Max-flow min-cut theorem

Max-flow = min-cut.

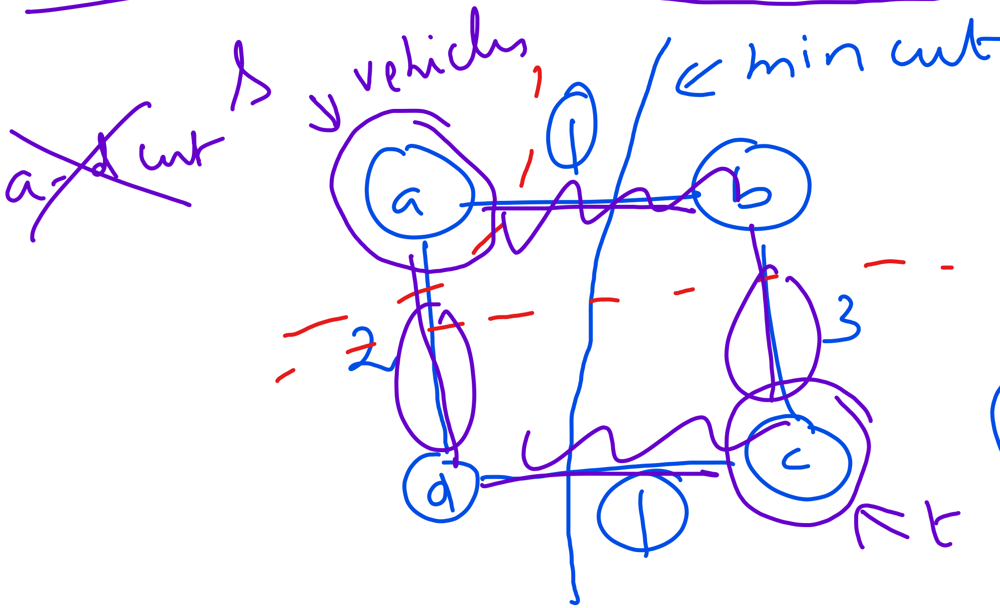
sink.

Source → s-t cut

s t

min a-c cut

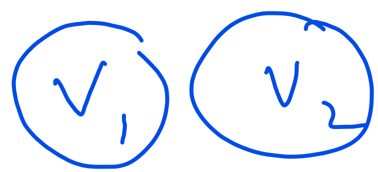
# Flow Networks




cut-value = 2

$w : E \to \mathbb{Z}^+$

$G(V, E, w)$




$V_1 \cup V_2 = V$

$V_1 \cap V_2 = \phi$

# What problem does it solve?

- Find maximum flow in a flow network from a single source s to a single sink t
- Ex. Practical applications:

Freight: find the maximum number of trucks we can send out to a location according to road map with different road capacities.

Other: Flow of water through pipes

- Flow network (G)
  - Simple Directed graph
  - 1 source s, 1 sink t
  - Non-negative capacity for each edge
- Residual Graph (Gf)
  - Only edges from G that can still have more flow
  - Cf(u,v) = c(u,v) – f(u,v)
  - Added edges: reverse edges to decrease flow *'cancellation'*

    cf(v,u)=cf(v,u)+f(u,v)
- Augmenting Paths (p)
  - Path from s to t in Gf
  - Residual capacity = smallest capacity of edges of p

# The algorithm

Fm = o

While there exist an augmenting path p in Gf

Find augmenting path p

Cf (p)= smallest capacity on p

Fm = Fm + Cf(p)

For each edge in p

cf(u,v) = cf(u,v)-Cf(p)

cf(v,u)= cf(v,u)+cf(p)

parametric time complexity

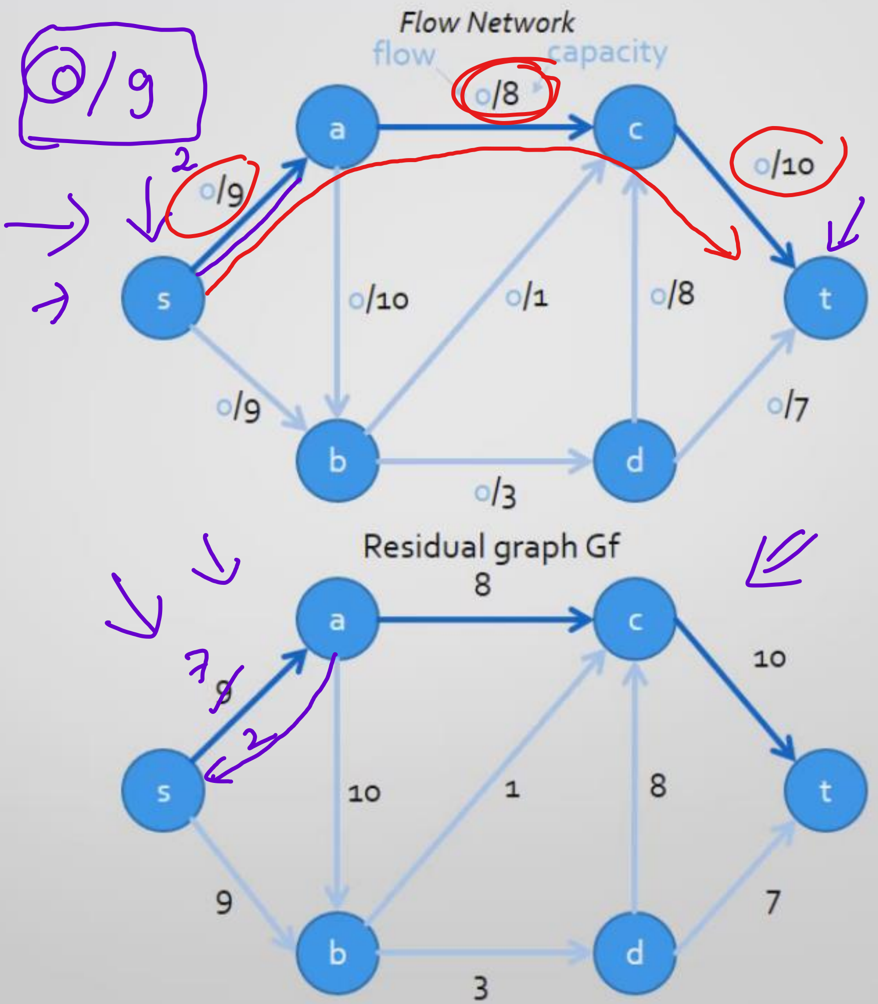
$O(|V|+|E|)$

$fm * O(|V|+|E|)$

Start:

F(u,v) = 0 for all edges

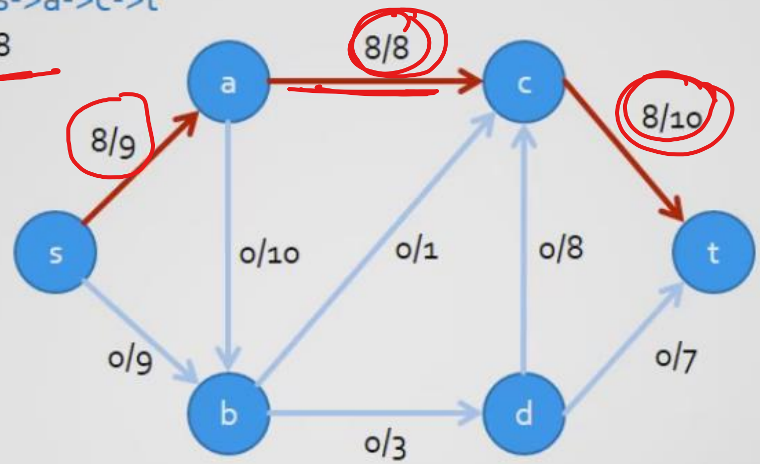
Fm = o



Find Augmenting path: s->a->c->t

Find Residual capacity: 8

Fm :- r

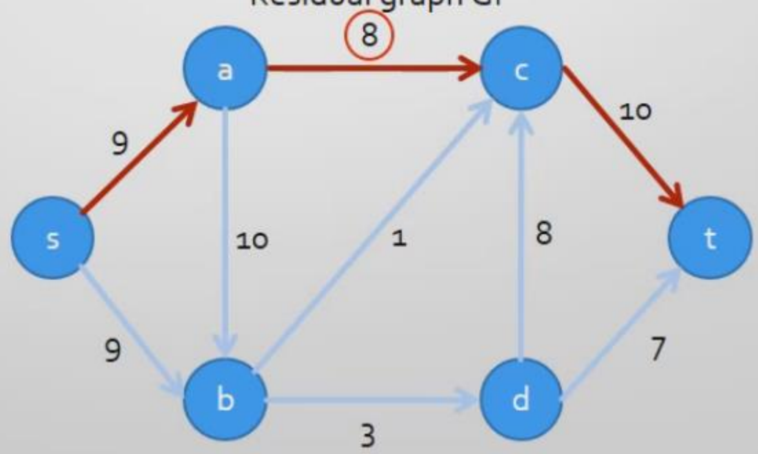



Fm = 0+ 8

$C_f(u,v)=c_f(u,v)-f(u,v)$

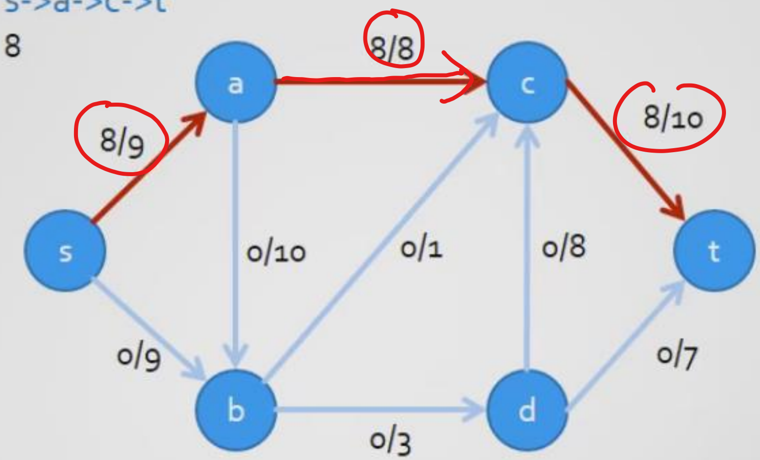
$C_f(v,u)=c_f(v,u)+f(u,v)$

Residual graph Gf



Find Augmenting path: s->a->c->t

Find Residual capacity: 8




Fm = 0+ 8

Cf(u,v)=cf(u,v)-f(u,v)

Cf(v,u)=cf(v,u)+f(u,v)

Cf(s,a)=9-8=1

Cf(a,c)=8-8=0

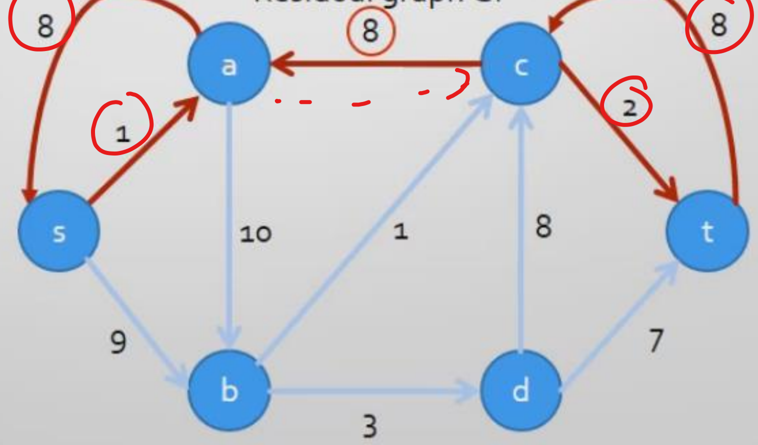
Cf(c,t)=10-8=2

Cf(a,s)=8

Cf(c,a)=8

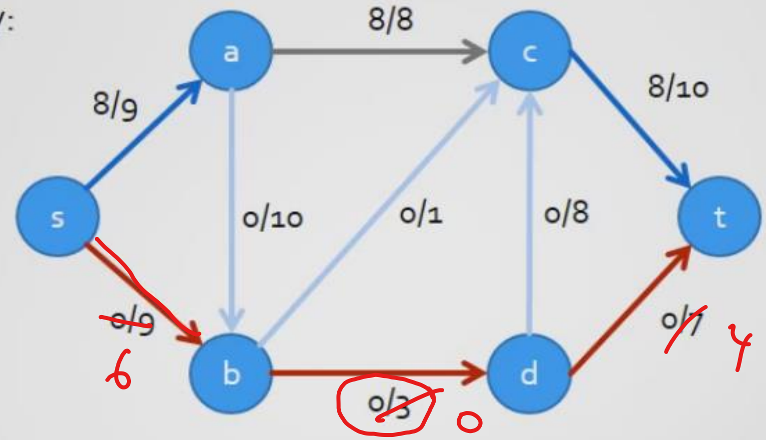
Cf(t,c)=8

Residual graph Gf



Find augmenting path: s->b->d->t

Find residual capacity:

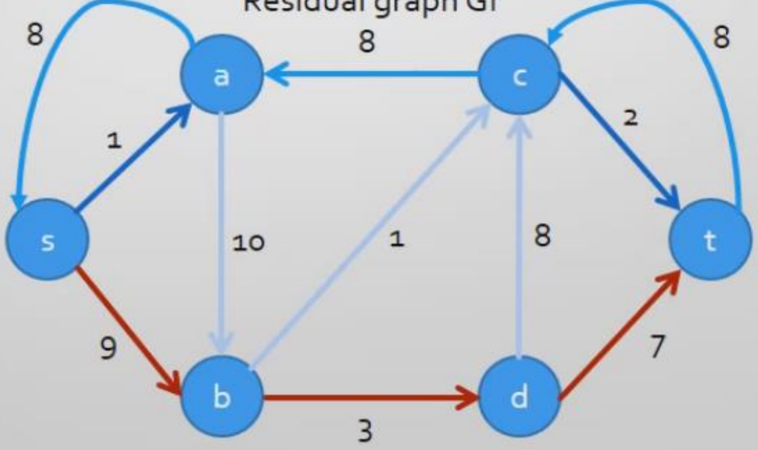



Fm = 0+8 + 3

$Cf(u,v)=cf(u,v)-f(u,v)$

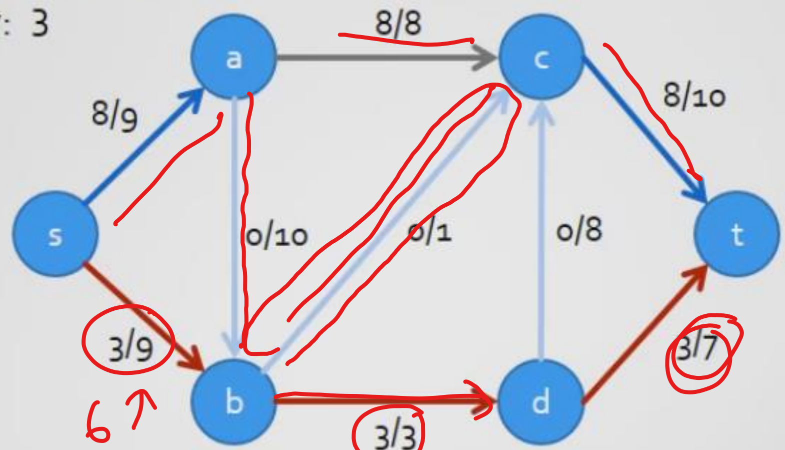
$Cf(v,u)= cf(v,u)+ f(u,v)$

Residual graph Gf



Find augmenting path: s->b->d->t

Find residual capacity: 3




Fm = o+8+3

Cf(u,v)=cf(u,v)-f(u,v)

Cf(v,u)= cf(v,u)+ f(u,v)

Cf(s,b)=9-3=6

Cf(b,d)=3-3=o

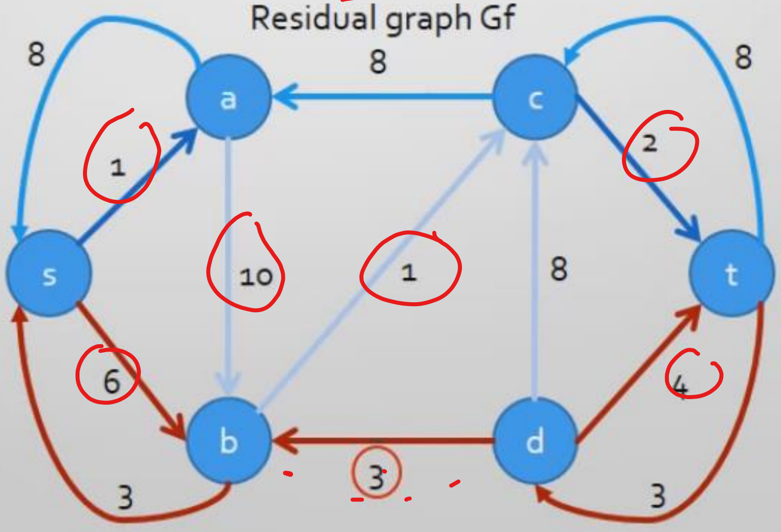
Cf(d,t)=7-3=4

Cf(b,s)=3

Cf(d,b)=3

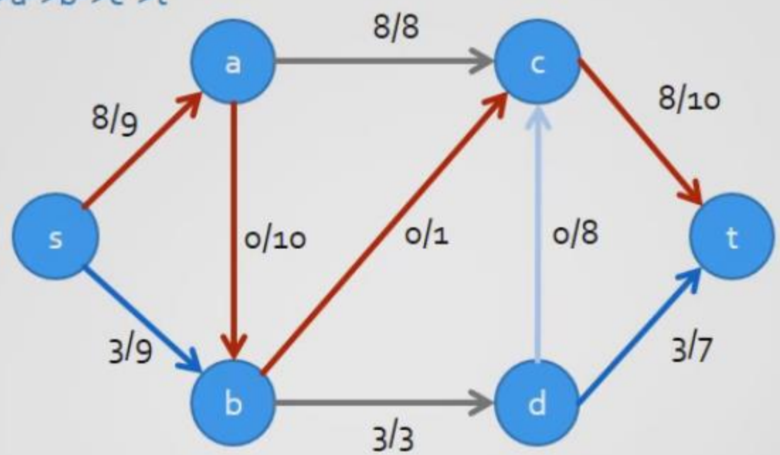
Cf(t,d)=3

Residual graph Gf



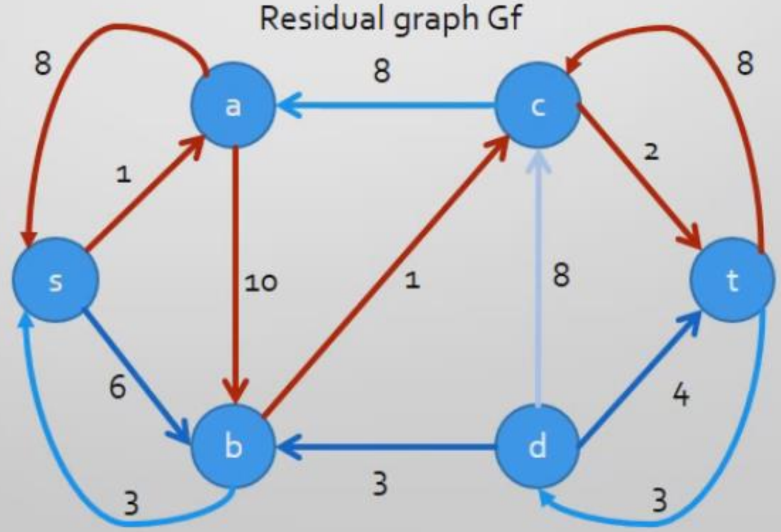
Find augmenting path: s->a->b->c->t

Find residual capacity:



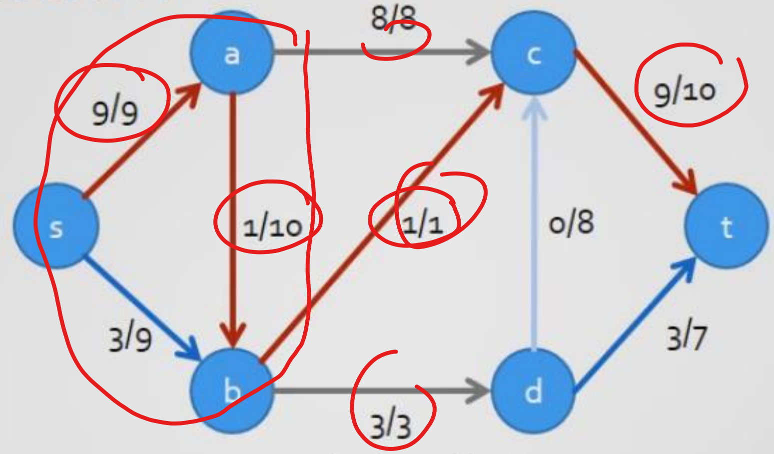

Fm = 0+8 + 3

Residual graph Gf



Find augmenting path: s->a->b->c->t

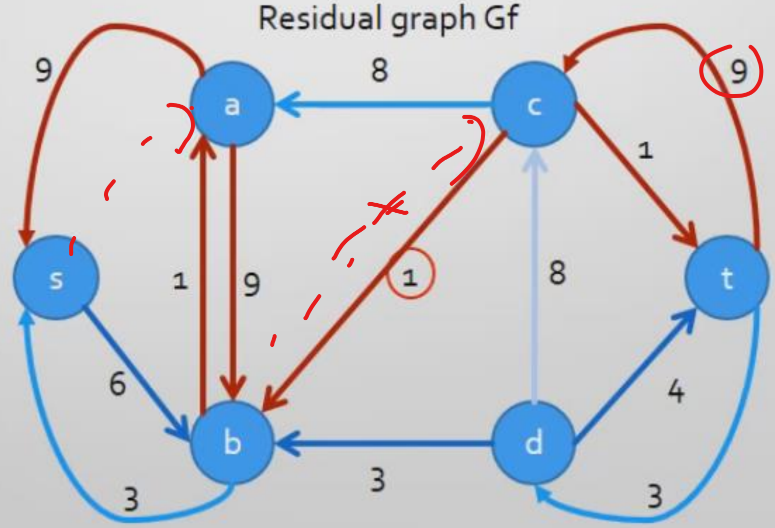
Find residual capacity: 1

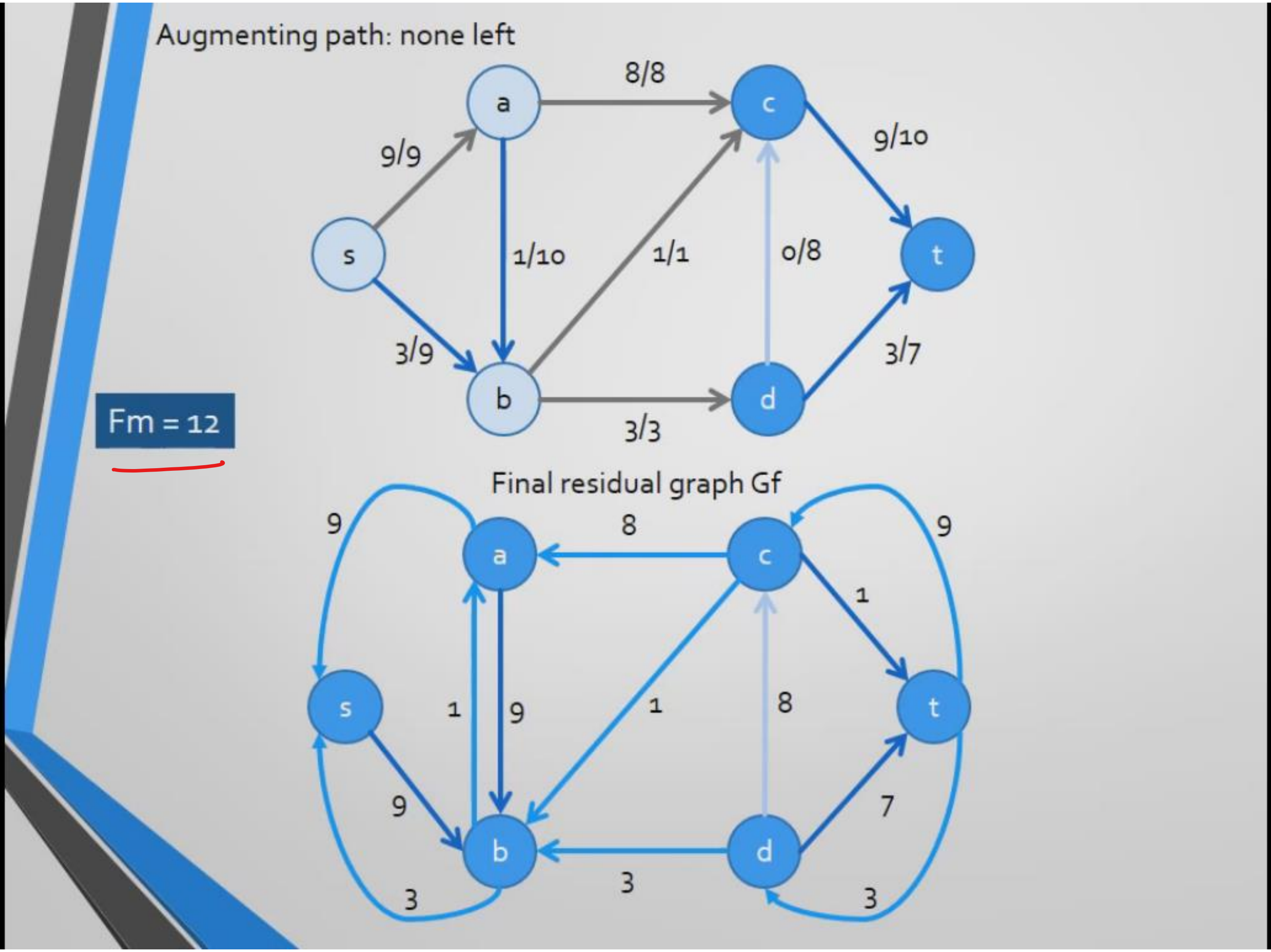



Fm = 0+8+3+1

12

Residual graph Gf




Augmenting path: none left
Fm = 12
8/8
9/9
9/10
1/10
1/1
0/8
3/9
3/7
3/3
Final residual graph Gf
9
8
9
1
1
9
1
8
9
7
3
3
3

m = 5

1 2 3 4 5 6 . . .

% $10^4$ 23456

1234

1 × $10^4$

# Analysis
# Complexity of the Algorithm

- Flow increased by at least 1 at every iteration, so the while loop is repeated Fm times at most, where Fm is the maximum flow
  - Finding an augmenting path : O(E) where E is the number of edges
  - Operations on value: O(1)
  - Each iteration: O(E)

This algorithm runs in time O(E*Fm)

$|V| << |E|$