Name
: Aagam Shah

USC ID :
5420023430

ALGORITHM - HW 5

Name :- Aagam Shah

USC ID : 5420023430

ALGORITHM HW-5

Ans1) consider the problem as a network flow problem where we need to validate if the arrangement between guards and timeslot exists or not.

Algorithm:-

① create a source vertex 's' and a sink vertex 't'

② create 'n' security guard vertex ranging from $\{g_1, g_2 \dots g_n\}$

③ create the 'm' time interval vertex ranging from $\{h_1, h_2 \dots h_m\}$

④ For each vertex g_i connect the source vertex 's' to g_i with lower bound L and upper bound M.

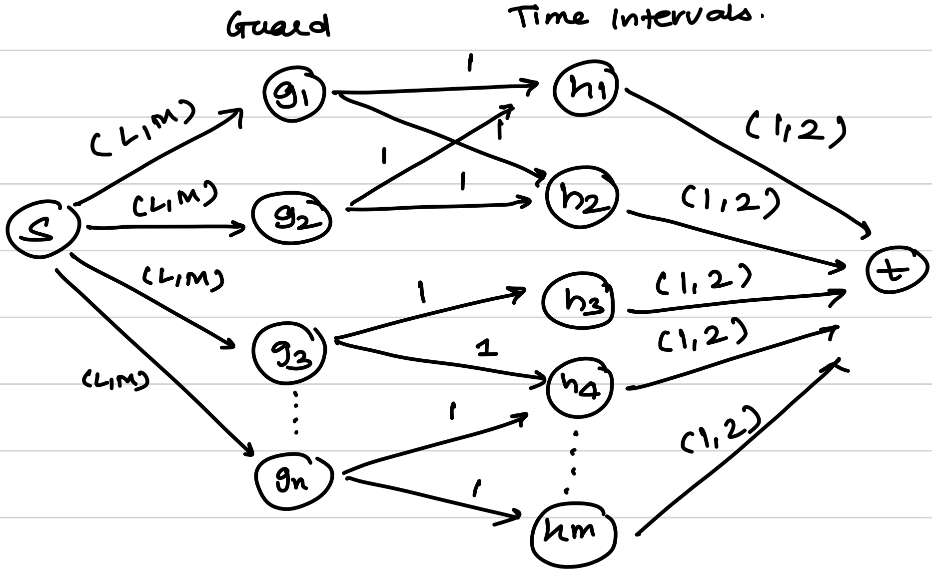
// since every guard must complete atleast L hours of duty and do at max M hours of duty.

⑤ Connect the vertex g_i with h_i based on the subset of time slot intervals that the guard wish to work into thus having the edge capacity = 1.

⑥ For all vertex of timeslot interval h_i connect them with sink node 't' with lowerbound = 1 and upper bound being 2. thus making the edge capacity $c(h_i, t) = (1, 2)$

// since at any particular timeslot there needs to be atleast one security guard and at most 2 security guard.

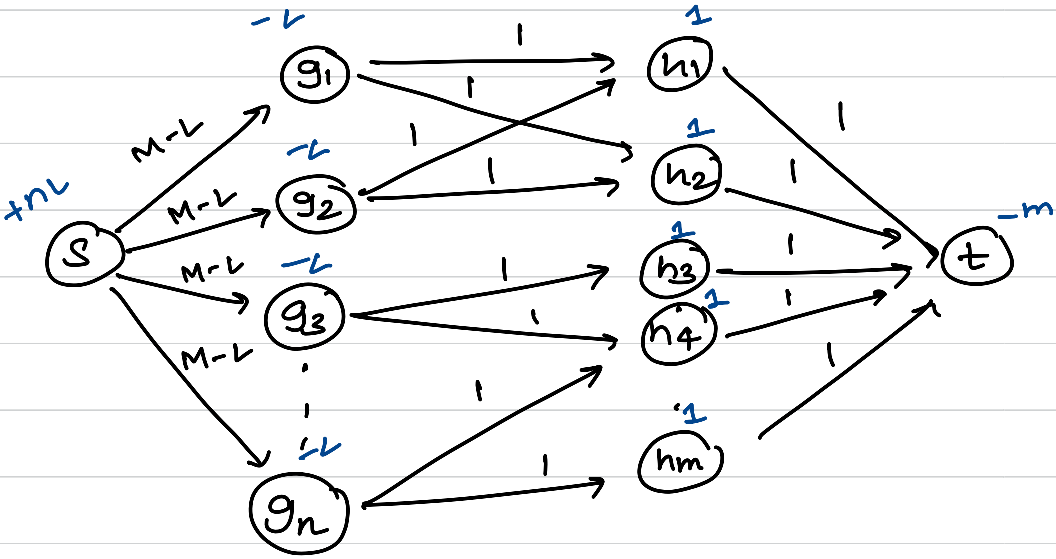
construction



7) Since here we have a circulation we will reduce it to network flow. where we reduce based on

Circulation with demands and lower bounds $\longrightarrow$ circulation with demands $\longrightarrow$ Network flow.

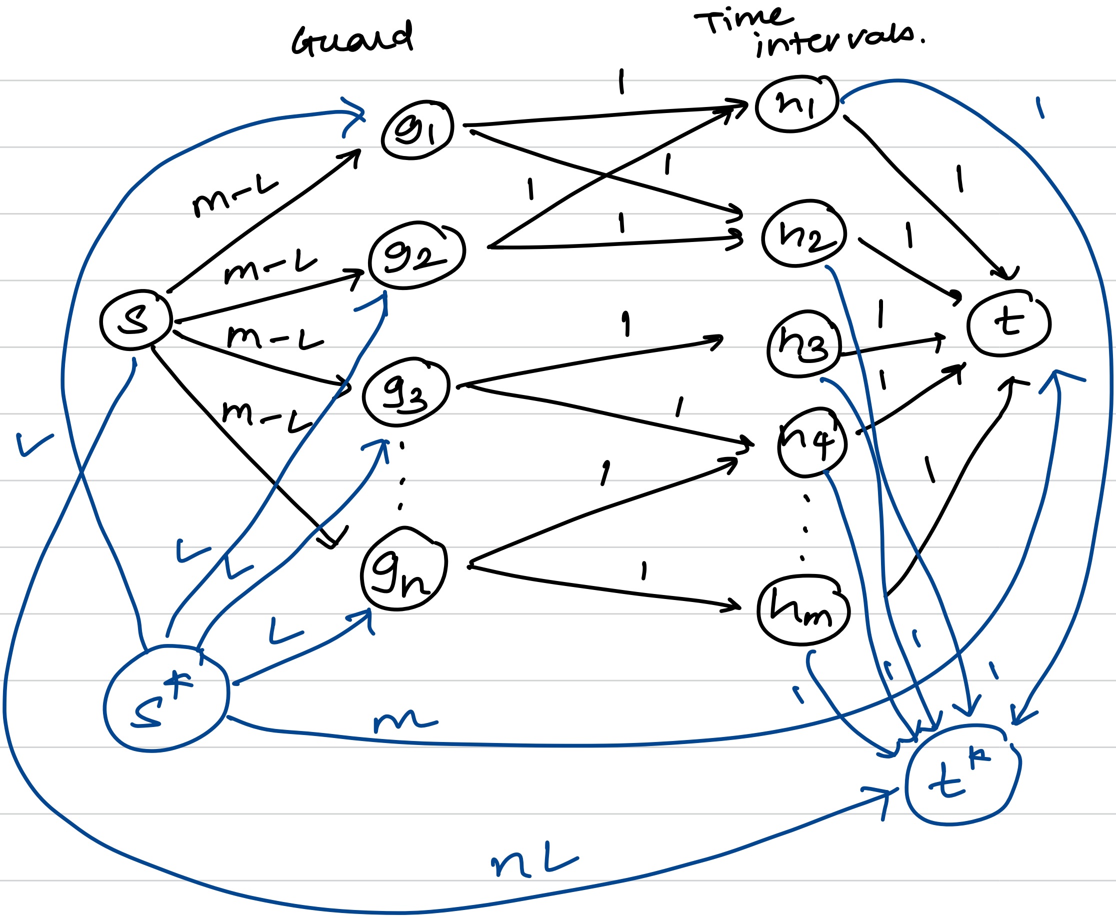
After removing the lowerbounds we get this graph.



(8) To this we will add super source s^* and super sink t^*

(9) we connect all the vertex having -ve vertex demand to source s^* with the edge capacity being the absolute value of demand

(10) we connect all the vertex having +ve demands to supersink t^* with the edge capacity being equal to the absolute value of demand.



(11) on this graph we will then find the max flow using any network flow algorithm like Ford Fulkerson and Edmond's Karp Algorithm.

(12) If there is a feasible solution to the circulation then $\sum_{d(v)>0} d(v) = D =$ max flow

then we can say that a deployment of guards exist.

Claim :- There exists a feasible solution in the original graph if and only if the value of maxflow in the reduced graph is equal to sum of lower bounds of the edges from s (nL) and sum of lower bounds of the edge to t (m) that means. each guard is deployed to atleast L time slots and each time slot must have 1 guard.

$\therefore$ maxflow = $nL+m$

Direction 1 => Given a feasible circulation exists in the original graph in which each gaurd is deployled to atleast L time slot and each time slot has atleast one guard. We need to prove that assignment satisfies the maxflow condition in Network flow.

Proof :- If there is a feasible circulation in the original graph, we can extend it to the flow in G'. by sending the flow along the new edges. The total flow entering the sink t^* will be equal the sum of the lower bounds of the edges from s. Thus the value of the flow in G' is equal to the sum of the lower bounds of the edges in s, and lower bound of the edges to t. Thus the value of flow in G' is equal to (nL+m)

Direction 2 => Given the value of maxflow in G' equals to the sum of the lower bounds of the edges from s(nL) and the lower bounds to t(m) in G, we need to prove that a feasible circulation exists in G in which assignment exists where each guard is deployed to atleast L timeslot and each slot has 1 guard.

Proof :- We can obtain a feasible circulation in G by removing the new source s^* and the new sink t^* from G' along side the added edges. The flow in G the will satisfy the capacity constraints and lower bound constraints.

Ans 2)

→ **Declaring variables:-**

Let x_{ij} be the no of units of i^{th} product that can be made in the j^{th} manufacturing unit.

→ **Objective function:-**

$$\text{Maximize } (10x_{11} + 8x_{12} + 6x_{13} + 9x_{14} + 18x_{21} + 20x_{22} + 15x_{23} + 17x_{24} + 15x_{31} + 16x_{32} + 13x_{33} + 17x_{34})$$

subject to :-

(conversion from hours to mins)

$$5x_{11} + 6x_{21} + 13x_{31} \le 35\times60$$
$$7x_{12} + 12x_{22} + 14x_{32} \le 35\times60$$
$$4x_{13} + 8x_{23} + 9x_{33} \le 35\times60$$
$$10x_{14} + 15x_{24} + 17x_{34} \le 35\times60$$

$\Rightarrow$

$$5x_{11} + 6x_{21} + 13x_{31} \le 2100$$
$$7x_{12} + 12x_{22} + 14x_{32} \le 2100$$
$$4x_{13} + 8x_{23} + 9x_{33} \le 2100$$
$$10x_{14} + 15x_{24} + 17x_{34} \le 2100$$

$$x_{11} + x_{12} + x_{13} + x_{14} \ge 100$$
$$x_{21} + x_{22} + x_{23} + x_{24} \ge 150$$

$x_{31} + x_{32} + x_{33} + x_{34} \geq 100$

$x_{11}, x_{12}, x_{13}, x_{14} \geq 0$

$x_{21}, x_{22}, x_{23}, x_{24} \geq 0$

$x_{31}, x_{32}, x_{33}, x_{34} \geq 0$

Converting the above formulation into standard form.

As we know the format of standard form of LP is given by

maximize $(c^T x)$

subject to

$Ax \leq B$

$x \geq 0$

We need to make the above stated equation in this format

Thus.

(1) $x_{11} + x_{12} + x_{13} + x_{14} \geq 100$ becomes.

$-x_{11} - x_{12} - x_{13} - x_{14} \leq -100$

(2) $X_{21} + X_{22} + X_{23} + X_{24} \geq 150$ becomes.

$$-X_{21} - X_{22} - X_{23} - X_{24} \leq -150$$

(3) $X_{31} + X_{32} + X_{33} + X_{34} \geq 100$ becomes

$$-X_{31} - X_{32} - X_{33} - X_{34} \leq -100$$

$\therefore$ Finally the equations become.

maximize $(10x_{11} + 8x_{12} + 6x_{13} + 9x_{14} + 18x_{21} + 20x_{22} + 15x_{23} + 17x_{24} + 15x_{31} + 16x_{32} + 13x_{33} + 17x_{34})$

subject to

$$5x_{11} + 6x_{21} + 13x_{31} \leq 2100$$

$$7x_{12} + 12x_{22} + 14x_{32} \leq 2100$$

$$4x_{13} + 8x_{23} + 9x_{33} \leq 2100$$

$$10x_{14} + 15x_{24} + 17x_{34} \leq 2100$$

$$-x_{11} - x_{12} - x_{13} - x_{14} \leq -100$$

$$-x_{21} - x_{22} - x_{23} - x_{24} \leq -150$$

$$-x_{31} - x_{32} - x_{33} - x_{34} \leq -100$$

$x_{11}, x_{12}, x_{13}, x_{14} \geq 0$

$x_{21}, x_{22}, x_{23}, x_{24} \geq 0$

$x_{31}, x_{32}, x_{33}, x_{34} \geq 0$

Converting to standard format.

$$c = \begin{bmatrix} 10 \\ 8 \\ 6 \\ 9 \\ 18 \\ 20 \\ 15 \\ 17 \\ 15 \\ 16 \\ 13 \\ 17 \end{bmatrix} \qquad X = \begin{bmatrix} X_{11} \\ X_{12} \\ X_{13} \\ X_{14} \\ X_{21} \\ X_{22} \\ X_{23} \\ X_{24} \\ X_{31} \\ X_{32} \\ X_{33} \\ X_{34} \end{bmatrix}$$

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccc} x_{11} & x_{12} & x_{13} & x_{14} & x_{21} & x_{22} & x_{23} & x_{24} & x_{31} & x_{32} & x_{33} & x_{34} \end{array} \\
A = & \begin{bmatrix}
5 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 13 & 0 & 0 & 0 \\
0 & 7 & 0 & 0 & 0 & 12 & 0 & 0 & 0 & 14 & 0 & 0 \\
0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 9 & 0 \\
0 & 0 & 0 & 10 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 17 \\
-1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1
\end{bmatrix}
\end{array}
$$

$$
B = \begin{bmatrix} 2100 \\ 2100 \\ 2100 \\ 2100 \\ -100 \\ -150 \\ -100 \end{bmatrix}
$$

Ans 3) We need to maximize $(-x_1 + 4x_2)$ subject to some constraints given below

$3x_1 + x_2 \leq 1$

$3x_1 + x_2 \geq -5$

$x_1 - x_2 \leq 4$

$x_1 - x_2 \geq -2$

$x_2 \leq 1$

① Plotting the graph in 2D

(i) $3x_1 + x_2 \leq 1 \longrightarrow 3x_1 + x_2 = 1$

when $x_1 = 0$

$\therefore x_2 = 1$ Pts $(0, 1)$

when $x_2 = 0$

$x_1 = 1/3 = 0.33$ Pts $(0.33, 0)$

(ii) $3x_1 + x_2 \geq -5 \longrightarrow 3x_1 + x_2 = -5$

when $x_1 = 0$

$x_2 = -5$ Pts $(0, -5)$

when $x_2 = 0$

$x_1 = -5/3 = -1.66$ Pts $(-1.66, 0)$

(iii) $x_1 - x_2 \leq 4 \longrightarrow x_1 - x_2 = 4$

when $x_1 = 0$

$x_2 = -4$ pt $(0, -4)$

when $x_2 = 0$

$x_1 = +4$ pt $(4, 0)$

(iv) $x_1 - x_2 \geq -2 \longrightarrow x_1 - x_2 = 2$

when $x_1 = 0$

$x_2 = +2$ pt $(0, 2)$

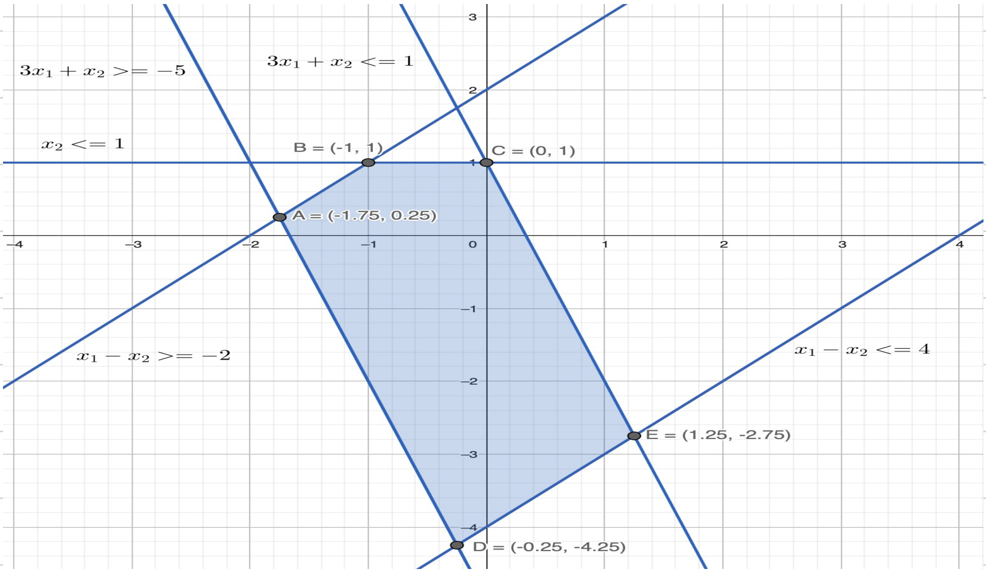
when $x_2 = 0$

$x_1 = -2$ pt $(-2, 0)$

summarizing the points needed to plot.

Sr.No	Equation	Point 1.	Point 2.
1	$3x_1 + x_2 \leq 1$	$(0, 1)$	$(0.33, 0)$
2	$3x_1 + x_2 \geq -5$	$(0, -5)$	$(-1.66, 0)$
3	$x_1 - x_2 \leq 4$	$(0, -4)$	$(4, 0)$
4	$x_1 - x_2 \geq -2$	$(0, 2)$	$(-2, 0)$

After plotting the graph we get this polygon as the feasible region.



Inorder to find the vertices of the feasible region we need to find the value at intersection of the 2 lines /region

① on solving eq 1: $3x_1 + x_2 \le 1$ and eq 2: $x_1 - x_2 \le 4$

$$\begin{array}{r} 3x_1 + x_2 = 1 \\ + \quad x_1 - x_2 = 4 \\ \hline 4x_1 \quad = 5 \end{array}$$

$$x_1 = 5/4 = 1.25$$

Putting x_1 in eq^n 2

$$x_1 - x_2 = 4$$

$$1.25 - x_2 = 4$$

$$x_2 = -2.75.$$

$\therefore$ The co-ordinate of point $E = (1.25, -2.75)$

2 on solving the eq^n 1 :- $3x_1 + x_2 \leq 1$ and eq^n 2 $x_2 = 1$

$$3x_1 + x_2 = 1$$

$$3x_1 + 1 = 1$$

$$3x_1 = 0 \qquad \therefore x_1 = 0$$

$\therefore$ The co-ordinates of Point $C = (0, 1)$

3 on solving the equations eq^n 1: $3x_1 + x_2 \geq -5$ and eq^n 2: $x - y \leq 4$

$$\begin{array}{r} 3x_1 + \cancel{x_2} = -5 \\ + \quad x_1 - \cancel{x_2} = 4 \\ \hline 4x_1 = -1 \end{array}$$

$$x_1 = -1/4 = -0.25$$

Putting x_1 in eq^n 2 :- $(-0.25) - x_2 = 4$

$$x_2 = -4.25$$

$\therefore$ The co-ordinates of Point $D(-0.25, -4.25)$

④ on solving the equation eqⁿ ① : $3x_1 + x_2 \geq 5$ and eqⁿ ② :- $x_1 - x_2 \geq -2$

$$
\begin{array}{r}
3x_1 + \cancel{x_2} = -5 \\
+ \quad x_1 - \cancel{x_2} = -2 \\
\hline
4x_1 = -7
\end{array}
$$

$$x_1 = -7/4 = -1.75$$

Putting x_1 in eqⁿ ②

$$(-1.75) - x_2 = -2$$

$$x_2 = 0.25$$

∴ The co-ordinates of pt $A(-1.75, 0.25)$

⑤ on solving the equation eqⁿ ① $x_1 - x_2 = -2$ and $x_2 = 1$

$$x_1 - x_2 = -2$$

$$x_1 - (1) = -2$$

$$x_1 = -2 + 1$$

$$x_1 = -1$$

∴ The co-ordinates of pt $B = (-1, 1)$

As we know the maximum value of obj function is always on the boundary of the feasible region.

Name of Point	Points of Intersection	Value of objective at vertex of intersection $(-x_1 + 4x_2)$
A	$(-1.75, 0.25)$	$(1.75 + 4(0.25)) = 2.75$
B	$(-1, 1)$	$(1 + 4(1)) = 5$
C	$(0, 1)$	$(0 + 4(1)) = 4$
E	$(1.25, -2.75)$	$(-1.25 + 4(-2.75)) = -12.25$
D	$(-0.25, -4.25)$	$(0.25 + 4(-4.25) = -16.75$

As, we can see we have achived the maximum value as 5 when we use value of $x_1 = -1$ and $x_2 = 1$

$$\boxed{\begin{array}{l} \text{Max value} = 5 \\ \text{when } (x_1, x_2) = (-1, 1) \end{array}}$$

Ans 4) We need to minimize the cost of buying the food j per day such that his requirement of getting bi nutrients gets satisfied.

Variables:- Let t_j denote the no of units of food bought by Andy per day for his consumption.

Objective function:-

$$\text{minimize } \left(\sum_{j=1 \text{ to } n} t_j * c_j \right)$$

subject to :-

(1) for each nutrient i from 1 to m

$$\sum_{j=1 \text{ to } n} t_j * a_{ij} \geq b_i$$

eg:- $t_1 a_{11} + t_2 a_{12} + \dots t_n a_{1n} \geq b_1$

$t_1 a_{21} + t_2 a_{22} + \dots t_n a_{2n} \geq b_2$

$\vdots$

$t_1 a_{n1} + t_2 a_{n2} + \dots t_n a_{mn} \geq b_m$

can also be stated as :- $t_1 a_{i1} + t_2 a_{i2} + \dots + t_n a_{in} \geq b_i$ for each nutrient $i = 1, 2 \dots m$

(2) $t_j \geq 0$ for all $j = 1$ to n

converting the given L.P into standard form

objective function

$$\text{Maximize} \left(-\sum_{j=1 \text{ to } n} t_j * c_j \right)$$

subject to :-

(1) $-\sum_{j=1,2..n} t_j * a_{ij} \leq -b_i$, for each nutrient $i = 1, 2 .. m$

$$-t_1 a_{i1} - t_2 a_{i2} - \ldots t_n a_{in} \leq -b_i$$

for all nutrients
$i = 1, 2 \ldots m$

(2) $t_j \geq 0$ for each food
$j = 1, 2 \ldots n$

converting into matrix form.

$\max (C^T X)$

subject to

$AX \leq B$

$X \geq 0$

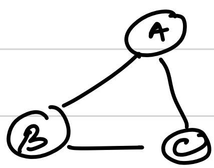
$$X = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} \qquad C = \begin{bmatrix} -c_1 \\ -c_2 \\ \vdots \\ -c_n \end{bmatrix}$$

$$A = \begin{bmatrix} -a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & \dots & \dots & -a_{2n} \\ \vdots & & & \\ -a_{m1} & -a_{m2} & \dots & -a_{mn} \end{bmatrix} \qquad B = \begin{bmatrix} -b_1 \\ -b_2 \\ \vdots \\ -b_m \end{bmatrix}$$

Ans 5) Lets consider a graph G having V vertex and E edges the vertex cover is defined as the a subset of vertex $S \subseteq V$ such that every edge in has atleast one endpoint in vertex cover.

consider example :-

①



vertex cover = {A, B}

∴ The min vertex cover is of size = 2

∴ Formulating an LP based on this

Variable :- Let X_V denote if the vertex 'v' has been taken in vertex cover or not. $X_V \in [0,1]$

Objective function :-

$$\text{Minimize } \left(\sum_{v \in V} x_v \right)$$

// since we need to minimize the no of vertex in vertex cover

subject to

$$x_u + x_v \geq 1$$

, for all edge $(u,v) \in E$, out of the 2 verticies - u, v, atleast one of them should be in vertex cover if they have and edge (u, v) between them.

$x_v \in (0,1)$, for all $v \in V$; since we can either select the vertex for V.C or reject it.

Converting to standard format.

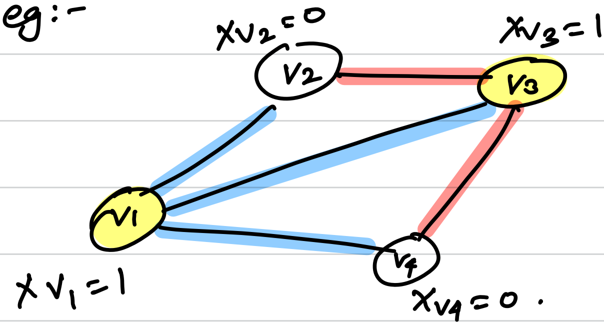
$\therefore$ maximize $(-\sum_{v\in V} x_v)$	$\therefore$ maximize $(-x_1 - x_2 \dots - x_n)$
subject to	subject to
$-x_u - x_v \leq -1$	$-x_u - x_v \leq -1$
$x \in (0, 1)$	$x \in (0, 1)$

Explaination:-

Here we will minimize the no of vertex considered in making vertex cover since x_v can take 0, 1 denoting that the vertex v has been taken in the vertex cover or not. now coming over for constraints we know in vertex cover every edge must be connected via a vertex. Thus we have put a constraint that for every edge (u, v) the value of $(x_u + x_v \geq 1)$ indicating that every edge will be connected by an vertex and that vertex will be taken in vertex cover

Also since x_i can take value's only $\{0,1\}$ we make this problem as integer linear programming.

eg:-



vertex cover = $\{v_1, v_3\}$

size of v.c = 2.

Here all the constraints are satisfied and we get the min value of vertex cover.

Ans 6) $\max(x_1 - 3x_2 + 4x_3 - x_4)$

subject to

$x_1 - x_2 - 3x_3 \leq -1 \quad \Rightarrow \quad x_1 - x_2 - 3x_3 + 0x_4 \leq -1$

$x_2 + 3x_3 \leq 5 \quad \Rightarrow \quad 0x_1 + x_2 + 3x_3 + 0x_4 \leq 5$

$x_3 \leq 1 \quad \Rightarrow \quad 0x_1 + 0x_2 + x_3 + 0x_4 \leq 1$

$x_1, x_2, x_3, x_4 \qquad\qquad x_1, x_2, x_3, x_4 \geq 0$

we will convert the given primal to standard format in matrix form first

format of Primal LP being

$$\max(c^T x)$$

subject to

$$Ax \leq B$$

$$x \geq 0$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \qquad c = \begin{pmatrix} 1 \\ -3 \\ 4 \\ -1 \end{pmatrix} \qquad A = \begin{bmatrix} 1 & -1 & -3 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{pmatrix} -1 \\ 5 \\ 1 \end{pmatrix} \qquad B^T = \begin{pmatrix} -1 & 5 & 1 \end{pmatrix} \qquad A^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 3 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Let y_1, y_2, y_3 be the variables for Dual form.

Primal Form		Dual Form
$\max(C^T X)$	converting from primal to Dual $\Longleftrightarrow$	$\min(B^T Y)$
subject to		subject to
$AX \leq C$		$A^T Y \geq C$
$X \geq 0$		$Y \geq 0$

$\therefore$ Dual Form

$$\min(B^T Y)$$

$$= \begin{pmatrix} -1 & 5 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = -y_1 + 5y_2 + y_3$$

subject to

$$A^T Y \geq C$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 3 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \geq \begin{pmatrix} 1 \\ -3 \\ 4 \\ -1 \end{pmatrix}$$

$$y_1 \geq 1$$

$$-y_1 + y_2 \geq -3$$

$$-3y_1 + 3y_2 + y_3 \geq 4$$

$$0y_1 + 0y_2 + 0y_3 \geq -1$$

$$y_1, y_2, y_3 \geq 0$$

$\therefore$ Final Dual of the Primal LP is

$$\boxed{\begin{aligned} &\text{minimize } (-y_1 + 5y_2 + y_3) \\ &\quad \text{subject to} \\ &\qquad y_1 \geq 1 \\ &\qquad -y_1 + y_2 \geq -3 \\ &\qquad -3y_1 + 3y_2 + y_3 \geq 4 \\ &\qquad 0y_1 + 0y_2 + 0y_3 \geq -1 \\ &\qquad y_1, y_2, y_3 \geq 0 \end{aligned}}$$

Ans 7)

(a) $\max(x_1 + x_2)$

$x_1 + 2x_2 \leq 3 \Rightarrow$ constraint C1

$3x_1 - x_2 \leq 2 \Rightarrow$ constraint C2

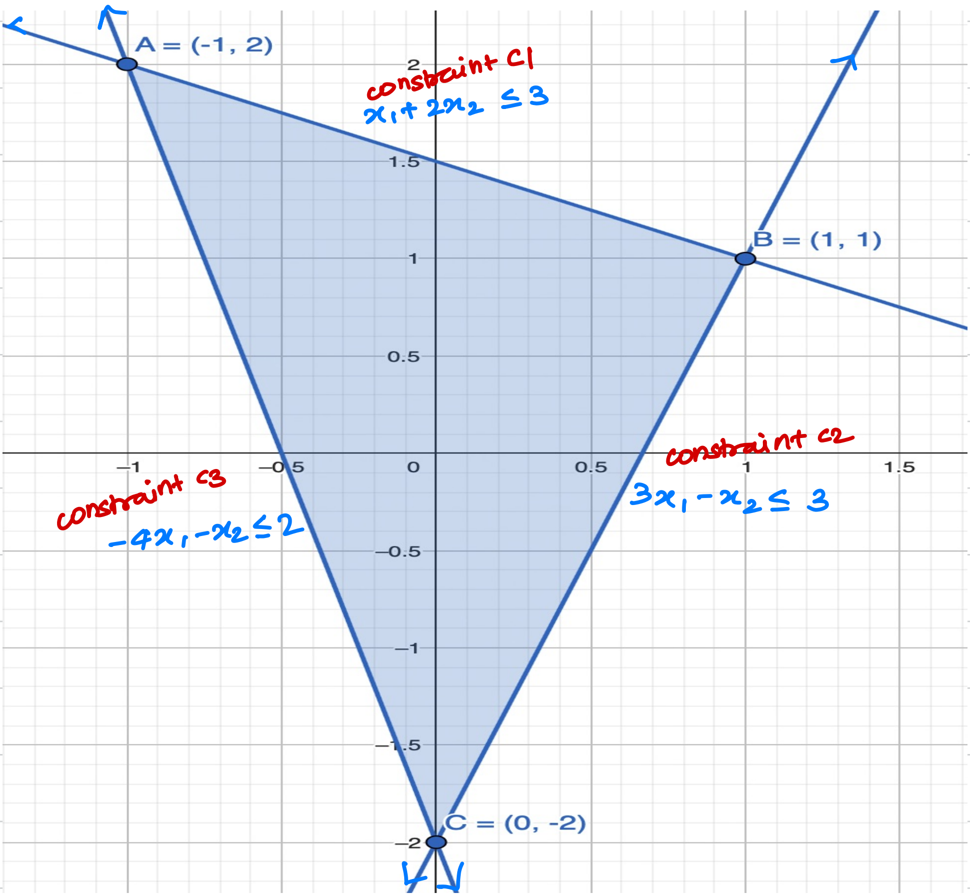
$-4x_1 - x_2 \leq 2 \Rightarrow$ constraint C3

In order to find if the solution of the LP is feasible bounded, feasible unbounded or unfeasible we plot the graph in 2D.

Equation	Point ($x_1 = 0$)	Point ($x_2 = 0$)
$x_1 + 2x_2 = 3$	$(0, 3/2)$	$(3, 0)$
$3x_1 - x_2 = 2$	$(0, -2)$	$(2/3, 0)$
$-4x_1 - x_2 = 2$	$(0, -2)$	$(-2/4, 0)$

using the following points we will plot the graph and check the feasibility of linear programming

On plotting the points we get the following graph



As we can see from the graph the lp is feasibly bounded.

Lets calculate the max value using the

intersection points.

Point (x_1, x_2)	Satisfies constraints C1	C2	C3	obj func $(x_1 + x_2)$ max value
$(1, 1)$	✓	✓	✓	2
$(-1, 2)$	✓	✓	✓	1
$(0, -2)$	✓	✓	✓	-2

Here the Point $(1, 1) = (x_1, x_2)$ satisfies all the constraints.

(C1) $x_1 + 2x_2 \leq 3$

$1 + 2(1) \leq 3$

$3 \leq 3$ satisfies.

(C2) $3x_1 - x_2 \leq 2$

$3(1) - 1 \leq 2$

$2 \leq 2$ satisfies.

(C3) $-4x_1 - x_2 \leq 2$

$-4(1) - 1 \leq 2$

$-5 \leq 2$ satisfies.

and also gives Max value of obj $(x_1 + x_2) = (1+1) = 2$.

$\therefore$ The Linear program is feasible bounded.

(b) $\max (x_1 + x_2)$

subject to

$$x_1 + 2x_2 \geq 3$$

$$3x_1 - x_2 \geq 2$$

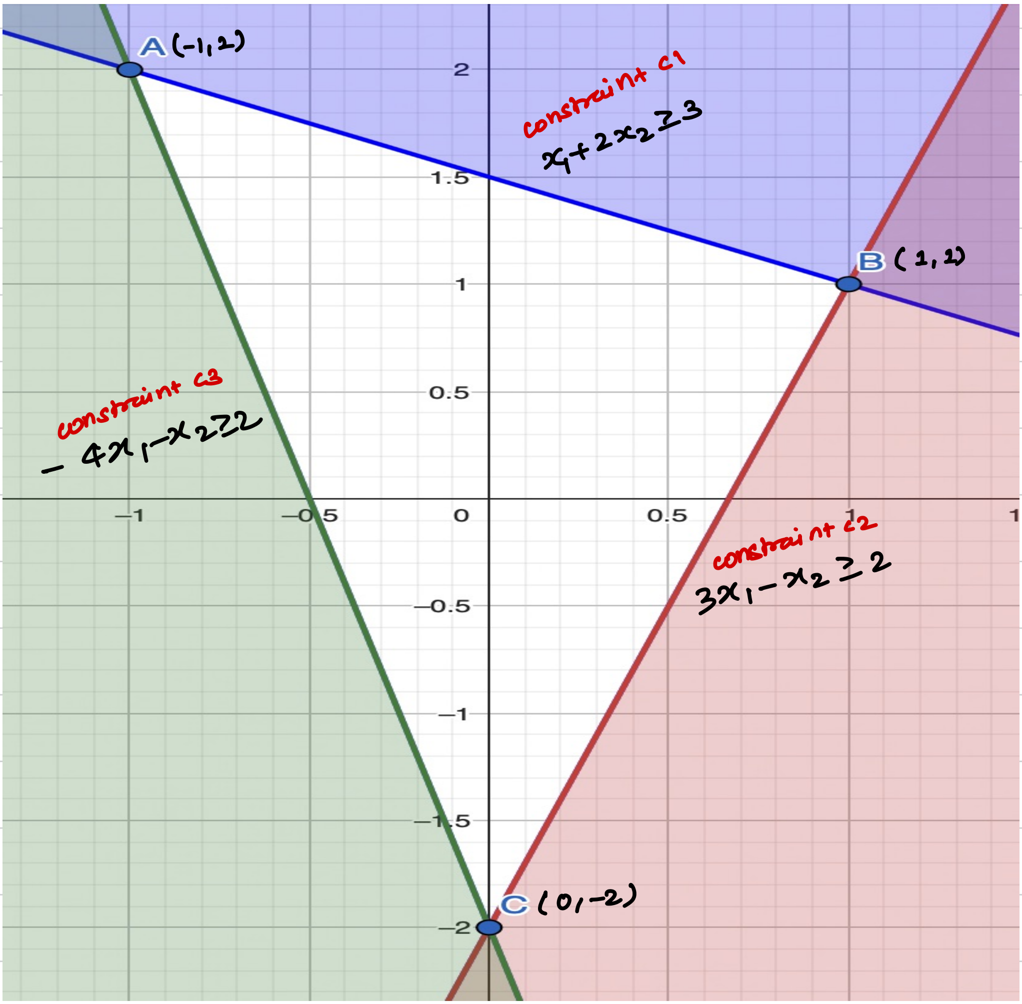
$$-4x_1 - x_2 \geq 2$$

In order to find if the solution of the LP is feasible bounded, feasible unbounded or unfeasible we plot the graph in 2D.

Equation	Point ($x_1 = 0$)	Point ($x_2 = 0$)
$x_1 + 2x_2 = 3$	$(0, 3/2)$	$(3, 0)$
$3x_1 - x_2 = 2$	$(0, -2)$	$(2/3, 0)$
$-4x_1 - x_2 = 2$	$(0, -2)$	$(-2/4, 0)$

using the following points we will plot the graph and check the feasibility of linear programming

Plotting the graph we get this



from this graph we can see we don't have any feasible solution. Lets reveify this using some points from the graph.

Points (x_1, x_2)	C1	C2	C3	Max value (x_1+x_2)
	constraint satisfied?			
$(1,1)$	✓	✓	✗	–
$(0,-2)$	✗	✓	✓	–
$(-1,2)$	✓	✗	✓	–

considering point $(1,1)$

C1 $x_1 + 2x_2 \geq 3$

$1 + 2(1) \geq 3$

$3 \geq 3$ satisfied.

C2 $3x_1 - x_2 \geq 2$

$3(1) - 1 \geq 2$

$2 \geq 2$ satisfied

C3 $-4x_1 - x_2 \geq 2$

$-4(1) - (1) \geq 2$

$-4 - 1 \geq 2$

$-5 \geq 2$ not satisfied.

$\therefore$ This linear program is infeasible

(c) $\max(x_1 + x_2)$

subject to

$$x_1 + 2x_2 \geq 3$$

$$3x_1 - x_2 \leq 2$$

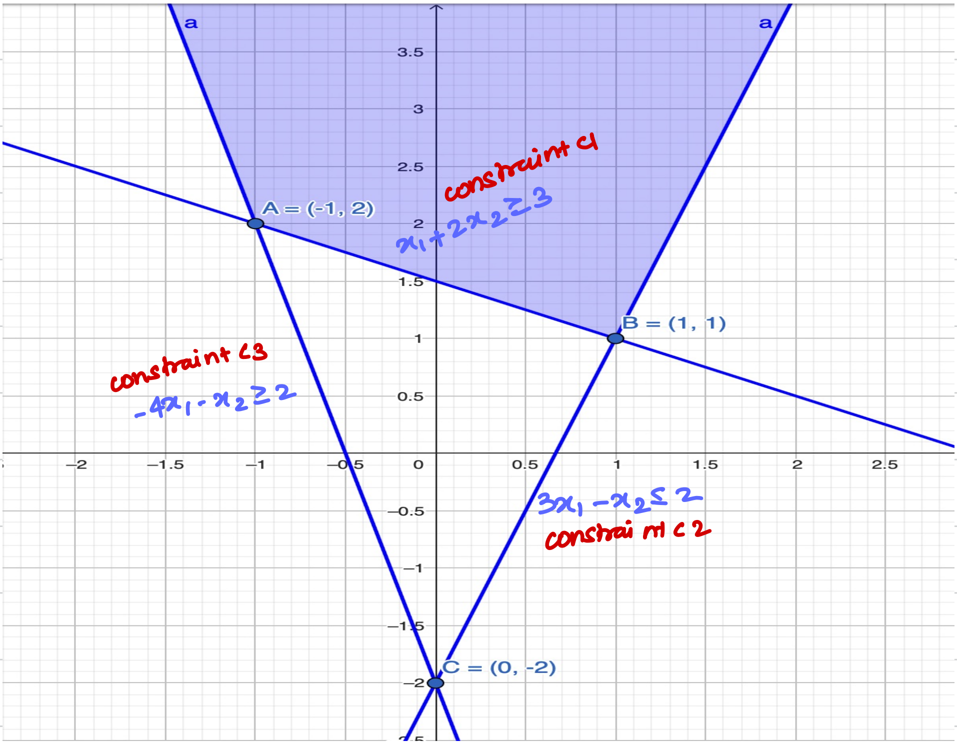
$$-4x_1 - x_2 \leq 2$$

In order to find if the solution of the LP is feasible bounded, feasible unbounded or unfeasible we plot the graph in 2D.

Equation	Point ($x_1 = 0$)	Point ($x_2 = 0$)
$x_1 + 2x_2 = 3$	$(0, 3/2)$	$(3, 0)$
$3x_1 - x_2 = 2$	$(0, -2)$	$(2/3, 0)$
$-4x_1 - x_2 = 2$	$(0, -2)$	$(-2/4, 0)$

Using the following points we will plot the graph and check the feasibility of linear programming

On plotting the equations we get such a graph



- As we can see from the graph the feasible region is unbounded in nature.

Let's consider some example to prove this

consider values of $(x_1, x_2) = (0, 60)$ and $(x_1, x_2) = (15, 65)$

constraint's satisfied				
values of (x_1, x_2)	C1	C2	C3	max value $(x_1 + x_2)$
$(0, 60)$	✓	✓	✓	$(0+60) = 60$
$(15, 65)$	✓	✓	✓	$60+15 = 75$

consider the points $(15, 65) = (x_1, x_2)$

(C1) $x_1 + 2x_2 \geq 3$

$15 + 2(65) \geq 3$ satisfied

$15 + 130 \geq 3$

$145 \geq 3$

max $(x_1 + x_2)$

$= 15 + 60 = 75$

(C2) $3x_1 - x_2 \leq 2$

$3(15) - 65 \leq 2$

∴ The LP is feasible

$45 - 65 \leq 2$

but it can go unbounded.

$20 \leq 2$ satisfied

(C3) $-4x_1 - x_2 \leq 2$

$-4(15) - 65 \leq 2$

$-125 \leq 2$ satisfied.

Thus we can see as we go above in the feasible region we will infinite value of (x_1, x_2) such that $\max(x_1 + x_2) \longrightarrow \infty$ ie tends to infinity.

lets consider some more points.

sr no	Points		satisfies constraint			obj function
	x_1	x_2	c1	c2	c3	x_1+x_2
1	-10	100	✓	✓	✓	90
2	10	100	✓	✓	✓	110
3	-30	150	✓	✓	✓	120
4	40	150	✓	✓	✓	190

As we can see here there are multiple solution to this optimization and the more above we go in the funnel shaped graph the better and better value we keep reciving. showing that this constraints set give us an unbounded region and we can have infinitely many solution.

∴ This Linear Program is considered to be Feasible unbounded

Ans 8) To show :- Vertex cover even is NP complete

In order to show that vertex cover even is NP Hard, we use reduction from vertex cover Vertex cover $\leq_p$ Vertex cover even.

Step 1 :- Show that vertex cover even is NP Hard

A certifier for VCE is an algorithm that takes input graph $G = (V, E)$, an integer K and a certificate which is proposed vertex cover for G. The certifier checks if C is a valid vertex cover of size K

If both condition are held true, then certifier accepts it, or else rejects it.

Algorithm of certifier :-

① Check if $|C| = K$

② For each edge (u, v) in E, check if either u, v is in C

③ For each vertex v in V, check if the degree of v is even.

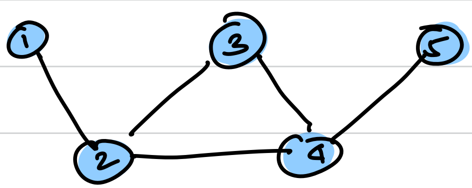
(3) If both condition hold true accept certificate else reject it.

This algorithm can check it in polynomial time making vertex cover even NP problem.

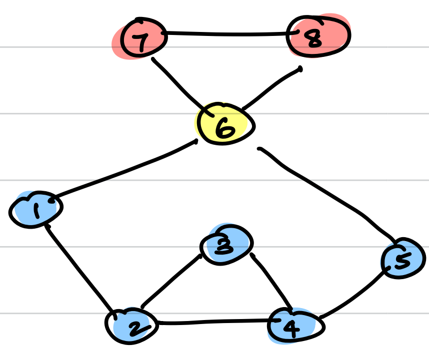
Construction:- We need to convert any graph G into a graph with all even degree vertices G'. (Note:- A simple fact that any undirected graph has an even number of odd degree verticies

There fore we construct the graph G' by adding 3 extra verticies (V6, V7, V8) forming a triangle and connecting one of them to vertices of odd degree.

Graph G



Graph G'



Claim:- G has vertex cover of size K if and only if G' has a vertex cover of size 'K+1'

Direction 1:- G has a vertex cover of size K then G' has a vertex cover of size 'K+2'

Proof:- By construction. Assume G has a vertex cover of size K. Then the vertex cover of G' created by adding two extra verticies (red and yellow). Thus the vertex cover size increases by 'K+2'

Direction 2: G' has a vertex cover of size 'K+2' then G has a vertex cover of size K.

Proof:- Assume G' has a vertex cover of size 'K+2'. In order to get the vertex cover of G, we have to remove two verticies. Those two vertices are easily identified; they must from the set of extra vertices. In our construction figure we will remove the yellow and red verticies to get the right vertex cover for G.

Ans 9) Let $\phi(x_1, x_2, x_3 \dots x_n) = c_1 \wedge c_2 \wedge c_3 \dots \wedge c_m$ be the boolean formulae for a valid 3-SAT instance where $c_1, c_2 \dots c_m$ represent the clauses and $x_1, x_2, x_3 \dots x_n$ represent the variables. Assume the algorithm ALG takes the CNF formula and will output 1. if it is satisfiable and does output 0 if it is not satisfiable.

If $ALG(\phi(x_1, x_2 \dots x_n)) = 0$, then return the instance is not satisfiable and if $ALG(\phi(x_1, x_2 \dots x_n)) = 1$ then the given ϕ is satisfiable, we can find an arrangement x_1 as follows. if $ALG(\phi(1, x_2, x_3 \dots x_n)) = 1$ then we can set $x_1 = 1$ and gaurentee that $\phi(x_1, x_2 \dots x_n)$ $\Rightarrow \phi(1, x_2, x_3 \dots x_n)$ is satisfiable. Else set $x_1 = 0$ and be gaurenteed that $\phi_1'(x_2, x_3 \dots x_n) = \phi(0, x_2 \dots x_n)$ is satisfiable. we can continue iterately with ϕ_1 and ϕ_1' to find subsiquent assignments of $x_2, x_3 \dots x_n$.

This will lead to n iterations with one call to ALG per iteration. In each iteration, finding the new formulae by plugging the value of a variable takes $O(m)$ time. Thus find the solution in polynomial time.

Example of how the solution will be formed

consider $\phi_1(x_2, x_3 \ldots x_n)$ ie $\phi(1, x_2 \ldots x_n)$ is satisfied making '$x_1 = 1$'. Then in 2^{nd} iteration we find an assignment for x_2 by computing $ALG(\phi_1(1, x_3, \ldots x_n))$, and setting $x_2 = 1$, if it returns 1, or $x_2 = 0$ otherwise and so on)