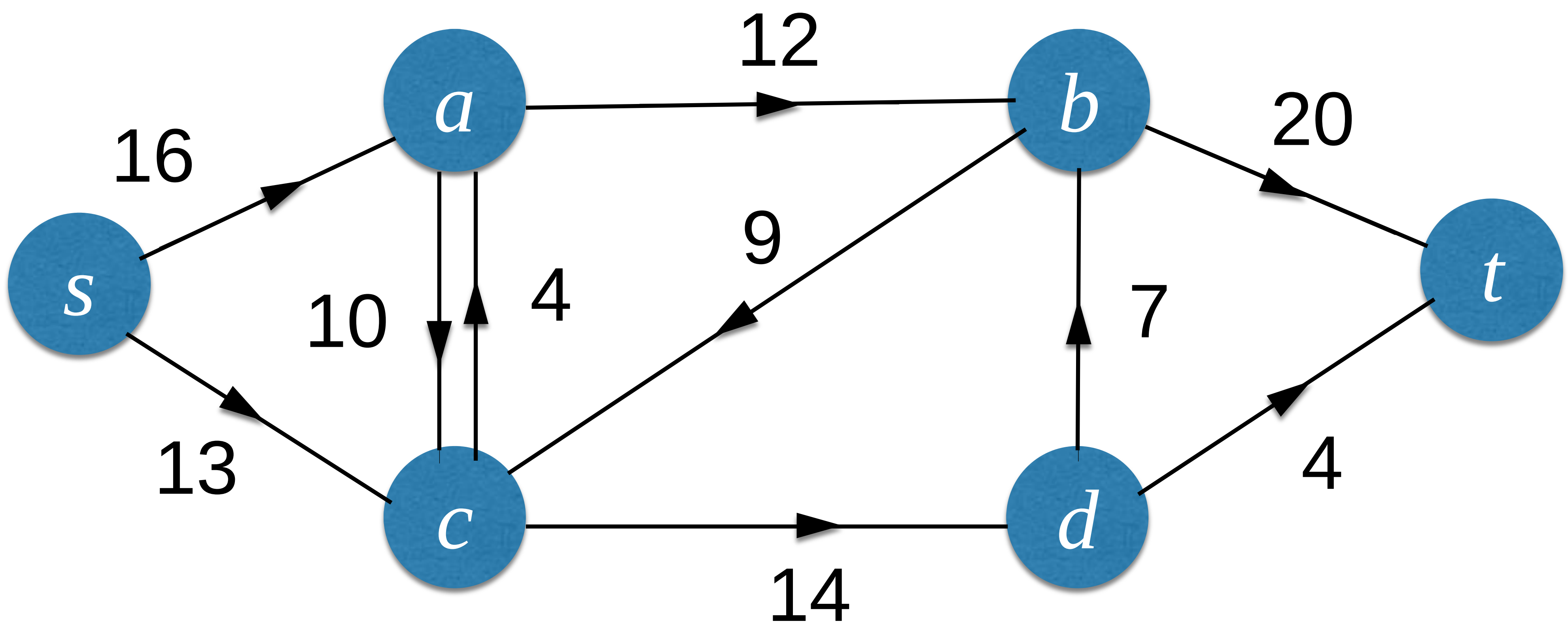
CSC 226

Algorithms and Data Structures: II

Rich Little
rlittle@uvic.ca
ECS 516

Network Flow

Example of an *st*-flow network



Network Flow (Definitions)

- A ***flow network*** is an edge-weighted, directed graph with positive edge weights, called ***capacities*** (capacities of non-existing edges are zero)

- An ***st-flow network*** is a flow network that has two identified vertices, namely the ***source*** ***s*** and the ***sink*** ***t***

- An ***st-flow*** in an *st*-flow network is a set of nonnegative values (*edge flows*) associated with each edge. Furthermore, we define

  - ***inflow***: total flow of edges into a specific vertex

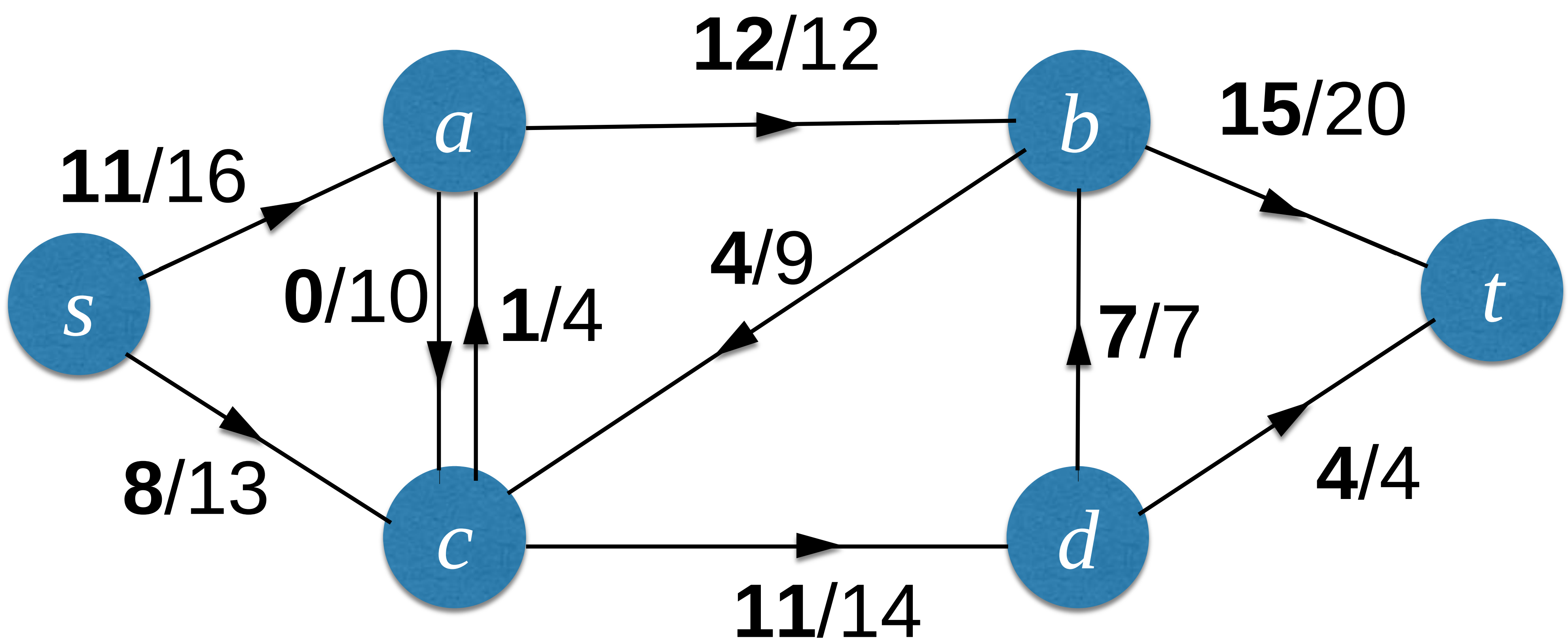
  - ***outflow***: total flow of edges from a specific vertex

  - ***netflow***: inflow minus outflow of a specific vertex

Flow Network (Definitions)

- An st-flow is *feasible* if it satisfies the conditions that
  - no edge's flow is greater than that edge's capacity and
  - the netflow of every vertex v (except s and t) in the st-flow network is zero: *inflow(v) = outflow(v)*
- st-flow ***value*** $|f|$ for st-flow network N with st-flow f : the sink's inflow (or the source's outflow)
- ***Maximum*** *st-flow* (or ***maxflow***): a feasible st-flow with maximum st-flow value over all feasible flows

Example of a feasible *st*-flow in an *st*-flow network



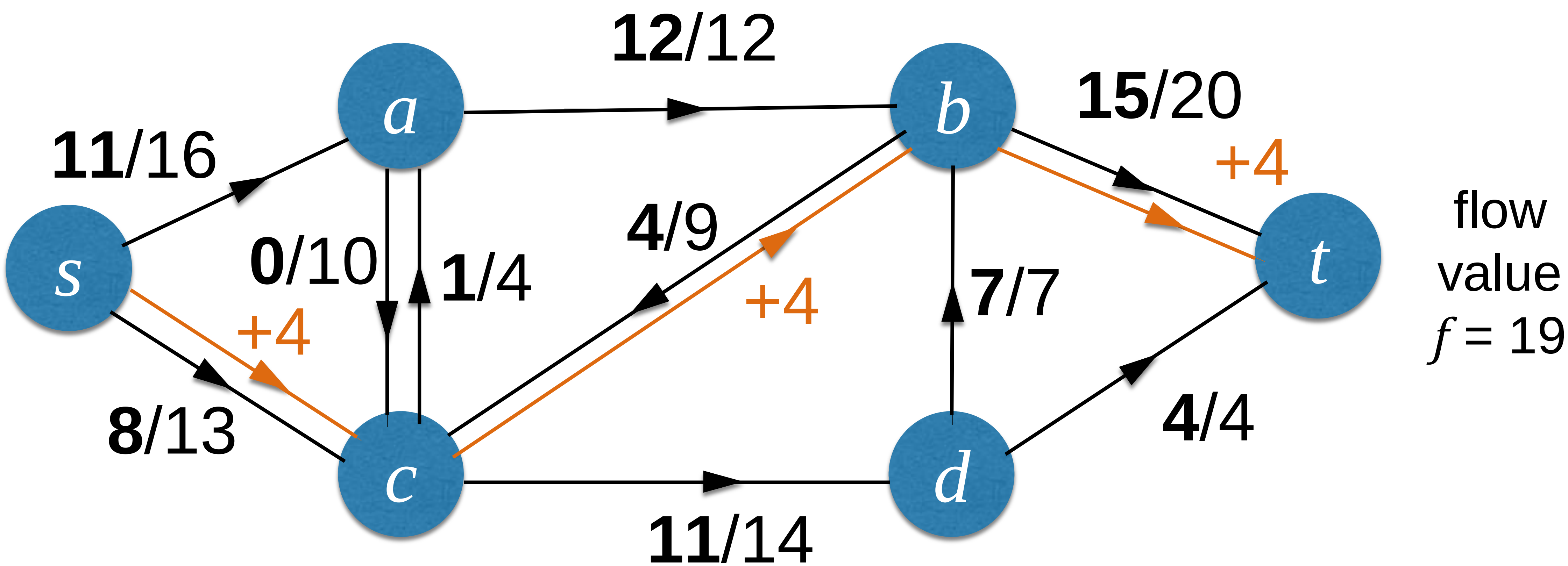
Maximum Flow Problem: *maxflow*

- Input: An *st*-flow network
- Output: A maximum *st*-flow

Key idea: Augmenting paths in *st*-flow networks

- An *augmenting path* in an *st*-flow network with feasible *st*-flow is an undirected path from source *s* to sink *t* along which we can push more flow, obtaining an *st*-flow with higher *st*-flow value.

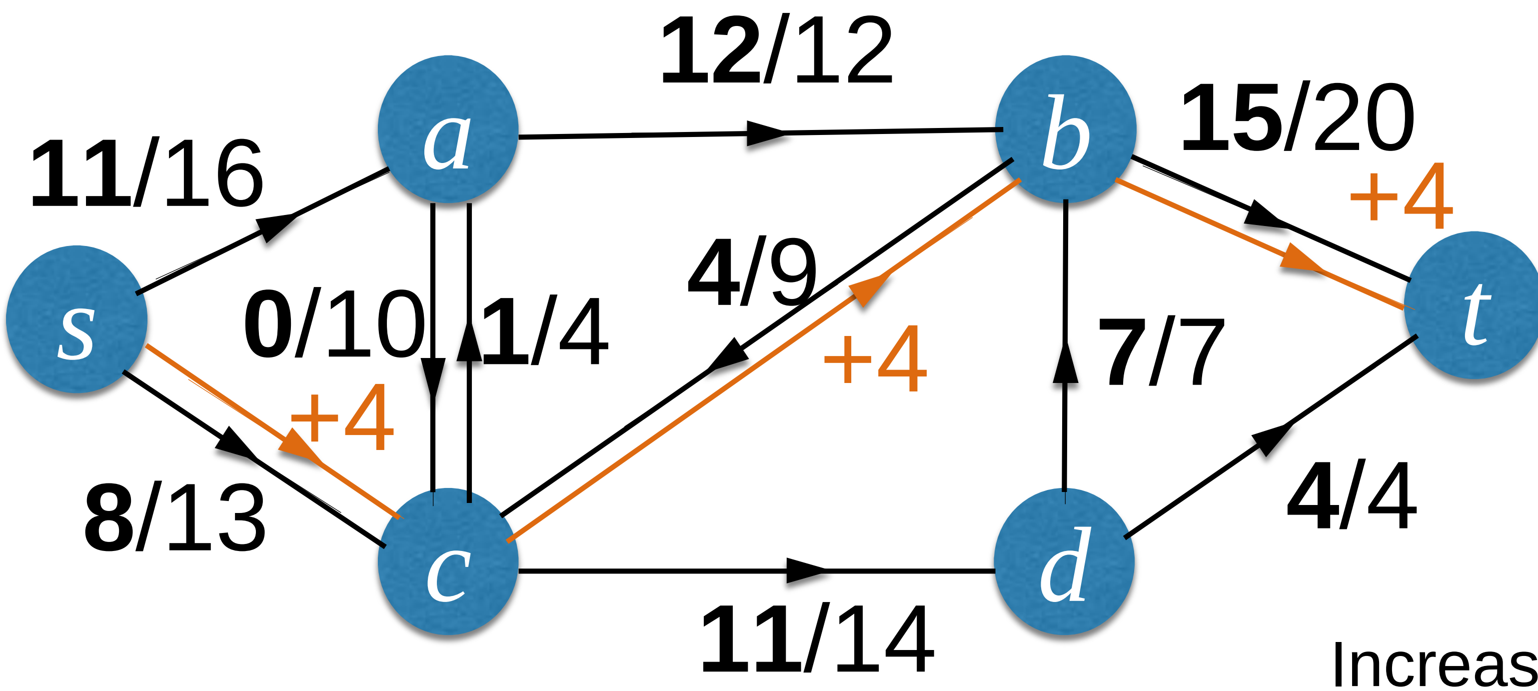
Example of an augmenting path that improves the flow: $scbt$



Arc bc is a *backward* arc on the path $scbt$.

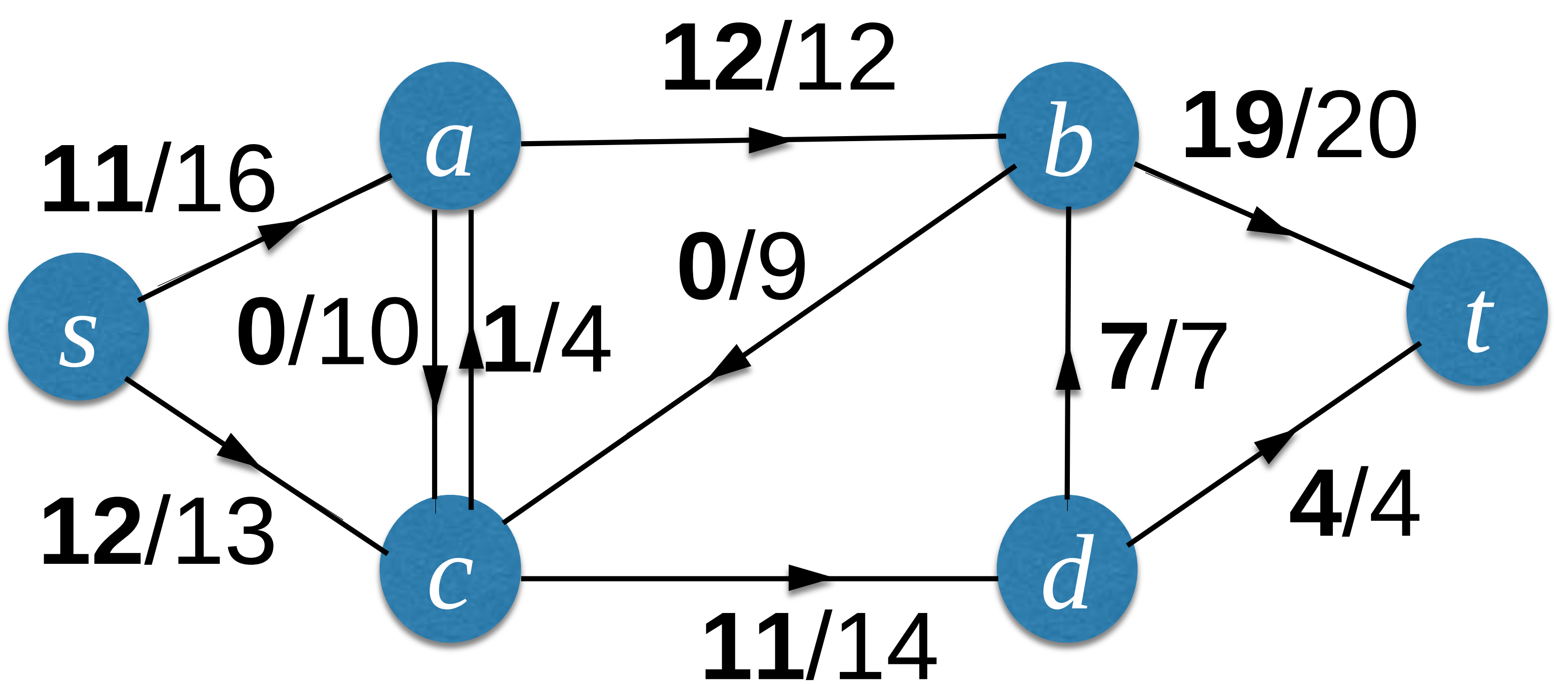
bottleneck capacity = min{(13-8), 4, (20-15)} = +4

Improved flow: can increase by +4



old flow
value
f = 19

Increase the flow by 4 in each forward arc and decrease the flow by 4 in each backward arc.



new flow
value
f = 23

Ford-Fulkerson's maxflow method

1. Initialize with a 0 flow: st-flow value $f = 0$

2. Increase the flow along any augmenting path from s to t

3. Repeat step 2 as long as an augmenting path exists

Finding Augmenting Paths: the residual network G_f of a flow f

- Consider an st-flow f in st-flow network G and a directed edge (u,v) in G

- The amount of additional flow we can push from u to v along (u,v) in G is called the *residual capacity* $c_f(u,v)$ of edge (u,v) -- it depends on f.

- That is: for edge (u,v) with capacity $c(u,v)$ and flow value $f(u,v)$ from u to v we have the residual capacity $c_f(u,v) = c(u,v) - f(u,v)$; this creates a directed edge (u,v) in the residual network G_f with capacity $c_f(u,v)$.

- Of course in G we could instead *reduce* the flow in (u,v) by as much as $f(u,v)$; this creates a directed edge (v,u) in the residual network G_f with capacity

$c_f(v,u) = f(u,v)$. (Note the order of the vertices.)

Residual Network

- Given an st-flow network $G = (V,E)$ and a flow f, the *residual network* of G *induced by* f is $G_f = (V, E_f)$ where $E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$

Question:

Consider an edge (u,v) in G. How many edges does (u,v) create in the residual network G_f?

A. 1 edge: (u,v)
B. 2 edges: (u,v) and (v,u)
C. Can't tell – it depends on the flow f, which can change

Question:

Consider an edge (u,v) in G , where

capacity $c(u,v) = 5$

and

flow $f(u,v) = 5$.

How many edges does (u,v) create in the residual network G_f?

A. 0
B. 1
C. 2

Question:

Consider an edge (u,v) in G , where

capacity $c(u,v) = 5$

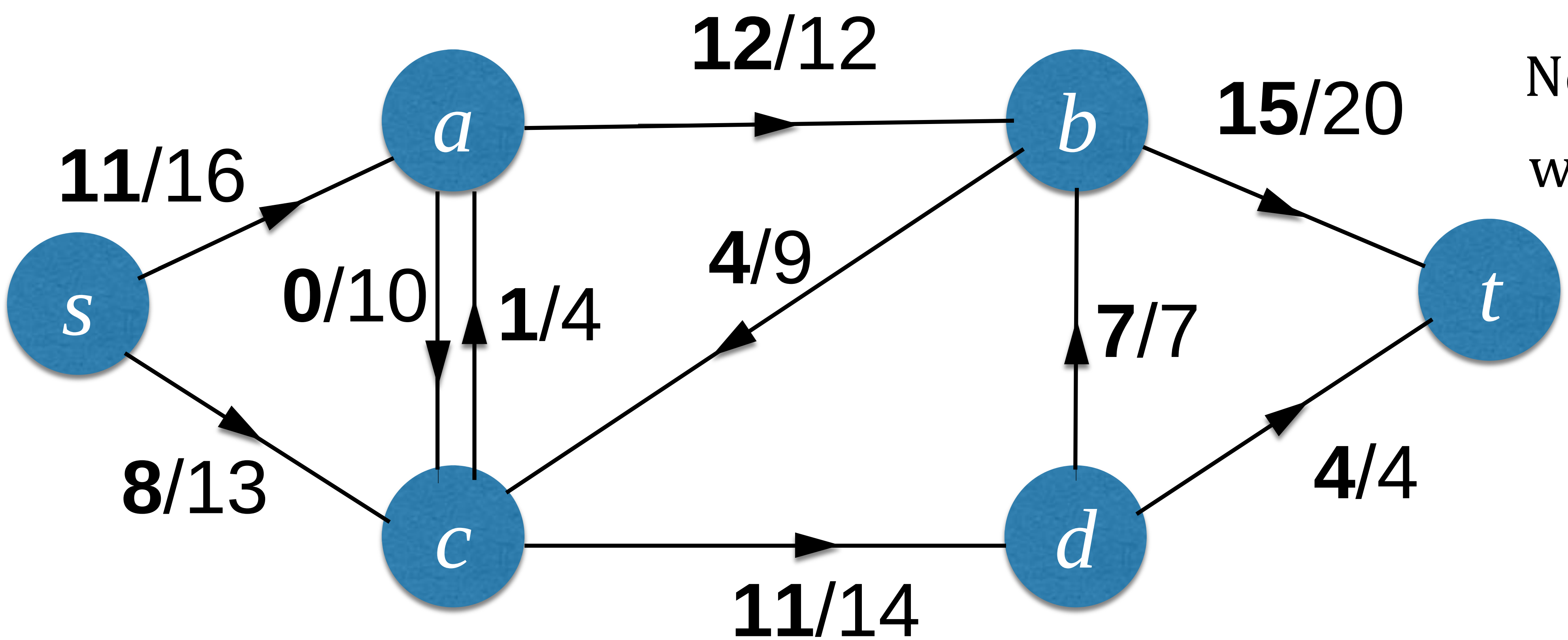
and

flow $f(u,v) = 3$.

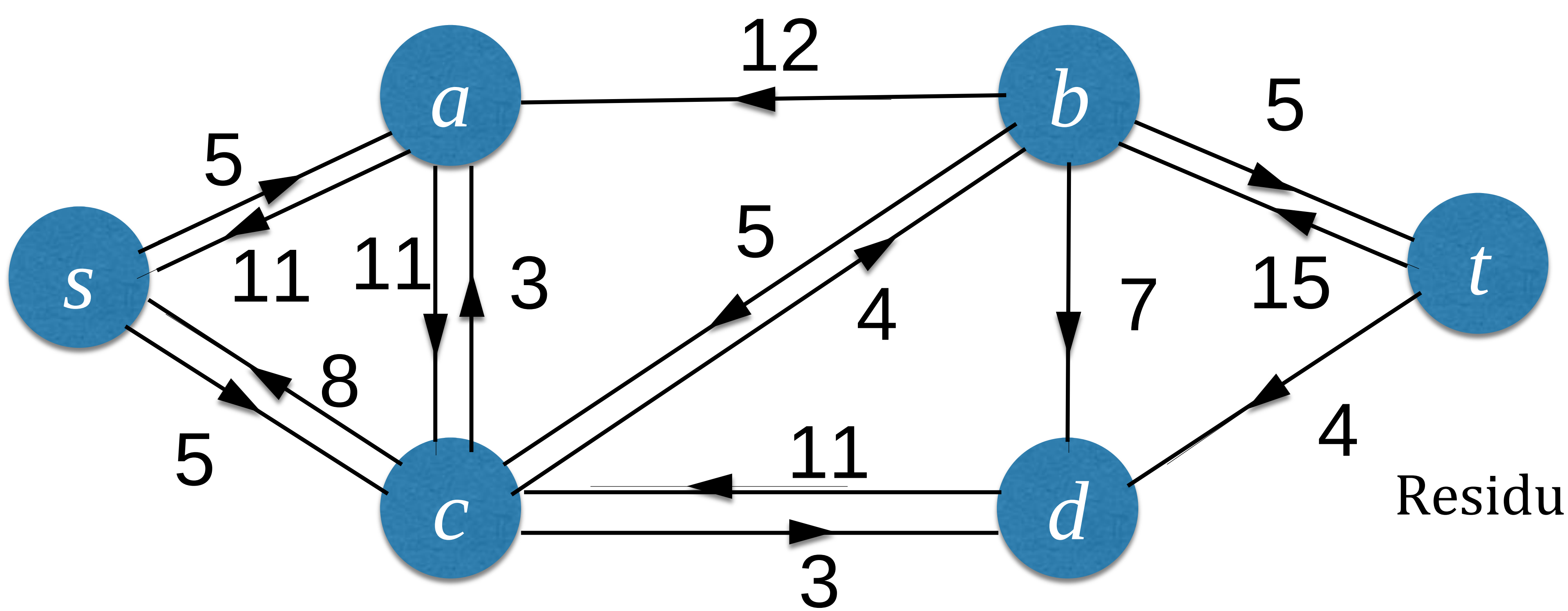
How many edges does (u,v) create in the residual network G_f?

A. 0
B. 1
C. 2

Example of a residual network



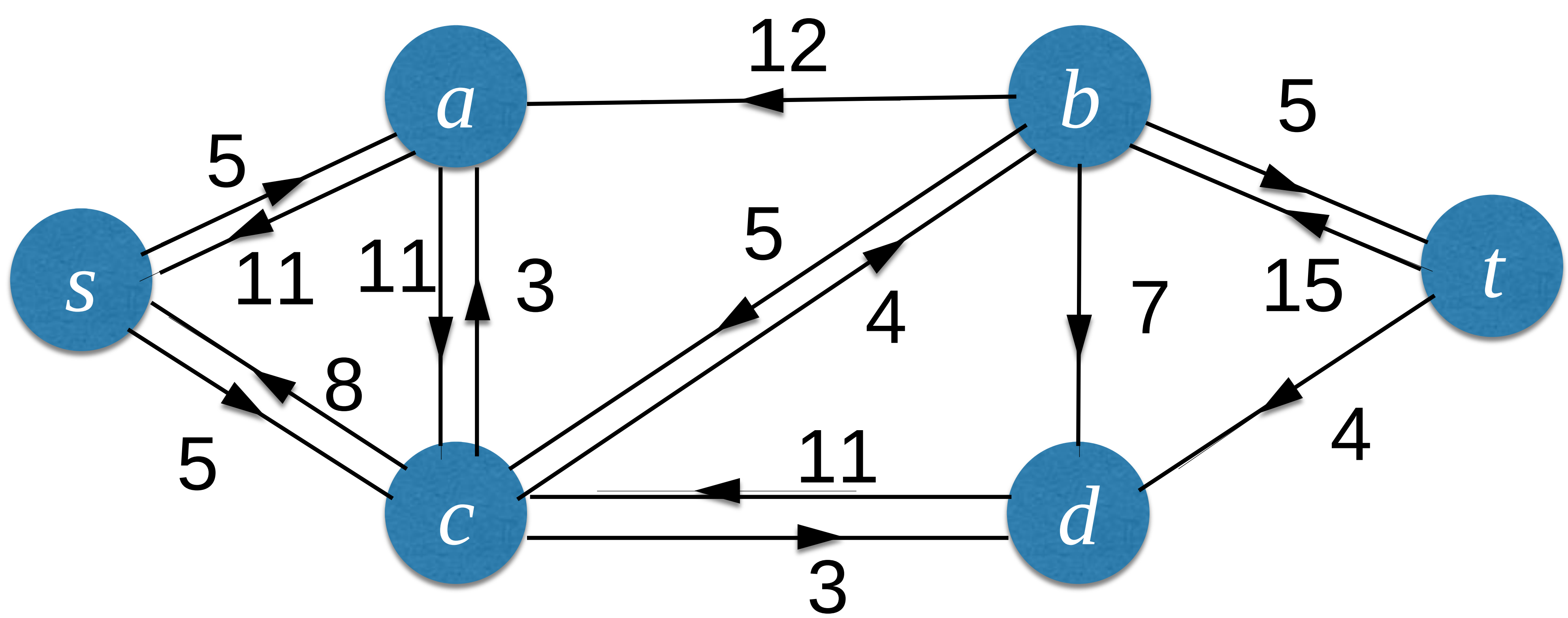
Network G with flow f



Residual network G_f

Augmenting path from s to t in residual network G_f : $scbt$

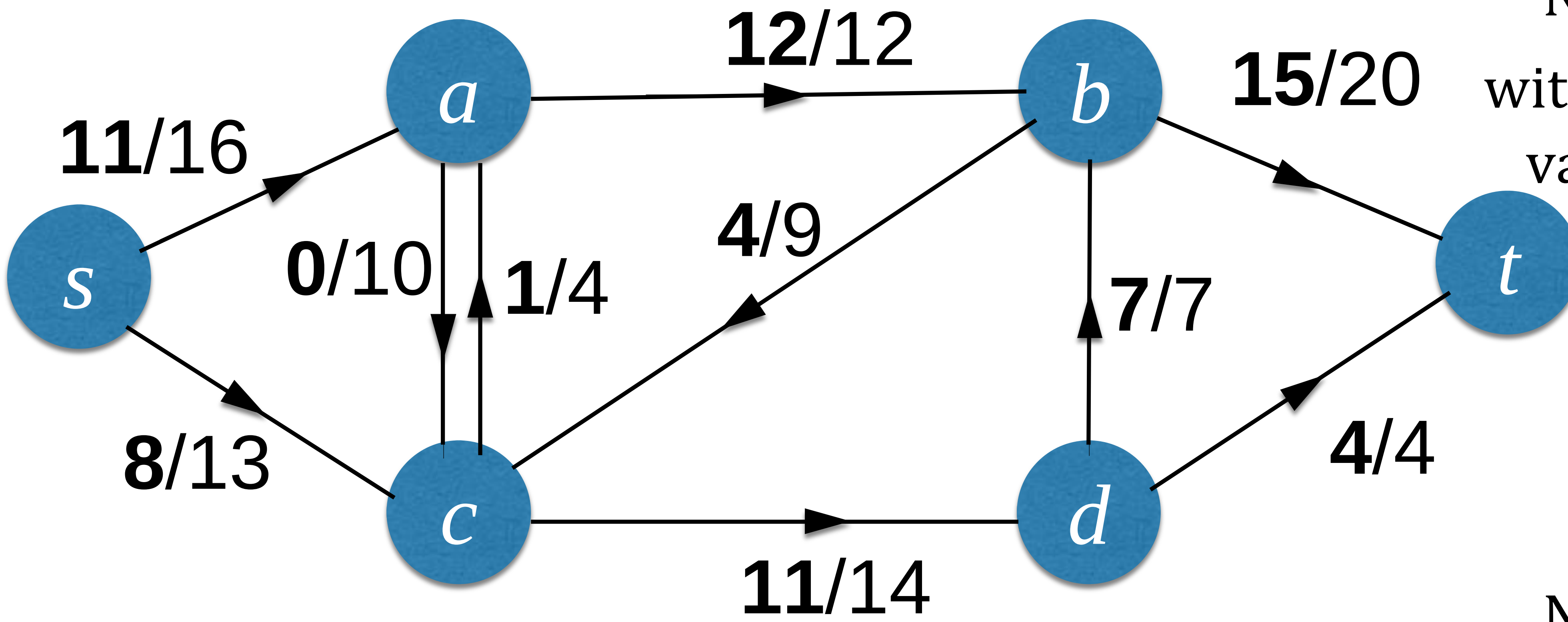
Residual network G_f



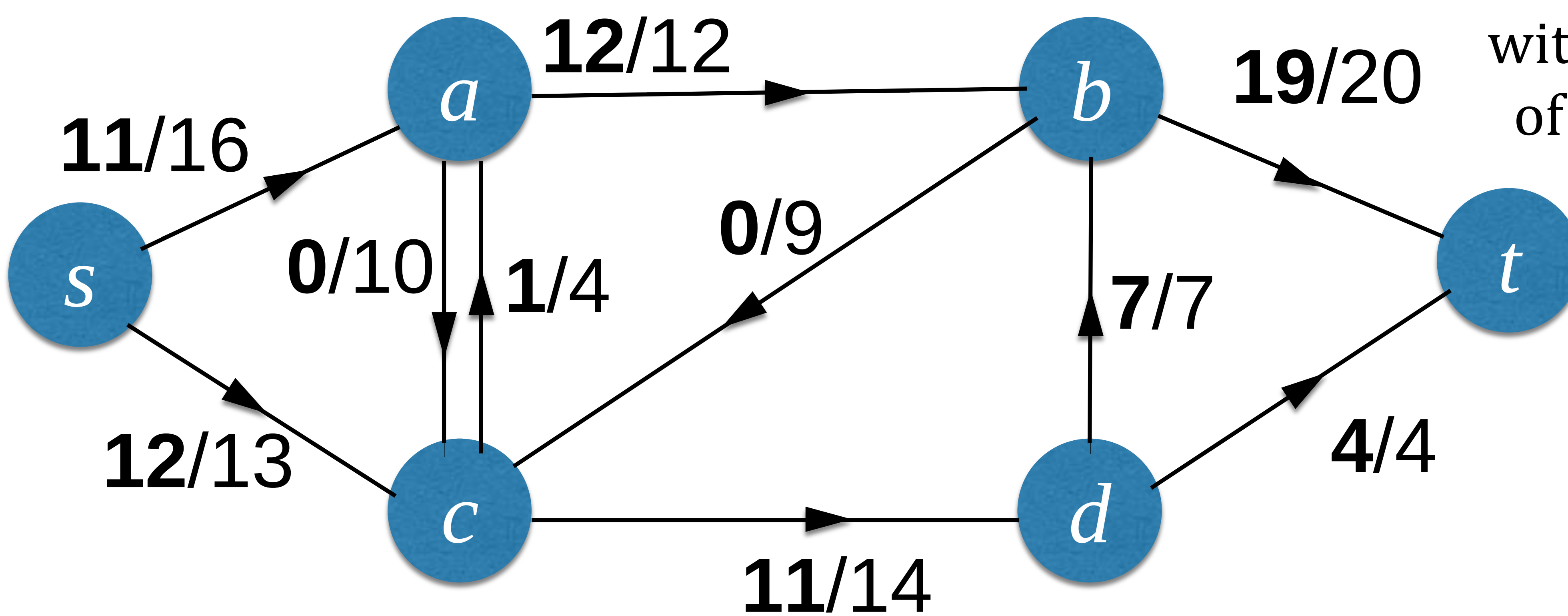
Residual capacity of path $scbt$ is $min\{5,4,5\} = 4$

In G , increase flow in each forward arc by 4, decrease flow in each backward arc by 4 to get new flow

Augment the flow along path *scbt* by 4

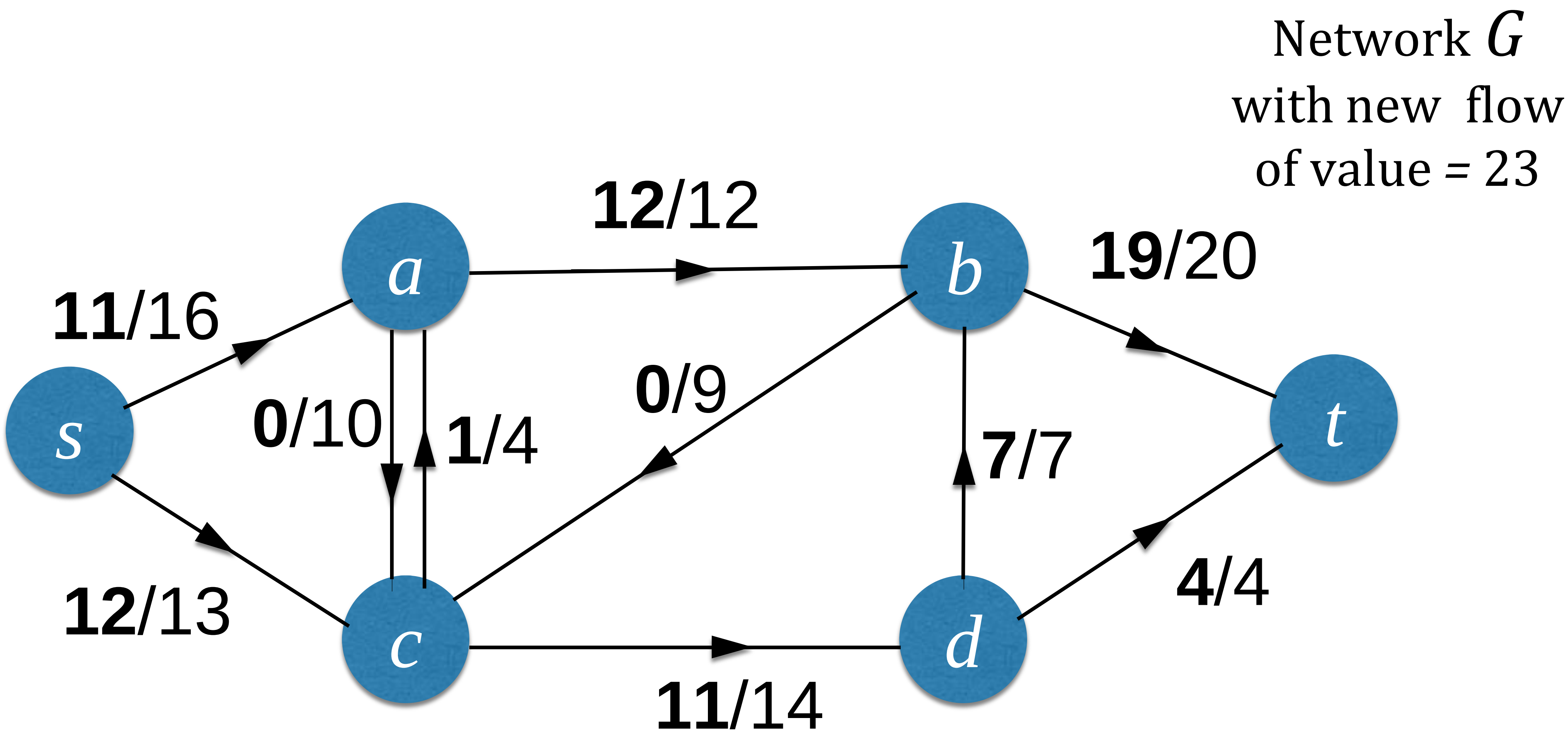


Network G
with old flow f;
value(f) = 19



Network G
with new flow
of value = 23

Claim: the new flow is a maxflow.



How would you prove this claim?
(two ways)

st-cuts

- Recall: A *cut* in a (directed) graph is a partition of the vertices into two disjoint subsets.
- The *cut edges* of a graph with a cut are the edges that have one endpoint in each subset of the partition.
- An st-*cut* is a cut that places vertex s in one of its subsets and vertex t in the other.

st-cuts (continued)

- *Capacity* of an *st*-cut in an *st*-network: sum of the capacities of the cut's edges from the subset containing s to the subset containing t

- *Flow across* an *st*-cut in an *st*-network: difference between the sum of the flows of cut's edges from the subset containing s to the subset containing t and the sum of the flows of cut's edges from the subset containing t to the subset containing s

minimum *st*-cut problem (or *mincut* problem)

- Given an *st*-network, find an *st*-cut such that the capacity of no other cut is smaller.

Properties of feasible *st*-flows in *st*-flow networks

1. For any *st*-flow, the flow across each *st*-cut is equal to the value of the flow

2. The outflow from s is equal to the inflow to t

3. No *st*-flow's value can exceed the capacity of any *st*-cut

4. Let f be an *st*-flow and let (S,T) be an *st*-cut whose capacity equals $|f|$. Then f is a maximum flow and (S,T) is a minimum cut.

Maxflow-Mincut Theorem

- Let f be an st-flow. The following three conditions are equivalent:

  A. there exists an st-cut whose capacity equals $|f|$

  B. f is a maximum flow

  C. there is no augmenting path with respect to f

Proving the flow is maximum

Compute the new residual network G_f and check it has no st-paths.

Find an S,T cut of capacity equal to the flow, 23. Such a cut provides a *certificate*, or *witness*, of optimality.

How do we find the S,T cut? Set S = { v such that there is an augmenting path from s to v} and $T = V - S$ (all the other vertices).

Ford-Fulkerson indeed computes a maximum flow

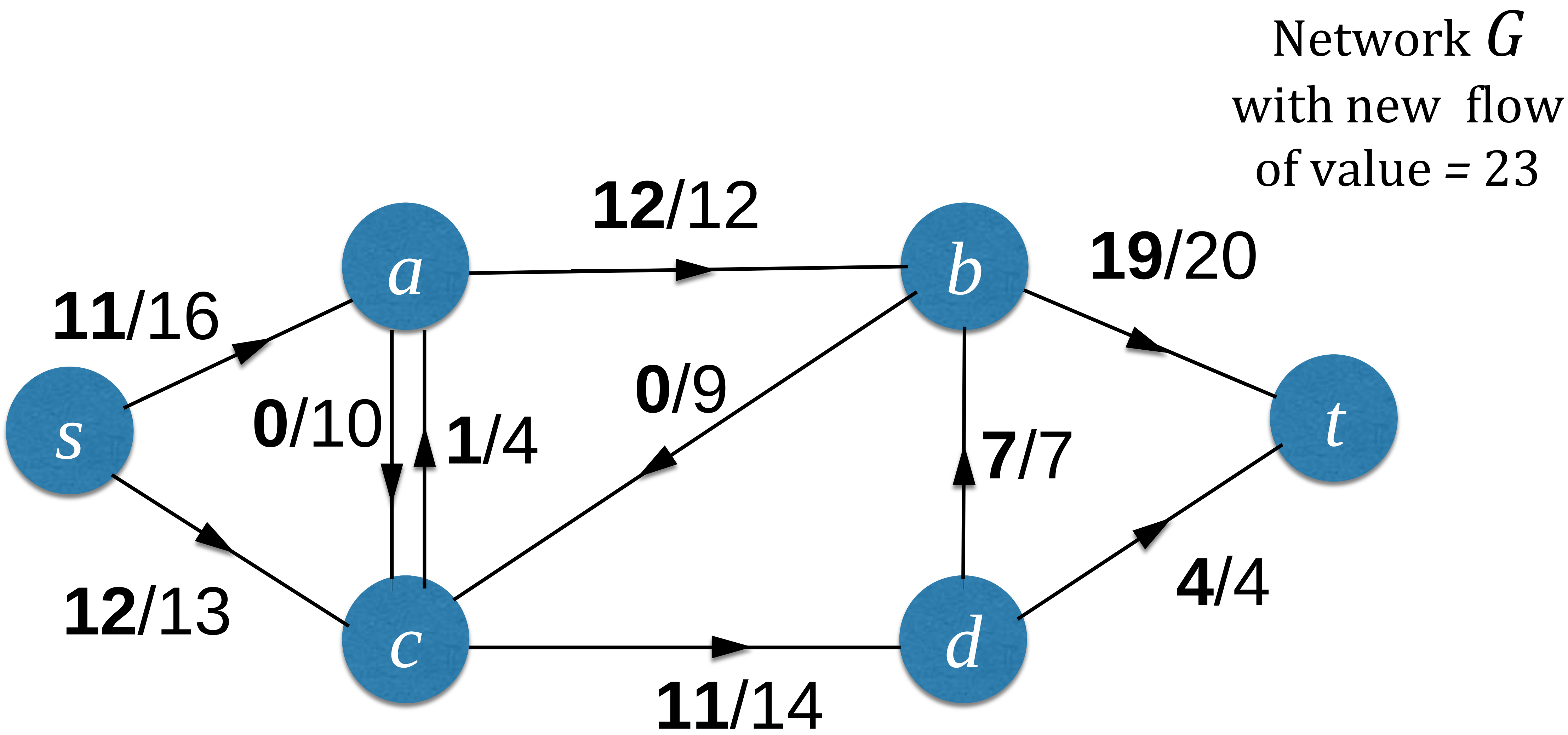
- by the Maxflow-mincut Theorem – can compute an S,T cut of capacity equal to the flow by taking

$S = \{v$ such that there is an augmenting path from s to $v\}$

and

$T = V - S$

Claim: the new flow is a maxflow.



What is the cut S,T of minimum capacity?

$S = \{ \ ?? \ \}$ $T = \{ \ ?? \ \}$

Check:
Are arcs leaving the S part full?
Are the arcs returning from the T part to S part empty?

Properties of the residual network G_f

- $|E_f| \leq 2|E|$
- The residual network G_f with capacities c_f of st-flow network G is an st-flow network

Definition of Augmenting Path

Given an st-flow f in st-flow network $G = (V, E)$ an augmenting path p is a directed path from s to t in the residual network G_f.

Pseudocode for Algorithm Ford-Fulkerson(G, s, t)

Initialize f as zero-flow

Compute residual network G_f

while there exists a path p from s to t in G_f **do**

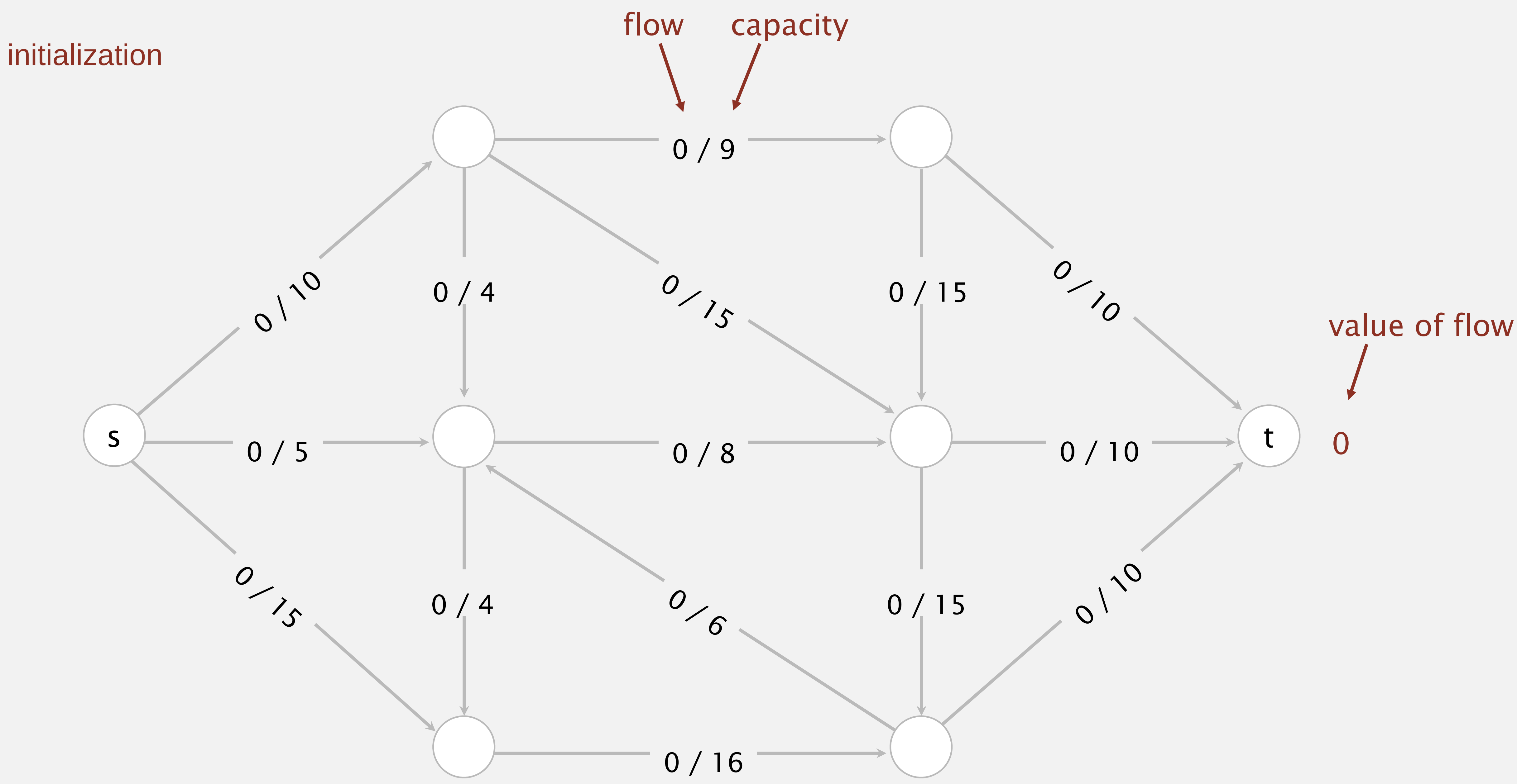
  Augment f using p

  Update G_f

return f

Ford-Fulkerson algorithm for solving MaxFlow

Initialization. Start with 0 flow.

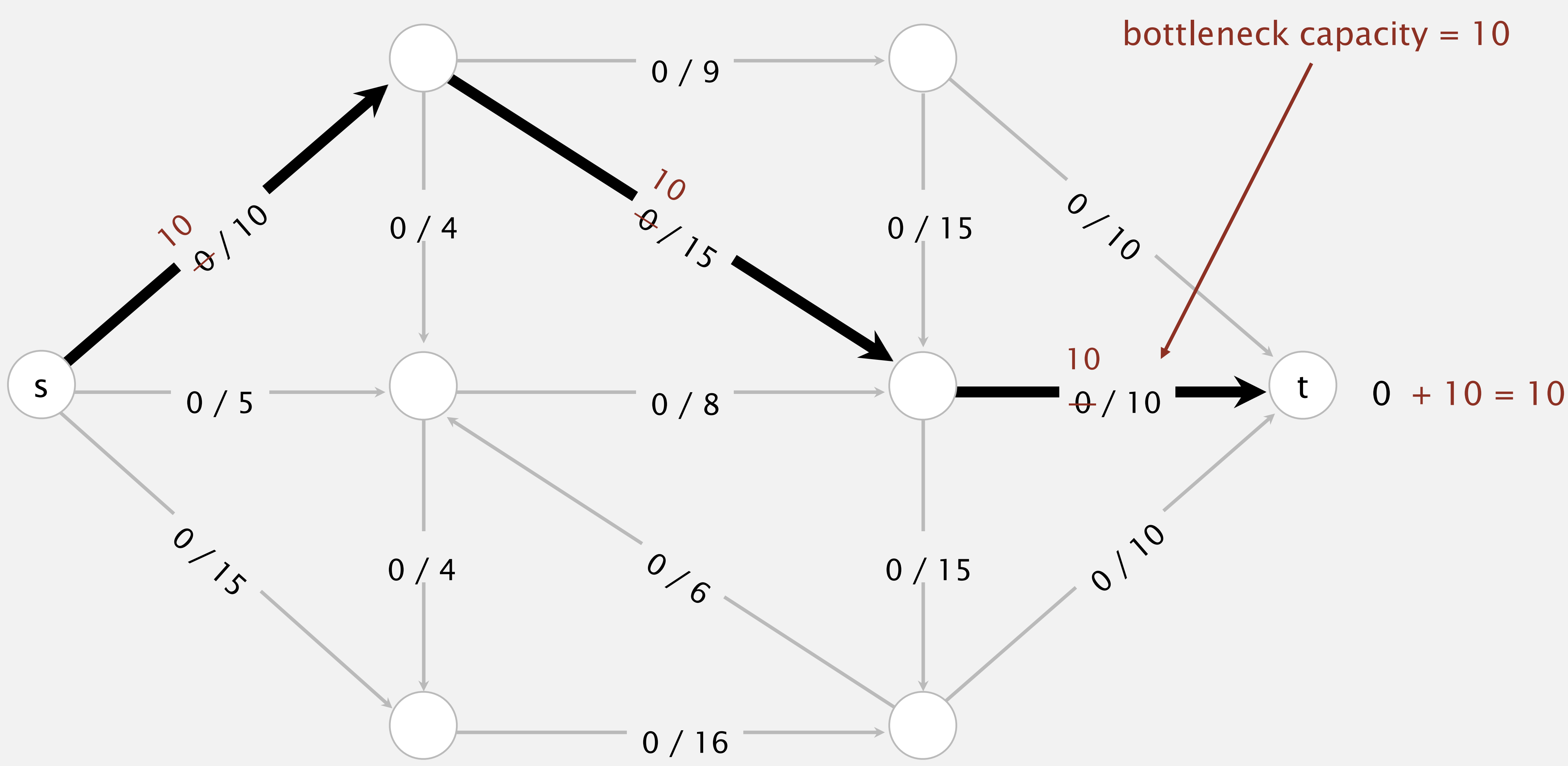


Idea: increase flow along augmenting paths

Definition: Augmenting path -- an undirected path from s to t such that:

- Can increase flow on forward edges (not full).
- Can decrease flow on backward edge (not empty).

1st augmenting path

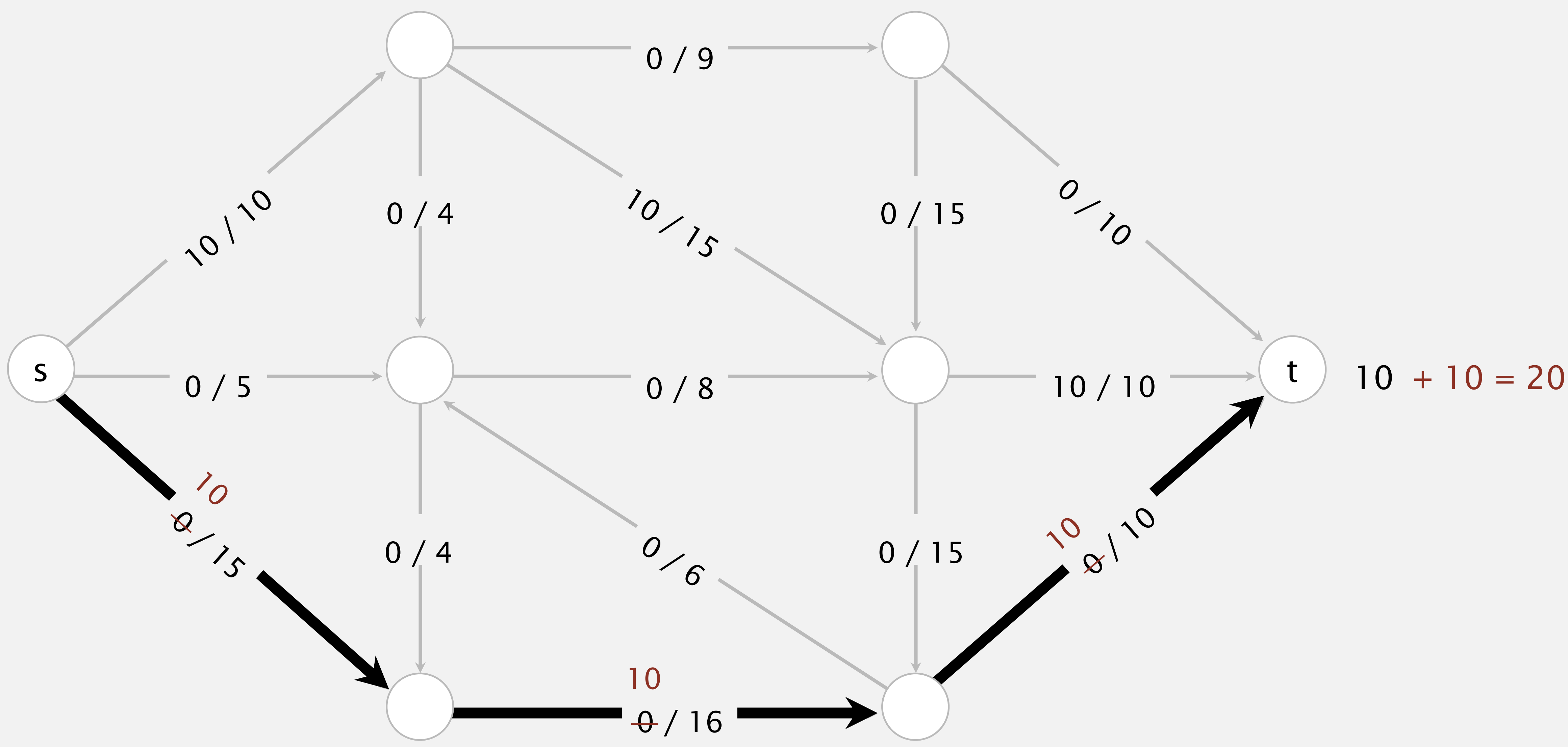


Idea: increase flow along augmenting paths

Augmenting path. Find an undirected path from s to t such that:

- Can increase flow on forward edges (not full).
- Can decrease flow on backward edge (not empty).

2nd augmenting path

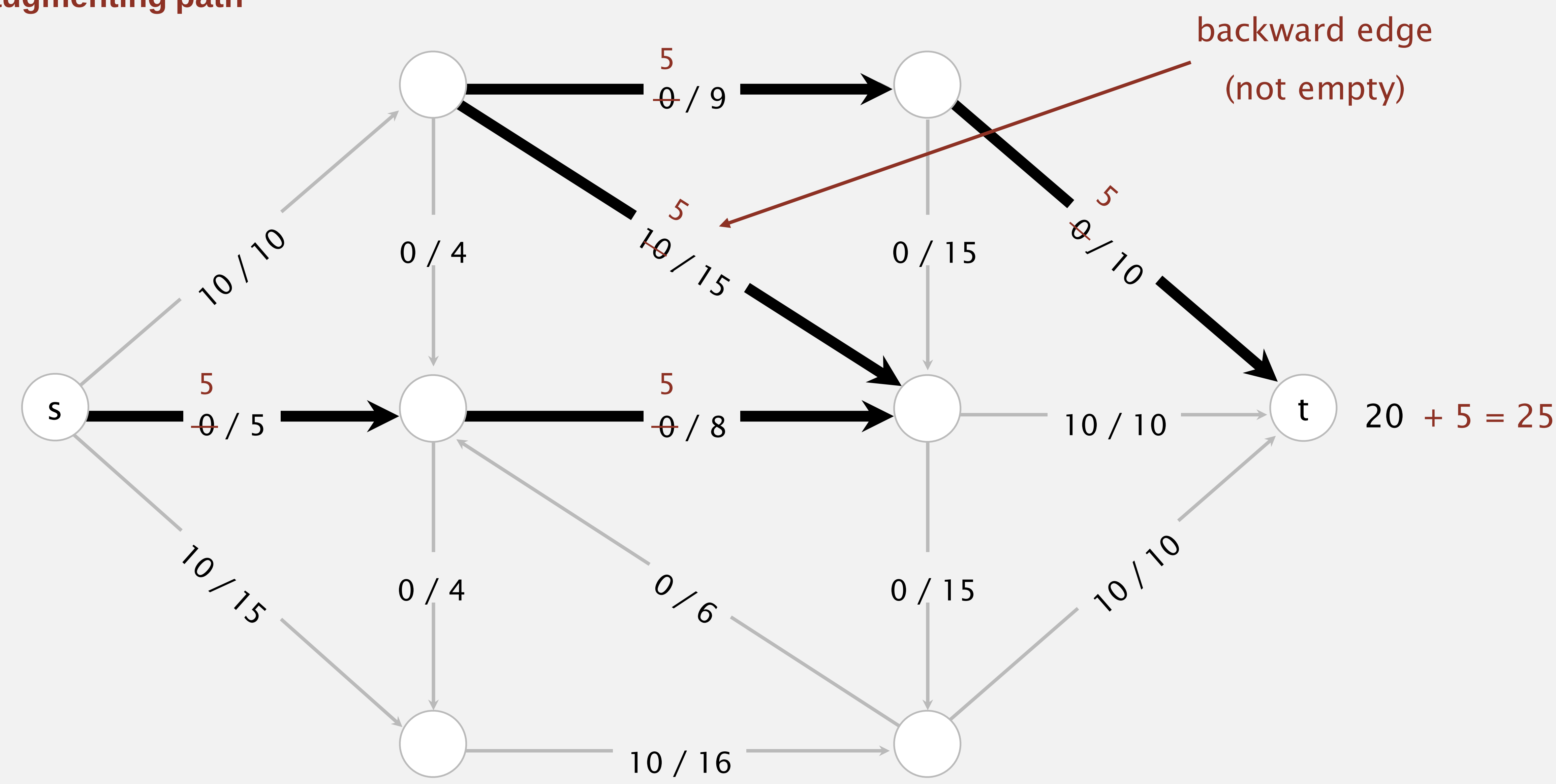


Idea: increase flow along augmenting paths

Augmenting path. Find an undirected path from s to t such that:

- Can increase flow on forward edges (not full).
- Can decrease flow on backward edge (not empty).

3rd augmenting path

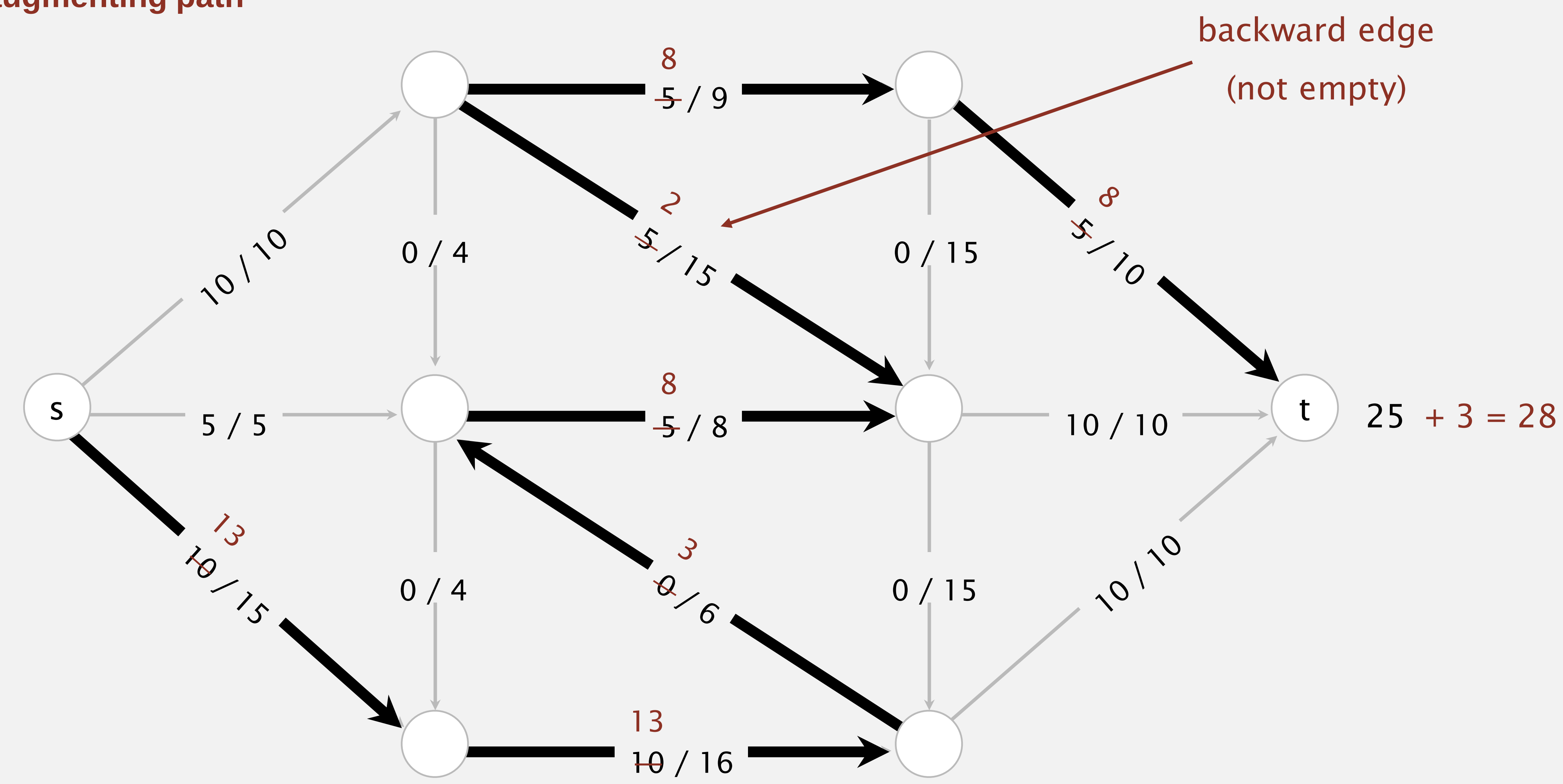


Idea: increase flow along augmenting paths

Augmenting path. Find an undirected path from s to t such that:

- Can increase flow on forward edges (not full).
- Can decrease flow on backward edge (not empty).

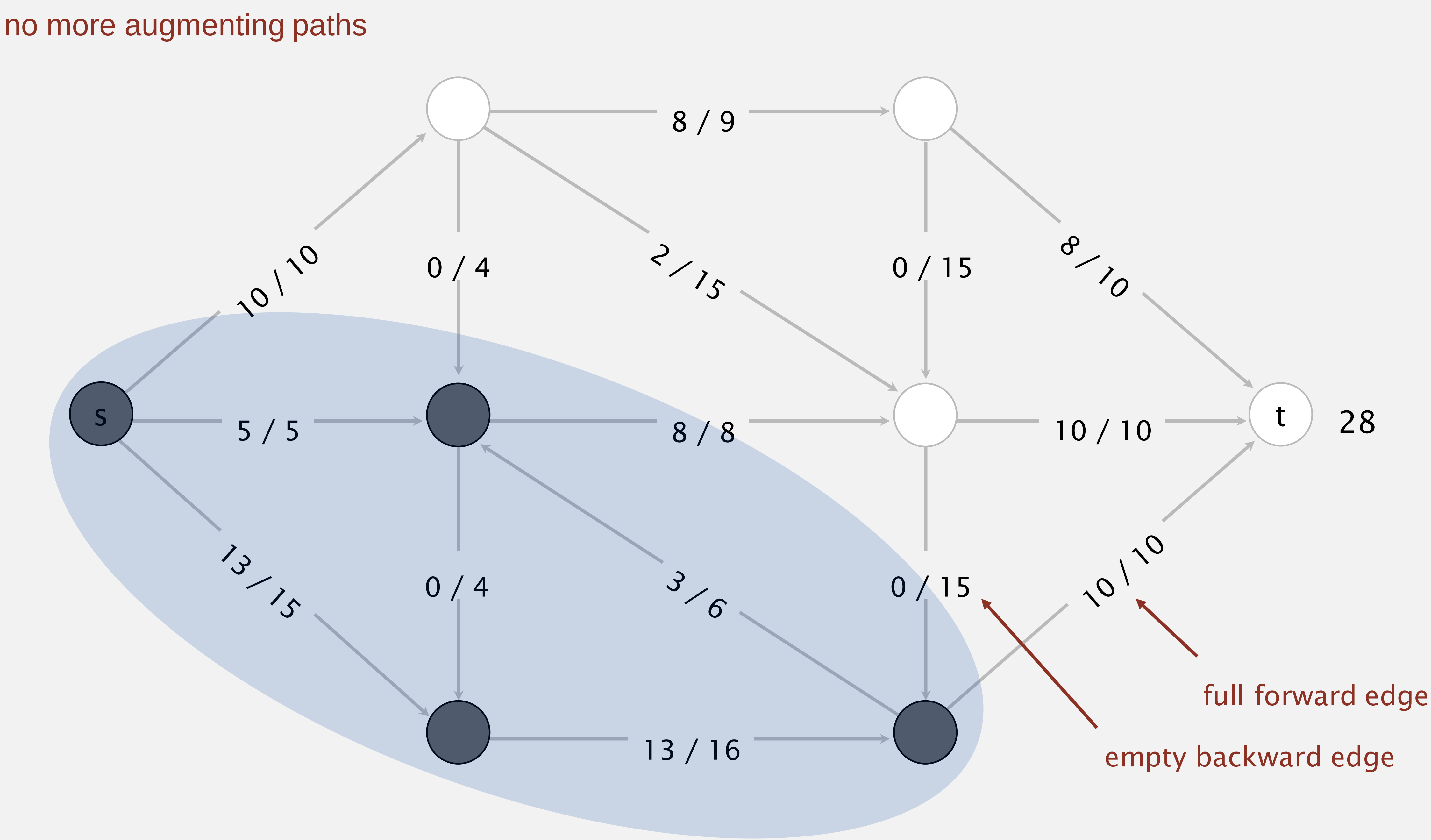
4th augmenting path



Idea: increase flow along augmenting paths

Termination. All paths from s to t are blocked by either a

- Full forward edge.
- Empty backward edge.



Running time of Ford-Fulkerson

- Building the residual network
- Finding an augmenting path in the residual network
- How many augmenting paths can be found in the worst case, if the capacities are positive integers?
  - value of maximum flow many (no more since the augmenting path has at least residual capacity 1)

EdmondsKarp(G, s, t)

Initialize f as zero-flow and residual network G_f with G
while there exists a path p from s to t in G_f **do**

Let p be a shortest path from s to t in G_f

Augment f using p

Update G_f

return f

Running Time Analysis of Algorithm by Edmonds & Karp

- Overall running time when using BFS (breadth first search) for determining augmenting path: $O(nm^2)$ $(m = |E|, n = |V|)$

  - An edge on an augmenting path in G_f is called *a bottleneck* if its capacity is equal to the path's residual capacity

  - Fact: an edge in G_f can be a bottleneck at most $O(n)$ times

  - The while loop therefore will not run more than $O(nm)$ times

  - The while loop's running time is $O(m)$

Baseball elimination problem

Q. Which teams have a chance of finishing the season with the most wins?

i	team		wins	losses	to play	ATL	PHI	NYM	MON
0		Atlanta	83	71	8	–	1	6	1
1		Philly	80	79	3	1	–	0	2
2		New York	78	78	6	6	0	–	0
3		Montreal	77	82	3	1	2	0	–

Montreal is mathematically eliminated.

- Montreal finishes with ≤ 80 wins.
- Atlanta already has 83 wins.

Baseball elimination problem

Q. Which teams have a chance of finishing the season with the most wins?

i	team		wins	losses	to play	ATL	PHI	NYM	MON
0		Atlanta	83	71	8	–	1	6	1
1		Philly	80	79	3	1	–	0	2
2		New York	78	78	6	6	0	–	0
3		Montreal	77	82	3	1	2	0	–

Philadelphia is mathematically eliminated.

- Philadelphia finishes with ≤ 83 wins.
- Either New York or Atlanta will finish with ≥ 84 wins.

Observation. Answer depends not only on how many games already won and left to play, but on whom they're against.

Baseball elimination problem

Q. Which teams have a chance of finishing the season with the most wins?

i	team		wins	losses	to play	NYY	BAL	BOS	TOR	DET
0		New York	75	59	28	–	3	8	7	3
1		Baltimore	71	63	28	3	–	2	7	4
2		Boston	69	66	27	8	2	–	0	0
3		Toronto	63	72	27	7	7	0	–	0
4		Detroit	49	86	27	3	4	0	0	–

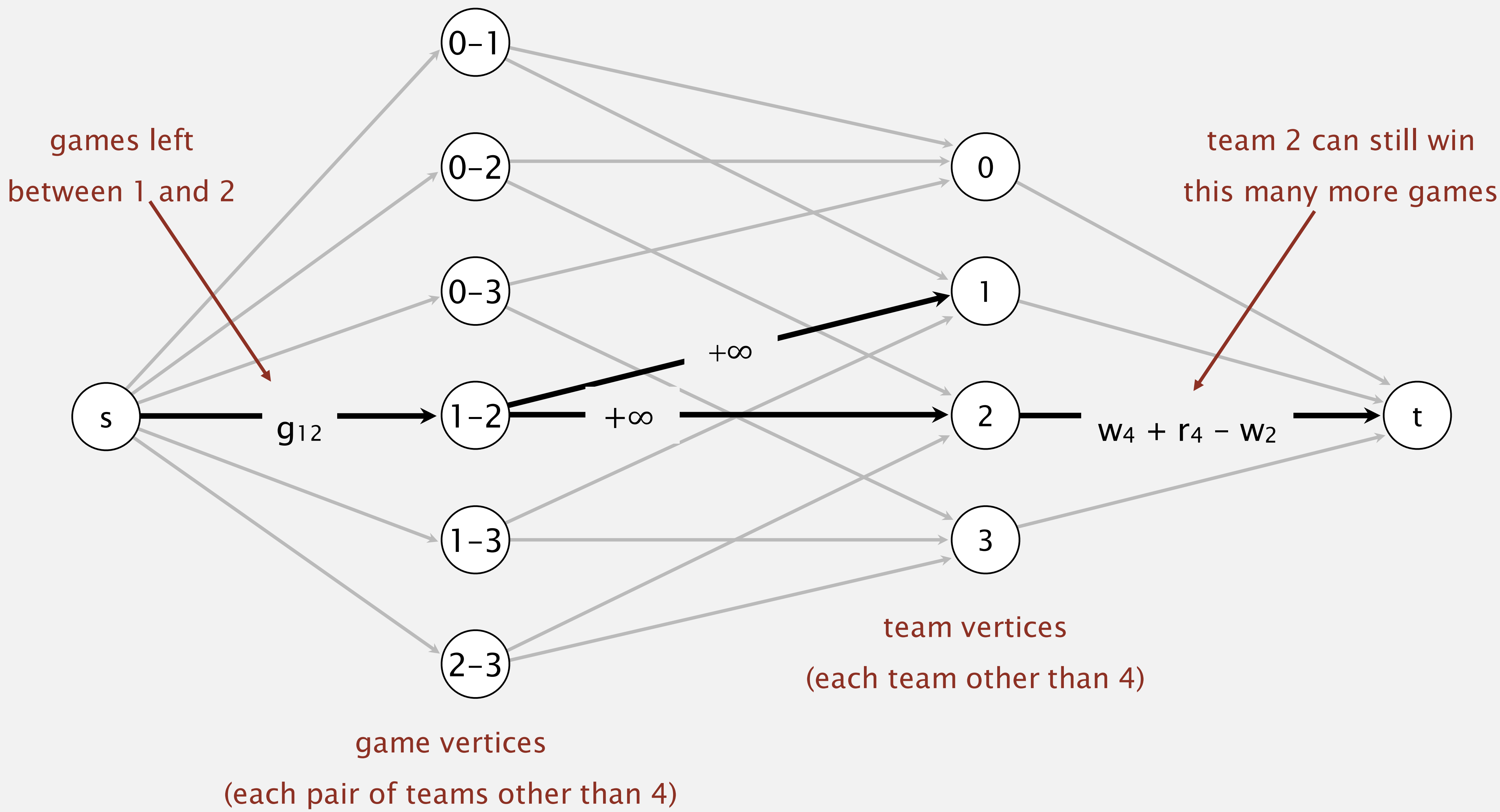
AL East (August 30, 1996)

Detroit is mathematically eliminated.

- Detroit finishes with ≤ 76 wins.
- Wins for $R = \{$ NYY, BAL, BOS, TOR $\} = 278$.
- Remaining games among $\{$ NYY, BAL, BOS, TOR $\} = 3 + 8 + 7 + 2 + 7 = 27$.
- Average team in R wins $305/4 = 76.25$ games.

Baseball elimination problem: maxflow formulation

Intuition. Remaining games flow from s to t.



Fact. Team 4 not eliminated iff all edges pointing from s are full in maxflow.