Design and Analysis of Algorithms
Digital Assignment - 1

Winter sem 2023 - 24.

Q1) Suppose you are about to have a meal. Value and calories of each food are given. Total calories consumed must be atmost 800

food	wine	beer	pizza	burger	fries	coke	apple	donut
value	90	90	30	50	90	79	90	10
calories	123	154	258	354	365	150	95	195

→

This problem is essentially just the 0/1 knapsack problem.
We have to maximize the total value of the meal, ~~whik~~ while ensuring that the calories are less than 800.
This is like choosing items such that the total profit is maximum but total weight does not exceed the capacity of the knapsack.

~~value~~ : The following mappings can be done.
value can be mapped to profit (P)
calories can be mapped to weight (w)
The max calories that can be consumed can be mapped to capacity of knapsack (m)

The table can be rewritten as:

Item ~~Food~~	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
P_i	90	90	30	50	90	79	90	10
W_i	123	154	258	354	365	150	95	195

Each food item is represented by x_i as:

wine $- x_1$
beer $- x_2$
pizza $- x_3$
burger $- x_4$
fries $- x_5$
coke $- x_6$
apple $- x_7$
donut $- x_8$

Now this can be solved by any method which is used to solve 0/1 knapsack problem i.e. Branch and Bound / Dynamic Programming / Backtracking.

Q2) Karatsuba Multiplication.

$\underline{9786 \times 7129}$.

→ 9786 can be written as $97 \times 10^2 + 86$
7129 can be written as $71 \times 10^2 + 29$
Total 4 digits in each number, ∴ $n = 4$
$n/2 = 2$

$\therefore 9786 \times 7129 = (10^{n/2}(\underset{a}{97}) + \underset{b}{86})(10^{n/2}(\underset{c}{71}) + \underset{d}{29})$

$(10^{n/2} a + b)(10^{n/2} c + d)$

$= 10^n ac + 10^{n/2}(bc + ad) + bd$

Now ~~for comp~~

$(a+b)(c+d) = ac + bc + ad + bd.$

$\therefore bc + ad = (a+b)(c+d) - ac - bd.$

Values of ac, bd, $(a+b)(c+d)$ need to be computed.

Hence, 4 multiplications got reduced to 3 multiplications

Also after each step, the no. of digits in the nos becomes half.

The algorithm calls itself recursively thrice on $n/2$ digit numbers.

$T(n) = 3T(n/2)$

Putting $n \to n/2$

$T(n/2) = 3T(n/4)$

$\therefore T(n) = 3^2 T(n/4)$

$T(n) = 3^3 T(n/8)$

$\vdots$

$T(n) = 3^i T(n/2^i)$ is general form of recurrence relation.

(4)

Applying master's Theorem.

$T(n) = a\,T(n/b) + f(n)$.

Here, $a = 3$, $b = 2$, $f(n) = 0 \Rightarrow k = 0$.

$a = 3$, $b^k = 2^0 = 1$.

$a > b^k$

$\therefore T(n) = O(n^{\log_b a})$

$= O(n^{\log_2 3})$

$T(n) \approx O(n^{1.58})$

Applying on given nos:

9786×7129.

$n = 4$, $n/2 = 2$.

$= (97 \times 10^{n/2} + 86)(71 \times 10^{n/2} + 29)$

$= 10^n (97)(71) + 10^{n/2}(97 \times 29 + 71 \times 86) + 86 \times 29$.

Now $(97 + 86)(71 + 29) = 97 \times 71 + 97 \times 29 + 71 \times 86 + 86 \times 29$

$\therefore 97 \times 29 + 71 \times 86 = (97+86)(71+29) - 97 \times 71 - 86 \times 29$.

$\therefore$ Req Ans $= 10^n \underbrace{(97 \times 71)}_{A} + 10^{n/2}\left(\underbrace{(97+86)(71+29)}_{B} - \underbrace{97 \times 71}_{A} - \underbrace{86 \times 29}_{C}\right) + \underbrace{86 \times 29}_{C}$ — (1)

A: 97×71:

$= (9 \times 10^1 + 7)(7 \times 10^1 + 1)$

$= 10^2 (9)(7) + 10(9 \times 1 + 7 \times 7) + 7 \times 1$

$= 10^2 \underbrace{(9 \times 7)}_{x} + 10\left(\underbrace{(9+7)(7+1)}_{y} - \underbrace{9 \times 7}_{x} - \underbrace{7 \times 1}_{z}\right) + \underbrace{7 \times 1}_{z}$

$x = 63$, $y = 16 \times 8 = 128$, $z = 7$.

$\therefore A = 10^2(63) + 10(128 - 63 - 7) + 7$

$= 6300 + 580 + 7$

$= 6887$

B: $(97+86) \times (71+29)$

$= 183 \times 100$

$n = 3, \quad \lfloor n/2 \rfloor = 1.$

$\therefore = (18 \times 10 + 3)(1 \times 10 + 0)$

$= 10^2(18 \times 1) + 10(18 \times 0 + 3 \times 1) + 3 \times 0$

$= 10^2 \underbrace{(18 \times 1)}_{x} + 10\left(\underbrace{(18+3)(10+0)}_{y} - \underbrace{18 \times 1}_{x} - \underbrace{3 \times 0}_{z}\right) + \underbrace{3 \times 0}_{z}$

$x = 180, \quad y = 21 \times 10, \quad z = 0$

$y: \quad 21 \times 10 \Rightarrow n = 2, \lfloor n/2 \rfloor = 1.$

$(2 \times 10 + 1)(1 \times 10 + 0)$

$= 10^2(2 \times 1) + 10(2 \times 0 + 1 \times 1) + 1 \times 0$

$= 10^2 \underbrace{(2 \times 1)}_{\alpha} + 10\left(\underbrace{(2+1)(1+0)}_{\beta} - \underbrace{2 \times 1}_{\alpha} - \underbrace{1 \times 0}_{\gamma}\right) + \underbrace{1 \times 0}_{\gamma}$

$\alpha = 2, \quad \beta = 3 \times 1 = 3, \quad \gamma = 0.$

$\therefore y = 10^2(2) + 10(3 - 2 - 0) + 0$

$= 200 + 10$

$= 210.$

put in B

$B = 10^2 x + 10(y - x - z) + z$

$= 100(180) + 10(210 - 180 - 0) + 0$

$= 18000 + 300$

$= 18300$

C: 86×29

$\Rightarrow n = 2, \quad n/2 = 1.$

$(8 \times 10 + 6)(2 \times 10 + 9)$

$= 10^2(8 \times 2) + 10(8 \times 9 + 6 \times 2) + 6 \times 9$

$= 10^2 \underbrace{(8 \times 2)}_{p} + 10\left(\underbrace{(8+6)(2+9)}_{q} - \underbrace{8 \times 2}_{p} - \underbrace{6 \times 9}_{r}\right) + \underbrace{6 \times 9}_{r}$

$p = 16, \quad q = 154, \quad r = 54$

$C = 10^2 \times 16 + 10(154 - 16 - 54) + 54$

$= 1600 + 840 + 54$

$= 2494$

6

Put values of A, B, C in (1).

$$Ans = 10^n A + 10^{n/2}(B - A - C) + C$$
$$= 10^4 \times 6887 + 10^2(18300 - 6887 - 2494) + 2494$$
$$= 68870000 + 891900 + 2494$$
$$= 69764394$$

$$\boxed{9786 \times 7129 = 69764394}$$

Q3) Apply backtracking for sum of subset problem.

$M = 10$

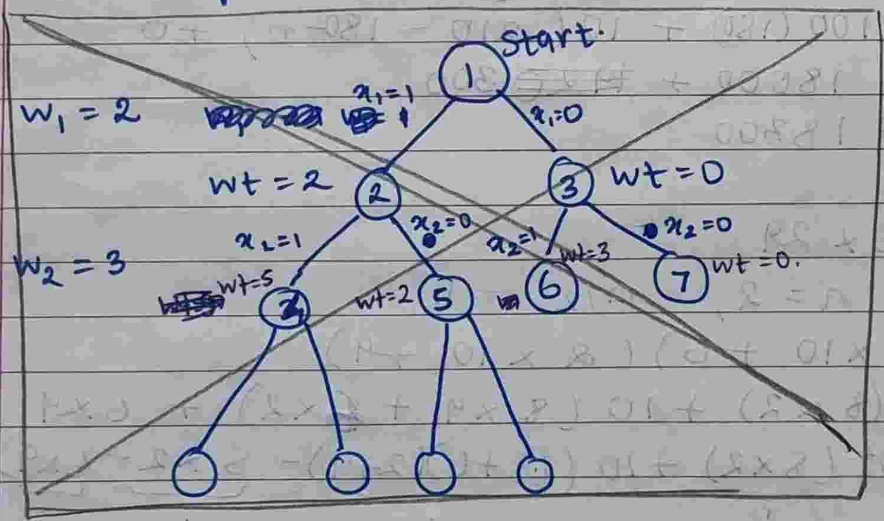
$w = \{2, 3, 5, 6, 8, 10\}$.

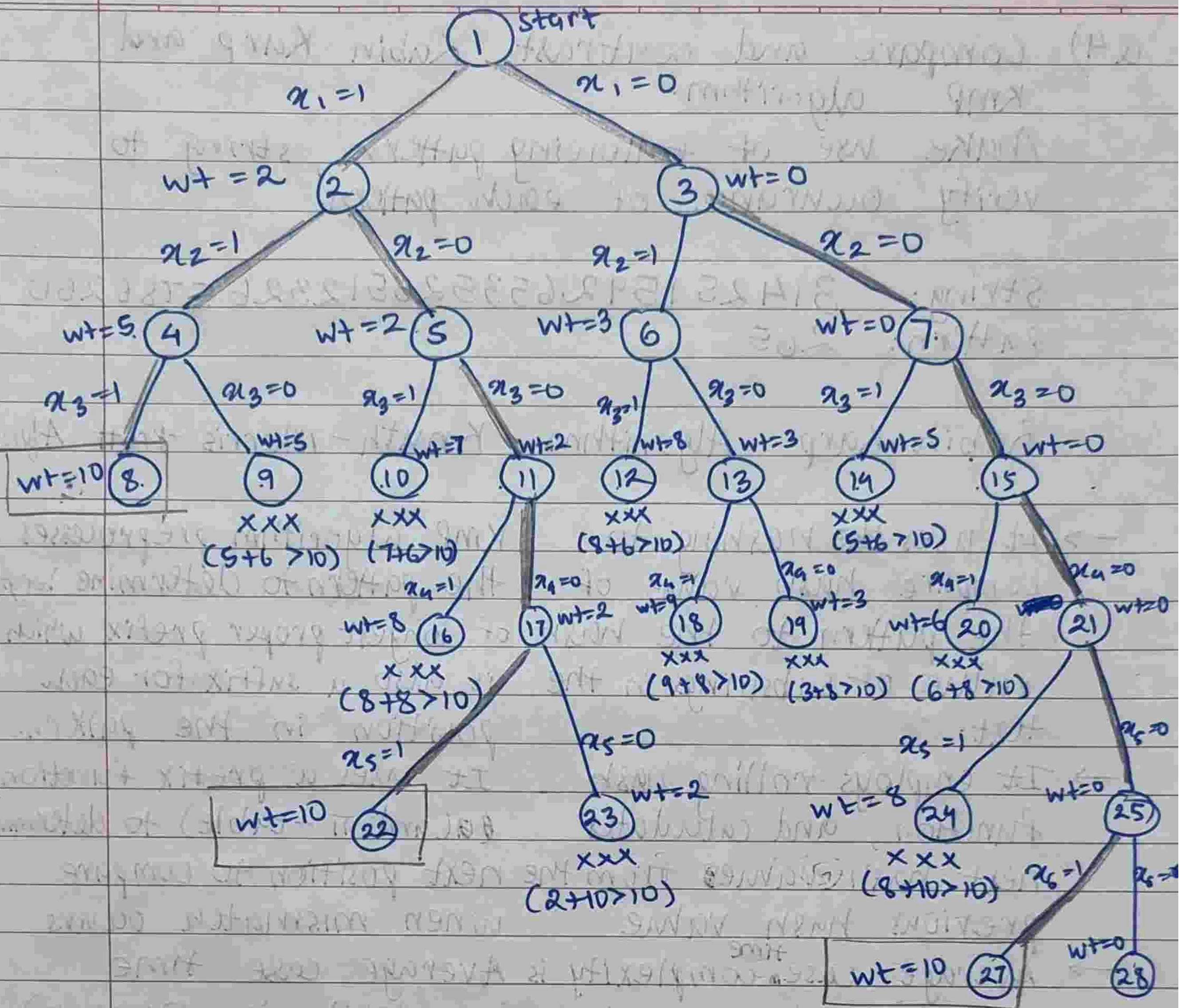
→ w is already sorted in ascending order.

$n = 3$

$m = 10$.

State space tree.





Hence, recieved solutions are:

1) $\{x_1, x_2, x_3\}$ (or) $\{1, 1, 1, 0, 0, 0\}$
2) $\{x_1, x_5\}$ (or) $\{1, 0, 0, 0, 1, 0\}$
3) $\{x_6\}$. (or) $\{0, 0, 0, 0, 0, 1\}$.

verifying:

1) $w_1 + w_2 + w_3 = 2+3+5 = 10$ ✓
2) $w_1 + w_5 = 2+8 = 10$ ✓
3) $w_6 = 10$ ✓

In the above state space tree, whenever adding ~~a new~~ the next element leads to excess (more than capacity), then ~~tree~~ ~~is~~ that branch is not explored further.

8

Q4) Compare and contrast Rabin Karp and KMP algorithm.
Make use of following pattern, string to verify occurance of each pattern.

String: 3142515926535265123265786260
Pattern: 265

Rabin Karp Algorithm	Knuth-Morris-Pratt Algo.
→ It uses ~~the~~ hashing to compare hash value of the pattern to the hash value of substrings in the text.	KMP algorithm preprocesses the pattern to determine length of longest proper prefix which is also a suffix for each position in the pattern.
→ It employs rolling hash function, and calculates next hash values from the previous hash value	It uses a prefix function, (and π-table) to determine next position to compare when mismatch occurs
→ Average case time complexity is O(m+n), where m is length of pattern, n is length of string, worst case time complexity is O(m*n)	Average case time complexity is O(m+n)
→ It can give false positives (spurious hits), hence each hit must be verified.	It will not give false positives.

Checking string for pattern.

A) Rabin Karp.

Pattern = "265"

Using base 10, $\Rightarrow d = 10$, $h = 10^2 = 100$

compute hash value as $10^2 \times 2 + 10 \times 6 + 5$
$= \underline{\underline{265}}$

S = $\underline{314}$2515926535265123265786260

Computing hash value for "314"
$\Rightarrow 3 \times 10^2 + 1 \times 10 + 4$
$= \underline{\underline{314}}$

Using rolling hash function: ~~$H_{k+1} = (H_k - S$~~
$t_{k+1} = d * (t_k - S[s+1] * h) + S[s+m+1]$

Shift	String	Hash value	Miss/Hit
0	314	314	M
1 ~~0~~	142	(314 - 300) × 10 + 2 = 142	M
2 ~~1~~	425	(142 - 100) × 10 + 5 = 425	M
3 ~~2~~	251	(425 - 400) × 10 + 1 = 251	M
4 ~~3~~	515	(251 - 200) × 10 + 5 = 515	M
5 ~~4~~	159	(515 - 500) × 10 + 9 = 159	M
6 ~~5~~	592	(159 - 100) * 10 + 2 = 592	M
7 ~~6~~	926	(592 - 500) × 10 + 6 = 926	M
8 ~~7~~	265	(926 - 900) × 10 + 5 = 265	H
9	653	(265 - 200) × 10 + 3 = 653	M
10	535	(653 - 600) × 10 + 5 = 535	M
11	352	(535 - 500) × 10 + 2 = 352	M
12	526	(352 - 300) × 10 + 6 = 526	M
13	265	(526 - 500) × 10 + 5 = 265	H
14	651	(265 - 200) × 10 + 1 = 651	M
15	512	(651 - 600) × 10 + 2 = 512	M
16	123	(512 - 500) × 10 + 3 = 123	M
17	232	(123 - 100) × 10 + 2 = 232	M

18	326	(232 - 200) x10 + 6 = 326	M
19	265	(326 - 300) x10 + 5 = 265	H
20	657	(265 - 200) x10 + 7 = 657	M
21	578	(657 - 600) x10 + 8 = 578	M
22	786	(578 - 500) x10 + 6 = 786	M
23	862	(786 - 700) x10 + 2 = 862	M
24	626	(862 - 800) x10 + 6 = 626	M
25	260	(626 - 600) x10 + 0 = 260	M

Verifying all the hits:

s=8) 265
265 ✓

s=13) 265
265 ✓

s=19) 265
265 ✓

Pattern found at 8, 13, 19 shifts.

0 spurious hits.

B) KMP Algorithm.

For pattern "265"

prefix table is

	2	6	5
	0	0	0
j 0	1	2	3

String: i

3	1	4	1	5	2	5	9	2	6	5	3	5	2	6	5	1	2	3	2	6	5	7	8	6	2	6	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

i↑

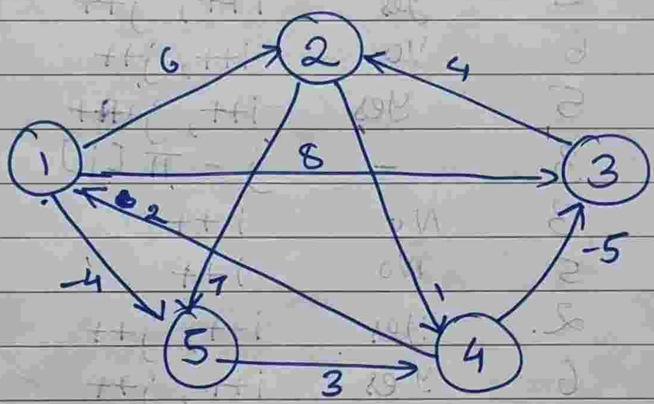
j	i	p[j+1]	s[i]	p[j+1] == s[i]?	Operation	j == m?
0	1	2	3	No	i++	✗
0	2	2	1	No	i++	✗
0	3	2	4	No	i++	✗
0	4	2	2	Yes	i++, j++	✗
1	5	6	5	No	j = π[j]	✗
0	5	2	5	No	i++	✗
0	6	2	1	No	i++	✗
0	7	2	5	No	i++	✗
0	8	2	9	No	i++	✗
0	9	2	2	Yes	i++, j++	✗
1	10	6	6	Yes	i++, j++	✗
2	11	5	5	Yes	i++, j++	✗
3	12	-	3	-	j = π[j]	✓
0	12	2	3	No	i++	✗
0	13	2	5	No	i++	✗
0	14	2	2	Yes	i++, j++	✗
1	15	6	6	Yes	i++, j++	✗
2	16	5	5	Yes	i++, j++	✗
3	17	-	1	-	j = π[j]	✓
0	18	2	2	Yes	j++, j++	✗
1	19	6	3	No	j = π[j]	✗
0	19	2	3	No	j++	✗
0	20	2	2	Yes	i++, j++	✗
1	21	6	6	Yes	i++, j++	✗
2	22	5	5	Yes	i++, j++	✗
3	23	-	7	-	j = π[j]	✓
0	23	2	7	No	i++	✗
0	24	2	8	No	i++	✗
0	25	2	6	No	i++	✗
0	26	2	2	Yes	i++, j++	✗
1	27	6	6	Yes	i++, j++	✗
2	28	5	0	No	END	✗.

For all times when $j == m$,
~~i == m-1~~ Pattern found at "$i-m-1$" shifts.

1) $12-3-1 = 8$. ✓
2) $17-3-1 = 13$ ✓
3) $23-3-1 = 19$ ✓

Hence, pattern found at 8, 13, 19 shifts.

Q5 Compute all pairs shortest path (Floyd Warshall Algorithm)



$$A^0 = \begin{array}{c|c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 6 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 3 & 0 \end{bmatrix} \end{array}$$

$$A^k(i,j) = \min\{A(i,j),\ A(i,k) + A(k,j)\}.$$

$$A^1 = \begin{bmatrix} 0 & 6 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 8 & -5 & 0 & -2 \\ \infty & \infty & \infty & 3 & 0 \end{bmatrix}$$ → considering 1 as middle vertex.

$$A^2 = \begin{bmatrix} 0 & 6 & 8 & 7 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 8 & -5 & 0 & -2 \\ \infty & \infty & \infty & 3 & 0 \end{bmatrix}$$ → considering 2 as middle vertex

$$A^3 = \begin{bmatrix} 0 & 6 & 8 & 7 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 3 & 0 \end{bmatrix}$$ — considering 3 as middle vertex

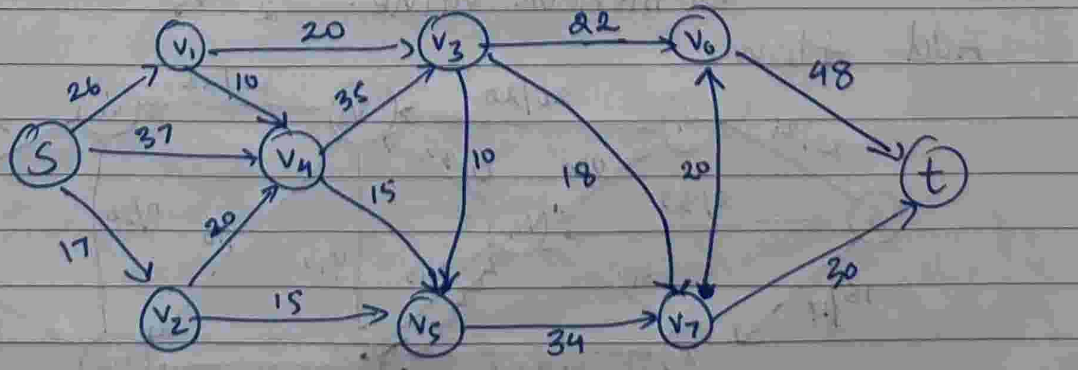
$$A^4 = \begin{bmatrix} 0 & 6 & 2 & 7 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 5 & 2 & -2 & 3 & 0 \end{bmatrix}$$ — considering 4 as middle vertex

$$A^5 = \begin{bmatrix} 0 & -2 & -6 & -1 & -4 \\ 2 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 5 & 2 & -2 & 3 & 0 \end{bmatrix}$$ — considering 5 as middle vertex

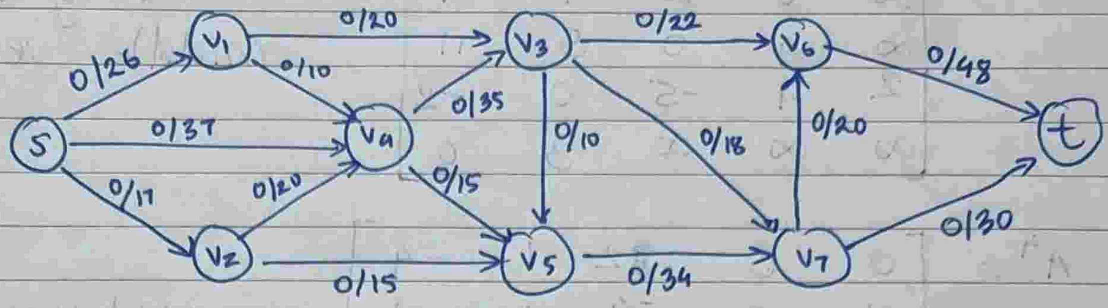
Final Answer

From \ To	1	2	3	4	5
1	0	-2	-6	-1	-4
2	2	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	5	2	-2	3	0

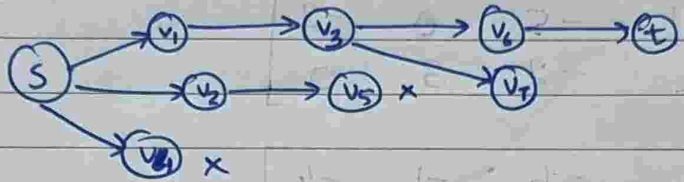
Q6) Compute max flow using Edmond Karp Algo.



Initialize all ~~B~~ flows as 0.



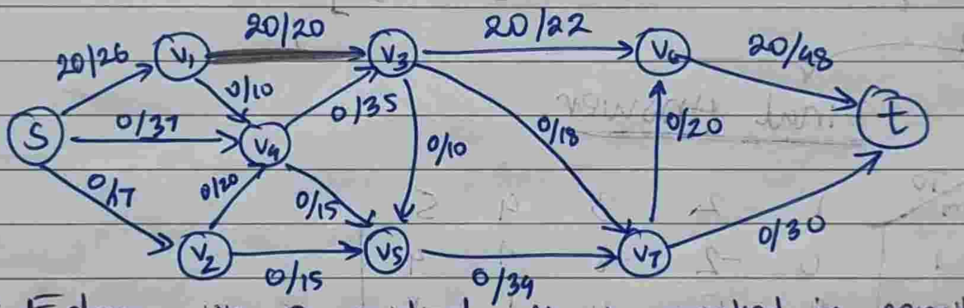
Doing B-F-S:



Augmenting path: $S - v_1 - v_3 - v_6 - t$

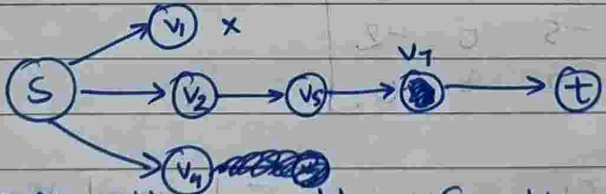
Bottleneck value: $v_1 - v_3 \rightarrow 20$.

Add flow.



* Edges with 0 residual flow marked in pencil

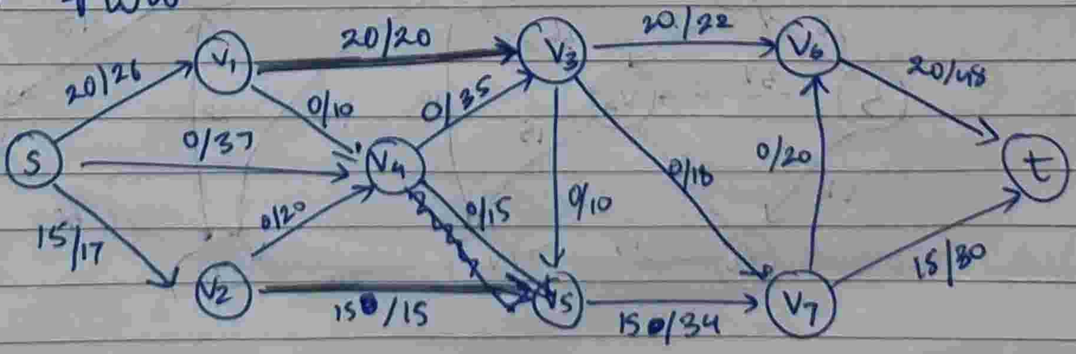
B·F·S:



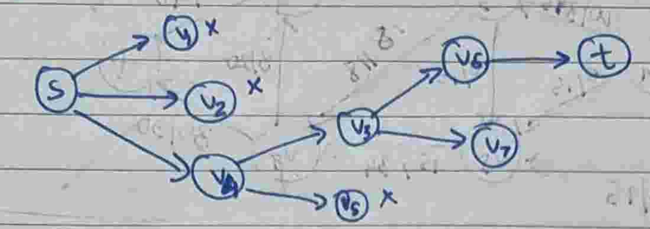
Augmenting path: $S - v_2 - v_5 - v_7 - t$

Bottleneck value: $v_2 - v_5 \rightarrow 15$

Add flow



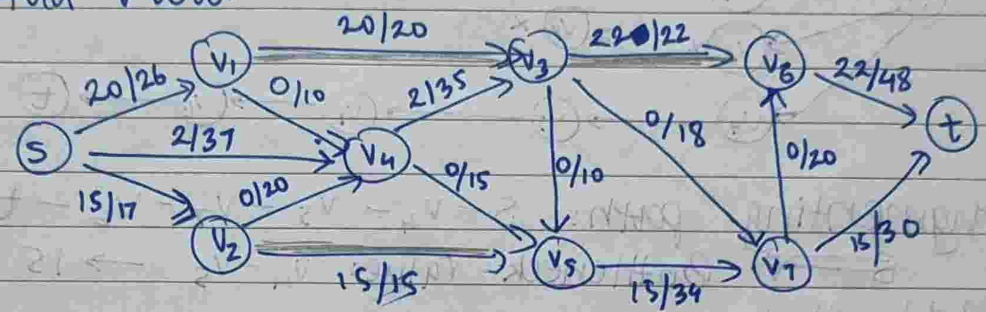
BFS. :



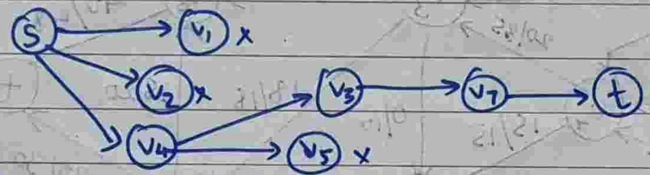
Augmenting path: $S - V_4 - V_3$ ~~[illegible]~~ $- V_6 - t$

Bottleneck value : $V_3 - V_6 \longrightarrow 2$.

Add flow.



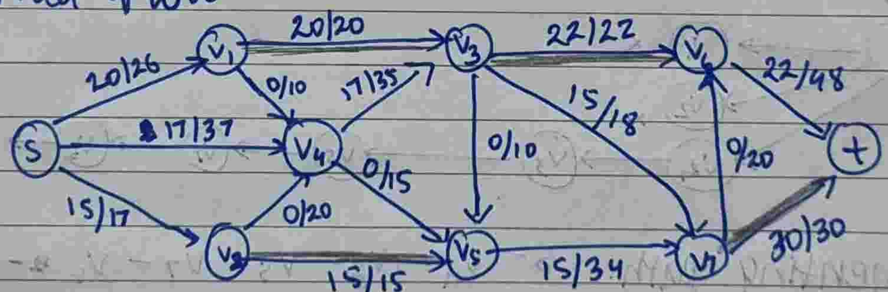
B.F.S:



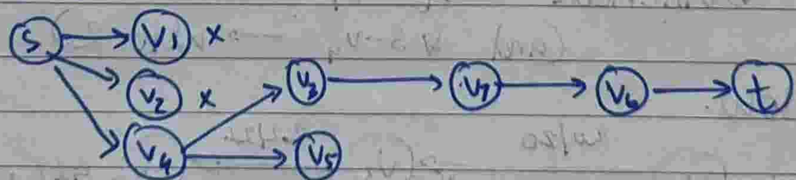
Augmenting path: $S - V_4 - V_3 - V_7 - t$

Bottle neck value: $V_7 - t \longrightarrow 15$

Add flow



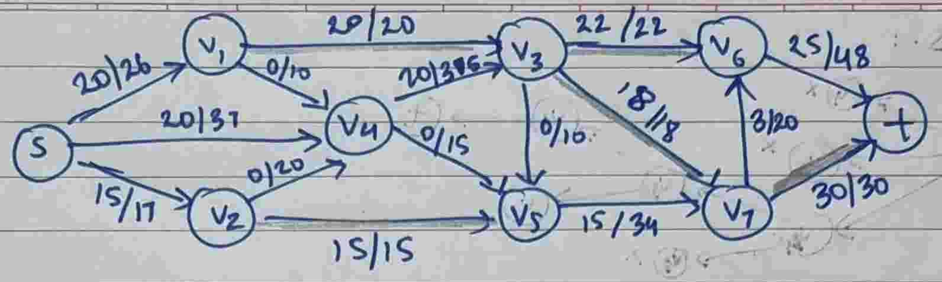
B. F. S:



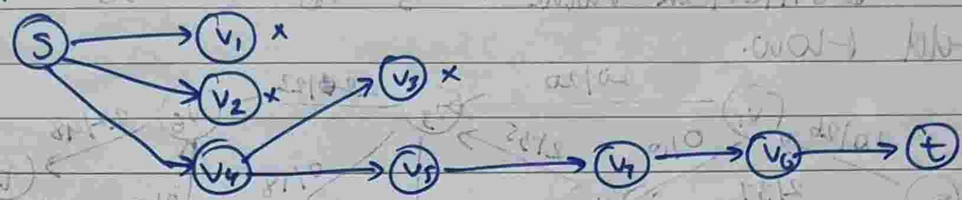
Augmenting path : $S - V_4 - V_3 - V_7 - V_6 - t$

Bottleneck value: $V_3 - V_7 \longrightarrow 3$.

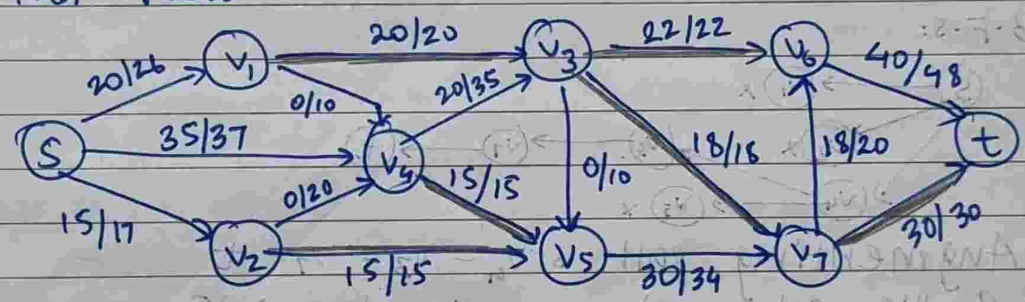
Add Flow.



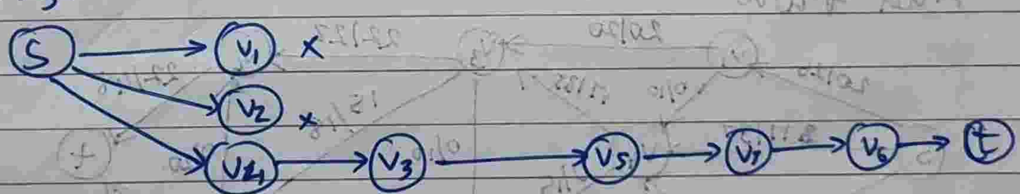
~~Augmenting path:~~
B.F.S.



Augmenting path: $S - v_4 - v_5 - v_7 - v_6 - t$
~~S – V~~ Bottleneck value: $v_4 - v_5 \rightarrow 15$
Add flow.

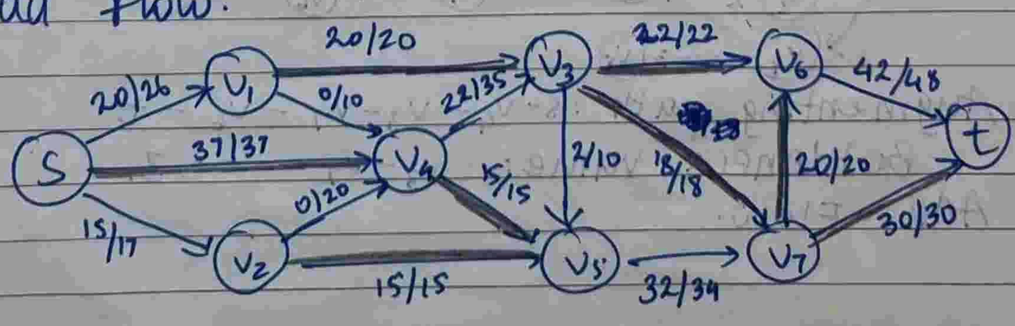


B.F.S.

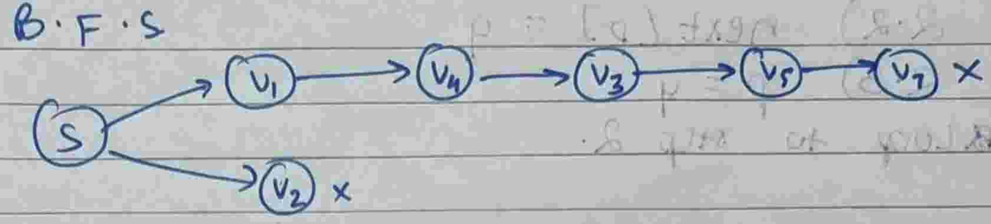


Augmenting path: $S - v_4 - v_3 - v_5 - v_7 - v_6 - t$
Bottleneck: $v_6 - v_7 \rightarrow 2$
(and $S - v_4 \rightarrow$ also 2).

Add flow.

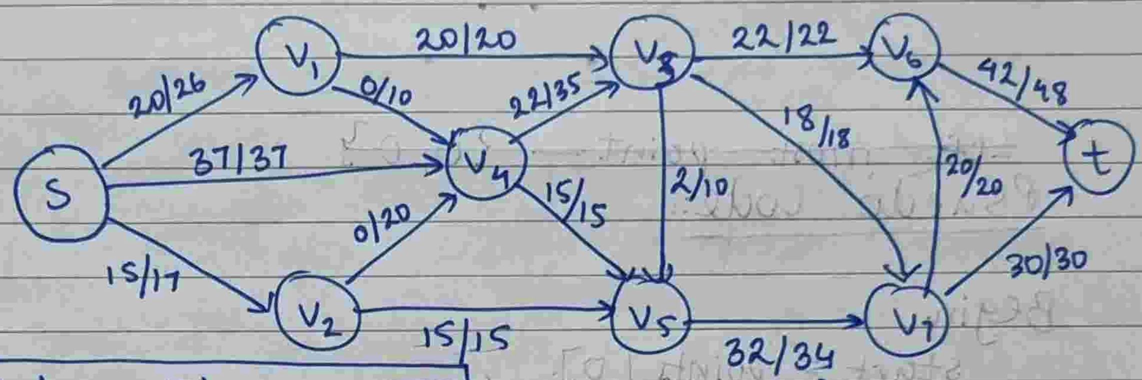


B.F.S



No more augmenting paths. possible.

Final Network:



$\boxed{\text{Max Flow} = 72}$ ..... (30+42)

Q7) Write a program to read the given coordinates and compute the Hull boundary by implementing Jarvis March Algorithm using given set of points and prove that your algorithm should print the boundary coordinates as given below.

{ {0,3}, {1,1}, {2,2}, {4,4}, {0,0}, {1,2}, {3,1}, {3,3}}

Boundary coordinates are:

{ {0,3}, {4,4}, {3,1}, {0,0}}.

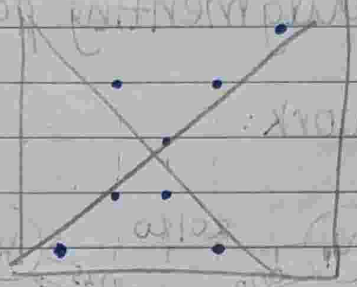
→ Algorithm:

Step 1: Initialize p as left most point

Step 2: Do while we dont come back to initial point:

2.1) Next pt is q such that (p, q, r) is is counter clockwise for any pt. r. To do this, initialize q as next pt and if pt i is more counter clockwise than q, update q as i.

2.2) next [p] = q,
2.3) p = q
Loop to step 2.



~~Left most point — {0, 0}~~

Psuedo code:

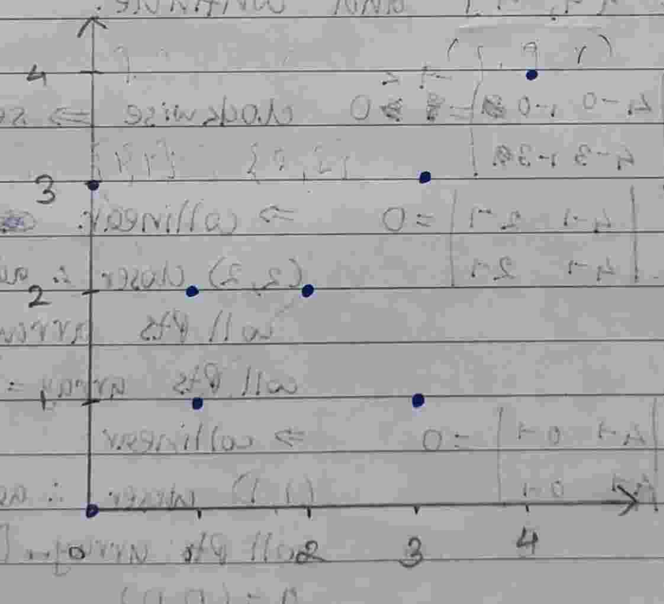
Begin
  start = points [0]
  for each point i,
    if points [i].x < start.x,
      then start = point [i]
  else ~~done~~ ~~if points [i].x == start.x and points[i].y < start.y,~~
      ~~start = points [i]~~
  current = start.
  add start point to result set.
  create coll Pts array for storing collinear pts.

  while true, do (Infinite Loop).
    next = points [i]
    for all points i except 0th point.
      if points [i] = current then,
        continue to next iteration
      val = cross product of current, next, points [i]
      if val > 0,
        next = points [i]
        clear coll Pts array
      else if val = 0,
        if next is closer to current than points [i]
          add next in the coll Pts

next = points [i]
else
add points [i] in coll Pts.
done
add all items in coll pts into result
if next = start, then
break from loop.
insert next into result
current = next
done
return result
End

→ For pts { {0,3}, {1,1}, {2,2}, {4,4}, {0,0}, {1,2}, {3,1} {3,3} }.

Solving acc to algorithm:
left most point is {0,3} ... (or {0,0})



$$(p-q)\times(r-q) = \begin{vmatrix} p \cdot x - q \cdot x & p \cdot y - q \cdot y \\ r \cdot x - q \cdot x & r \cdot y - q \cdot y \end{vmatrix}$$

20

p	q	r	$(q-q) \times (r-q)$	(p,q,r)
{0,3}	{1,1}	{2,2}	$\begin{vmatrix} (0-1) & (3-1) \\ (2-1) & (2-1) \end{vmatrix} = -3<0$	clockwise ⇒ set q=r
{0,3}	{2,2}	{4,4}	$\begin{vmatrix} 0-2 & 4-2 \\ 3-2 & 4-2 \end{vmatrix} = -6<0$	clockwise ⇒ set q=r
{0,3}	{4,4}	{0,0}	$\begin{vmatrix} 0-4 & 0-4 \\ 3-4 & 0-4 \end{vmatrix} = 20>0$	counterclockwise. ⇒ No change in q
{0,3}	{4,4}	{1,2}	$\begin{vmatrix} 0-4 & 1-4 \\ 3-4 & 2-4 \end{vmatrix} = 5>0$	counter cw
{0,3}	{4,4}	{3,1}	$\begin{vmatrix} 0-4 & 3-4 \\ 3-4 & 1-4 \end{vmatrix} = 11>0$	counter cw
{0,3}	{4,4}	{3,3}	$\begin{vmatrix} 0-4 & 3-4 \\ 3-4 & 3-4 \end{vmatrix} = 3>0$	counter cw.

Now (p, q) = ({0,3}, {4, 4}) is such that (p,q,r) is counter clockwise for all pts r.
Now put p = {4, 4} and continue.

{4,4}	{0,3}	{1,1}	$\begin{vmatrix} 4-0 & 1-0 \\ 4-3 & 1-3 \end{vmatrix} = -9<0$	clockwise ⇒ set q=r
{4,4}	{1,1}	{2,2}	$\begin{vmatrix} 4-1 & 2-1 \\ 4-1 & 2-1 \end{vmatrix} = 0$	⇒ collinear. (2,2) closer ∴ add (2,2) in coll pts array. coll pts array = [{2,2}]
{4,4}	{1,1}	{0,0}	$\begin{vmatrix} 4-1 & 0-1 \\ 4-1 & 0-1 \end{vmatrix} = 0$	⇒ collinear (1,1) closer ∴ add (1,1) to coll pts array → [(2,2), [1,1)] q = (0,0)
{4,4}	{0,0}	{1,2}	$\begin{vmatrix} 4-0 & 1-0 \\ 4-0 & 2-0 \end{vmatrix} = 4>0$	counter cw.
{4,4}	{0,0}	{3,1}	$\begin{vmatrix} 4-0 & 3-0 \\ 4-0 & 1-0 \end{vmatrix} = -8<0$	~~counter~~ cw ⇒ q=r → clear coll pts array.
{4,4}	{3,1}	{3,3}	$\begin{vmatrix} 4-3 & 3-3 \\ 4-1 & 3-1 \end{vmatrix} = 2>0$	counter ~~clockwise~~ clockwise. ~~→ clear coll pts array~~

(3, 1) is the next pt
p = {3, 1}.

p	q	r	(p-q) × (q-r)	(p, q, r)
{3,1}	{0,3}	{1,1}	$\begin{vmatrix} 3-0 & 1-0 \\ 1-3 & 1-3 \end{vmatrix} = -4 < 0$	clockwise ⇒ q = r
{3,1}	{1,1}	{2,2}	$\begin{vmatrix} 3-1 & 2-1 \\ 1-1 & 2-1 \end{vmatrix} = 2 > 0$	counter cw.
{3,1}	{1,1}	{4,4}	$\begin{vmatrix} 3-1 & 4-1 \\ 1-1 & 4-1 \end{vmatrix} = 6 > 0$	counter cw.
{3,1}	{1,1}	{0,0}	$\begin{vmatrix} 3-1 & 0-1 \\ 1-1 & 0-1 \end{vmatrix} = -1 < 0$	clockwise ⇒ q = r.
{3,1}	{0,0}	{1,2}	$\begin{vmatrix} 3 & 1 \\ 1 & 2 \end{vmatrix} = 2 > 0$	counter cw
{3,1}	{0,0}	{3,3}	$\begin{vmatrix} 3 & 3 \\ 1 & 3 \end{vmatrix} = 6 > 0$	counter cw.

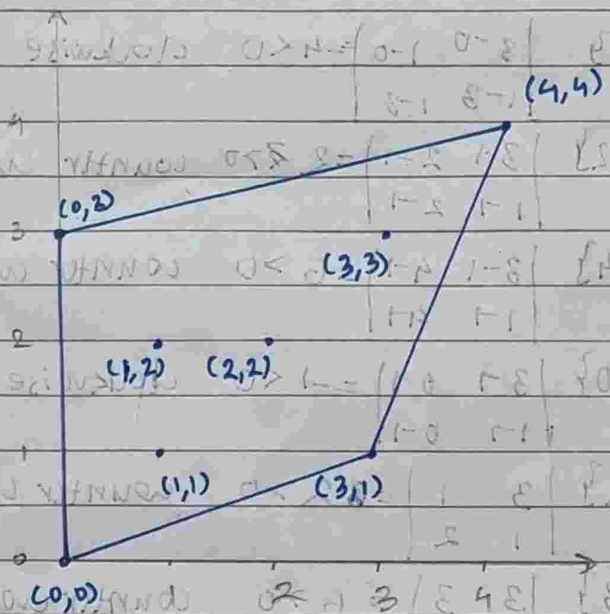
{0,0} added in the solution set.
continue with p = {0,0}

{0,0}	{0,3}	{1,1}	$\begin{vmatrix} 0-0 & 1 \\ 0-3 & -2 \end{vmatrix} = 3 > 0$	counter cw
{0,0}	{0,3}	{2,2}	$\begin{vmatrix} 0-0 & 2 \\ 0-3 & -1 \end{vmatrix} = 6 > 0$	countercw
{0,0}	{0,3}	{4,4}	$\begin{vmatrix} 0 & 4 \\ -3 & 1 \end{vmatrix} = 12 > 0$	countercw
{0,0}	{0,3}	{1,2}	$\begin{vmatrix} 0 & 1 \\ -3 & -1 \end{vmatrix} = 3 > 0$	countercw
{0,0}	{0,3}	{3,1}	$\begin{vmatrix} 0 & 3 \\ -3 & -2 \end{vmatrix} = 9 > 0$	countercw
{0,0}	{0,3}	{3,3}	$\begin{vmatrix} 0 & 3 \\ -3 & 0 \end{vmatrix} = 9 > 0$	countercw.

Next pt → {0,3}.
Hence, {0,3} (starting pt is reached again}.

Final solution:

{ {0,3}, {4,4}, {3,1}, {0,0} }.



Final convex Hull.