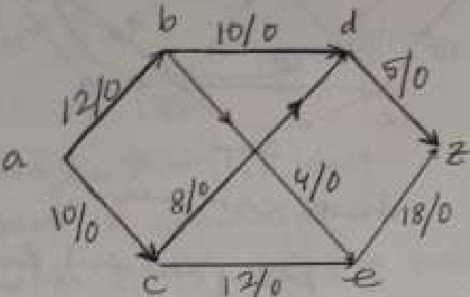
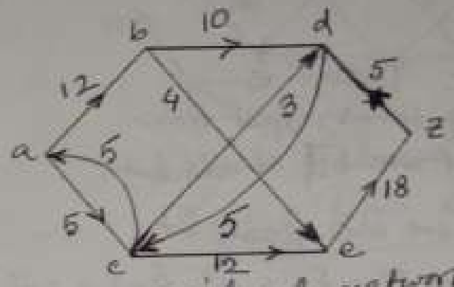
20/20

Sandipan Dey Math-685 HW 4.3

2.(a) Start with 0 flow

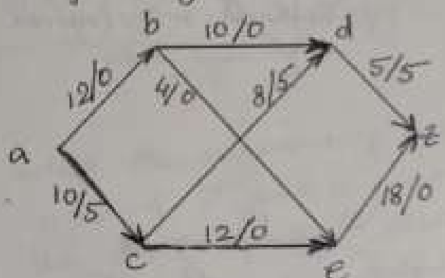


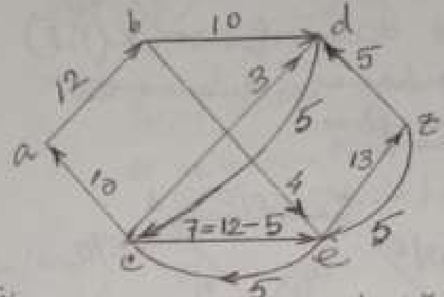


Choose path a-c-d-z with min (bottleneck) slack of 5 units (along $\vec{dz}$) and augment the flow by 5 additional units

residual network after selecting the augmenting path a-c-d-z

Next select the augmenting path a-c-e-z



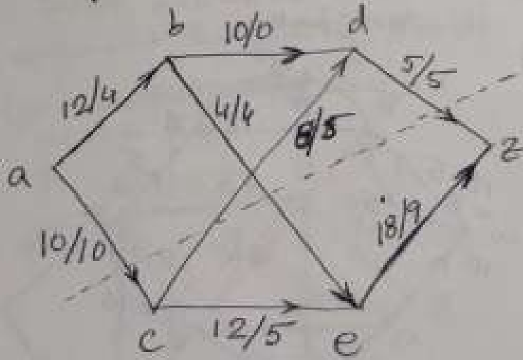


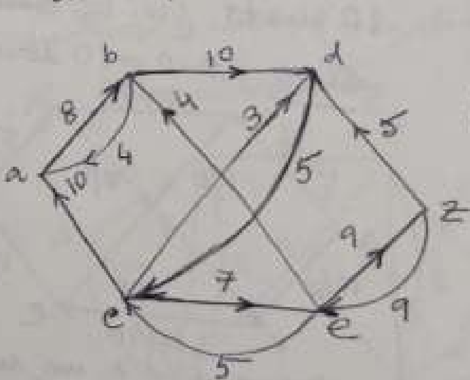
10/10

· Choose path a-c-e-z with bottleneck residual capacity of 5 units (along $\vec{ac}$, 10-5) and send 5 additional units to get the next residual network

residual network.

Next choose the path a-b-e-z as augmenting path with bottleneck residual critical capacity as 4 (on $\vec{be}$) and augment the flow by 4 additional units.





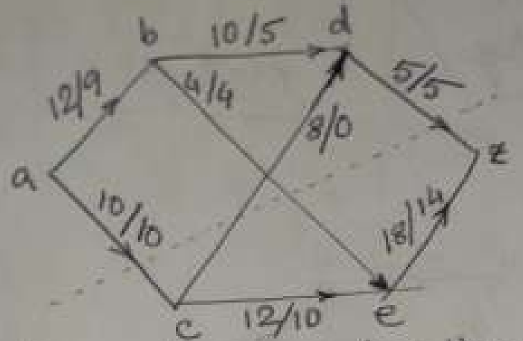
In residual graph as can be seen ∃ a path a-b-d-c-e-z of residual capaci[illegible] 5 units ($\kappa(\vec{dc})$ [illegible]

Next time choose this path and augment the flow by another 5 units

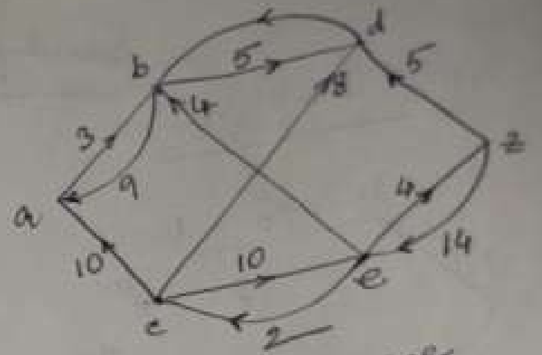
Choose path a-b-e-z and consider $P=\{a,b,d\}$ $\bar{P}=\{c,e,z\}$

Hence the cut $(P,\bar{P})$ containing edges $\vec{ac}, \vec{be}, \vec{dz}$ is saturated with $\kappa(P,\bar{P}) = 10+4+5=19$, but flow is $5+5+4=14$ units! some path must have been chosen by m[illegible]

By sending 5 units of flow back through cd

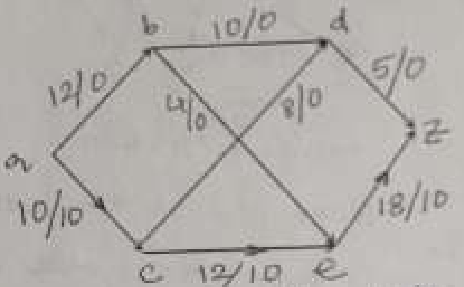


As can be seen, last time we did not use $\overline{bd}$ at all but this time we are using it. and again $P = \{a, b, d\}$, $\bar{P} = \{c, e, z\}$, the cut $(P, \bar{P})$ ~~becomes~~ remains saturated, with $K(P, \bar{P})$ but this time with flow $= 5 + 14 = 19$, which is equal to the cut size $K(P, \bar{P})$ hence we have achieved the maximum flow.

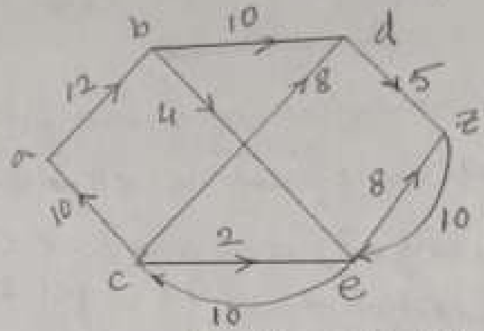


As we can see $\nexists$ a path (directed) from a to z in the residual network, hence the flow is indeed maximum.

Choosing alternative path

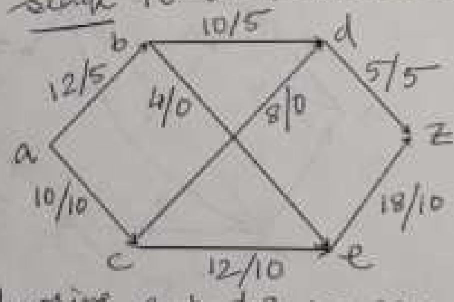


Choose augmenting path a-c-e-z first, in order to send an additional flow of 10 units. (with ~~max~~ min. slack $10-0=10$ across $\vec{ac}$)

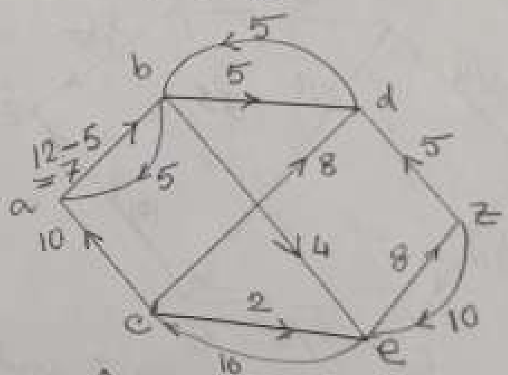


residual network,

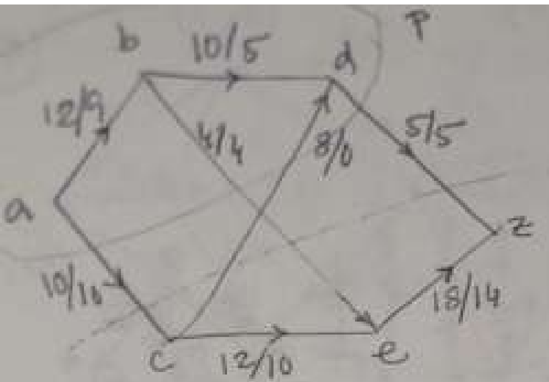
Choose a-b-d-z as the next path

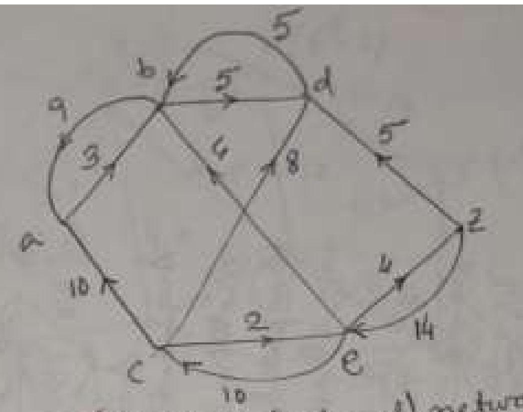


Choosing a-b-d-z, we see that the residual capacity is 5 units (along $\vec{dz}$)



As we can see $\exists$ a path a-b-e-z from a to z with min residual capacity 4 (along $\vec{be}$) and we can still send an additional 4 units

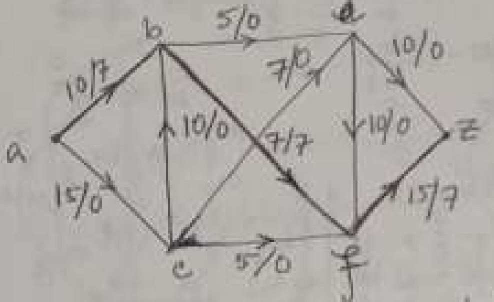




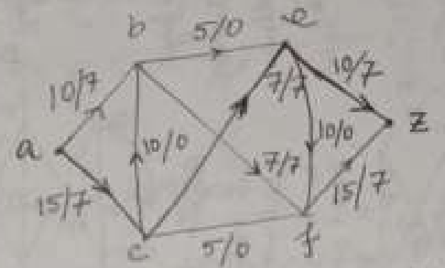
We see that $(P, \bar{P})$ cut becomes saturated now, with $K(P, \bar{P}) = f = 19$ units

Residual (reduced) network, $\nexists$ a path from a to z.

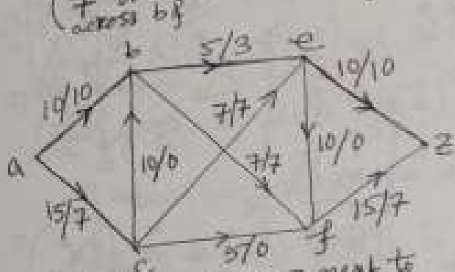
(b)



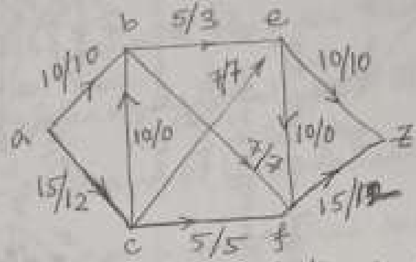
Select the augmenting path a – b – f – z and augment the flow by min slack (7 units across $\vec{bf}$) and augment the flow



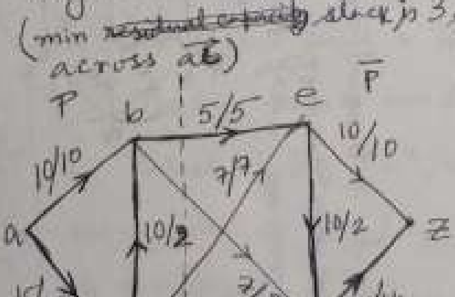
Next chose a-c-e-z to get another additional flow of 7 units (min slack is across $\vec{ce}$)



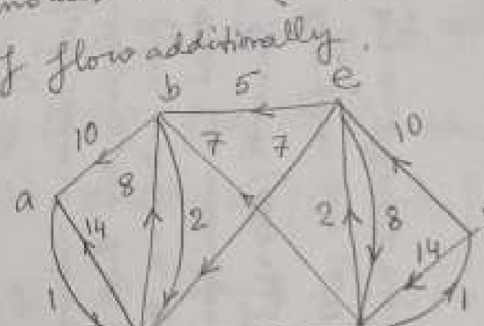
Select a-b-e-z next to augment the flow by 3 units (min ~~residual capacity~~ slack is 3, across $\vec{ab}$)



Next select path a-c-f-z to get another 5 units (min slack $\vec{cf}$ being 5 units) of flow additionally.

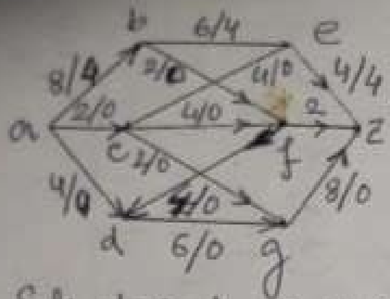


Choose a-c-b-e-f-z as next augmented path and send another 2 units of flow across it. Now consider the cut $(P, \bar{P})$ with $P = \{a, b, c\}$. $K(P, \bar{P}) = 24$ units $= f$

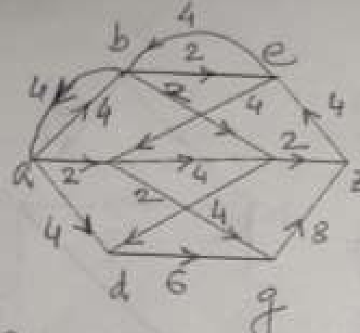


Max flow achieved

Residual (reduced) network does not have any directed path from a to z. Also, $f = 10 + 14 = 24$ units $= K(P, \bar{P})$, hence max flow

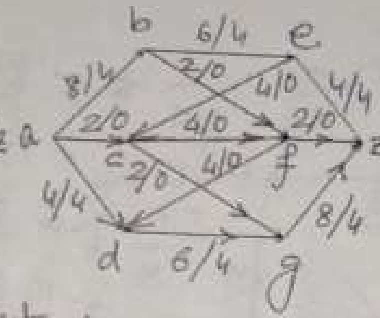


Select a-b-e-g as augmenting path with min slack as $\min\{8,6,4\} = 4$ units as flow to augment

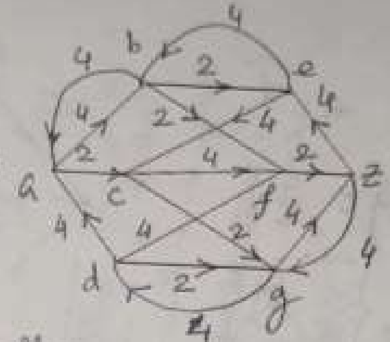


residual network

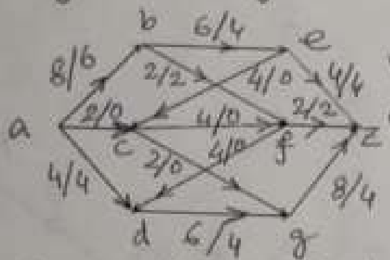
Select the path a-d-g-z as next augmenting path with min slack 4



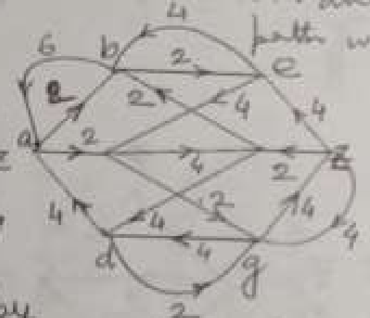
Augment flow across the path a-d-g-z by 4 units



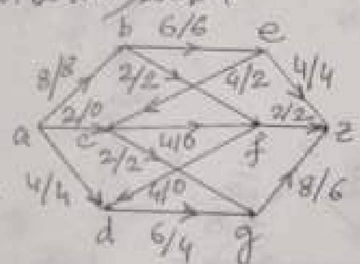
Now from the reduced network choose the path a-b-f-z



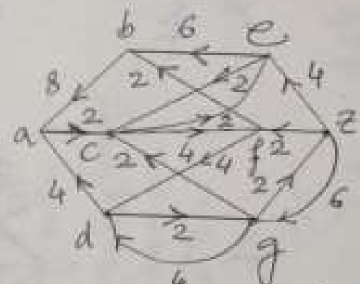
Augment the flow by another 2 units through the path abfz (with min slack $\vec{bf}$ or $\vec{fz}$)



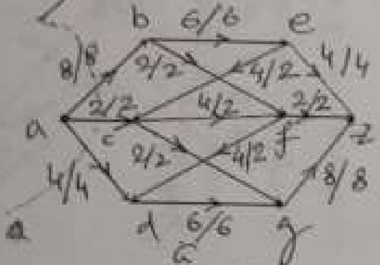
Again from the reduced network choose a-b-e-c-g-z as next augmenting path



Augment flow by another 2 units through the chosen path

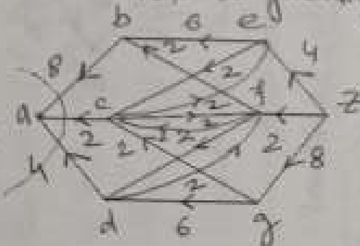


from reduced network we can see that ∃ a path from a to z, i.e. a-c-f-d-g-z that has minimum ~~flow~~ slack 2 units (across $\vec{dg}, \vec{gz}$), hence the flow can still be augmented by another 2 units



Choose $P = \{a\}$

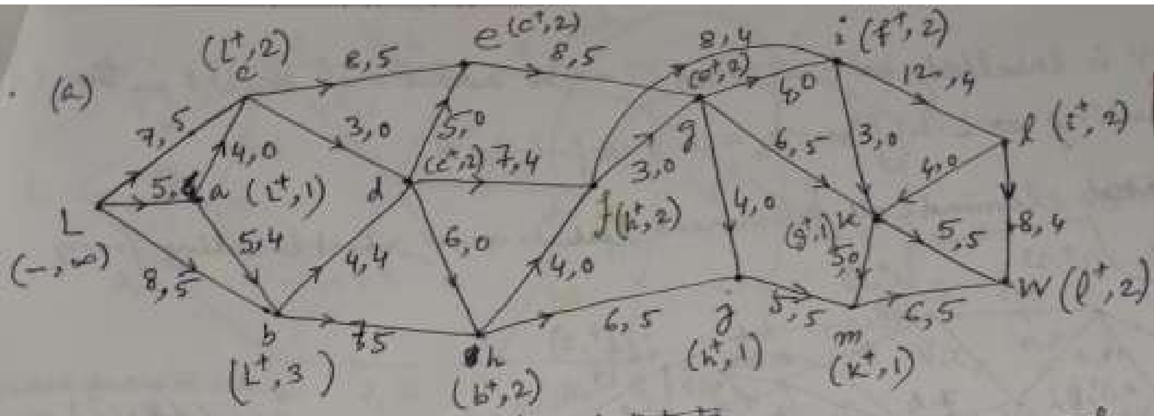
$\bar{P} = G - \{a\}$



∄ a path from a to z in the residual network.

$K(P, \bar{P}) = 8 + 2 + 4 = 14$ units, which is achieved here. Hence this is the max flow, since $K(P, \bar{P})$ is equal to the flow value $= 14$ units.

(a)



10/10

~~Select the first augmenting path to be~~

By inspection, let the initial flow be $f = 5f_{K_1} + 4f_{K_2} + 5f_{K_3}$, where

$K_1 = L-c-e-g-K-W$, $K_2 = L-a-b-d-f-i-l-W$,

$K_3 = L-b-h-j-m-W$

Now apply labeling algorithm to the flow f.

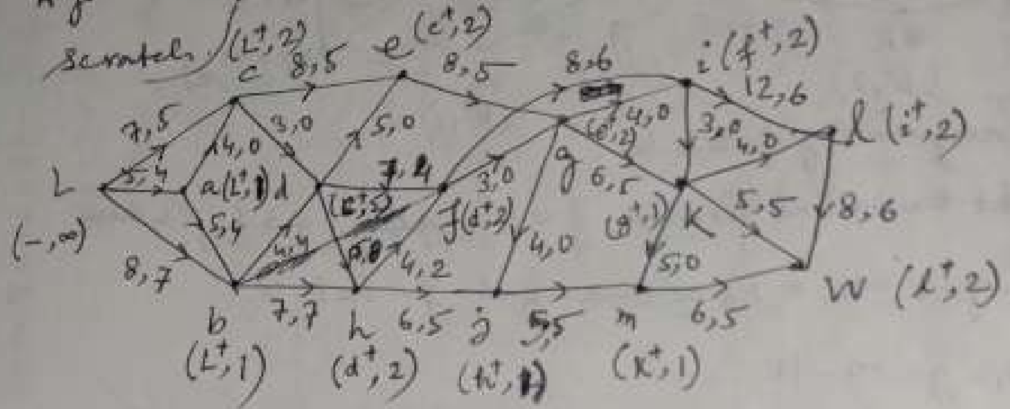
Scanning edges at L, there can't be incoming edges at L (source) with flow, but there are outgoing edges $\vec{La}, \vec{Lb}, \vec{Lc}$, having positive slacks $s(\vec{L,a}) = 5-4=1$, $s(\vec{L,b}) = 8-5=3$, $s(\vec{L,c}) = 7-5=2$ respectively and unlabelled, hence label by $(L^+,1)$, $(L^+,3)$ and $(L^+,2)$ respectively, since $\Delta L = \infty$ and $\min(\Delta(a), s(a)) = 1$ etc. Next scan a (in order the vertices were labelled) the incoming edge $(\vec{L,a})$ with $s(\vec{L,a}) = 1$ is already labelled and ~~that~~ the outgoing edges $\vec{a,c}$ has slack $s(\vec{a,c}) = 4-0 > 0$ and c is already labelled, so ~~label c by~~ $(a^+,$ ~~$\min[\Delta(a), s(c)]$~~$)$ is b. Next scan b and since incoming edges are already labelled, consider outgoing edges $\vec{b,d}$ with $s(\vec{b,d}) = 0$ but $s(\vec{b,h}) = 7-5 = 2$ and h is unlabelled, hence label h by $(b^+, \min[\underset{3}{\Delta b}, \underset{2}{s(h)}])$

h e d
f j g
i k
l m

Similarly, scan c and label e by $(c^+, 2)$, d by $(c^+, 2)$. Next scan h, label f by $(h^+, \min[\underset{2}{\Delta h}, \underset{4}{s(f)}])$, label j by $(h^+, \min[\underset{2}{\Delta h}, \underset{1}{s(j)}])$. Next scan e and label g by $(e^+, 2)$. Scan d next and there are no unlabelled vertices adjacent to d. Scan f and label i by $(f^+, 2)$. Scan j and ~~label m by~~ since slack $s(\vec{j,m}) = 0$ can't label m. Scan g and label K by $(g^+, 1)$. Scan i and label l ~~do nothing~~. Scan K next, can't label l with incoming flow & label m by $(K^+, 1)$, W ~~by~~ can't be labelled with slack 0

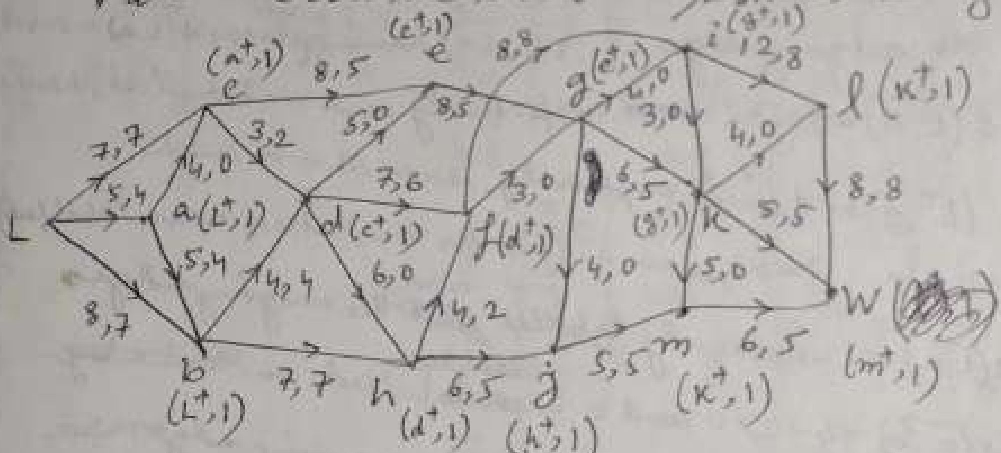
Since W is labelled, we can send 2 = ΔW additional units with augmenting L-W flow $k_4 = L-b-h-f-i-l-W$

Again ~~start~~ eliminate all the current labels and start labeling from scratch

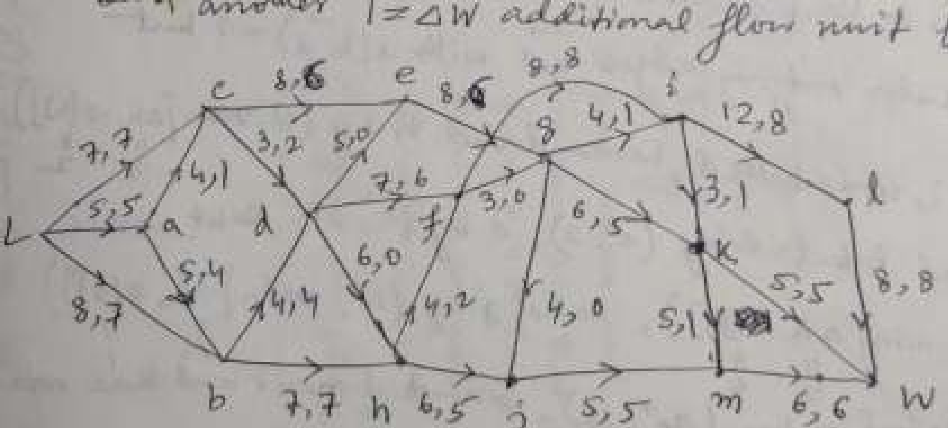


Again send another 2 = ΔW additional units through $W_5 = L-c-d-f-i-l-W$

Eliminate all the labels and restart labelling



Send another 1 = ΔW additional flow unit through $W_6 = L-a-c-e-g-K-m-W$



Consider $W = \overline{P}$

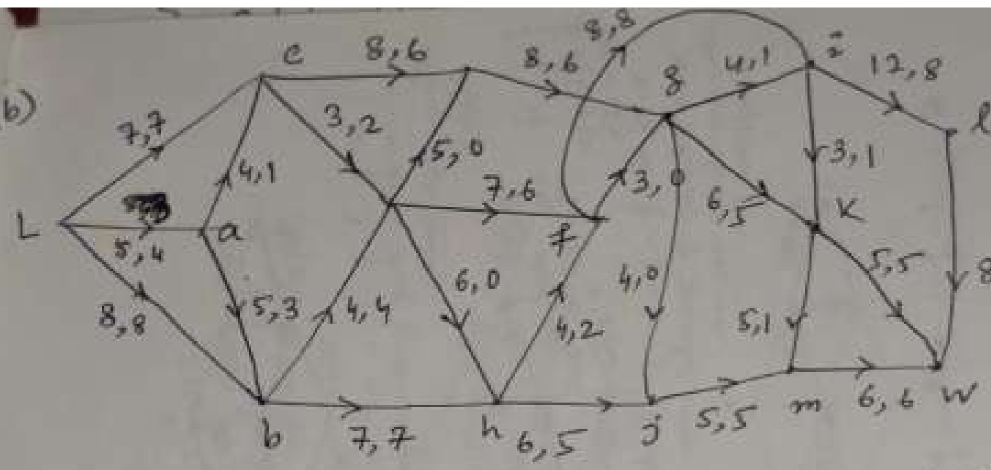
We can see

$K(P, \overline{P}) = f = 5+6+8$

$= 19$ units

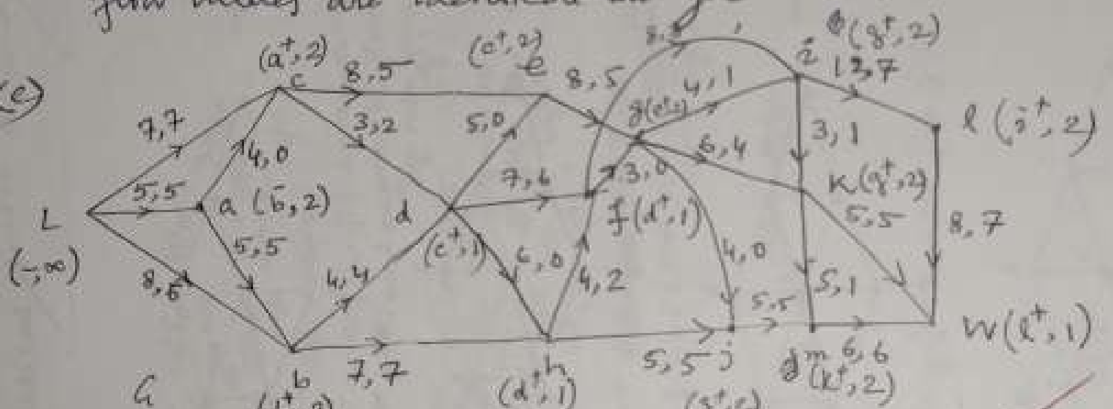
and maximum flow is achieved.

(b)



This is a different max-flow, since it differs from the previous one in $f(L,a)=4 \neq 5$, $f(L,b)=8 \neq 7$, $f(a,b)=3 \neq 4$, all other flow values are identical through

(c)

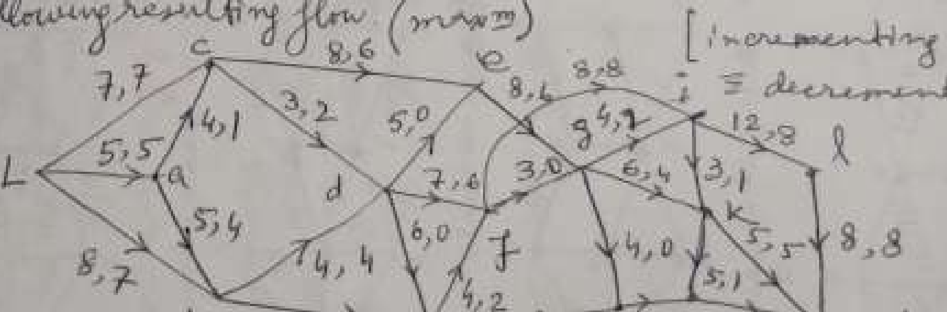


~~b d f d k f i g k j~~ order in which the vertices are labelled

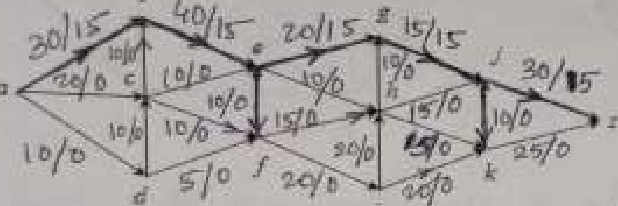
Consider $P=\{L,b\}$, $\bar{P}=G-P$ and $(P,\bar{P})$ cut which has edges $\vec{Lc}$, $\vec{La}$, $\vec{bd}$, $\vec{bh}$ all are saturated, with $K(P,\bar{P}) = 7+5+4+7 = 23$ units, but $f = 7+5+6 = 18$ units

(d) Applying the algorithm, we can ~~sed~~ send $1=\Delta W$ additional unit in the augmenting L-W flow $K = L-b-a-c-e-g-i-l-W$ and we get the following resulting flow (max^m)

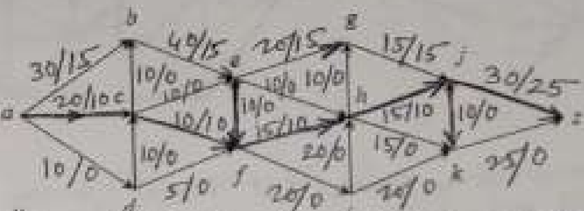
[incrementing flow in $\vec{ba}$ direction ≡ decrementing in $\vec{ab}$]



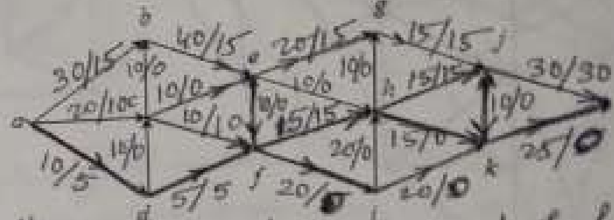
8. (a) Extend the network by adding hypothetical source a and sink z



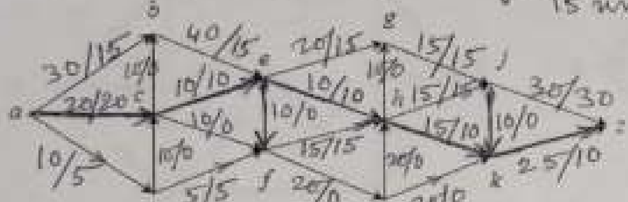
Select a-b-e-g-j-z as augmenting path with min slack 15 (across $\overrightarrow{gj}$) and send 15 units flow



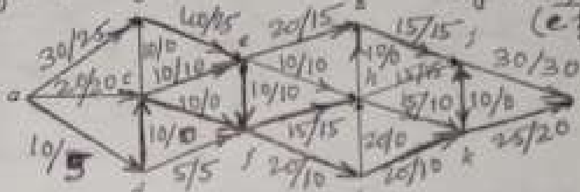
Select a-c-f-h-j-z as augmenting ~~flow~~ path and augment the flow by min slack 10 units ($\overrightarrow{ef}$)



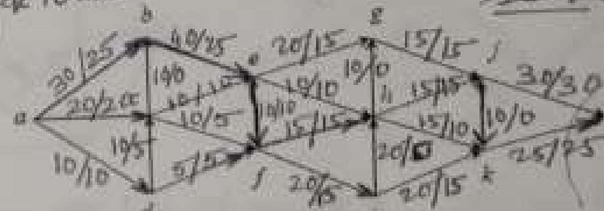
Next choose a-d-f-h-j-z as augmenting path with min slack as 5 units ($\overrightarrow{df}$)



Next choose ~~[illegible]~~ a-c-e-h-K-z (min slack 10)



Next select ~~[illegible]~~ a-b-e-f-i-K-z (min slack ~~[illegible]~~ 10 units along $\overrightarrow{ef}$)



$\bar{P} = \{z\}$

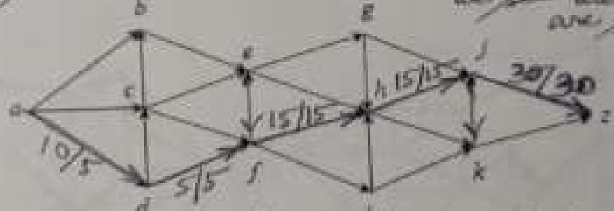
$\kappa(P, \bar{P}) = f$
$= 30 + 25$

Next select a-d-c-f-i-K-z with min slack 5 units and we see the demands are satisfied.

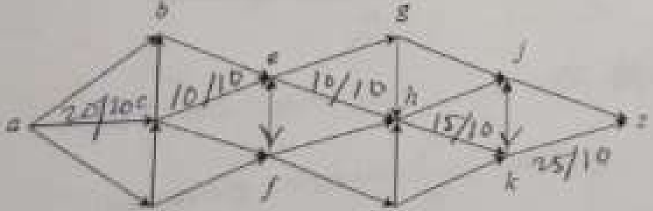
(b)

augmenting path a-b-e-g-j-z

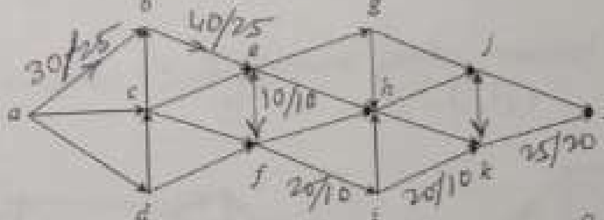
augmenting path a-c-f-h-j-z



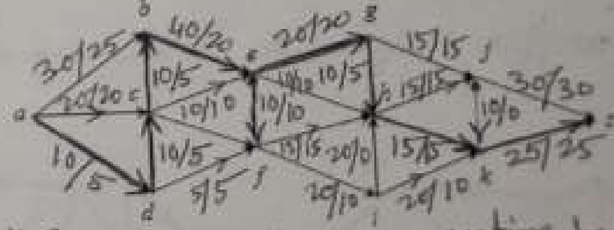
augmenting path a-d-f-h-j-z



augmenting path a-c-e-h-K-Z



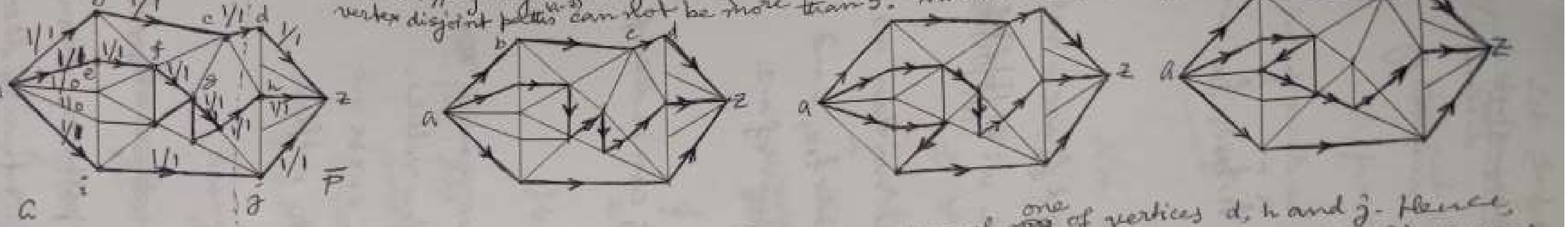
augmenting path a-b-e-f-i-K-z



∃ another augmenting path now after changing direction of (h, g)

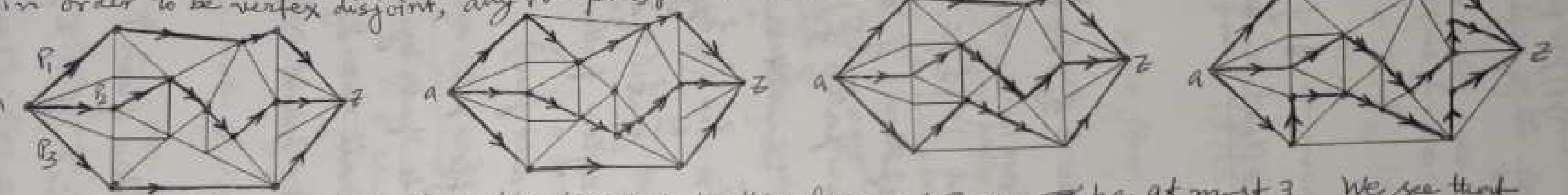
namely a-d-c-b-e-g-h-K-z.

14. As we can see, total # of edges coming into z (i.e. in-degree of z) is 5, size of the largest set of vertex disjoint paths$^{(a-z)}$ can not be more than 5. Also notice that if we let $\bar{P} = \{z, d, h, j\}$

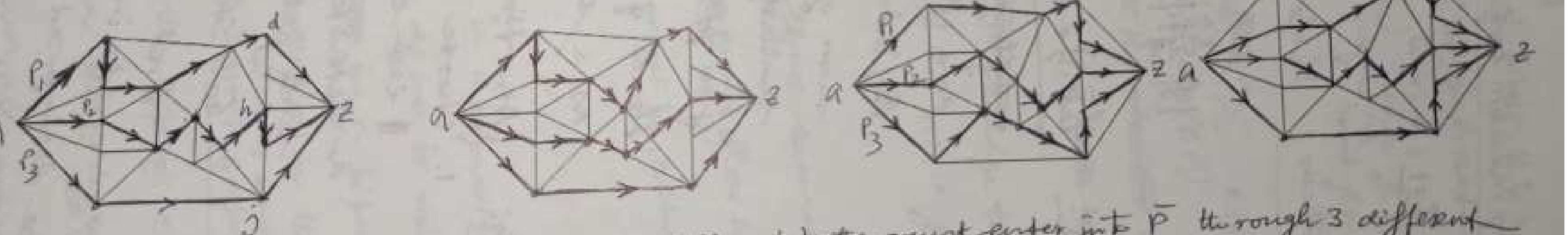


G

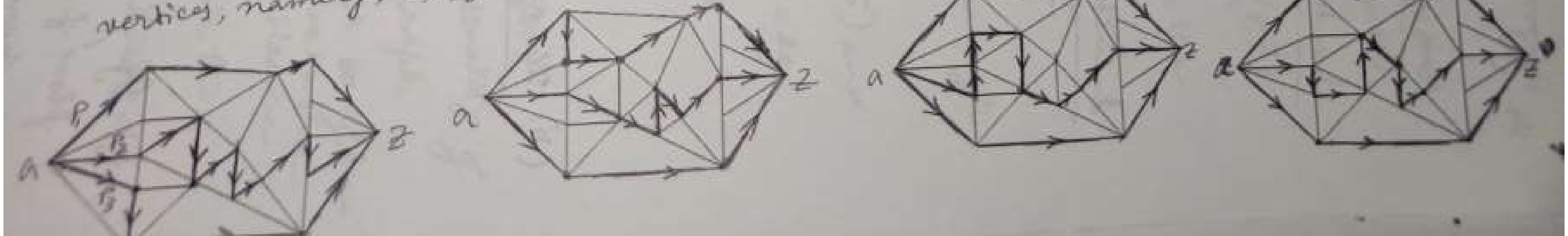
and let $P = G - \bar{P}$, all the edges coming from P to $\bar{P}$ must come through one of vertices d, h and j. Hence, in order to be vertex disjoint, any two paths from a in P must enter $\bar{P}$ in two different vertices, i.e., in $\binom{3}{2} = 3$ ways.



Hence, the largest set of vertex disjoint paths from a to z can be at most 3. We see that there exist many such 3-paths set, as shown in the example



As we can see, in all the cases, 3 different paths (P_1, P_2, P_3) must enter into $\bar{P}$ through 3 different vertices, namely, d, h, j



24. (a) Proof (by induction on the value of f): (∀f) f is circuit free (assign unit capacity ... direction)

Base case: $f=1$: trivial, ~~...~~ source and sinks are connected by an edge.

Induction hypothesis: Let's ~~assign~~ assume ~~f~~ f can be decomposed ... a ~~...~~ sum of a-z flow paths ~~...~~ $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_K$, wh... $|f| = \sum_{i=1}^{K} \hat{f}_i| \leq n$.

Induction Step: Let $f = n+1$. Then $\exists f_1, f_2 \mid f_1$ and f_2 are valid ~~a-z flow paths~~ flows in the network with $|f| = |f_1| + |f_2|$ and $1 \leq |f_1| \leq n$ and $1 \leq |f_2| \leq n$.

By induction hypothesis f_1 can be written as sum of a-z flow paths $\hat{f}_{11}, \hat{f}_{12}, \ldots, \hat{f}_{1K_1}$ and similarly f_2 can be written as sum of a-z flow ~~paths~~ paths $\hat{f}_{21}, \hat{f}_{22}, \ldots, \hat{f}_{2K_2}$.

Hence $f = f_1 + f_2 = \sum_{i=1}^{K_1} f_{1i} + \sum_{j=1}^{K_2} f_{2j}$ can be decomposed as a sum of a-z flow paths.

(b) The algorithm: (Assumption: f is circuit free).

1. Choose any a-z flow path and decrement ~~the~~ ~~f~~ each edges on the path by the flow value.

~~2. Go to Step 1 until all the~~ If there exists no such path, goto step 3 and output the net flow.

2. Go to Step 1.

3. End.

(c) If $\nexists$ ~~a flow~~ circuits in any of the flows f, the ~~above algorithm~~ flow f can be decomposed into a set of a-z flow paths as shown in part (a). If the ~~...~~ flows contains circuits, f can be eventually decomposed into the a-z flow paths plus the set of circuits

Let C be a given cut, $C = (P, \bar{P})$
27. Each unit flow path ~~crosses~~ f_K where K is a path crosses the cut C exactly once. Also C is saturated.

To prove: $f = \sum_{K \in \mathcal{K}} f_K$ is a maxflow, where $\mathcal{K}$ is set of all unit flow a-z paths.

Proof We know $|f| \le \sum_{e \in (P,\bar{P})} f(e) \le \sum_{e \in (P,\bar{P})} \kappa(e) = \kappa(P, \bar{P})$.

Now each unit flow path ~~cuts~~ K ~~cuts P,P~~ crosses the cut $(P, \bar{P})$ exactly once $\Rightarrow \exists\, e \in (P, \bar{P})$ ~~s.t.~~ $\forall K \in \mathcal{K}$ and $e \in K$.

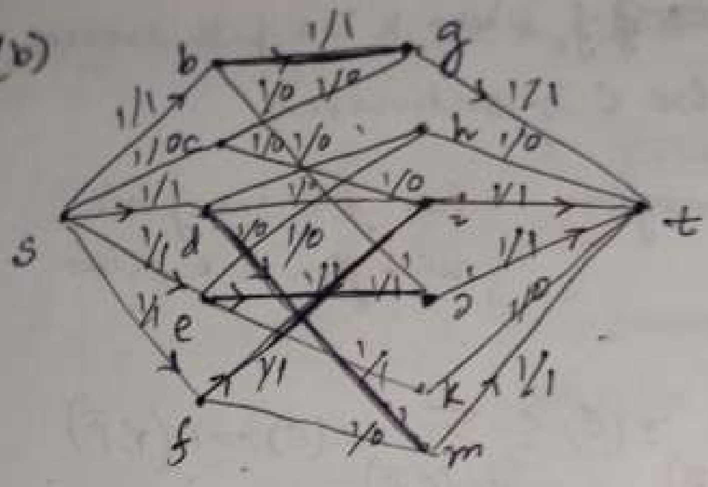
$\Rightarrow \forall$ unit flow path K $\exists$ at least one edge that is already saturated. can't send more flow through ~~the~~ K (can't augment K)

Moreover sum of ~~Hence~~ all ~~the~~ such unit flows saturates $(P, \bar{P})$ and $f = \kappa(P, \bar{P})$, since every unit flow has an edge $e \in (P, \bar{P})$ and that edge must be saturated.

Hence, f is a max-flow.

32. (a) By König-Egervary theorem in a bipartite graph, max cardinality of matching is equal to min-cardinality of a vertex cover. [Max-Matching ~~...~~ in a bipartite graph can be found by finding maxflow in the graph ~~to~~ after assigning each of its edge a 1 unit capacity on adding ~~a sink~~ a source & a sink]. Hence ~~first~~ find a max-flow in the bipartite graph, the flow ~~...~~ value will be equal to the ~~...~~ min vertex cover (min number of edges reqd to cover all vertices)

(b)



As can be seen,

~~the s-t flow~~ there are 4-units of flow through s-b-g-t, s-d-m-t, s-e-j-t and s-f-i-t. Hence min vertex cover will be the set ~~Ed~~ of edges

{(b, g), (e, j), (d, m), (f, i)}