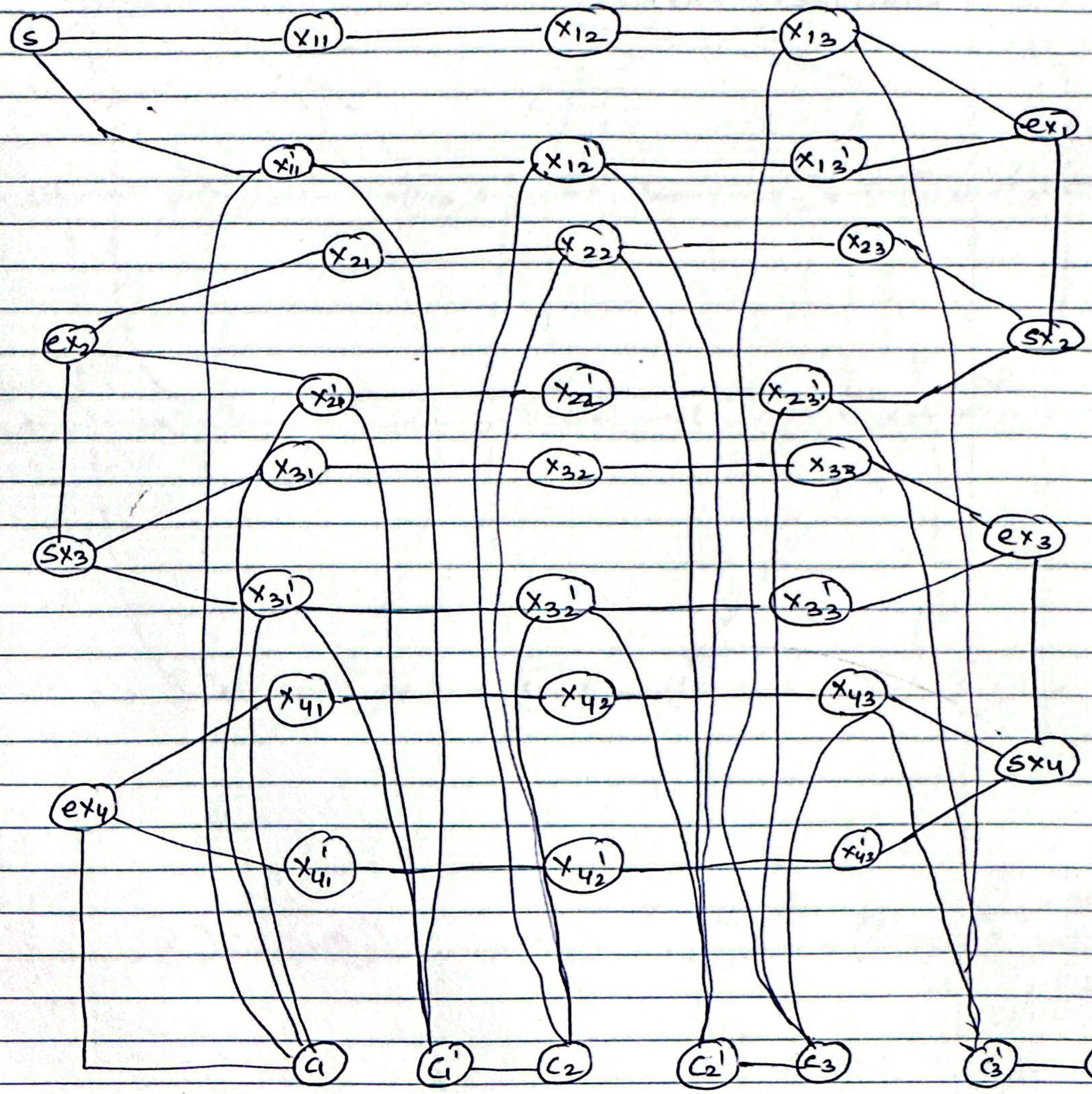
TTFF

1.a) Given $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_4)$

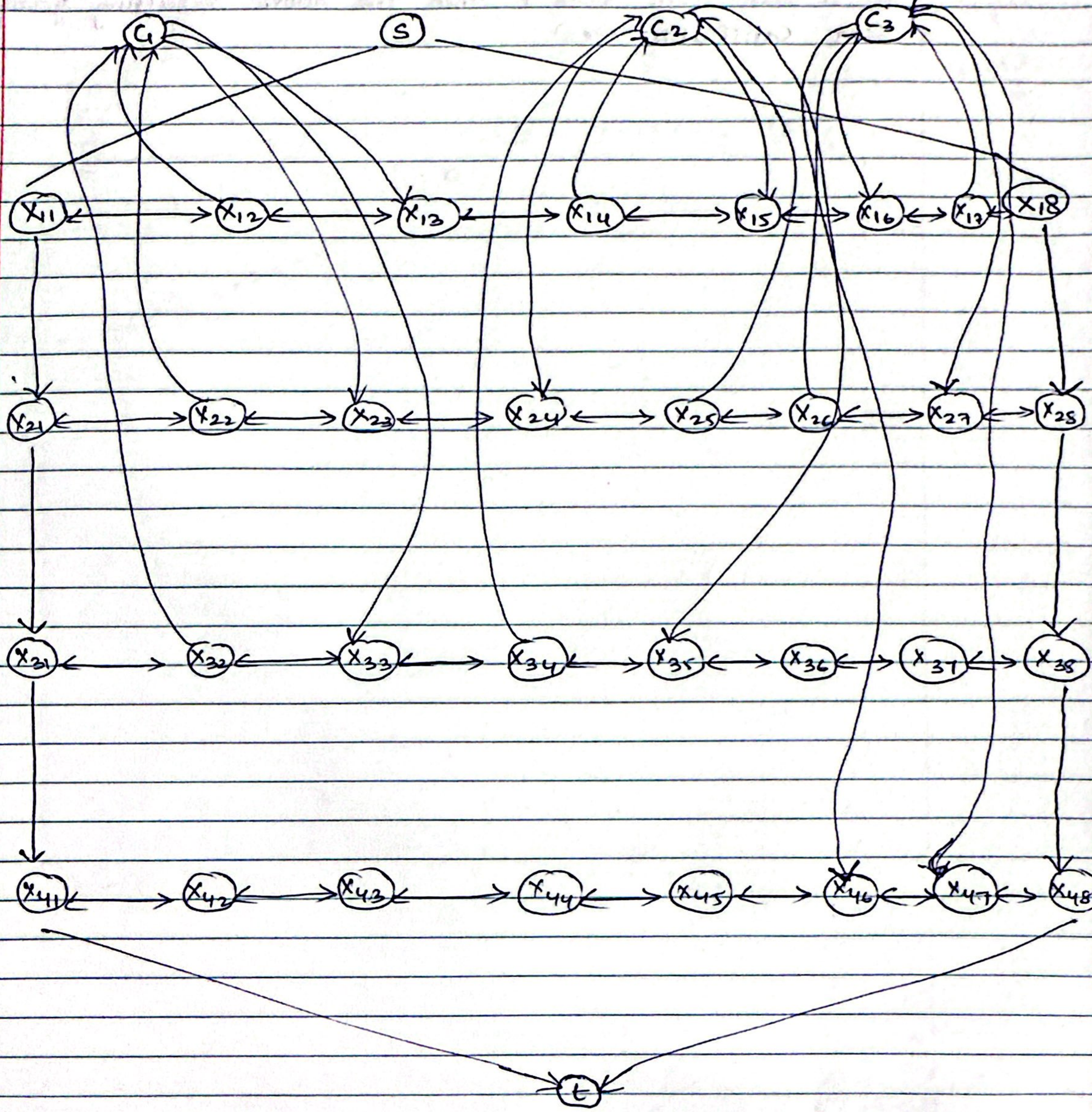



$x_1 = T, \ x_2 = T, \ x_3 = F, \ x_4 = F$

1.b) For the above assignment $x_1 = T$, $x_2 = T$, $x_3 = F$ and $x_4 = F$ if there a path from S to T then the resulting gadget is 3SAT satisfiable. (Yes)

TTTT

2.a) Given, $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_4)$




DIRECTED-HAM CYCLE

$x_1 = T,\ x_2 = T,\ x_3 = T,\ x_4 = T$

2.b) For the above assignment $X_1 = T$, $X_2 = T$, $X_3 = T$ and $X_4 = T$ if there is a path from S to T then the above resulting gadget is 3SAT satisfiable. (Yes)

Chandana Ganasale
ID : 3013823394.

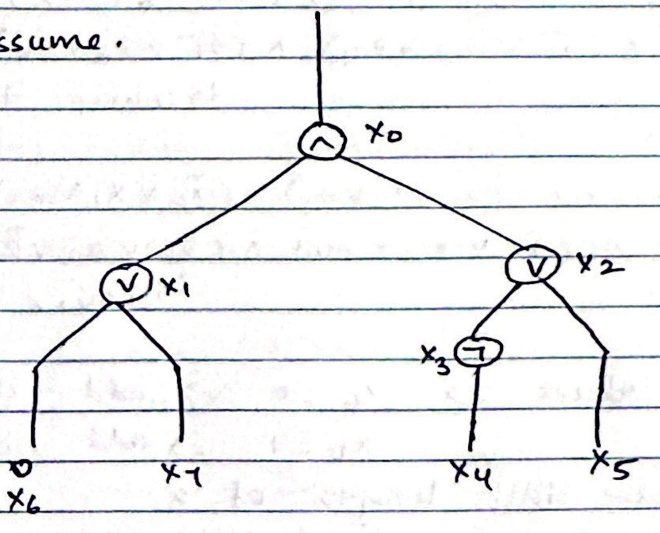
# Assignment - 3

3.a) CIRCUIT SAT $\leq_p$ 3-SAT

proof:

1. Let K be any circuit
2. Create a 3-SAT variable $x_i$ for each circuit element $i$.
3. Make circuit complete correct values at each node:
   - $x_3 = \neg x_4$ => add 2 clauses : $x_3 \vee x_4$, $\bar{x}_3 \vee \bar{x}_4$
   - $x_2 = x_3 \vee x_5$ => add 3 clauses : $x_2 \vee \bar{x}_3$, $x_2 \vee \bar{x}_5$, $\bar{x}_2 \vee x_3 \vee x_5$
   - $x_1 = x_6 \vee x_7$ => add 3 clauses : $x_1 \vee \bar{x}_6$, $x_1 \vee \bar{x}_7$, $\bar{x}_1 \vee x_6 \vee x_7$
   - $x_0 = x_1 \wedge x_2$ => add 3 clauses : $\bar{x}_0 \vee x_2$, $\bar{x}_0 \vee x_1$, $x_0 \vee \bar{x}_1 \vee \bar{x}_2$
4. Hard-coded input values and output value.
   - $x_6 = 0$ => add 1 clause : $\bar{x}_6$
   - $x_0 = 1$ => add 1 clause : $x_0$
5. Finally turn clause of length less than 3 into clauses of length exactly 3.

b) let's Assume.




$x_3 = \neg x_4$ => add 2 clauses : $x_3 \vee x_4$, $\bar{x}_3 \vee \bar{x}_4$

$x_2 = x_3 \vee x_5$ => add 3 clauses : $x_2 \vee \bar{x}_3$, $x_2 \vee \bar{x}_5$, $\bar{x}_2 \vee x_3 \vee x_5$

$x_1 = x_6 \vee x_7$ => add 3 clauses : $x_1 \vee \bar{x}_6$, $x_1 \vee \bar{x}_7$, $\bar{x}_1 \vee x_6 \vee x_7$

$x_0 = x_1 \wedge x_2$ => add 3 clauses => $\bar{x}_0 \vee x_2$, $\bar{x}_0 \vee x_1$, $x_0 \vee \bar{x}_1 \vee \bar{x}_2$

Now we have to turn clause of length < 3 to clauses of length exactly 3.

For $x_3 = \neg x_4$ => $(x_3 \vee x_4 \vee z_i)$, $(\bar{x}_3 \vee \bar{x}_4 \vee z_i)$,
$(\bar{z}_i \vee \bar{z}_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee \bar{z}_4) \wedge (\bar{z}_i \vee \bar{z}_3 \vee \bar{z}_4)$
↳ where $z_i$ ($i = 1$ and $2$)

Total we have 10 clauses.

For $x_2 = x_3 \vee x_5$ => $(x_2 \vee \bar{x}_3 \vee z_1)$, $(x_2 \vee \bar{x}_5 \vee z_1)$, $(\bar{x}_2 \vee x_3 \vee x_5)$
now here we already know $x_3$ have 2 clauses initially those will be considered here
now total we have 15 clauses along with $(\bar{z}_i \vee \bar{z}_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee \bar{z}_4) \wedge (\bar{z}_i \vee \bar{z}_3 \vee \bar{z}_4)$
↳ where $z_i$ ($i = 1$ & $2$)

For $x_1 = x_6 \vee x_7$ => $(x_1 \vee \bar{x}_7 \vee z_1)$, $(x_1 \vee \bar{x}_6 \vee z_1)$, $(\bar{x}_1 \vee x_6 \vee x_7)$
$(\bar{z}_i \vee \bar{z}_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee \bar{z}_4) \wedge (\bar{z}_i \vee \bar{z}_3 \vee \bar{z}_4)$
↳ where $z_i$ ($i = 1$ & $2$)

For $x_0 = x_1 \wedge x_2$ => $(\bar{x}_0 \vee x_2 \vee z_1)$, $(\bar{x}_0 \vee x_1 \vee z_1)$, $(x_0 \vee \bar{x}_1 \vee \bar{x}_2)$
$(\bar{z}_i \vee \bar{z}_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee \bar{z}_4) \wedge (\bar{z}_i \vee \bar{z}_3 \vee \bar{z}_4)$
↳ where $z_i$ ($i = 1$ & $2$)

for Hardcoded values => $x_6 = 0$ => add 1 clause : $\bar{x}_6$
$x_0 = 1$ => add 1 clause : $x_0$

now turn clauses into length of 3.
$(\bar{x}_6 \vee z_1 \vee z_2)$ and $(x_0 \vee z_1 \vee z_2)$

=> we have $(\bar{z}_i \vee \bar{z}_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee z_4) \wedge (\bar{z}_i \vee z_3 \vee \bar{z}_4) \wedge (\bar{z}_i \vee \bar{z}_3 \vee \bar{z}_4)$
where $z_i$ ($i = 1$ & $2$) in common we can just avoid writing every time.

4) a. $C = \begin{bmatrix} 30 \\ 40 \\ 50 \\ 10 \\ 10 \end{bmatrix}$

$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$

Representing Edges $e_{ij}$ if edge from vertex $v_i$ to $v_j$ in graph.

A =

| 0 | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $e_1$ | 1 | 0 | 0 | 0 | 1 |
| $e_2$ | 1 | 0 | 0 | 1 | 0 |
| $e_3$ | 1 | 1 | 0 | 0 | 0 |
| $e_4$ | 0 | 0 | 1 | 1 | 0 |

$b = [1, 1, 1, 1]$

4. b)

$A_{11} * x_1 + A_{15} * x_5 >= 1 \Rightarrow x_1 + x_5 >= 1$

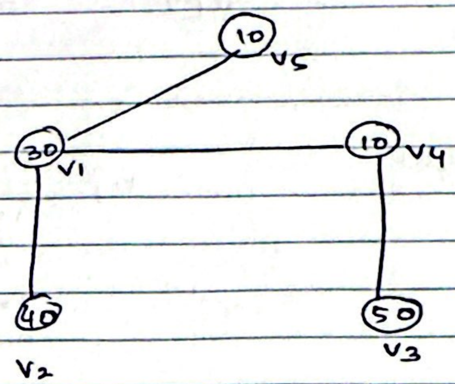
$A_{21} * x_1 + A_{24} * x_4 >= 1 \Rightarrow x_1 + x_4 >= 1$

$A_{31} * x_1 + A_{32} * x_2 >= 1 \Rightarrow x_1 + x_2 >= 1$

$A_{43} * x_3 + A_{44} * x_4 >= 1 \Rightarrow x_3 + x_4 >= 1$

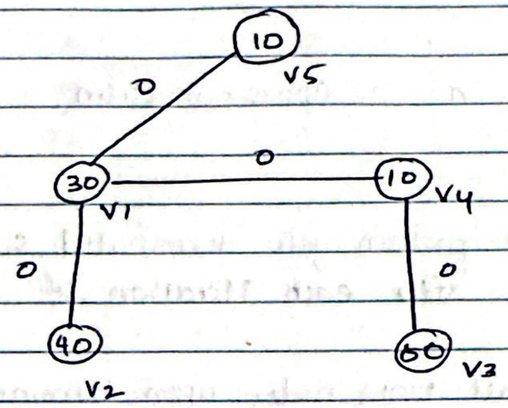
5)

a,b Given,



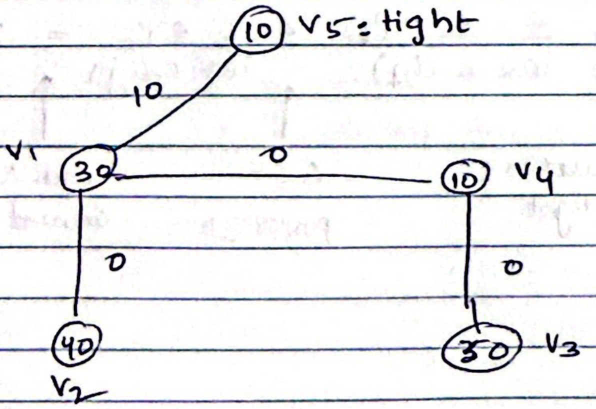

now let's Assume edge weigth's zero in every case.

step1:

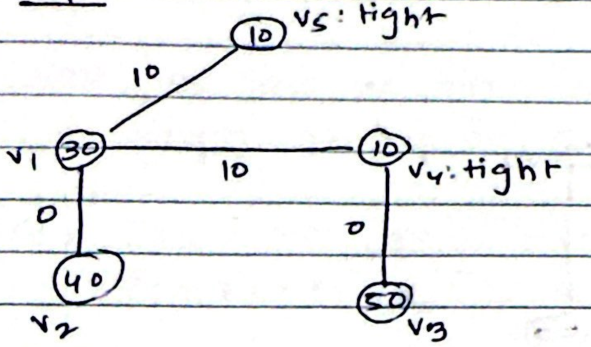



step2:

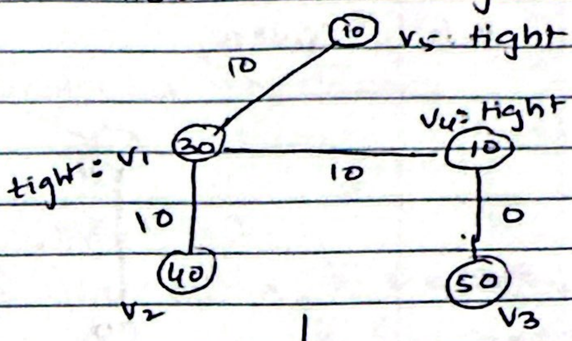
Now start from any one of the edge where neither $V_1$ or $V_5$ are tight and select such an edge e and increase $P_e$ as much as possible until $V_1$ (or) $V_5$ are tight



Step3: now make $v_4$ tight




Step4: now make $v_1$ tight




Final step
we can't make $v_3$ tight

5.c) pricing Method is a 2-Approximation

proof:

→ we know that Algorithm gets terminated since atleast one new node becomes tight after each iteration of while loop.

→ Let $S$ = Set of all tight nodes upon termination of algorithm. $S$ is a Vertex cover: if some edge $i$-$j$ is uncovered, then neither $i$ nor $j$ is tight. But then while loop wouldn't Terminate.

→ Let $S^*$ be optimal vertex cover. We show $w(S) \le 2w(S^*)$

$$w(S) = \underbrace{\sum_{i \in S} w_i = \sum_{i \in S} \sum_{e=(i,j)} P_e}_{\text{All nodes in } S \text{ are tight}} \le \underbrace{\sum_{i \in V} \sum_{e=(i,j)} P_e}_{S \subseteq V,\ \text{prices} \ge 0} = \underbrace{2 \sum_{e \in E} P_e}_{\text{each edge counted twice}} \le \underbrace{2w(S^*)}_{\text{fairness lemma.}}$$