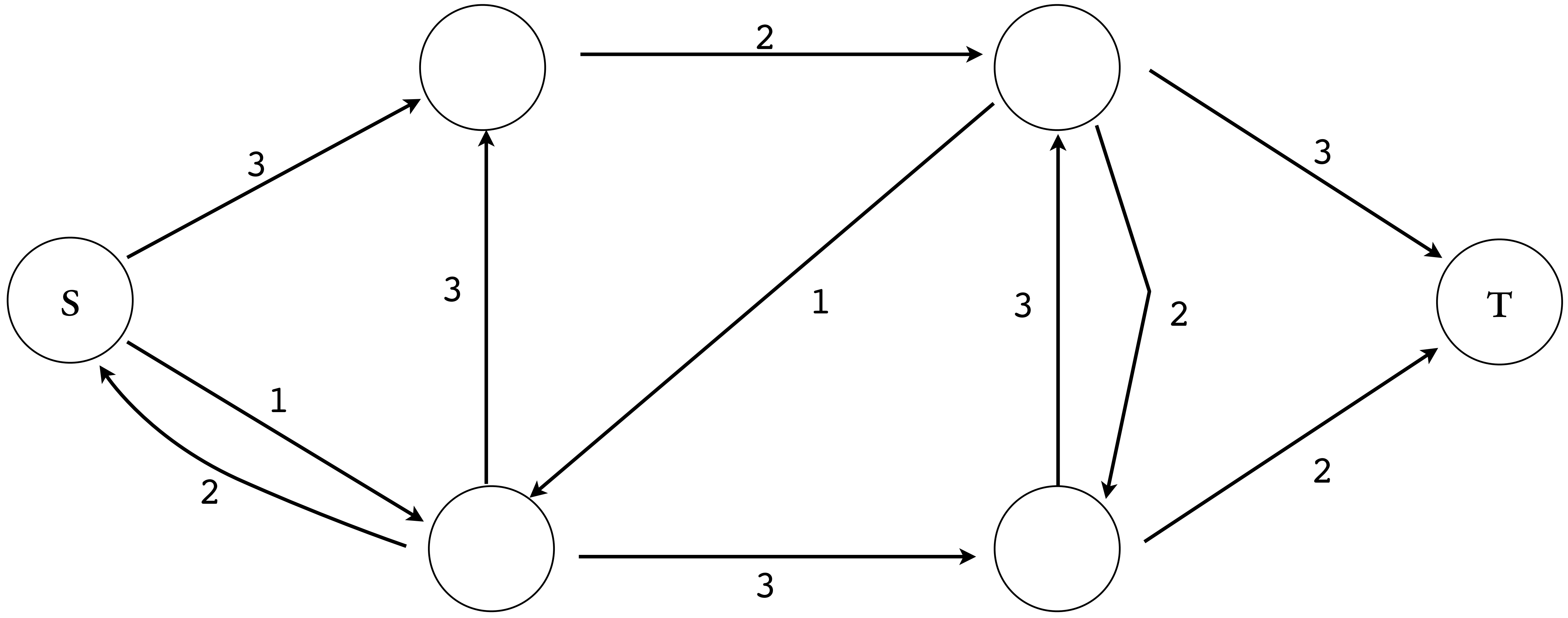
$$\max_f |f| = \min_{S,T} ||S, T||$$

FORD-FULKERSON

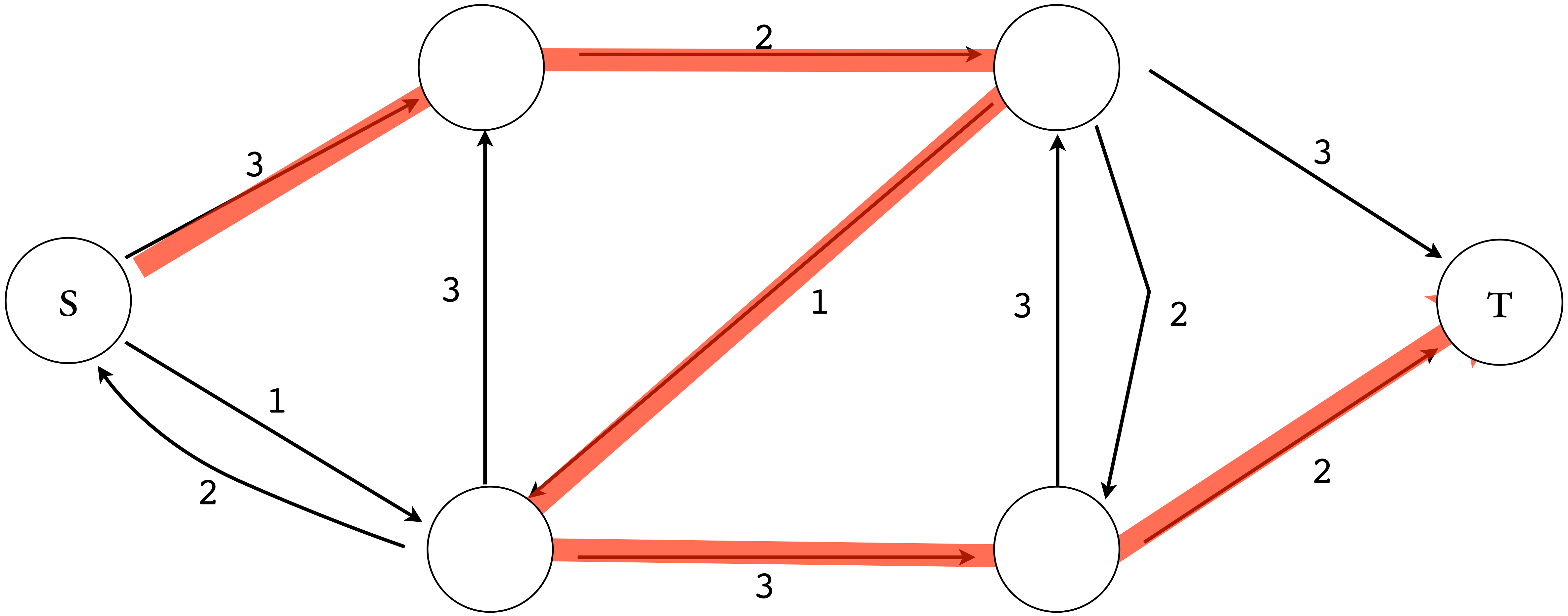
INITIALIZE $f(u,v) \leftarrow 0 \; \forall u,v$

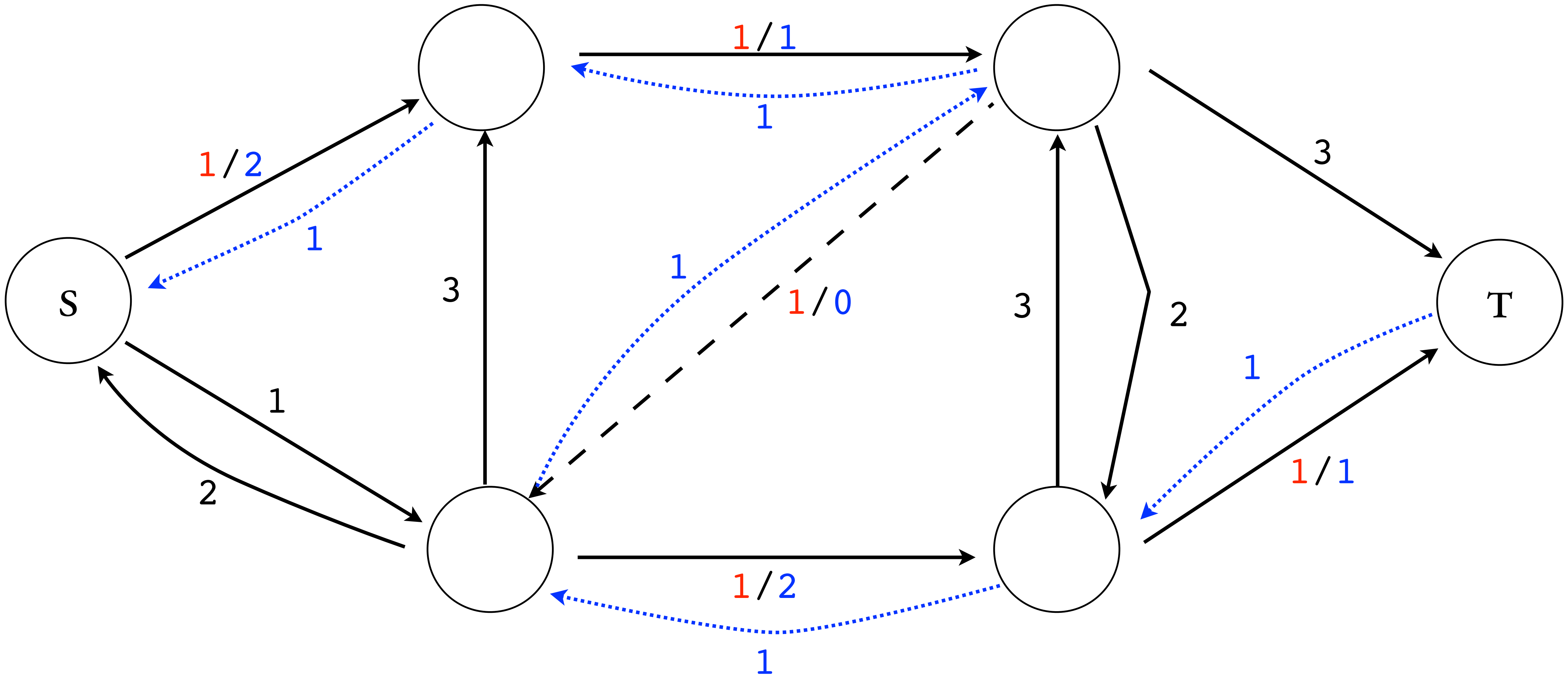
WHILE EXISTS AN AUGMENTING PATH p IN G_f

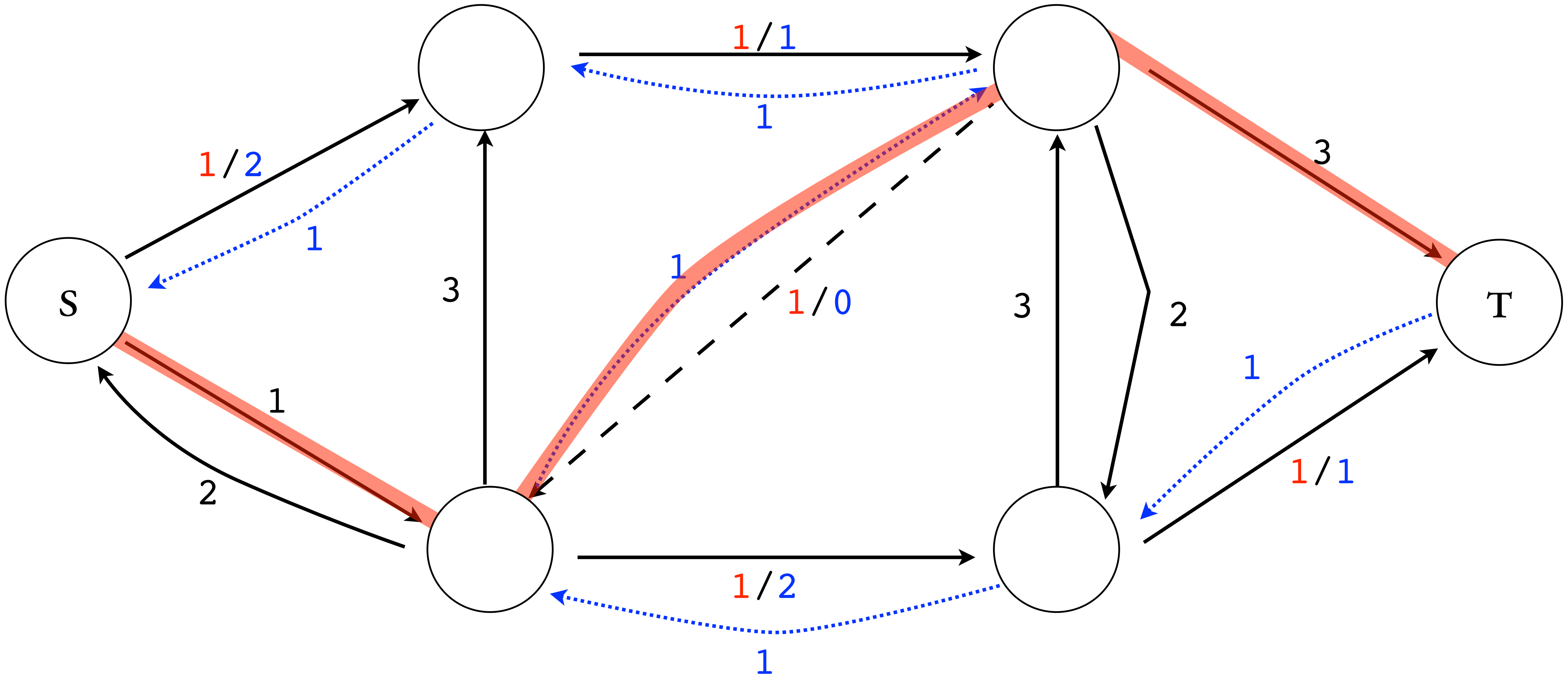
AUGMENT f WITH $c_f(p) = \min_{(u,v)\in p} c_f(u,v)$



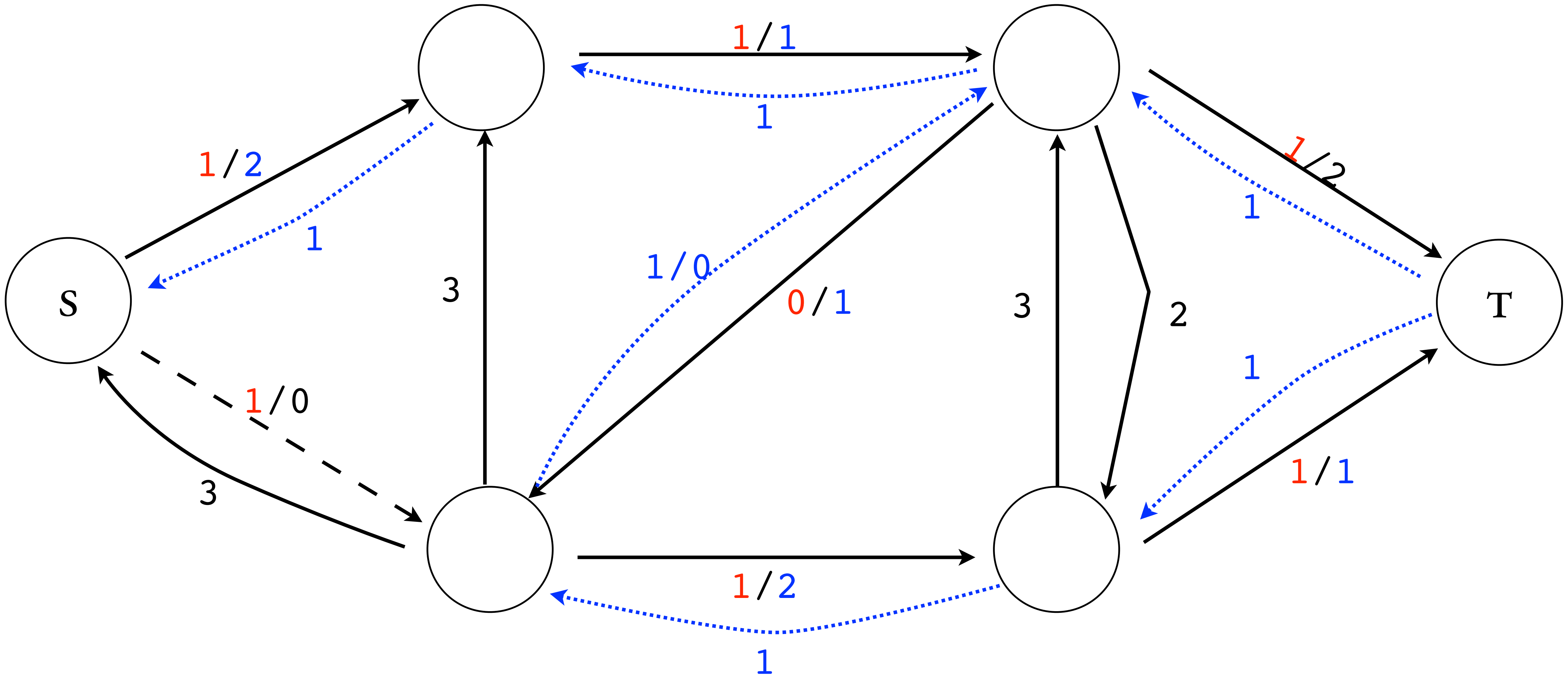
S
T
3
1
2
3
2
1
3
3
2
3
2


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

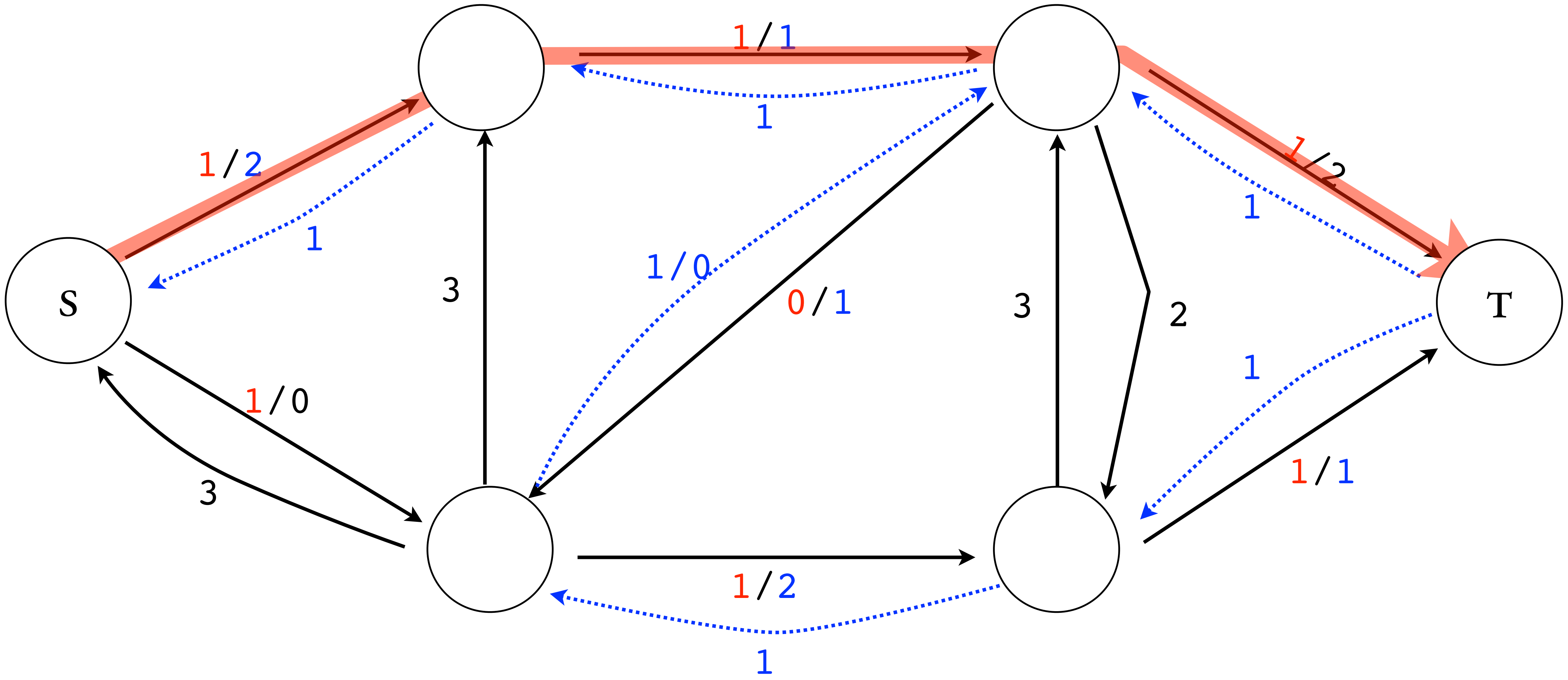

S
T
1 / 2
1
1 / 1
1
3
1
1 / 0
3
3
2
1
2
1 / 2
1
1
1 / 1
3

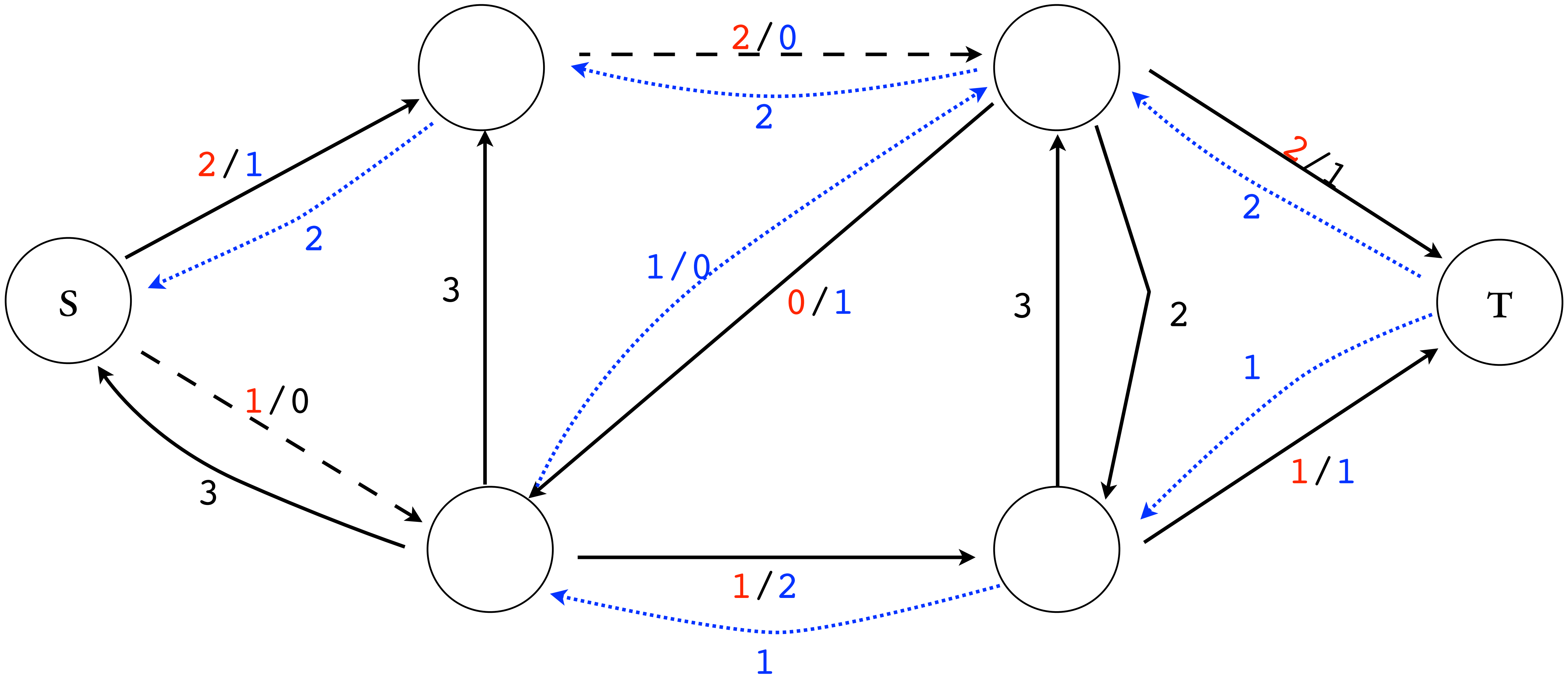


S
T
1/1
1
1/2
1
3
1
1/0
3
3
2
1
1
2
1/1
1/2
1



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


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


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

FORD-FULKERSON

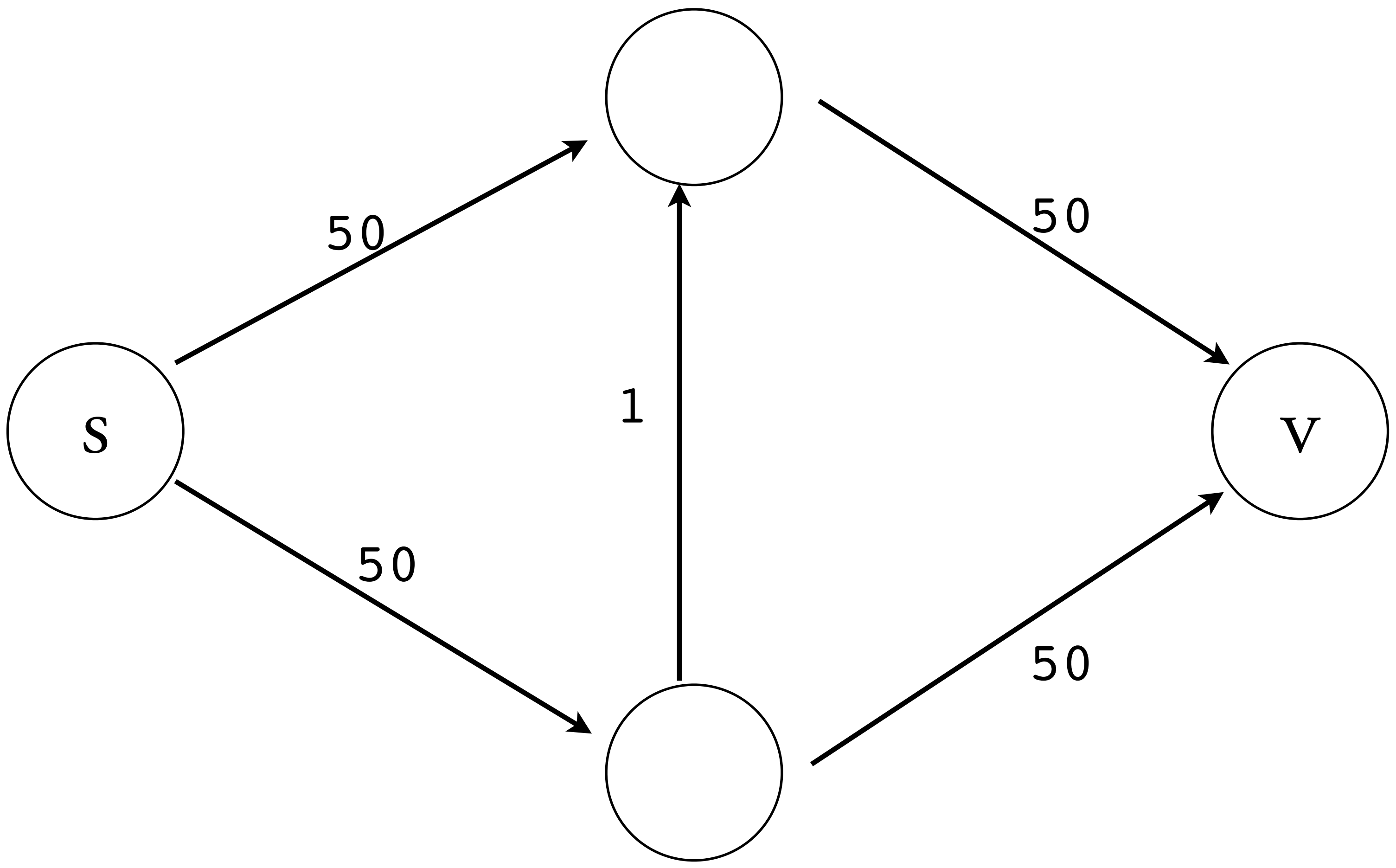
INITIALIZE $f(u,v) \leftarrow 0 \; \forall u,v$

WHILE EXISTS AN AUGMENTING PATH p IN G_f

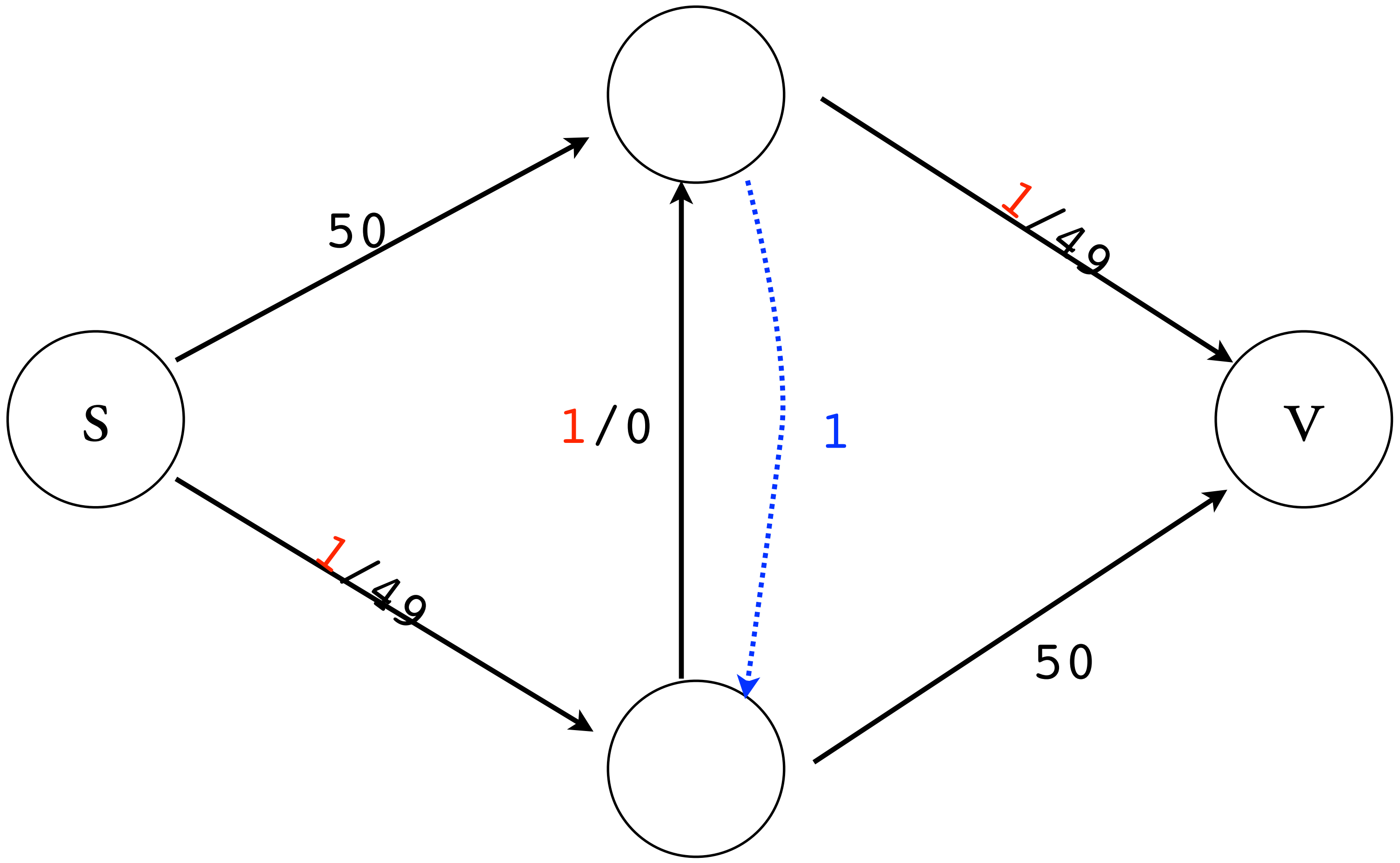
AUGMENT f WITH $c_f(p) = \min_{(u,v)\in p} c_f(u,v)$

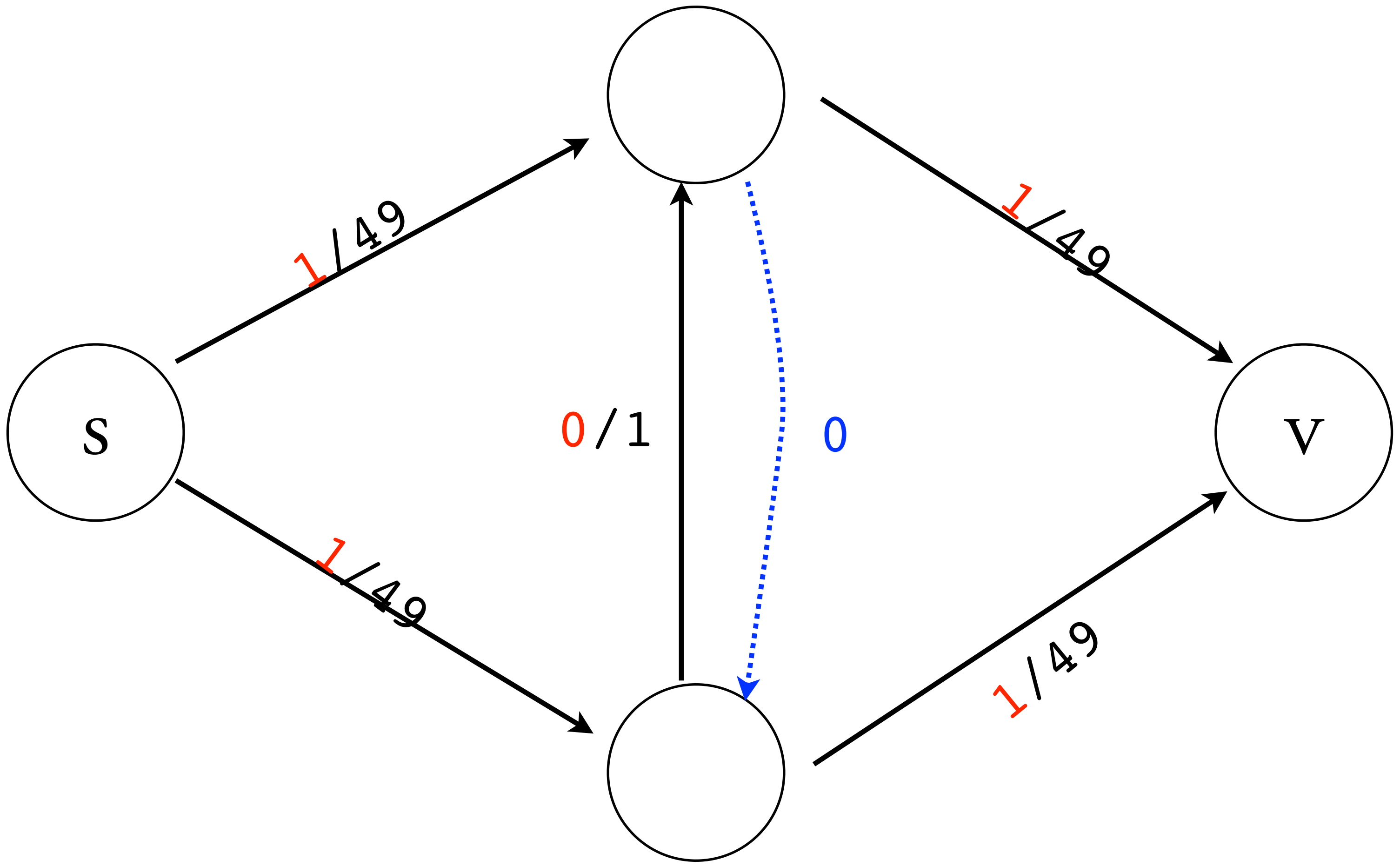
TIME TO FIND AN AUGMENTING PATH:

NUMBER OF ITERATIONS OF WHILE LOOP:

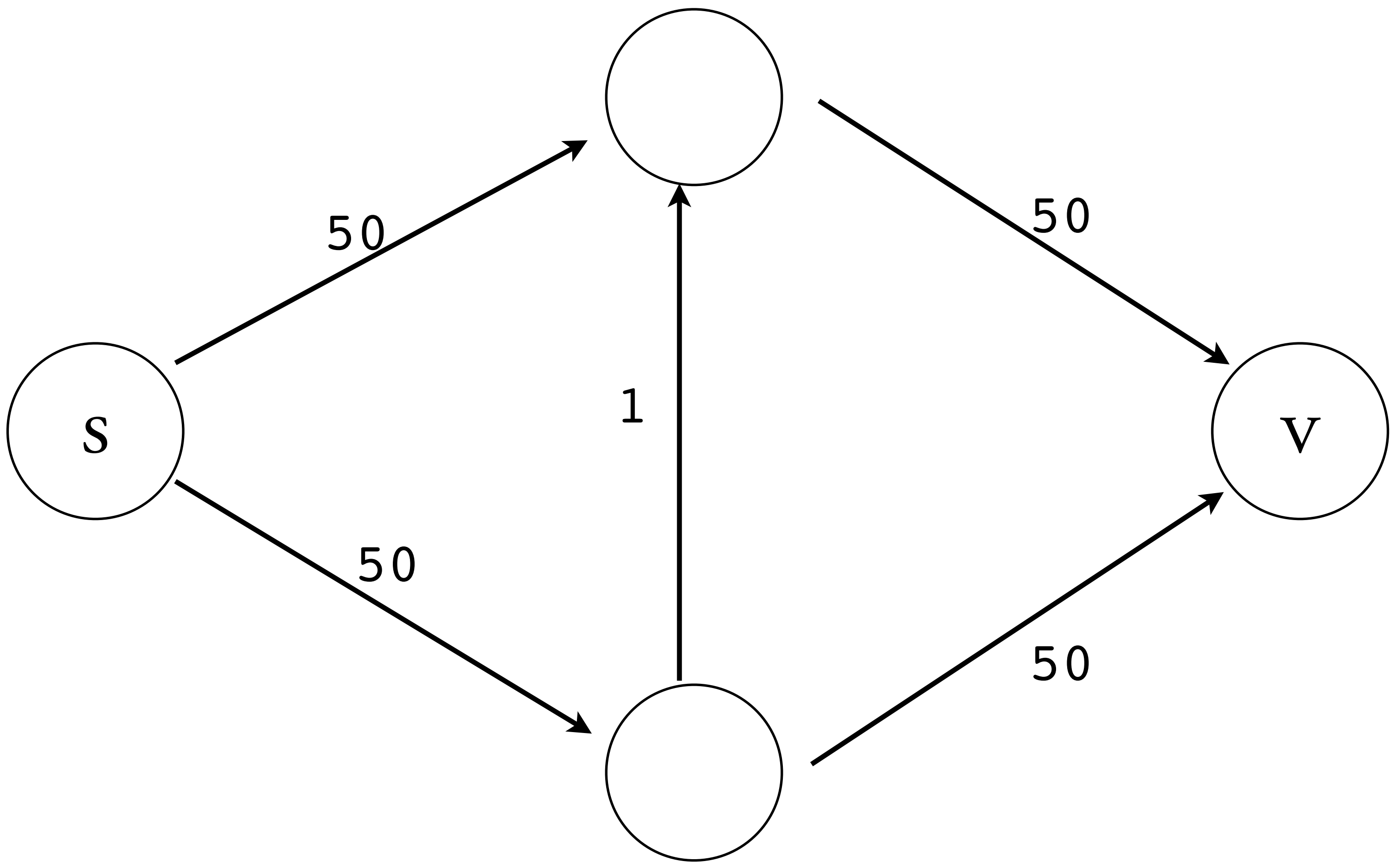


S
V
50
50
1
50
50


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


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

ROOT OF THE PROBLEM



EDMONDS-KARP 2

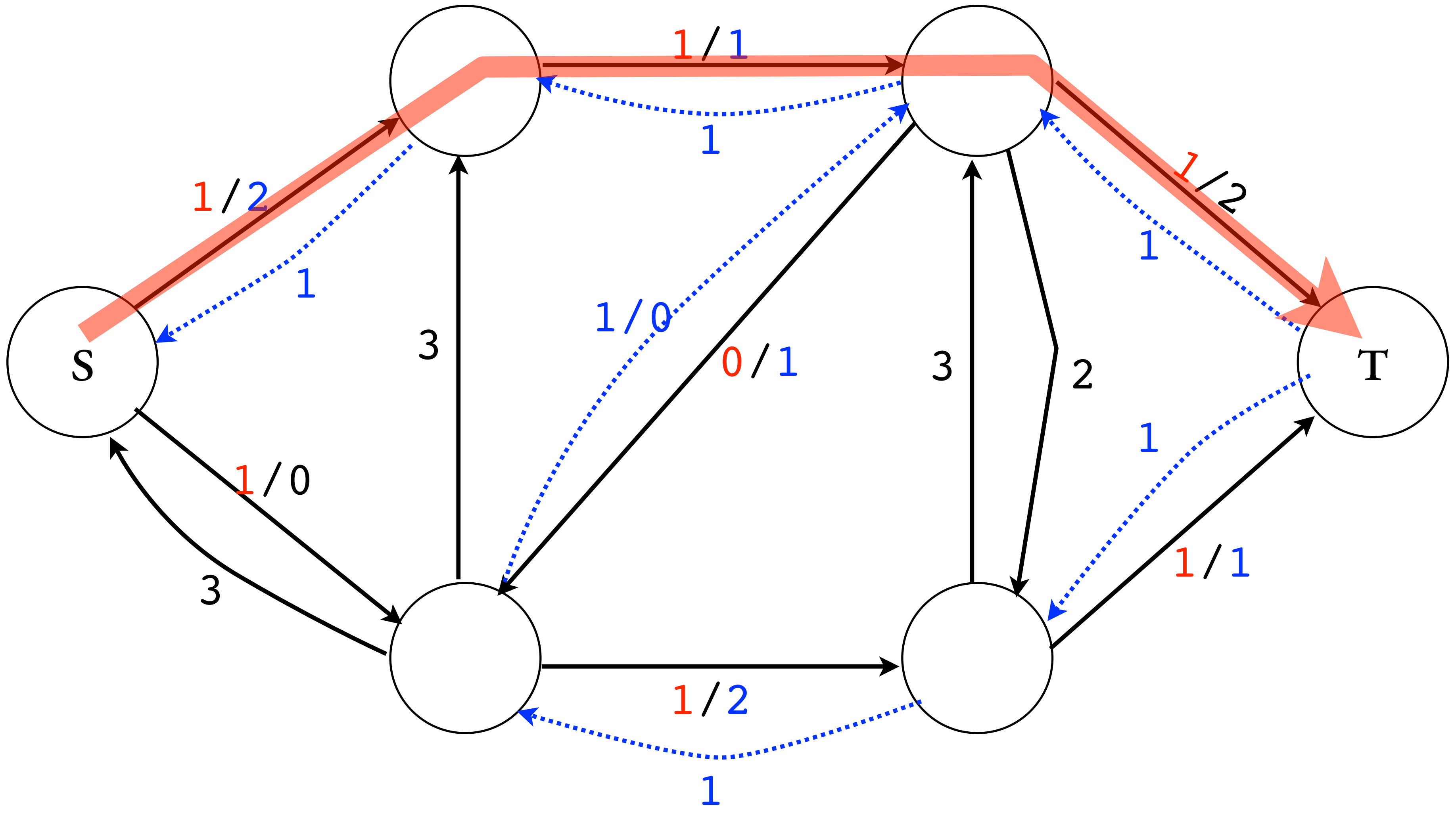
CHOOSE PATH WITH FEWEST EDGES FIRST.

$\delta_f(s, v)$:

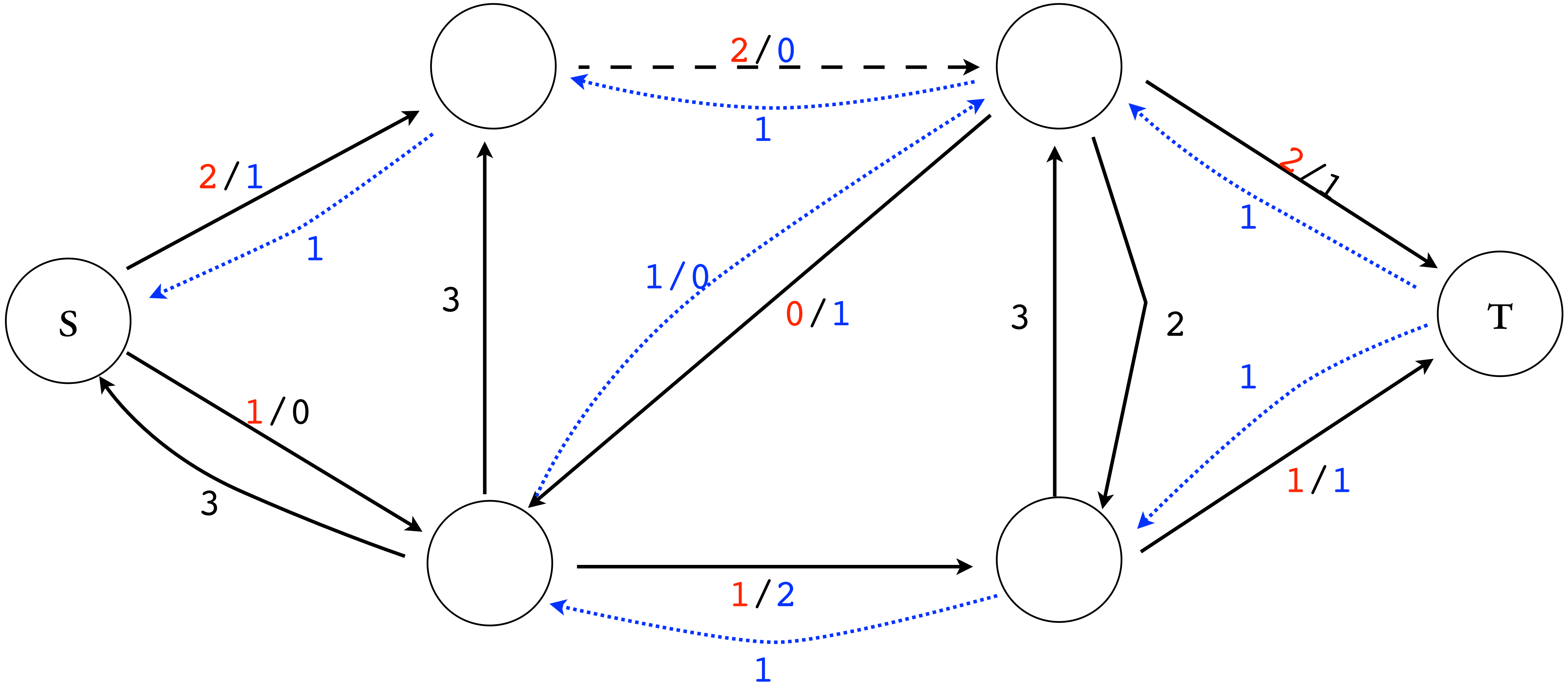
LEMMA:

$\delta_f(s, v)$ INCREASES MONOTONICALLY THRU EXEC

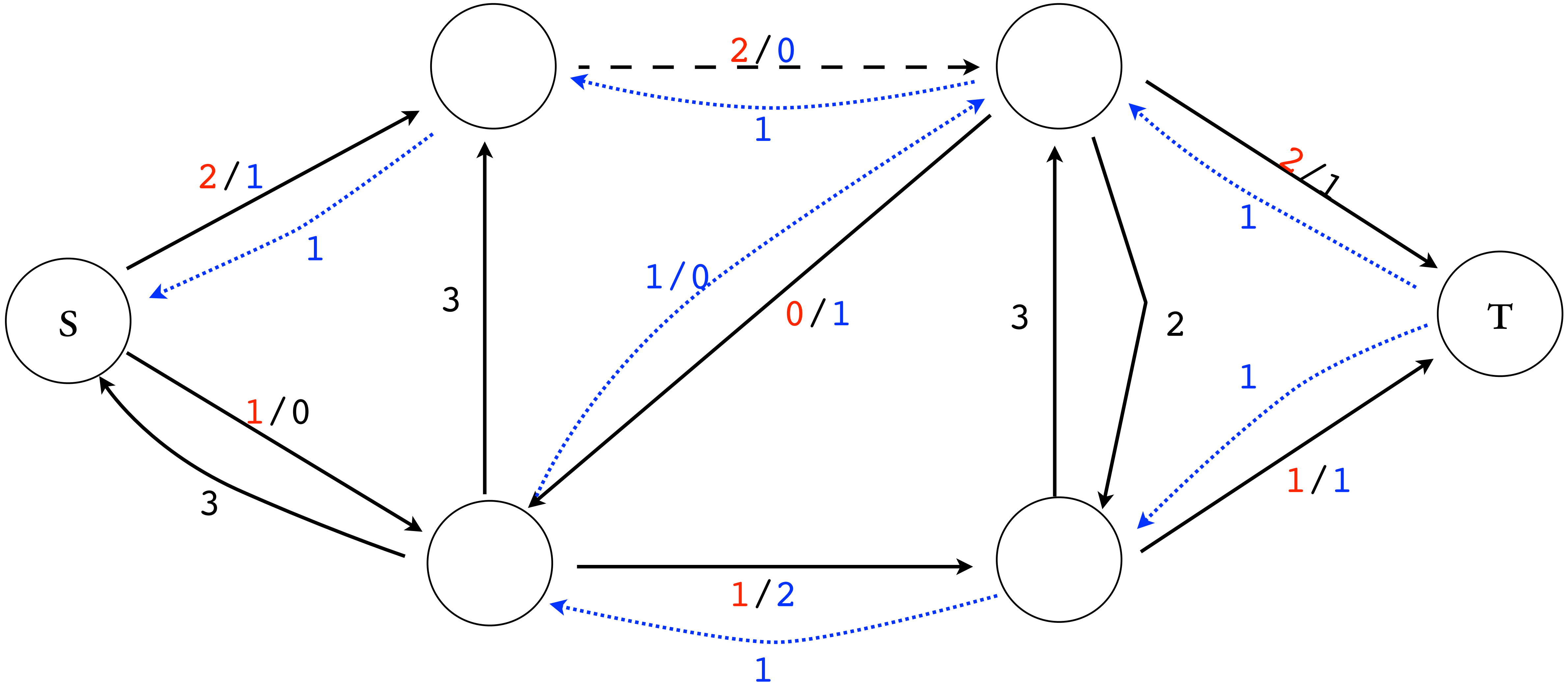
$$\delta_{i+1}(v) \geq \delta_i(v)$$



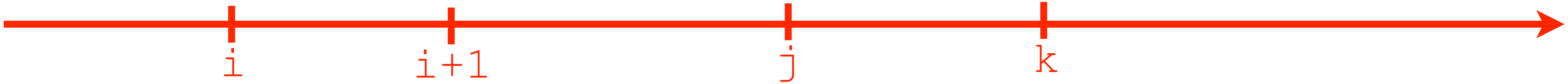
FOR EVERY AUGMENTING PATH, SOME EDGE IS CRITICAL.



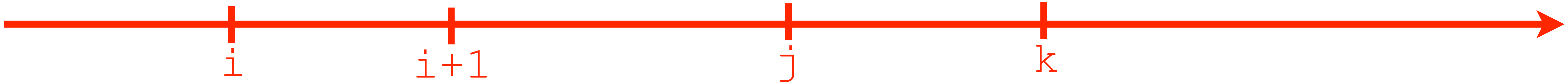
CRITICAL EDGES ARE REMOVED IN NEXT RESIDUAL GRAPH.



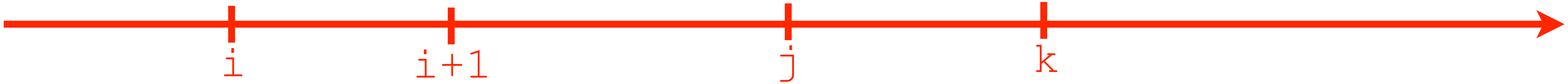
KEY IDEA: HOW MANY TIMES CAN AN EDGE BE CRITICAL?



Outline of the argument



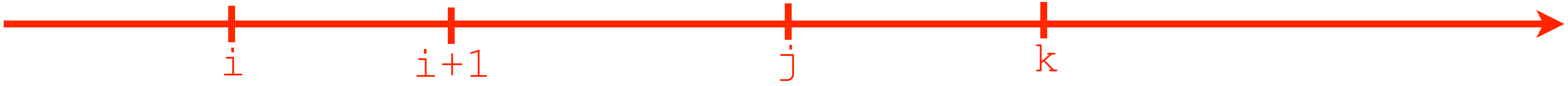
first time (u,v) is critical:



time i: (u,v) is critical:

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

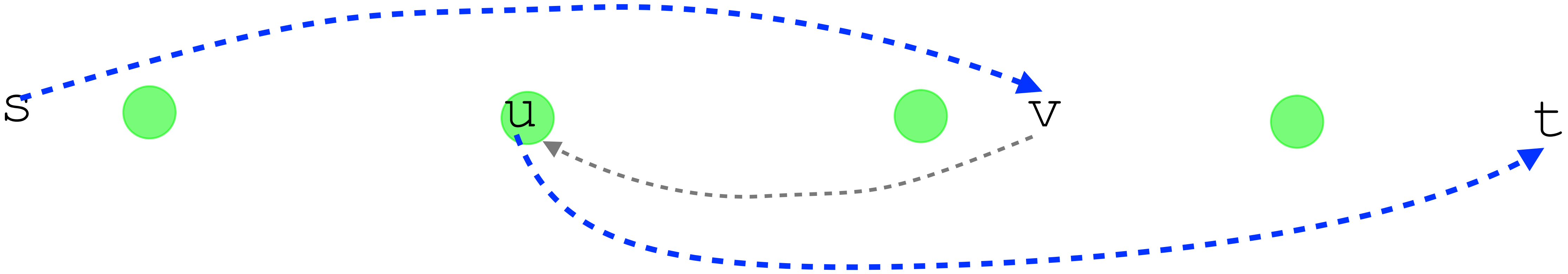
time j: Edge (u,v) STRIKES BACK



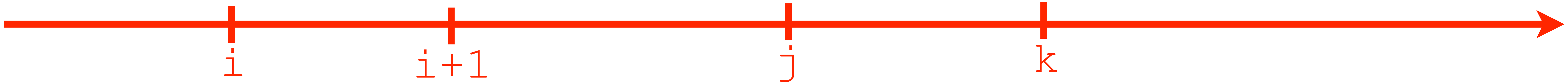
time i: (u,v) is critical:

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

time j: Edge (u,v) STRIKES BACK



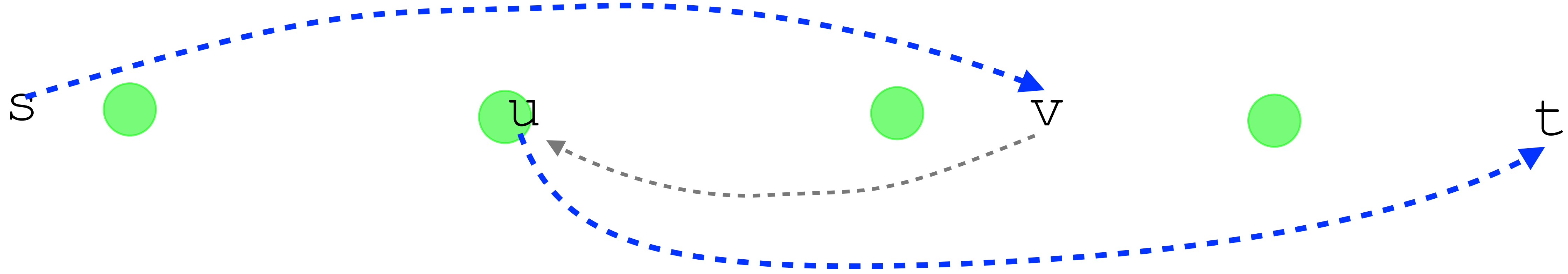
$$\delta_j(s,u) = \delta_j(s,v) + 1$$

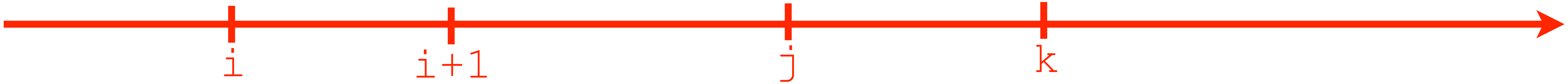


time j: Edge (u,v) STRIKES BACK

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

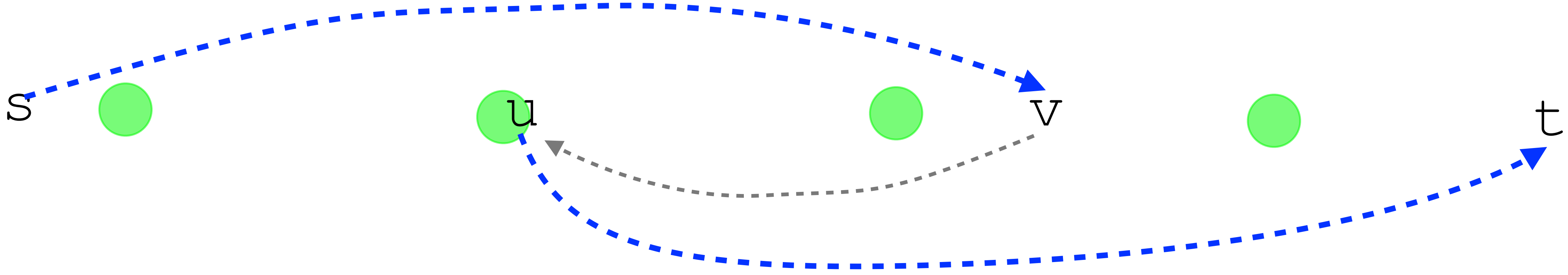
$$\delta_j(s,u) = \delta_j(s,v) + 1$$





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	