Homework 4
Colorado CSCI 5454

Sai Siddhi Akhilesh Appala

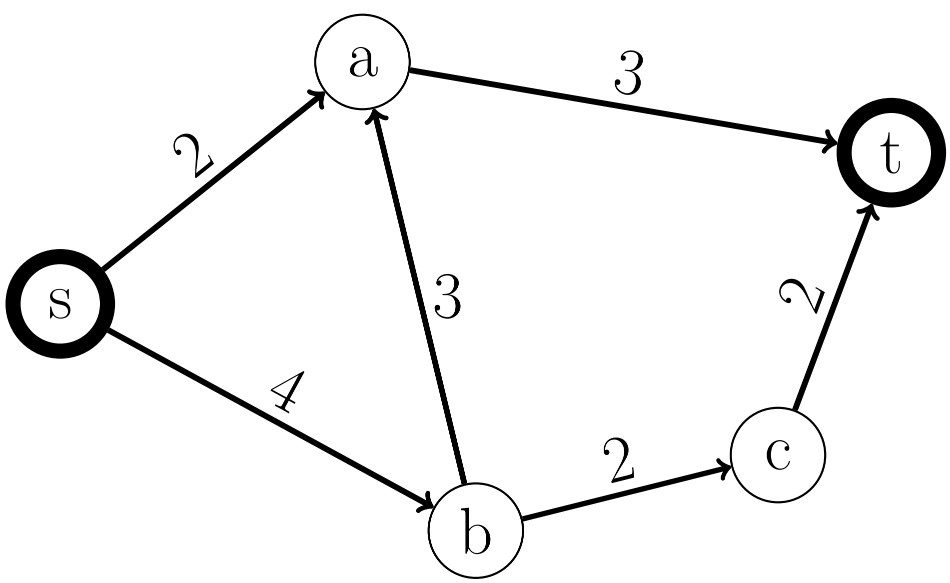
November 30, 2022

People I studied with for this homework: None

Problem 1

Question

part a



Flow graph: Before starting the **Net flow($|f|$)**: 0 will be 0 according to the algorithm. so there will be no flow between the edges.

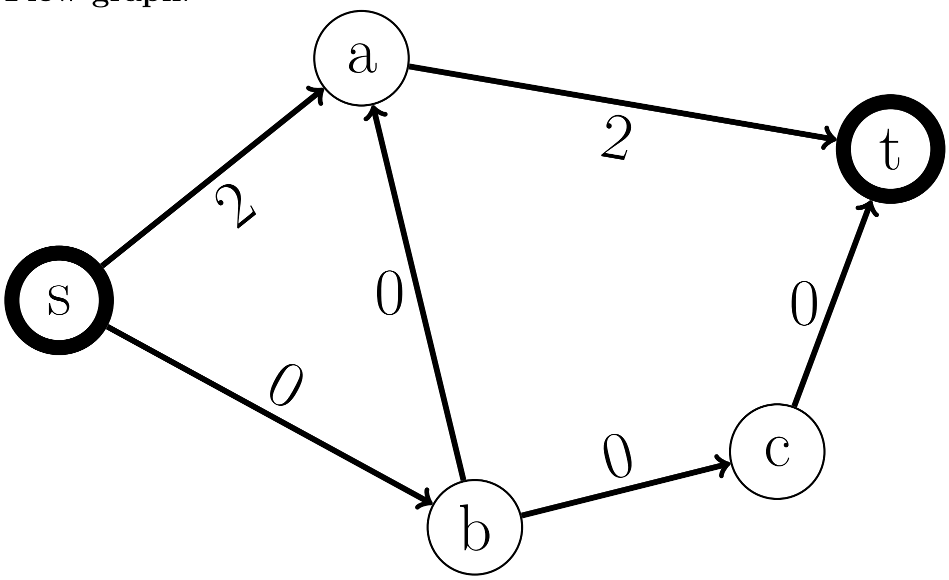
Net flow : ($|f|$): 0
Residual values (r_f). will be present in the below residual graph
Residual Graph(G_f) : Same graph as given.

Step 1

From the above iteration of the Residual Graph let's select a path from $S \to a \to t.s$

To apply the augment we need to take the min(2,3) = 2 from the above path. Now the augment is 2 we need to send 2 for both $S \to a \to t.s$

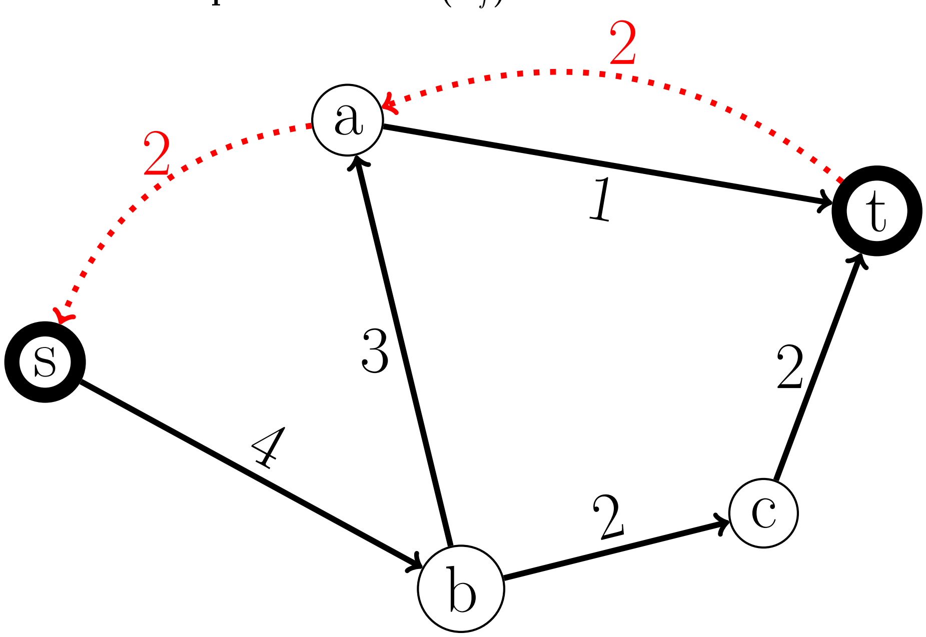
Flow graph:



Net flow : ($|f|$): 2

Residual values (r_f) will be present in the below residual graph

Residual Graph with values(G_f) :

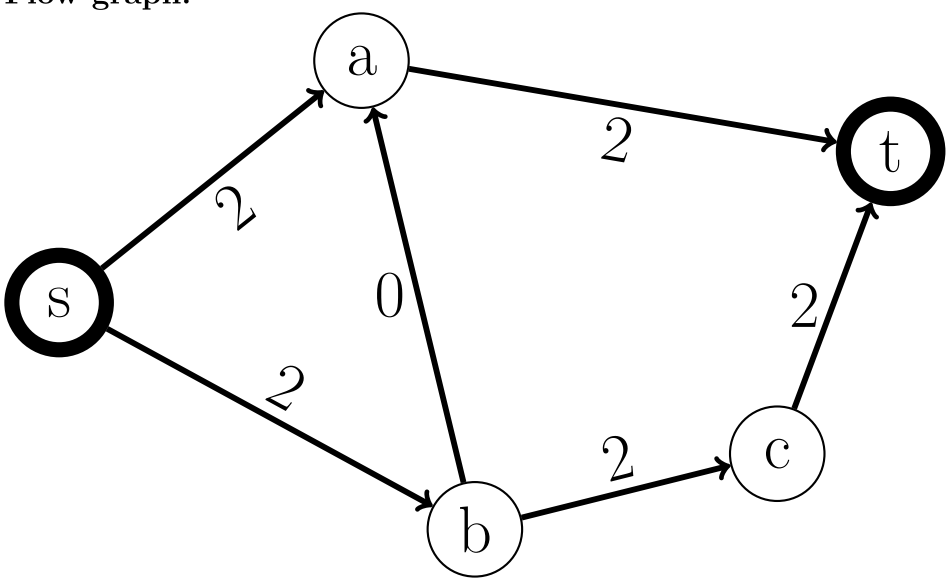


Step 2

From the above iteration of the Residual Graph let's select a path from $S \rightarrow b \rightarrow c \rightarrow t$

To apply the augment we need to take the min(4,2,2) = 2 from the above path. Now the augment is 2 we need to send 2 for both $S \rightarrow b \rightarrow a \rightarrow t$

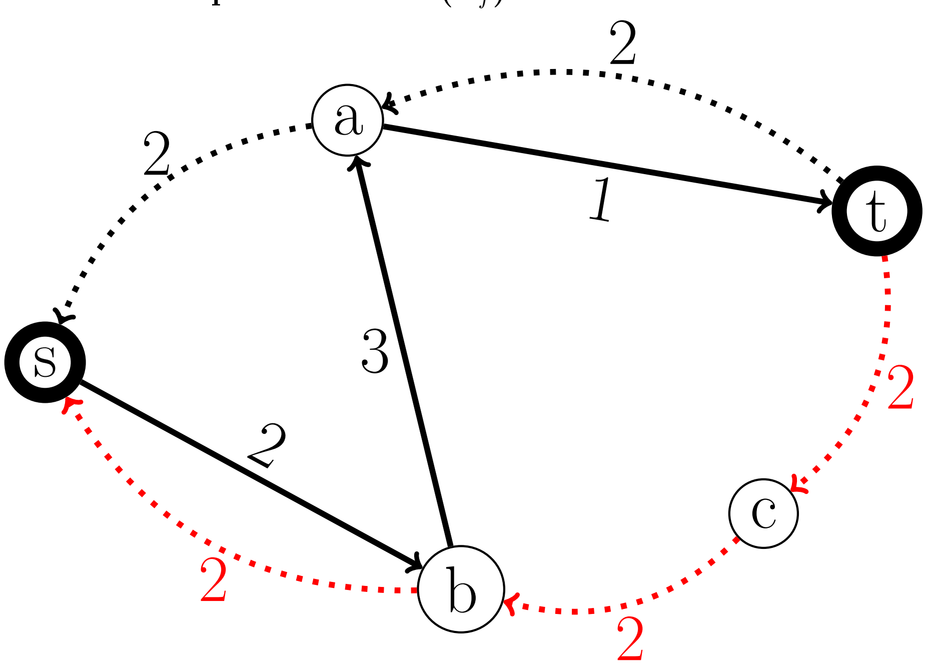
Flow graph:



Net flow : ($|f|$): 4

Residual values (r_f) will be present in the below residual graph

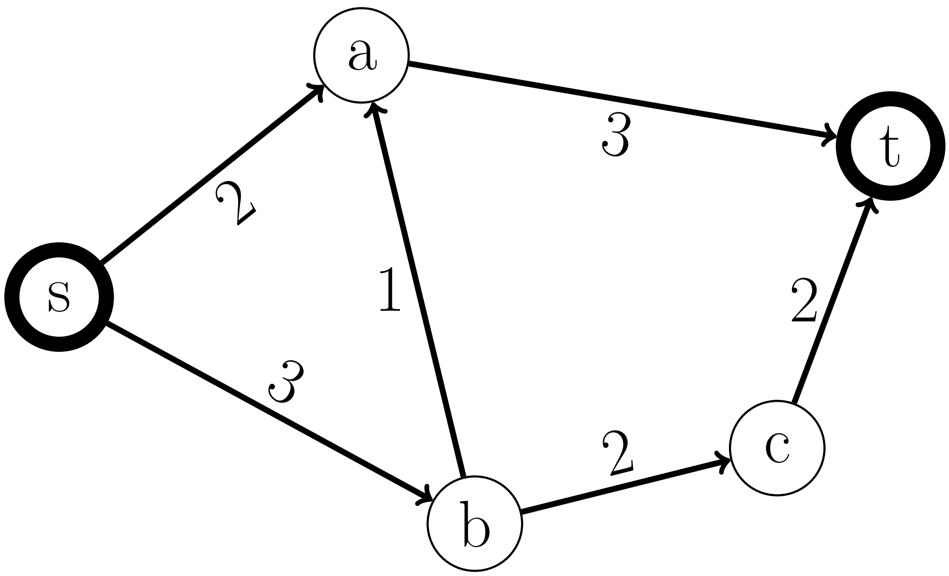
Residual Graph with values(G_f) :



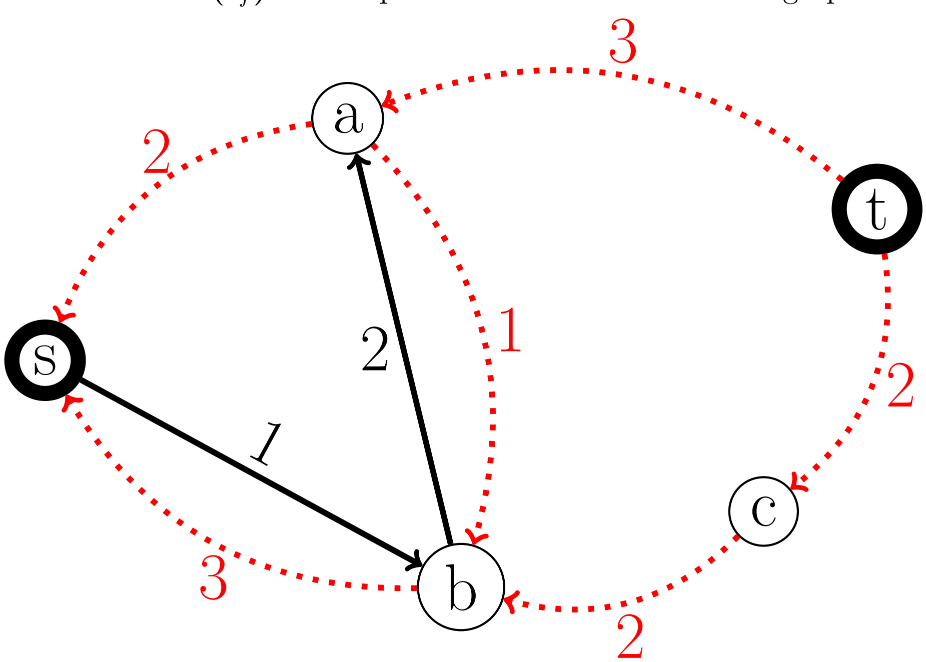
Step 3

From the above iteration of the Residual Graph let's select a path from $S \to b \to a \to t$

To apply the augment we need to take the min(2,3,1) = 1 from the above path. Now the augment is 1 we need to send 1 for $S \to b \to a \to t$

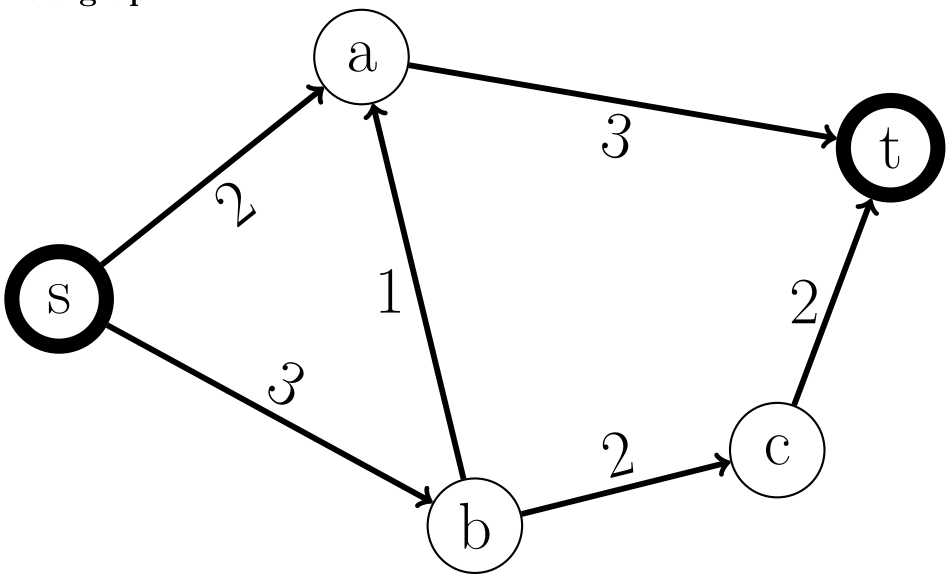


Net flow : ($|f|$): 5
Residual values (r_f) will be present in the below residual graph



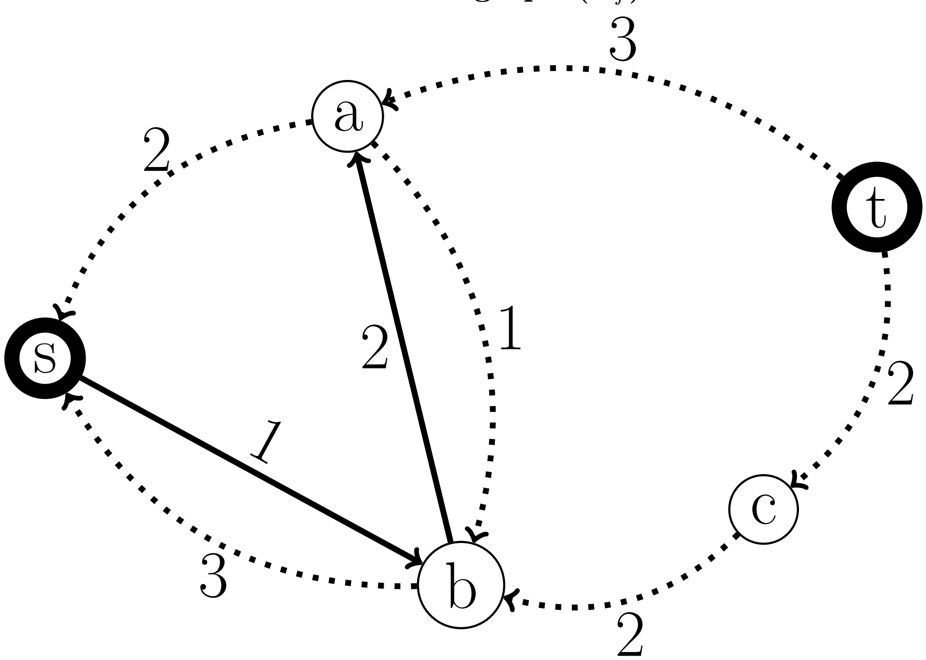
Step 4
From the above iteration of the Residual Graph, we can say that there is no path from s to t. Because the capacity is not available from a to t.

flow graph



So the max-flow will be 5 till this iteration.

Residual values with residual graph (G_f)



part b

Min $s - t$ **cut graph theorem explanation** : Given a flow f (u, v) and a cut (S, T), define the net flow across the cut

F(S, T)= $\sum_{u \in S} \sum_{v \in T} C(u, v)$

The cut that which can minimize the value of the cutting the graph, which means separating into 2 sets S and T were s in set S which is T = t and t in set T which is S =s and the value of the cut is $\sum_{u \in S} \sum_{v \in T} C(u, v)$.

Net=flow across the cut: From the above, we can conclude that the reachable vertices from s in the residual graph are one set of vertices of a cut. Then from the above residual graph clearly the vertices s, a, b are reachable from S. So one set $S = \{s, b, a\}$ and the other set $T = \{c, t\}$.

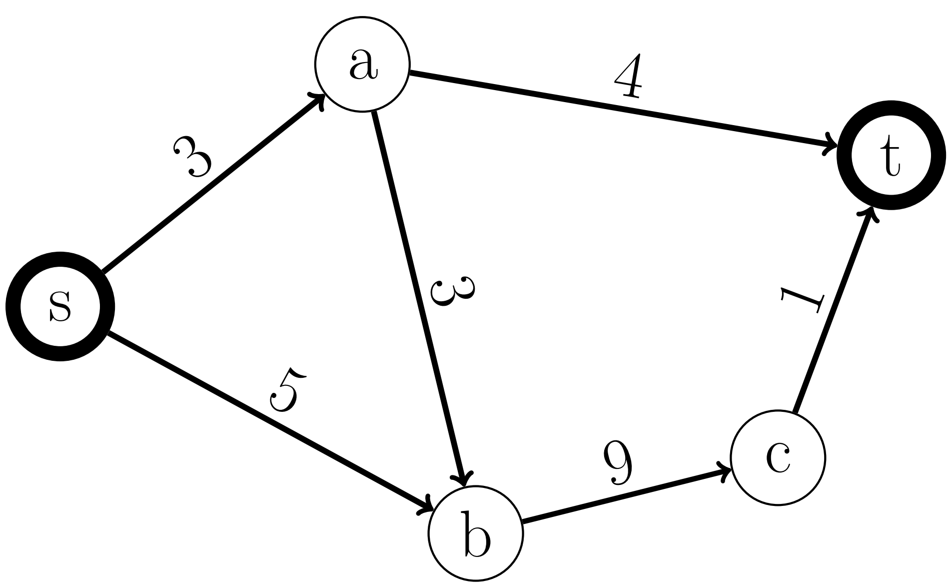
From the 2 sets, we can say that we have 2 cross edges namely $a \to t$ and $b \to c$, so the From the 2 cross edges the sum of the capacities for min-cut value for $S = \{s, a, b\}$ and $T = \{c, t\}$ as 3+2 = 5.

part C

The above cutting of the graph is $S = \{s, a, b\}$ and $T = \{c, t\}$ was a minimum cut, but it's not the only minimum, we also have another one.

Now we can check for another set i.e sets $S = \{s, a, b, c\}$ and $T = \{t\}$. here the cross edges are $a \to t$ and $c \to t$, which have edges capacities of 3 and 2. now the cut value is 5, which is equal to the previous part evaluation for different sets of S and T.

Problem 2



We are going to disprove that the above algorithm is false. From the min S-T cut, given that Initial cut is S = {s} and $T = \{a,b,c,t\}$.

For this step the min Cut is the cut is over the $S \rightarrow a, S \rightarrow b$ which are 3 and 5 respectively

$$K(S,T) = C(s,a) + C(s,b) = 3 + 5 = 8$$

Below, we will check if we add a/b/c to the set S, and whether we obtain a min-cut or not.

Adding a to the set S

let's say if we consider an into the set of S, i.e cut is S = {s,a} and $T = \{b,c,t\}$

For this step the min Cut is the cut is over the $a \rightarrow t$ as 4, $S \rightarrow b$ as 5 , $a \rightarrow b$ as 3

$$K(S,T) = C(a,t) + C(s,b) + C(a,b) = 4 + 5 + 3 = 12$$

Adding b to the set S

let's say if we consider an into the set of S, i.e cut is S = {s,b} and $T = \{a,c,t\}$

For this step the min Cut is the cut is over the $S \rightarrow a$ as 3, $b \rightarrow c$ as 9 , $a \rightarrow b$ as 3

$$K(S,T) = C(S,a) + C(b,c) + C(a,b) = 3 + 9 + 3 = 15$$

Adding c to the set S

let's say if we consider c into the set of S, i.e S = {s,c} and $T = \{a,b,t\}$

For this step the min Cut is the cut is over the $S \rightarrow a$ as 3, $S \rightarrow b$ as 5 , $c \rightarrow t$ as 1

$$K(S,T) = C(a,t) + C(s,b) + C(a,b) = 3 + 5 + 1 = 9$$

So none of them reduce the cut value. **So Algorithm stops**.

So this algorithm concludes the minimum cut value as 8, but this is false. Consider the cut $S = \{s,a,b,c\}$ and $T = \{t\}$, this has a cut value of 4+1=5, which is the minimum.
Therefore the proposed algorithm is wrong.

Problem 3

Given useful information

Here —f (u, v)— is the absolute value of the flow from u to v. This adds up every unit of flow through u twice: once for negative flow (i.e. incoming flow), and once for positive flow (i.e. outgoing flow). Therefore, we divide by 2 to get the net flow through the vertex.

solution :

From the above information, we can say that the vertex capabilities should satisfy the case for incoming and outgoing flow at that particular vertex should be less than the capacity of that vertex. (Both incoming and outcoming).

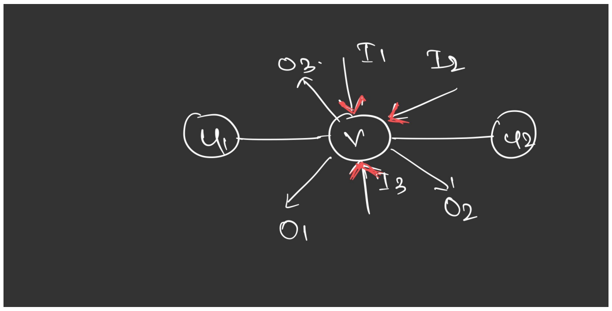
Algorithm

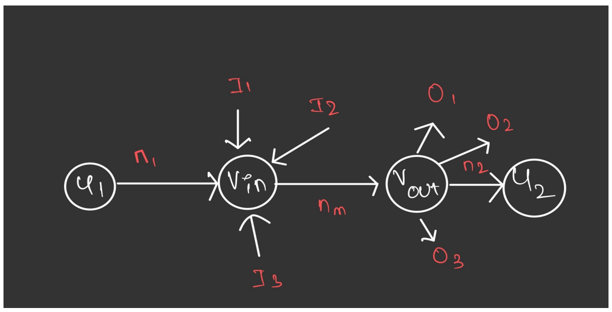
- The reduction is as follows We build a graph $G*$, which, for each node v in G with vertex capacity (C_v), has two nodes v1 and v2.
- All incoming edges to v connect to v1, and all outgoing edges from v connect from v2. Finally, add the edge (v1, v2) with edge capacity which will be the solution
- Where even the new graph will be satisfying the vertex capacities (C_v). the conversion of the graph now from G* and f* can be possible and transformed to G and f.

0.1 Reducing the graphs into f* and G*

Follow the same process for all the vertices except source (S) and Sink(t) of the given graph.

- As mentioned in the above Algorithm, the reduction is as follows We build a graph $G*$, which, for each node v in G with vertex capacity (C_v), has two nodes v1 and v2.
- To convert f to f *, set f*(u2, v1) = f (u, v) for any edge (u, v) G. we will be introducing 2 nodes for V, as v1 and v2.
- v1 can be defined as all the edges which are incoming to the V can be connected to the v1. f*(u, v1) to be the total incoming flow at node v.As the node capacity constraint on v is satisfied in G,
- v2 can be defined as all the edges which are outgoing to the V can be connected to the v2. f*(v2,u) to be the total outgoing flow at node v.As the node capacity constraint on v is satisfied in G.
- If we check, even the new graph will be satisfying the vertex capacities (C_v). As the node capacity constraint on v is satisfied in G, $f * (v1, v2) \leq C_v$,
- Now finds the maxflow of the graph G* and f*. The below step is after finding the max-flow of the graph G*.
- This step will be done after we find the max value of the G* and f*, the conversion of the graph now from G* and f* can be possible and transformed to G and f.





I1, I2, and I3 are the incoming edges, O1, O2, and O3 are the outgoing edges, and n_m is the new edge connecting the v1 and v2 in the graph which satisfies the capacity constraint. n1 and n2 are the new edges that satisfy the capacity constraint

Proof

- **capacity Constraints of edges** Now any flow f in G which respects the node capacity constraints can be converted to a flow f* in G* which also respects the edge capacity constraints. To convert f to f *, set f*(u2, v1) = f (u, v) for any edge (u, v) G, and f*(v1, v2) to be the total incoming flow at node v. As the node capacity constraint on v is satisfied in G, $f*(v1, v2) \leq cv$, and hence f* satisfies the edge capacity constraints in G*. f* also satisfies conservation constraints because f satisfies conservation constraints and because of the way we set f*(v1, v2).
  Conclusion : $f * (v1, v2) \leq cv$,

- **vertex Capacity constraints** From the above graph, consider the (v_{in},v_{out}) edge. The max flow algorithm on G* ensures that must be less than the $k(v)$ as it flows through it for the transformation. Now consider the node v_{in}, it has all incoming edges of v in G and only one outgoing node. Since the net incoming flow is equal to net outgoing at vertex v_{in}, and the only outgoing from it is the edge (v_{in},v_{out}), the net incoming flow to v_{in} should have a bound of k(v), which is equivalent to saying that net incoming flow to v in G under a constraint of $k(v)$.

- **Algorithm termination** The residual capacity of each edge declines with each iteration and eventually reduces to zero.No further flow exists if the residual capacity reaches zero.

- **max flow** In the graph, the source s and sink t are not changed and remain the same G* as well. There is no change in outgoing and incoming edges for s and t for new and old graphs, Only intermediate vertex only changes.