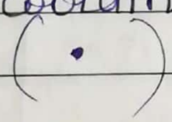
Computational Geometry

→ It is the branch of computer science that studies algorithms for solving geometric problems.

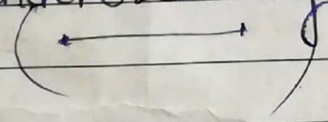
areas are → robotics, computer aided design, molecular modelling, metallurgy, manufacturing, textile layout, forestry and statistics.

Geometric Objects:

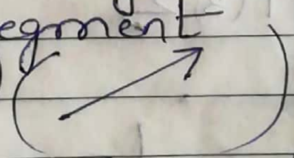
• Points: a point (x,y) is represented using two cartesian coordinates: $x, y \in R$. The origin $(0,0)$ is point.



• Lines: Line $\overline{p} = (p1, p2)$ is represented by two points $p1 \neq p2$ which represent the two end points of a line. Sometimes we may represent a line using only one point, with the understanding that second point is the origin.



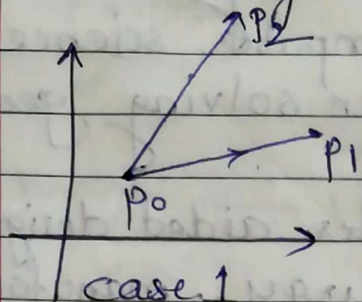
Vectors: A vector $\vec{v} = (v_1, v_2)$ is a direction. A vector v and a point (x,y) may represent the directed line segment (x,y) $(x+v_1)$ $(y+v_2)$



Polygons: A polygon is a closed shape made up of straight edges $(e_1, e_2, \ldots, e_k)$ where e_i is a line for $i = 1, \ldots, k$ and the start point of e_1 is the end point of e_k.

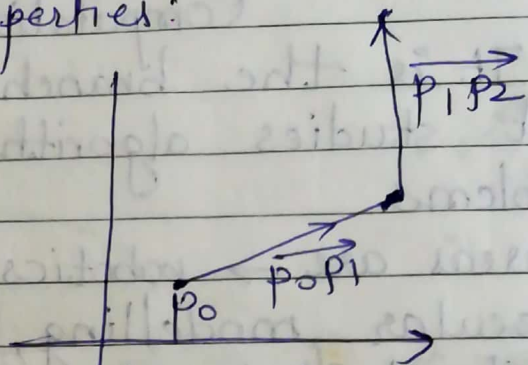
Scanned by CamScanner

Line segment Properties:



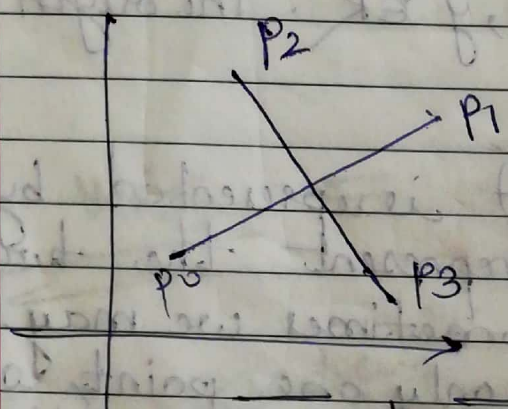
case 1

Is $\overrightarrow{p_0p_2}$ clockwise or anticlockwise from $\overrightarrow{p_0p_1}$



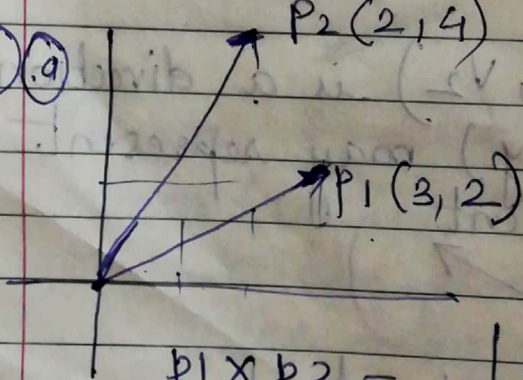
case 2:

If we traverse from p_0p_1 to p_1p_2, do we make left or right.



Do $\overline{p_0p_1}$ & $\overline{p_2p_3}$ intersect

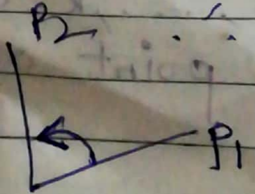
1. (a)



$\overrightarrow{p_0p_1} = (3,2)$ $\overrightarrow{p_0p_2} = (2,4)$

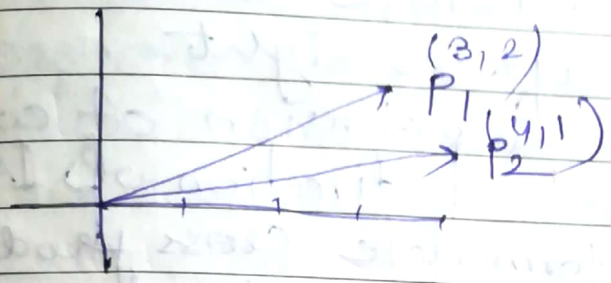
$$p1 \times p2 = \begin{vmatrix} 3 & 2 \\ 2 & 4 \end{vmatrix} = 12 - 4 = 8 > 0$$

$\therefore \overrightarrow{p_0p_2}$ is anticlockwise to $\overrightarrow{p_0p_1}$.



left turn

(b)



$P_1 \times P_2$

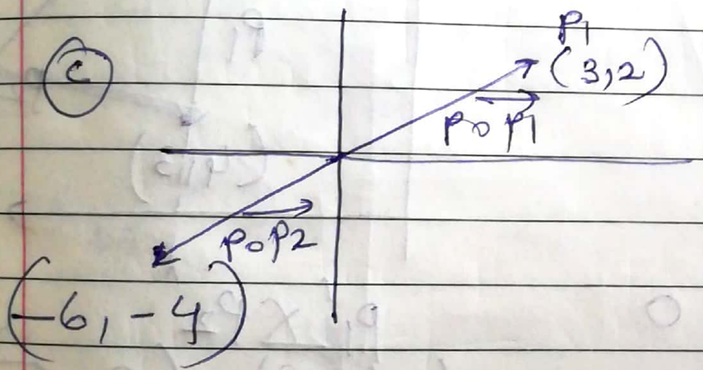
$$\begin{vmatrix} 3 & 2 \\ 4 & 1 \end{vmatrix}$$

$= 3 - 8 = -5$

$= < 0.$

$\therefore$ P_2 is clockwise to P_1

(c)



$P_1 \times P_2$

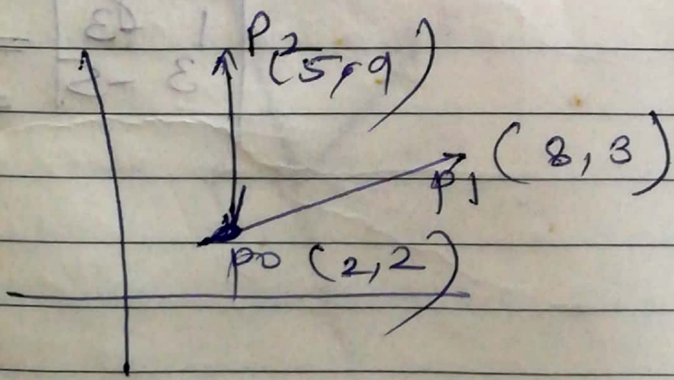
$$= \begin{vmatrix} 3 & 2 \\ -6 & -4 \end{vmatrix}$$

$= -12 + 12$

$= 0$

$\overrightarrow{P_0P_1}$ & $\overrightarrow{P_0P_2}$ are collinear.

(d)

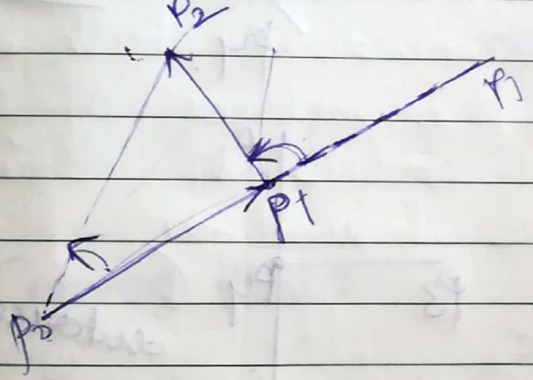


$\overrightarrow{P_0P_1}$ (6, 1) $\overrightarrow{P_0P_2}$ (3, 7)

$$\begin{vmatrix} 6 & 1 \\ 3 & 7 \end{vmatrix} = 42 - 3$$

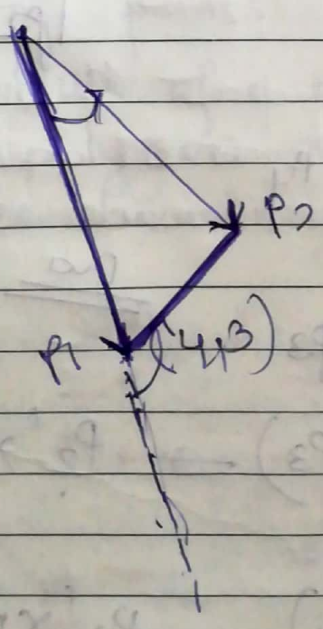
$= 39.$

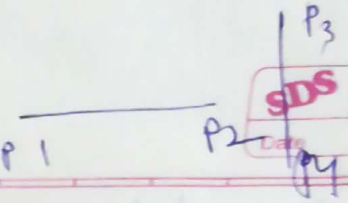
case2: determine whether consecutive segments turn left or right.
This is extension of case 1
1. Extend direction of 1st vector
2. Formulate cross products of extended vector & 2nd vector!
3. Apply Axioms



$p_0p_1 \times p_0p_2 > 0$ left turn.
anticlockwise

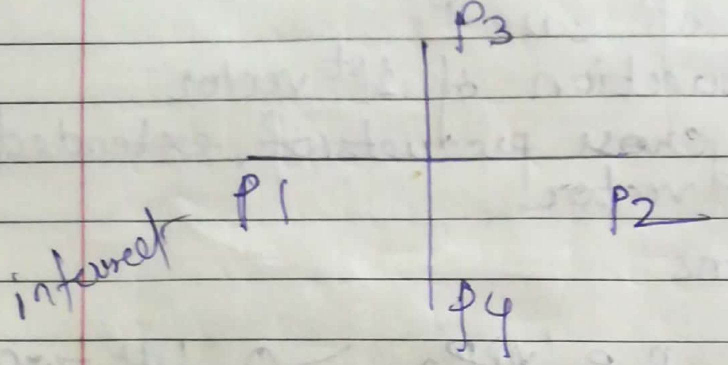
$\overrightarrow{p_0p_1} \times \overrightarrow{p_0p_2}$ ...

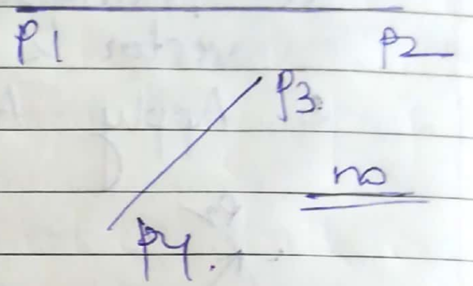


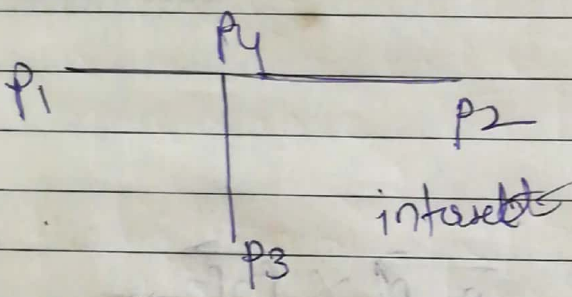


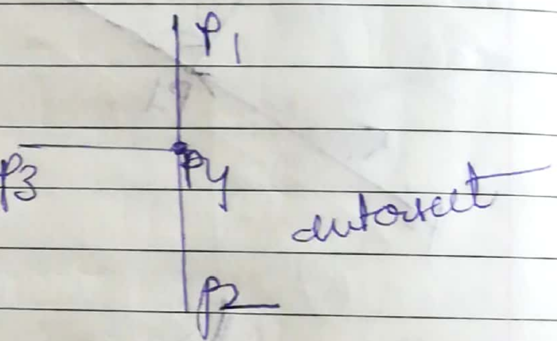
Case 3

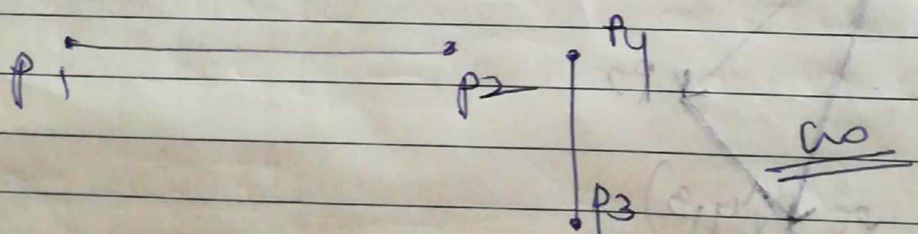
Determine whether two line segments intersect











direction$(P_1, P_2, P_3) \rightarrow P_2' \times P_3' \rightarrow +ve$
→ anticlockwise
→ left turn

direction$(P_1, P_2, P_4) \rightarrow P_2' \times P_4' \rightarrow -ve$
→ clockwise
→ right turn

IIIly direction$(P_3, P_4, P_1) \rightarrow P_4' \times P_1' \rightarrow$ ~~+~~ve

direction$(P_3, P_4, P_2) \rightarrow P_4' \times P_2' \rightarrow +ve$

onsegment (p_1, p_2, p_4)

$\min(x_1, x_2) \leq x_4 \leq \max(x_1, x_2)$
or $\min(y_1, y_2) \leq y_4 \leq \max(y_1, y_2)$

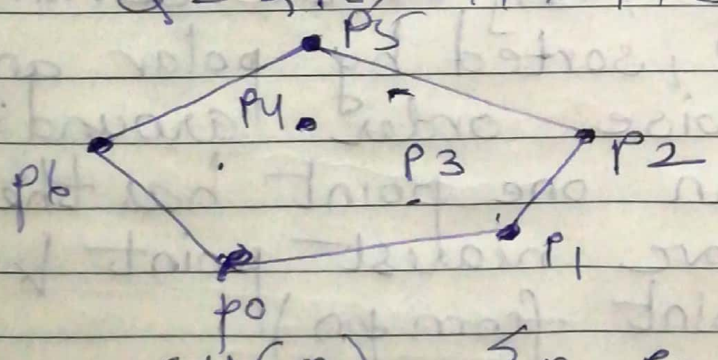
direction $(p_i, p_j, p_k) \longrightarrow p_j' \times p_k'$
origin at p_i

return $\underbrace{(p_j - p_i)}_{p_j'} \times \underbrace{(p_k - p_i)}_{p_k'}$

Convex Hull:

Q set of points
Convex Hull of a set Q is the smallest convex polygon p for which each point in Q is on the boundary or its interior.

$Q = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6\}$



$CH(Q) = \{p_0, p_1, p_2, p_5, p_6\}$

Intuitively, we can think of each point in Q as being a nail sticking out from a board. The convex hull is then the shape formed by a tight rubber band that surrounds all the nails.

Scanned by CamScanner

Graham's scan:

It solves the convex hull problem by maintaining a stack S of candidate points. It pushes each point of input set Q onto the stack one at a time, and it eventually pops from the stack each point that is not on boundary of polygon.

When algorithm terminates, stack S contains exactly the vertices of CH(Q) in counterclockwise order of their appearance on the boundary.

Precondition $|Q| \geq 3$

Graham_scan(Q)

1. Let p_0 be the point Q with minimum y-cordinate or leftmost such point in case of tie. (minimum x)

2. Let $\{p_1, \ldots, p_m\}$ be the remaining points in Q, sorted by polar angle in counterclockwise order around p_0 (if more than one point has the same angle, remove nearest point & keep farthest point from p_0)

3. push(S, p_0)
4. push(S, p_1)
5. push(S, p_2)

Scanned by CamScanner

```
for i=3 to n
{
  while(true)
  {
    ~~pk = s_top_peek(s)~~
    ~~pj = peek(s);~~
    ~~if(~~

       pj = peek(s)
       pk = peek_second_top(s)

       if(direction(pk, pj, pi) < 0) //right turn
                                         cw.
            pop pj; // pj is interior point
       else
       {    push(pi)
            break;
       }
  }
}
```