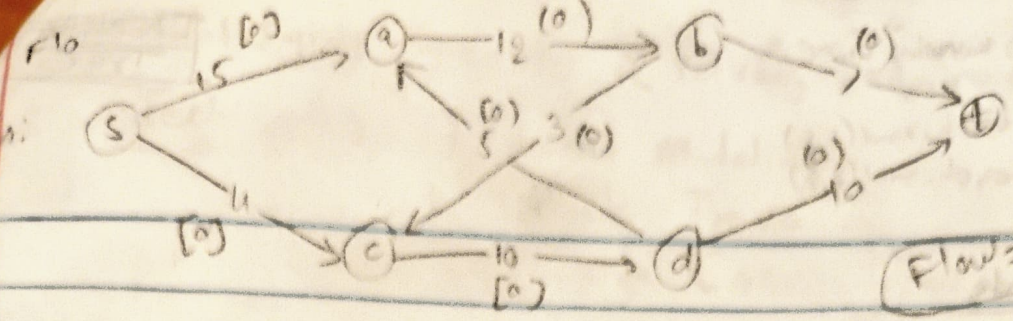
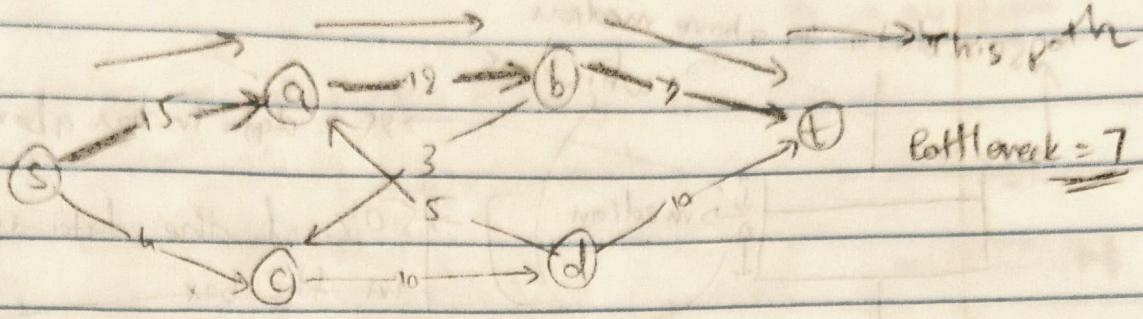
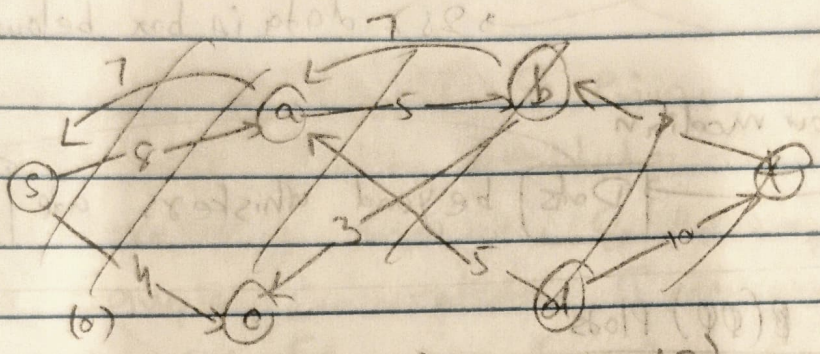
f 10

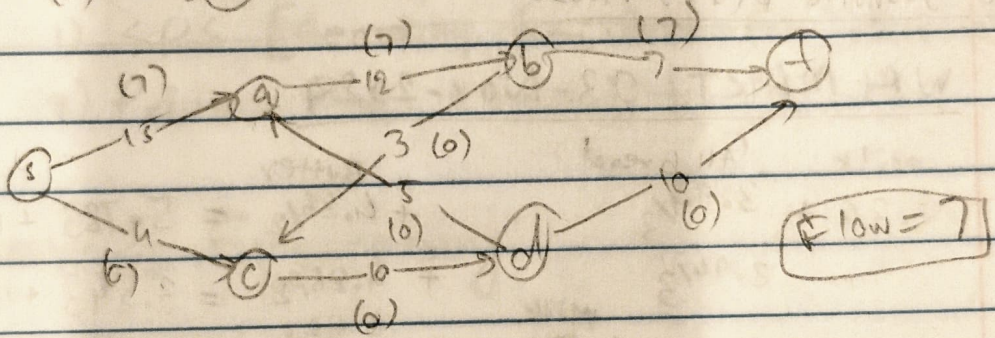

s
a
b
c
d
t
15
12
10
4
10
3
5
10
Flow = 0

g:

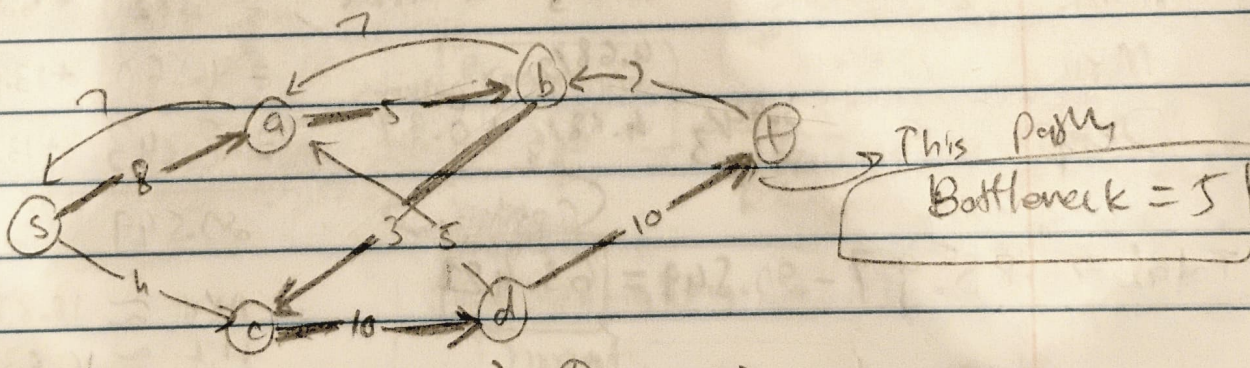

This path
s
a
b
c
d
t
15
12
3
5
4
10
10
Bottleneck = 7


s
a
b
c
d
t
7
8
5
3
4
10

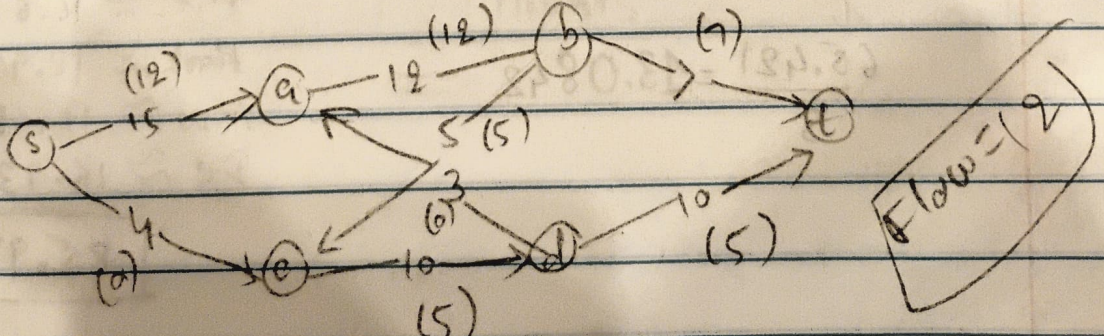
h:


s
a
b
c
d
t
15
12
10
3
5
4
10
Flow = 7

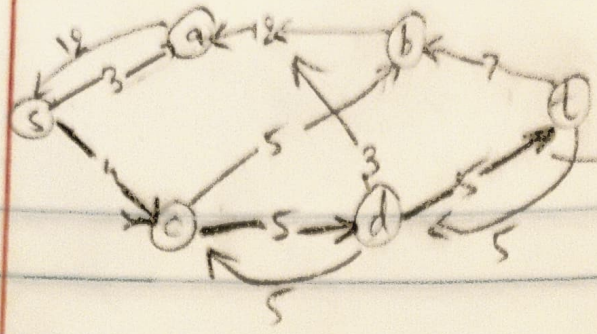
i:


s
a
b
c
d
t
8
5
3
5
4
10
10
This path
Bottleneck = 5

j:

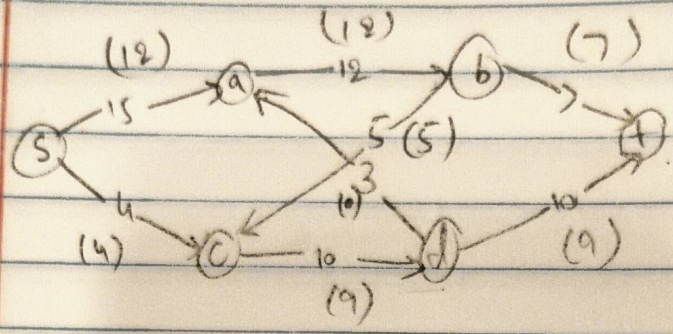

s
a
b
c
d
t
15
12
5
3
4
10
10
Flow = 9

g:



→ Bottleneck = 4

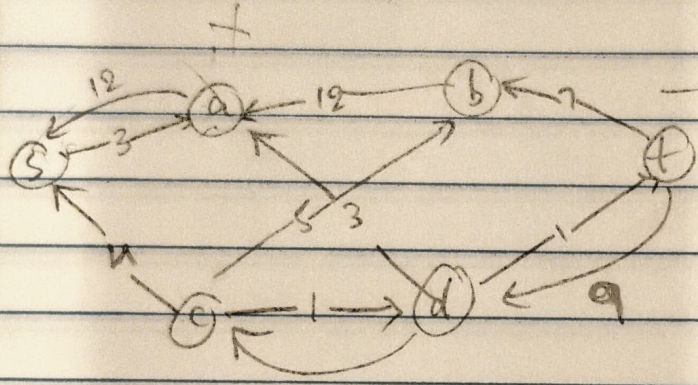
h:



7+5=12+4

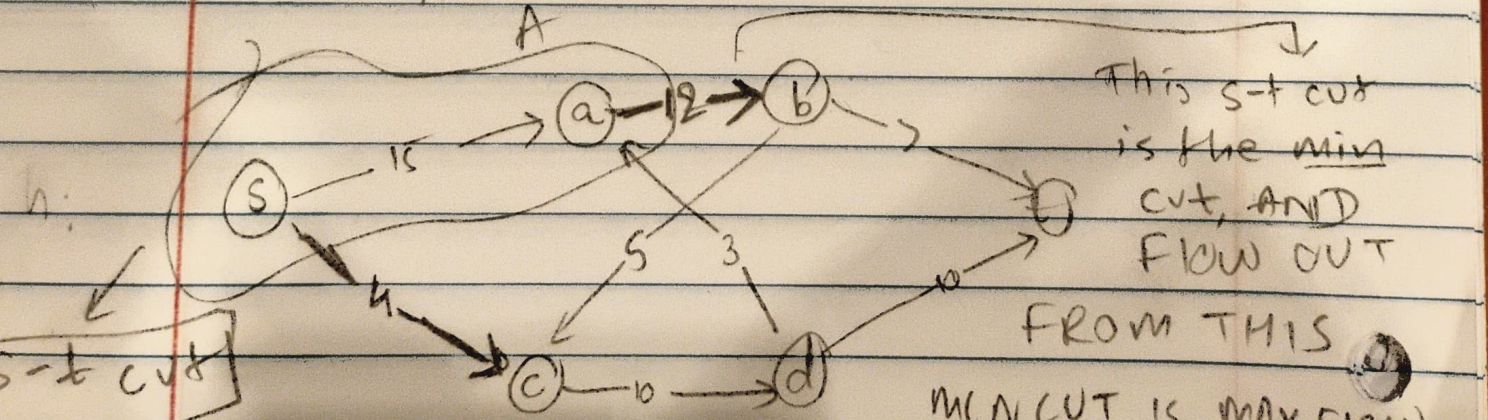
FLOW = 16

G_f:



→ At this point we are stuck thus means we cannot go from source (s) to sink (t) anymore, hence (s-t) cut

we found the optimal PATH! Also, the minimum cut will include all nodes reachable from source (s) in G_f i.e. residual graph G_f. It is clear that we can only reach (a) from (s), hence the s-t cut will look like



This s-t cut is the min cut, AND FLOW OUT FROM THIS MIN CUT IS MAX FLOW.

Which is 4 → t — 12 → = 16 ← This was the MAX FLOW!!

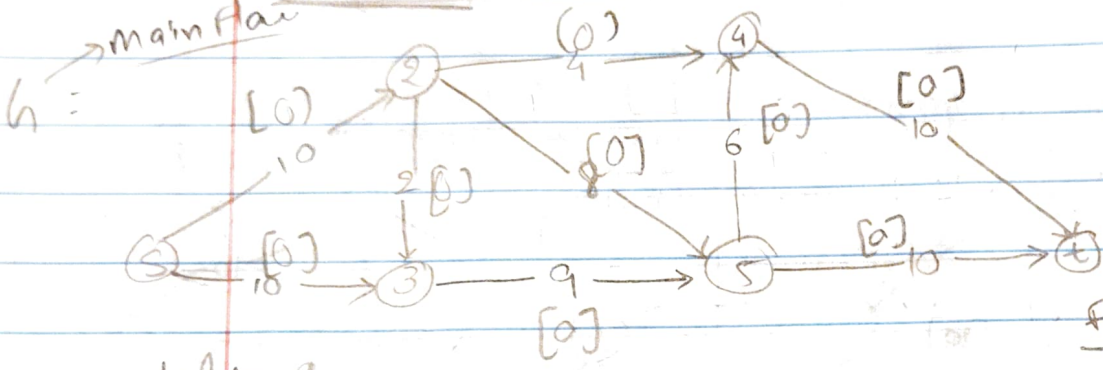
amount equal to (capa
and remove flow part

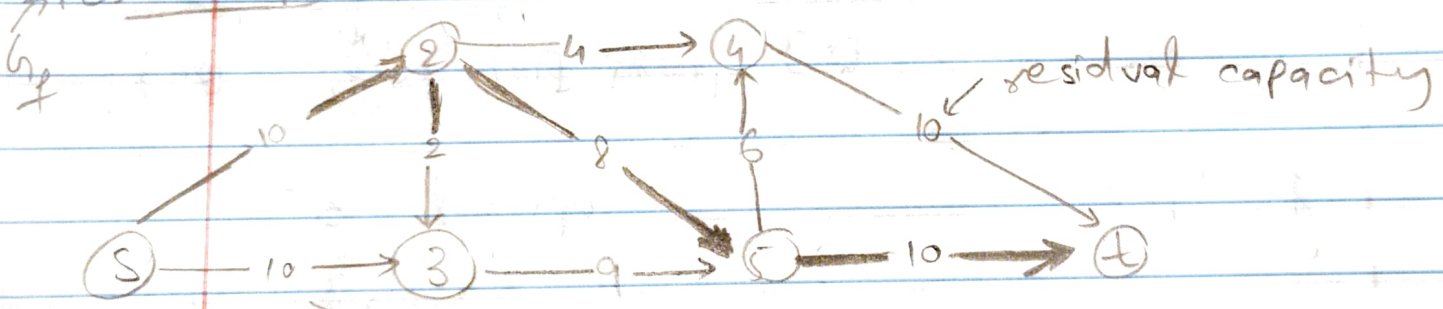
Residual Graph

⇒ All such edges with positive residual capacity will make the Residual Graph

→ Now we can run an algorithm on this residual graph, i.e to find MAX FLOW.

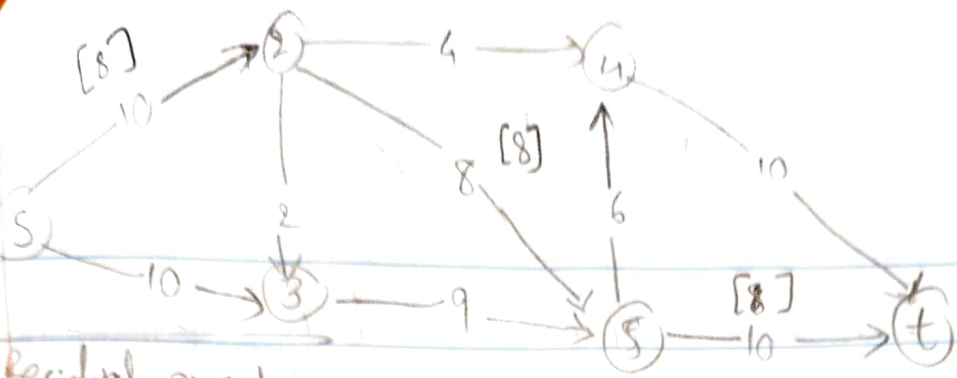
→ FORD-FULKERSON ALGORITHM





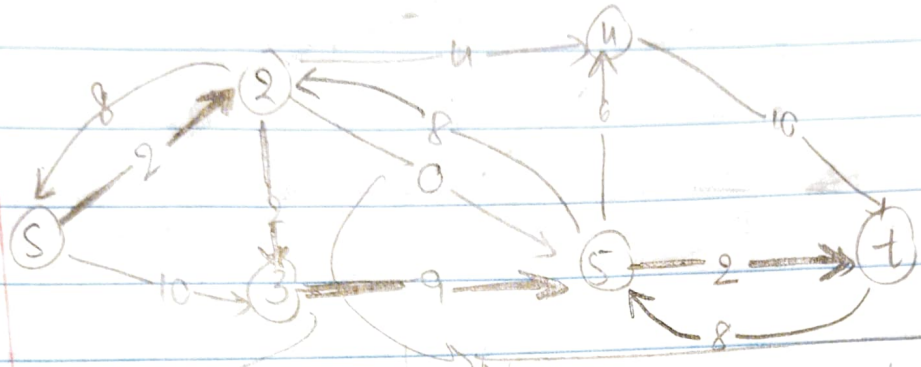
Let's choose this path, (you can choose any!)

The bottleneck = edge with lowest capacity as that is the MAX flow. we can send in this p = 8 in this path, So take this and pass flow of 8 through this path in the main flow.



Residual graph

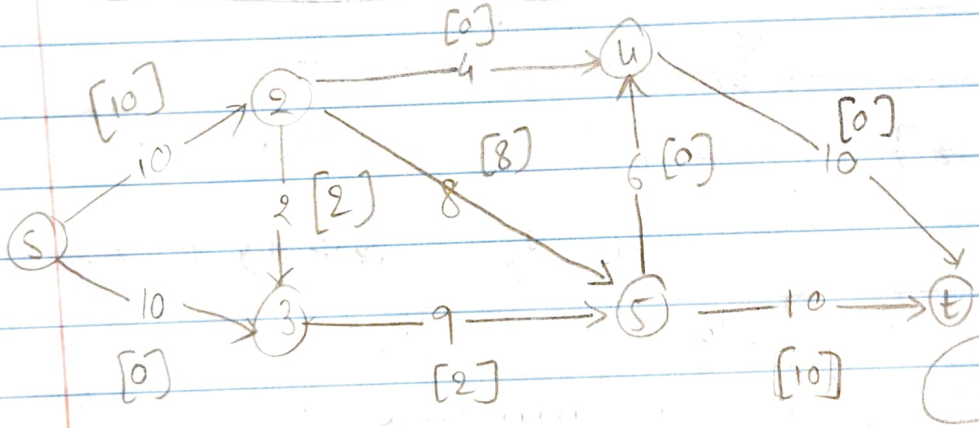
flow = 8



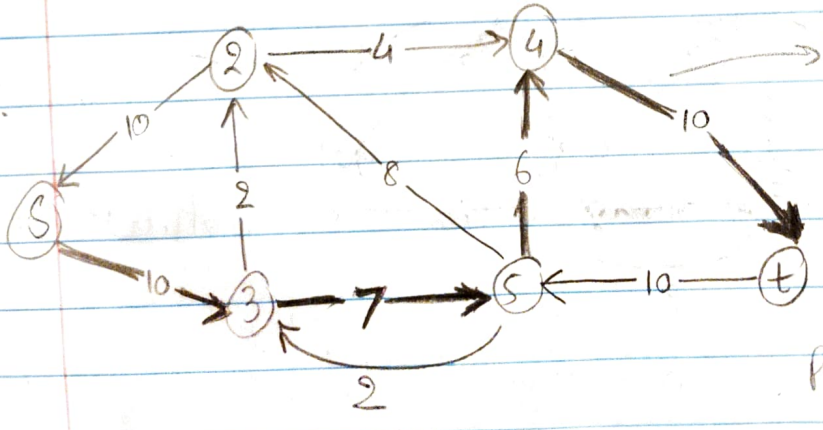
No need to draw edge if cap = 0, it does not mean it DOES NOT EXIST

Afterwards, lets choose this path

Bottleneck = 2 → So take '2' flow on this path



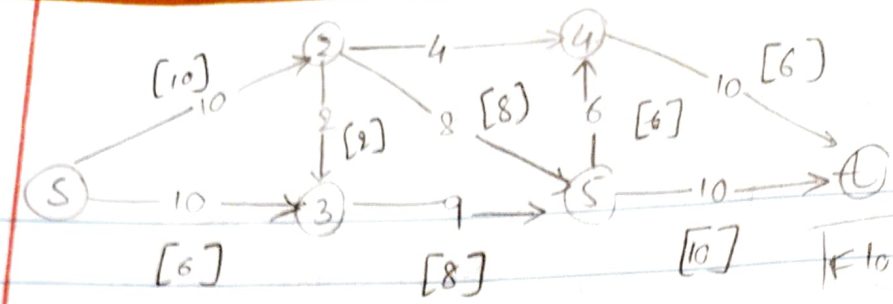
Flow = 10



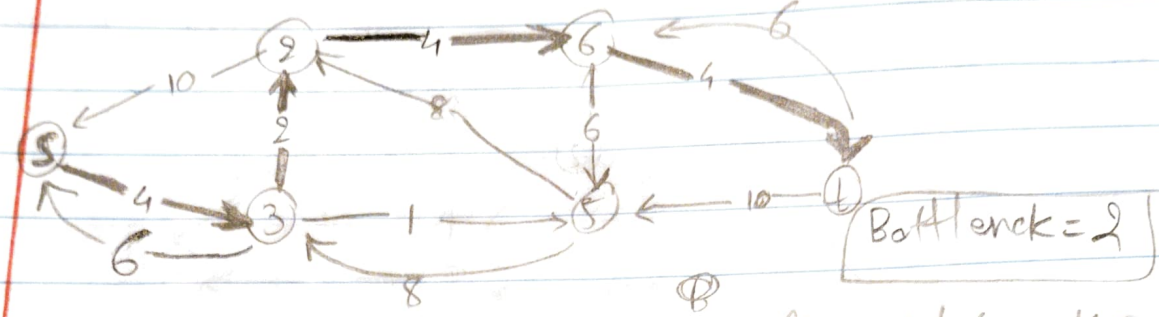
→ Next path

Bottleneck = 6

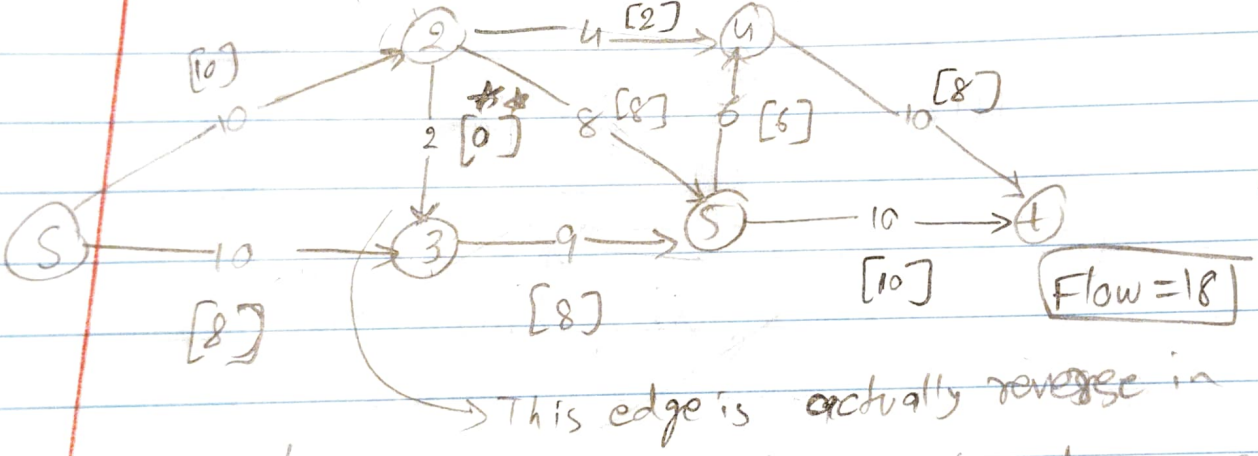
Add flow of '6' on this path in original "G".



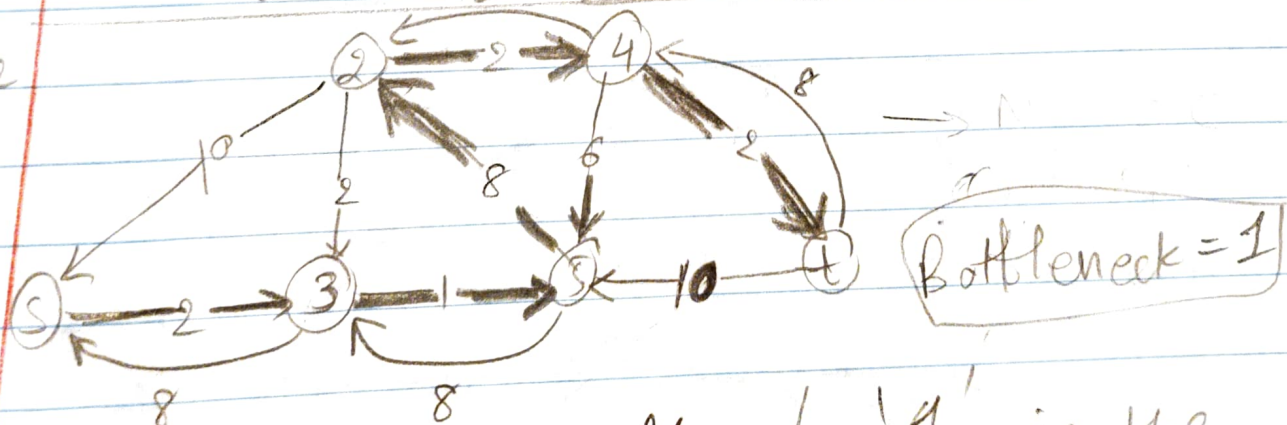
Update G_f:



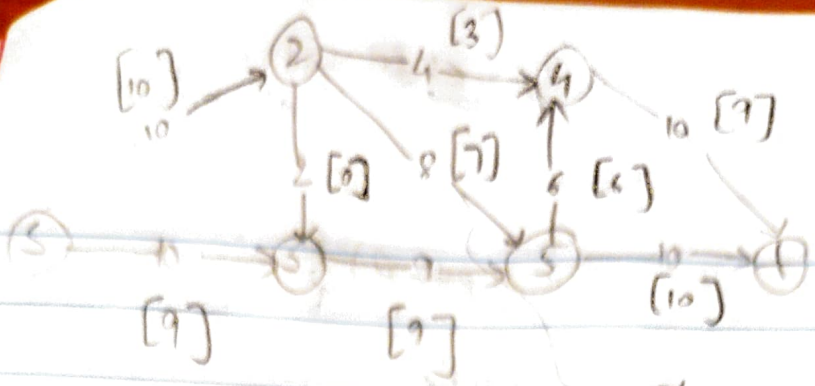
Add flow of '2' on this path in the original 'G'



the 'G_f', hence it is telling us to decrease the flow by 'bottleneck amount'



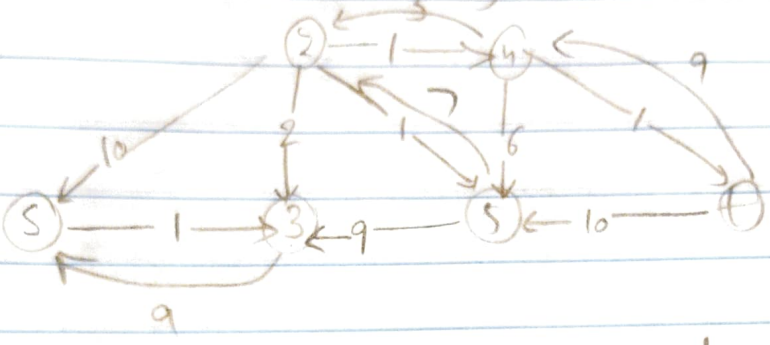
Increase flow by '1' in the main flow Graph 'G'



Flow = 19

→ This is reverse edge in G_f, hence decrement flow by bottleneck = 1, Flow = 8 − 1 = (7)

Update G_f



→ NOW WE CANNOT GO FROM SOURCE (S) to SINK (t)

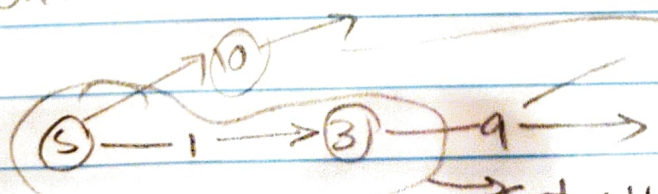
→ When this occurs, it means we found our optimal flow OR MAX FLOW OF THE SYSTEM Hence max flow of this example = (19)

→ Cut: All the nodes at this point of time that can be reached by sink (s), are the "part" of (s-t) cut. which is NOTHING BUT THE MINIMUM CUT OR MAXIMUM FLOW → (minimum cut required to restrain maximum flow of the system

s-t cut

In our case, only (3) is reacheable from (s)

Hence the s-t cut is



See the flow from s-t cut = 10 + 9 = 19 i.e. the MAX FLOW