Name: Rohit Kudache USN: 1BM18CS083

Course: Advance Algorithm Code:

Date: 06/01/2021 Time: 1:00 - 2:15 pm

Part-C

4b)

```
#include <isotream>
using namespace std;
struct point {
        int x ;
        int y; } ;
bool onSegment (Point p, Point q, Point r) {
    if (q.x <= max (p.x, r.x) &&
            q.x >= min (p.x, r.x) &&
        q.y <= max (p.y, r.y) &&
            q.y >= min (p.y, r.y))
        return true;
    return False; }

int orientation (Point p, Point q, Point r) {
    int val = (q.y - p.y) * (r.x - q.x) -
              (q.x - p.x) * (r.y - q.y);
    if (val ==0) return 0;
    return (val >0) ? 1 : 2 ; }

bool doInterest (Point P1, point q1, Point P2, point q2)
{
    int o1 = orientation (P1, q1, P2);
    int o2 = orientation (P1, q1, q2);
    int o3 = orientation (P2, q2, q1);
    int o4 = orientation (P2, P1, q2);

    if (o1 != o2 && o3 != o4)
        return true ;
```

PTO

Rohit

Name: Rohit Kudache

USN: 1BM18CS083

4b continued ...

```
if (O1==0 && onSegment(P1, P2, q1)) return true;
if (O2==0 && onSegment(P1, q2, q1)) return true;
if (O3==0 && onSegment(P2, P1, q2)) return true;
if (O4==0 && onsegment(P2, P1, q2)) return true;
}

int main() {
    struct point P1 = {10,10}, q1 = {10,30};
    struct point P2 = {30,30}, q2 = (40,10);

doInterest (P1, q1, P2, q2) ? cout << "yes"; cout << "NO";
    P1 = {10,0}, q1 = {0,10};
    P2 = {0,0}, q2 = {10,10};
doinferest (P1, q1, P2, q2) ? cout << "yes"; cout << "NO";

    return 0;
}
```

// algorithm

```
SEGMENT-INTERSECT (P1, P2, P3, P4)
  d1 = DIRECTION (P3, P4, P1)
  d2 = DIRECTION (P3, P4, P2)
  d3 = DIRECTION (P1, P2, P3)
  d4 = DIRECTION (P1, P2, P4)
  if ((d1 > 0 and d2 < 0) or (d1 < 0 and d2 > 0) and
          ((d3 > 0 and d4 < 0) or (d3 < 0 and d4 > 0))
      return TRUE;
  elseif d1 == 0 and ON-SEGMENT (P3, P4, P1)
      return TRUE;
```


Scanned by TapScanner

Name : Rohit Kudache

USN : 1BM18CS083

4b continued

```
elseif d3 == 0 and ON-SEGMENT (P1, P2, P3)
        return TRUE
elseif d4 == 0 and ON-SEGMENT( P1, P2, P4)
        return TRUE
else
        return FALSE

DIRECTION (Pi, Pj, Pk)
    return (Pk - Pi) x (Pj - Pi)

ON-SEGMENT (Pi, Pj, Pk)
    if min (xi, xj) <= xk <= max (xi, xj) and min (yi, yj) <=
                                          yk <= max (yi, yj)
        return TRUE
    else
        return FALSE
```

Name : Rohit Kudache

USN : 1BM18CS083

Part B

2 a) minimize $x_1 + 2x_2$

subjected to $x_1 + x_2 > 40$

$x_1 - x_2 = 14$

$x_1 - 2x_3 < 3$

* maximize $-(x_1 + 2x_2) \Rightarrow -x_1 - 2x_2$
* x_1 & x_2 has no negative constraints

replace x_1 with $(x_1' - x_1'')$ and

x_2 with $(x_2' - x_2'')$

add constraints $x_1', x_1'', x_2', x_2'' \geq 0$

$\therefore$ maximize $= -(x_1' - x_1'') - 2(x_2' - x_2'')$

$\Rightarrow -x_1' + x_1'' - 2x_2' + 2x_2''$

Subjected to

$x_1' - x_1'' + x_2' - x_2'' > 40$

$x_1' - x_1'' + x_2'' - x_2' = 14$

$x_1' - x_1'' - 2x_2' + 2x_2'' < 3$

$x_1', x_1'', x_2', x_2'' \geq 0$

* Replace equality constraints with pair of inequality constraints

maximise $= x_1'' - x_1' + 2x_2'' - 2x_2'$

subjected to $x_1' - x_1'' + x_2' - x_2'' > 40$

$x_1' - x_1'' + x_2'' - x_2' \geq 14$

$x_2' - x_1'' + x_2'' - x_2' \leq 14$

$x_1' - x_1'' - 2x_2' + 2x_2'' < 3$

$x_1', x_2', x_1'', x_2'' \geq 0$

* convert $>=$ by $<=$ by multiplying -1

$\therefore$ maximise $= x_1'' - x_1' + 2x_2'' - 2x_2'$

Subjected to $x_1'' - x_1' + x_2'' - x_2' \leq -40$

$x_1'' - x_1' - x_2'' + x_2' \leq -14$

$x_2' - x_1'' + x_2'' - x_2' \leq 14$

$x_1' - x_1'' - 2x_2' + 2x_2'' < 3$

$x_1', x_1'', x_2', x_2'' \geq 0$

Name : Rohit Kudache

USN : 1BM18CS083

Part - B

2b) Let P_1 be (10, 25), P_2 be (30, 30)

$$P_1 \times P_2 = \det \begin{pmatrix} 10 & 30 \\ 25 & 30 \end{pmatrix}$$

$$= -450$$

As $P_1 \times P_2 < 0$
hence point P_1 (10, 25) is to the left of point P_2 (30, 30),

2c) Assume $c(u,v) = 0$ if $(u,v) \notin E$ and there are no antiparallel edges, we can express the max flow problem as a LP.

$$\text{maximize} \sum_{v \in V} f_{xv} - \sum_{v \in V} f_{vs}$$

subjected to $f_{uv} \leq c(u,v) \quad \forall u,v \in V$

$$\sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv} \quad \forall u \in V - \{s-t\}$$

$$f_{uv} \geq 0 \quad \forall u,v \in V.$$

This LP has $|V|^2$ variables, corresponding to the flow between each pairs of vertices and it has $2|V|^2 + |V| - 2$ constraints, so the more efficient to recurite the LP so that, it has $O(V+E)$ constraints

$$\max Z = F_1 + F_2 + F_3 + \cdots + F_n$$

subjected to $\sum_{(1 \ldots k_i)} F_i \leq U_e \quad \forall e = E(G)$ when $e \in P_i$,

the capacity constraintt, U_e - capacity fe

The constraints indicates that the sum of flow on on edge e must be bounded by U_e (capacity fe).

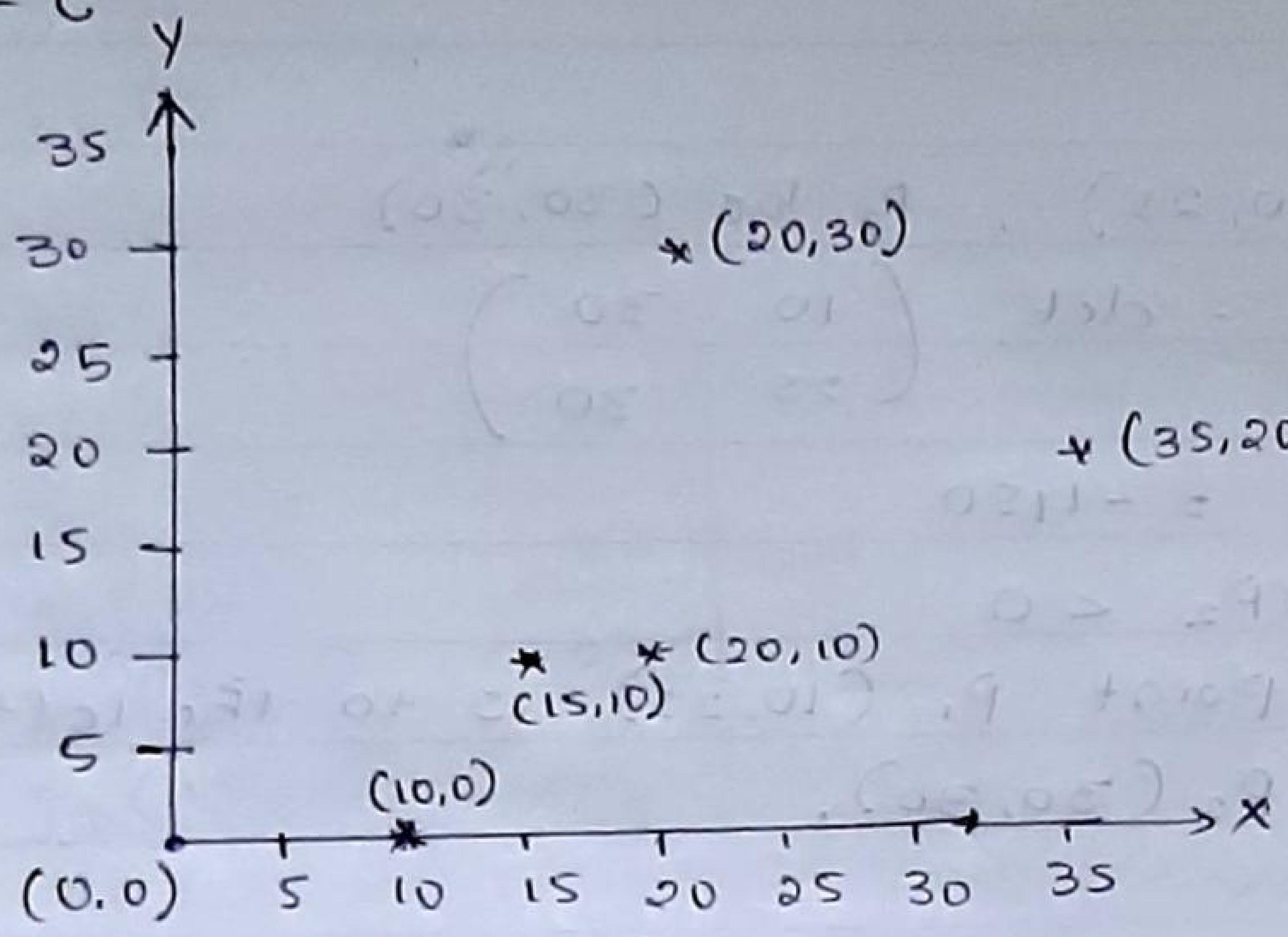
Rohit

Name : Rohit Kudache

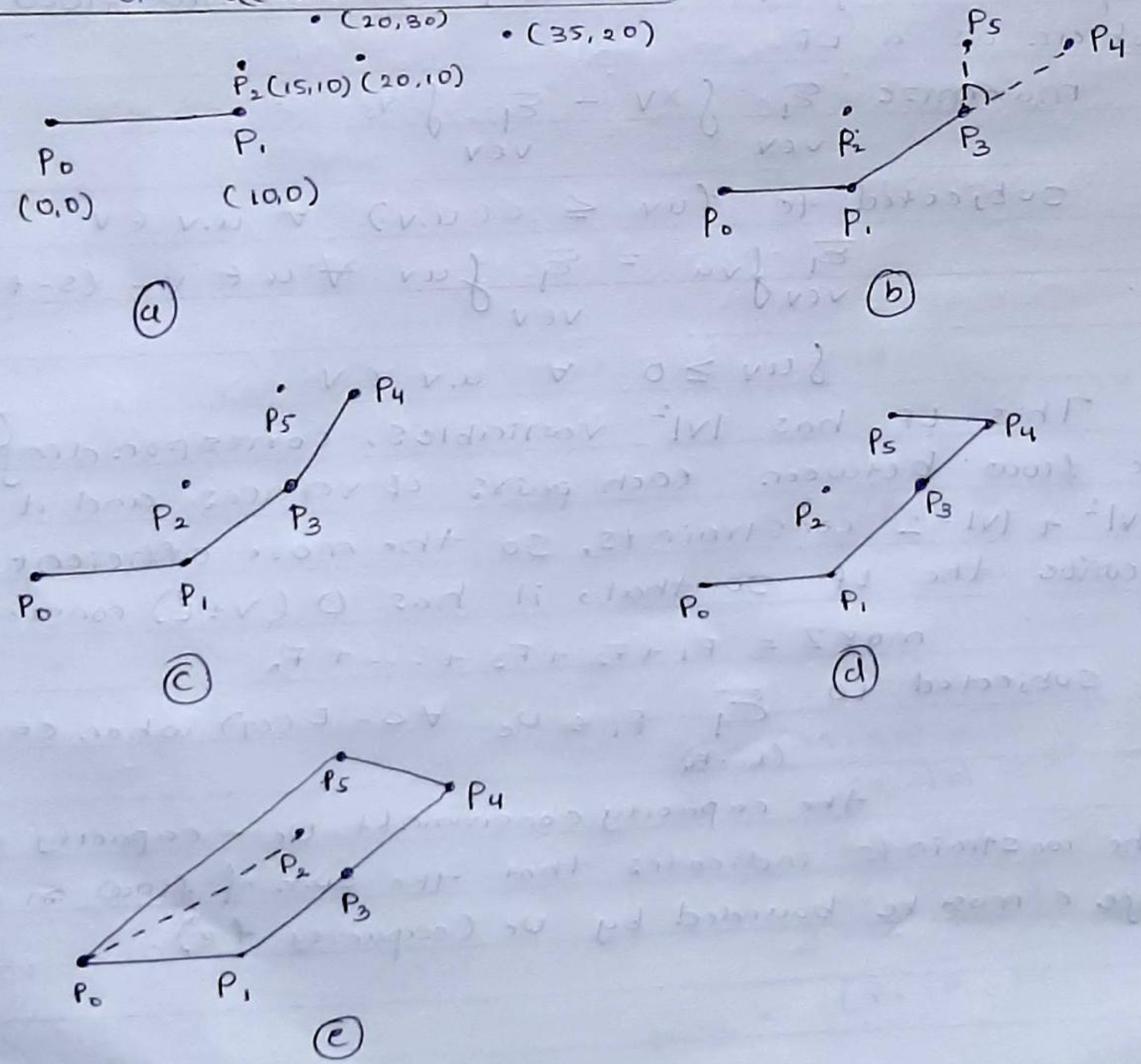
USN : 1BM18CS083

Part - C

4a)



Q = {((0,0), (10,0), (20,10), (15,10), (20, 30), (35,20))}



Rohit

Part C

4a) $P_0(0,0)$

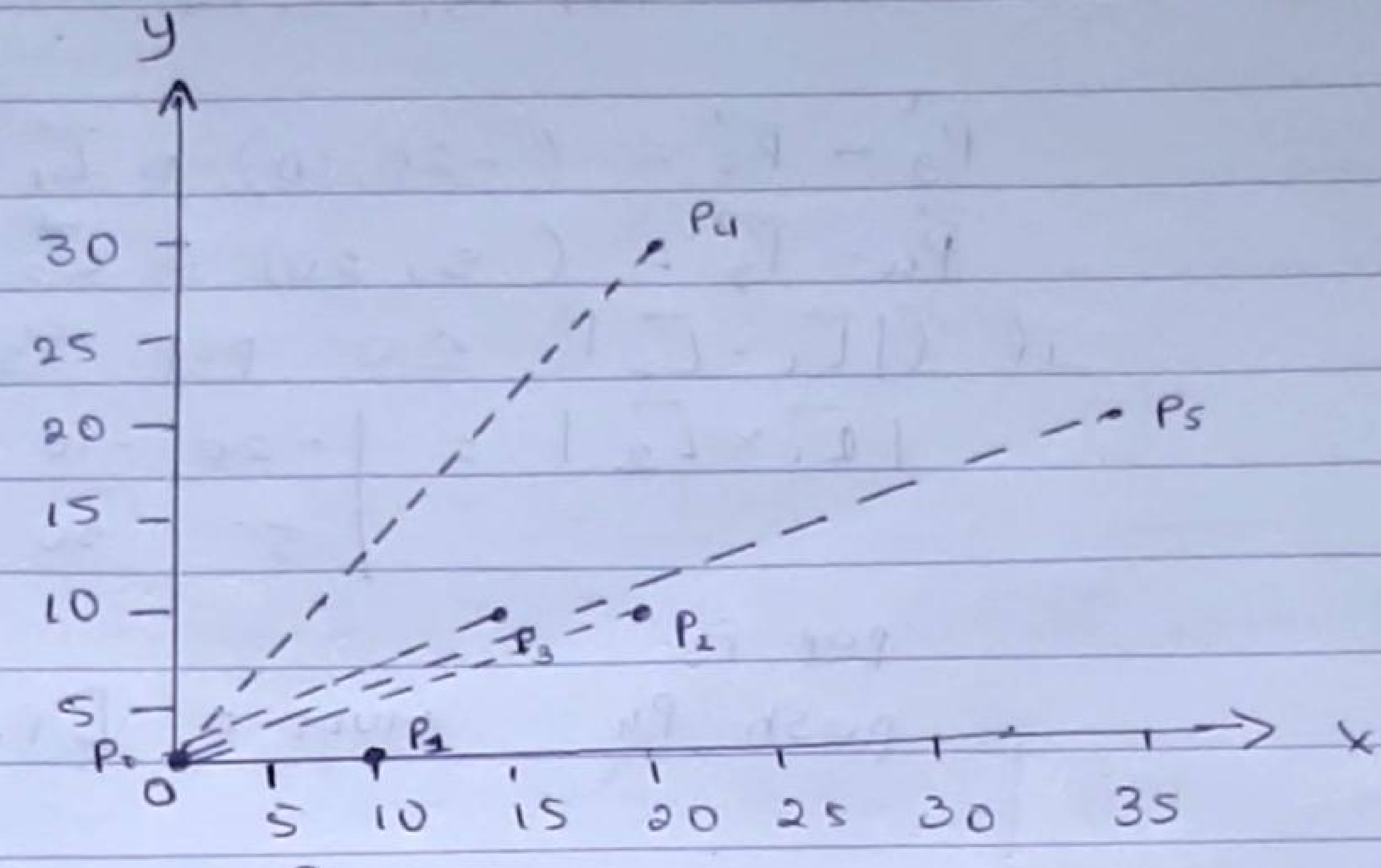
$P_1(10,0)$

$P_2(20,10)$

$P_3(15,10)$

$P_4(20,30)$

$P_5(35,20)$



Sorted according to polar angles

points $[P_0, P_1, P_2, P_5, P_3, P_4]$

step 1: push P_0, P_1, P_2 to stack (NULL)

Step 2: $\vec{P_2} - \vec{P_4} = (10,10) \equiv \vec{L_1}$

$\vec{P_5} - \vec{P_2} = (15,10) \equiv \vec{L_2}$

if $|\vec{L_1} \times \vec{L_2}| \leq 0$ pop P_2

$|\vec{L_1} \times \vec{L_2}| = \begin{vmatrix} 10 & 10 \\ 15 & 10 \end{vmatrix} = 100 - 150 = -50$

$\vec{L_2}$ is is on right of $\vec{L_1}$

so pop P_2

push P_5

Step 3: Check $P_1 P_5$, $P_5 P_3$

$\vec{P_5} - \vec{P_1} = (25-20) \equiv \vec{L_1}$

$\vec{P_3} - \vec{P_5} = (-20,-10) \equiv \vec{L_2}$

if $(\vec{L_1} \times \vec{L_2}| \leq 0$, pop P_5

$|\vec{L_1} \times \vec{L_2}| = \begin{vmatrix} 25 & 20 \\ -20 & -10 \end{vmatrix} = -250 + 400$

$\Rightarrow 150$

HULL = $[P_0, P_1, P_5, P_3]$

4a continued

Step 4 : check P_5, P_3, P_3, P_4

$\vec{P_3} - \vec{P_5} = (-20, -10) = \vec{L_1}$

$\vec{P_4} - \vec{P_3} = (5, 20) = \vec{L_2}$

if $(|\vec{L_1} \cdot \vec{L_2}|) \leq 0$ pop P_3

$|\vec{L_1} \times \vec{L_2}| = \begin{vmatrix} -20 & -10 \\ 5 & 20 \end{vmatrix} = -400 + 50$

$= -350$

pop P_3

push P_4 HULL = $[P_0, P_1, P_5, P_4]$

Step 5 : check P_1, P_5, P_5, P_4

$\vec{P_5} - \vec{P_1} = (25, 20) = \vec{L_1}$

$\vec{P_4} - \vec{P_5} = (-15, 10) = \vec{L_2}$

if $|\vec{L_1} \cdot \vec{L_2}| \leq 0$ pop P_4

$|\vec{L_1} \cdot \vec{L_2}| = \begin{vmatrix} 25 & 20 \\ -15 & 10 \end{vmatrix} = 250 + 300$

$= 550$

HULL = $[P_0, P_1, P_5, P_4]$

Step 6 : check P_5, P_4, P_4, P_0

$\vec{P_0} - \vec{P_4} = (-20, -30) = \vec{L_2}$

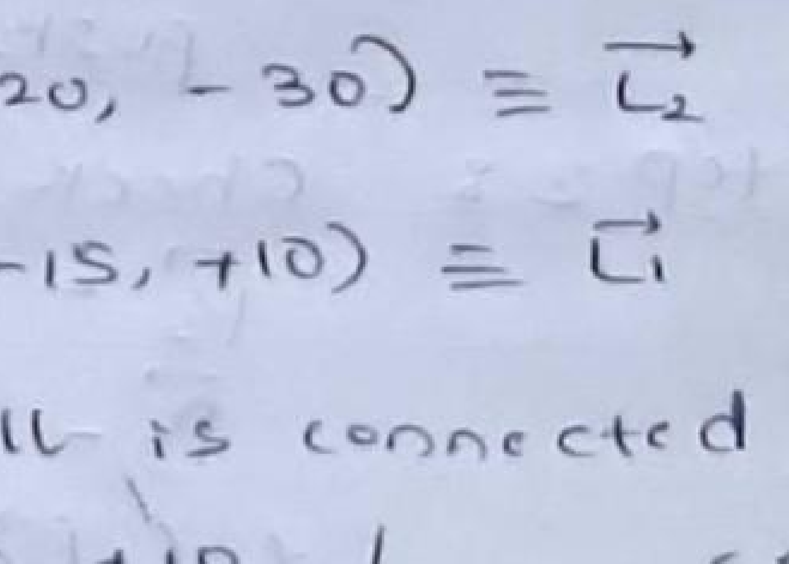
$\vec{P_4} - \vec{P_5} = (-15, +10) = \vec{L_1}$

if $|\vec{L_1} \cdot \vec{L_2}| \geq 0$ HULL is connected & Final

$|L_1 \times L_2| = \begin{vmatrix} -15 & +10 \\ -20 & -30 \end{vmatrix} = 650$

$\vec{L_1} \times \vec{L_3} > 0 \Rightarrow$ HULL is obtained

HULL = $[P_0, P_1, P_5, P_4]$



Preet

Part C

3a) maximise $z = 7x_1 + 5x_2$

subject to = $2x_1 + x_2 <= 100$

$4x_1 + 3x_2 <= 240$

Standard form $2x_1 + x_2 + S_1 = 100$ ——— (1)

$4x_1 + 3x_2 + S_2 = 240$ ——— (ii)

$\max z = 7x_1 + 5x_2 + 0S_1 + 0S_2$

$S_1, S_2 \geq 0$

optimizer condition for max $\forall\ C_j - z_j \leq 0$

* initialize simplex table

CBi	Cj Basic variable	7 x_1	5 x_2	0 S_1	0 S_2	Solution	Ratio
0	S_1	2	1	1	0	100	100/2 = 50 ←
0	S_2	4	3	0	1	240	240/4 = 60
($Z_j = \Sigma C_B a_{ij}$)	Z_j	0	0	0	0	0	
	$C_j - Z_j$	7 ↑	5	0	0		

$R_1 \Rightarrow R_1/2$ $R_2 \rightarrow R_2 - R_1$

~~iteration 1~~

~~CBi~~ ~~Cj~~ ~~7~~ ~~5~~ ~~0 0~~ ~~Solution~~ Ratio

7 ~~x_1~~ ~~1~~ ~~0.5~~ ~~0.5~~ ~~0~~

~~0~~ ~~S_2~~ ~~2.5~~ ~~-0.5~~ ~~1~~

PTO

3a continued

iteration 1

CB_i	C_j	7	5	0	0	Solution	Ratio
7	x_1	1	0.5	0.5	0	50	100
0	S_2	3	2.5	-0.5	1	190	76 ← key row
	Z_j	7	3.5	3.5	0		
	$C_j - Z_j$	0	2.5	0.5	1		

↑ key column

$R_1 \to R_1/2$

iteration 1

CB_i	C_j	7	5	0	0	Sol^n	Ration
7	x_1	1	0.5	0.5	0	50	100
0	S_2	0	1	-2	1	40	40 ← key row
	Z_j	7	3.5	3.5	0		
	$C_j - Z_j$	0	1.5	-3.5	0		

↑ key row

iteration 2

CB_i	C_j	7	5	0	0	Sol^n	Ratioo
7	x_1	1	0	1.5	-0.5	30	–
5	x_2	1	0	-2	1	40	–
	Z_j	7	5	0.5	1.5	410	
	$C_j - Z_j$	0	0	-0.5	-1.5		

3 a continued

a_{11} $c_j - z_j \leq 0$

Z is optimal at

$x_1 = 7$, $x_2 = 5$, $z = 410$ //

Part A

1. Linear programming

solving the problem finding real values for the values, satisfying the constrains & that gives the maximum possible value for object function.

(i) feasible solution:

Ex: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ Statisfis all the constraints & is called feasible solution.

⇒ it's objective function value, obtained by evaluating the object at $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ is $0 + 1 - 2(1) = -1$

(ii) optimal Solution:

max possible objects function value in the case of maximization problem is called optimal sol^n.

Ex: max $x + y$

st $2x + 2y \leq 1$

⇒ x and y cant exceeds $1/2$

Now, both $[x/y] = \begin{bmatrix} 1/2 \\ 0 \end{bmatrix}$ & $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}$

are feasible solution

Peat

1 continued.

3) Infeasible : Not all LP problem have optimal solution

Ex : min x

st $x \leq 1$

$x \geq 2$

$\Rightarrow$ There is no value for x i.e., at the same time at most 1 and atleast 2

Scanned by TapScanner