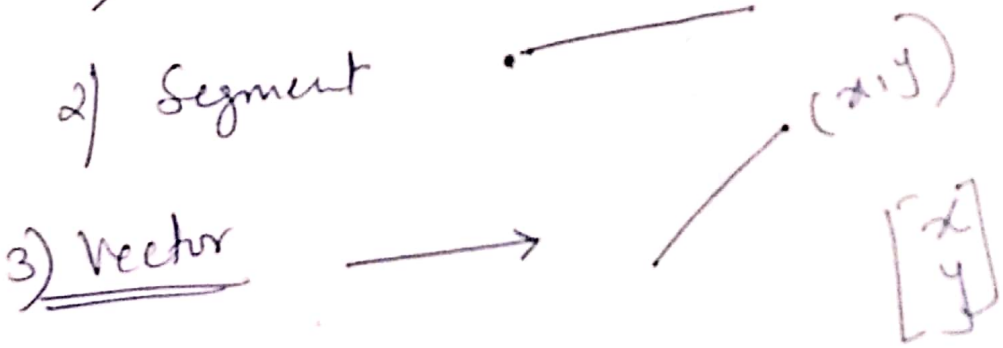
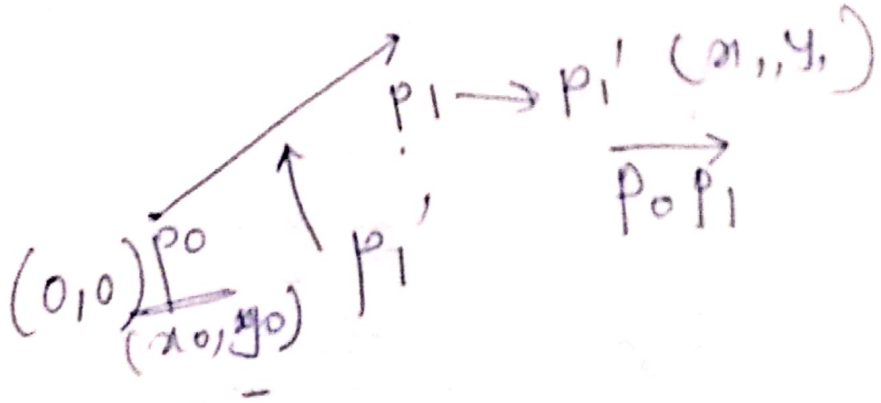
1) Line

2) Segment

3) Vector





$(x_1, x_2, x_3, \ldots x_d)$

AA R^d $R^2(x,y)$

Push-Relabel & Relabel to front $T(n) = O(V^2)$

Computational Geometry :—

Q1] Given 2 line segments $\overrightarrow{P_0P_1}$ $\overrightarrow{P_0P_2}$ (common end point i.e. P_0). We need to check whether segment P_0P_2 is clockwise or anticlockwise w.r.t. P_1.

Soln: By Right hand thumb rule.

Take $P_1 \times P_2$ (Cross product)

If sign of resultant is +ve, then from P_1 to P_2 rotation is anticlockwise or counterclockwise

If sign of resultant is −ve, then from P_1 to P_2 rotation is clockwise.

If sign (value) is 0. Both are collinear

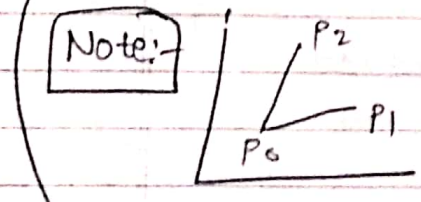
Ex. $P_1(3,2)$ & $P_2(-6,-4)$.

$$\begin{vmatrix} x & y \\ 3 & 2 \\ -6 & -4 \end{vmatrix} \quad -12 + 12 = 0.$$

∴ They are collinear

To move from P_0 to P_1 $(\overrightarrow{P_0P_1})$ to $(\overrightarrow{P_1P_2})$ whether there is left turn or right turn.

Note:-



$\Rightarrow$ Bring to Origin.

$P_1' = (P_1 - P_0)$ $P_2' = (P_2 - P_0)$

If $P_1' \times P_2' \Rightarrow$ Resultant is +ve.

Then Anticlockwise $\Rightarrow$ left turn.

If $P_1' \times P_2' \Rightarrow$ Resultant is −ve

Then Clockwise $\Rightarrow$ Right turn.

If it is '0' $\Rightarrow$ Straight way

P_1' & P_2' are used since, always P_0 won't

required = "required" not required.

Jarvis March ⇒ $T(n) = O(nh)$ (No. of points, h)

~~$T(n) = O(n \log n)$~~

Graham-scan (Q)

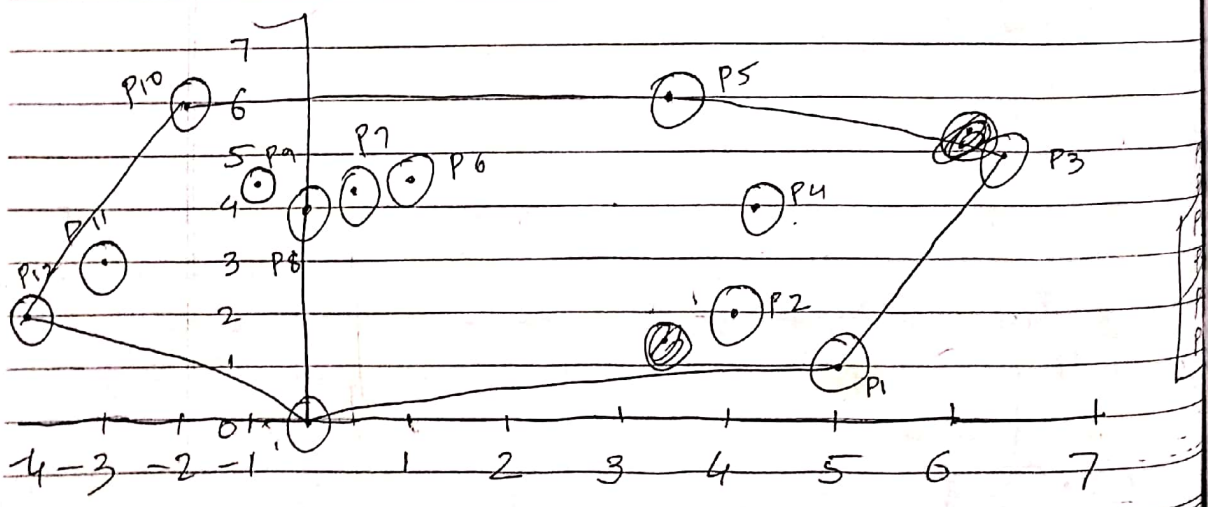
1] Select anchor point (leftmost & smallest)

2] $\{P_1, \ldots, P_m\}$ w.r.t. P_0 ⇒ sort

3] push (S, P_0), push (S, P_1), push (S, P_2)

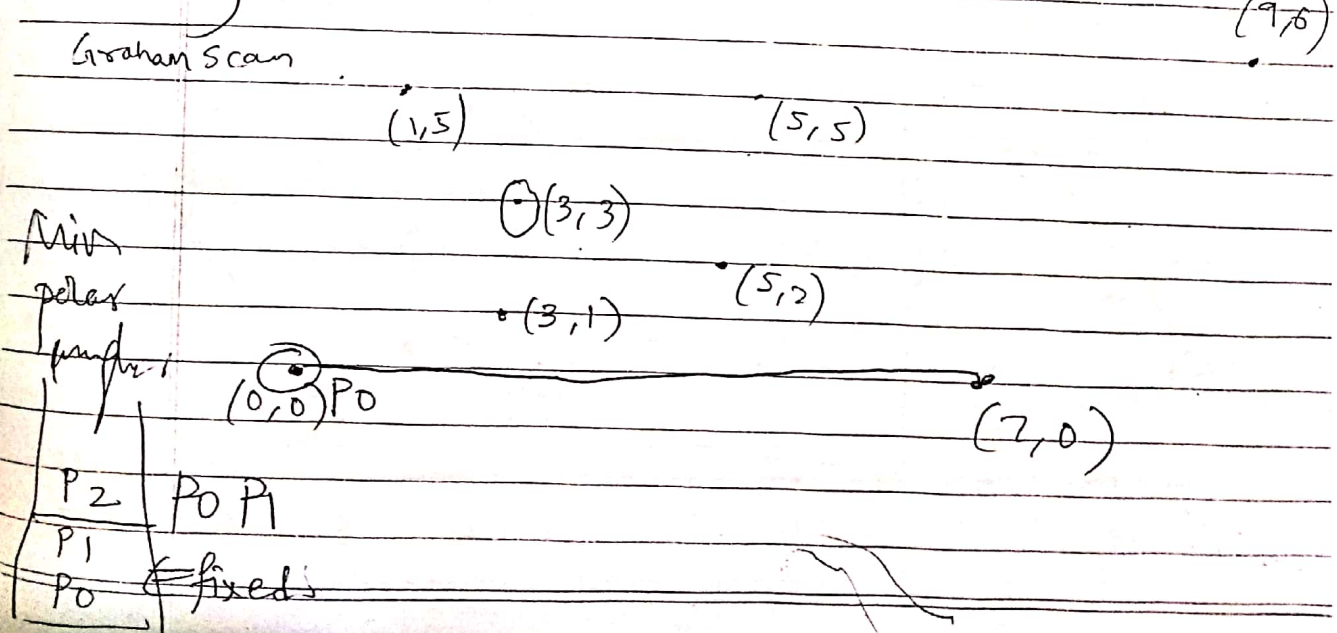
```
for i=3 to m
  while (true)
    { pk = s.top-peek(s);
      pj = peek(s);
      if (direction (pk, pj, pi) < 0)
          pop (s);
      else
        { push (pi);
          break;
        }
    }
}
```

(0,0), (5/6, 1), (4,2) (6,5), (4,4)
(3,5 6, 1 55), (1,4.5), (0,4) (-1,4.4)
(-2,6), (-3,3), (-4,2), (0.5, 4.5)



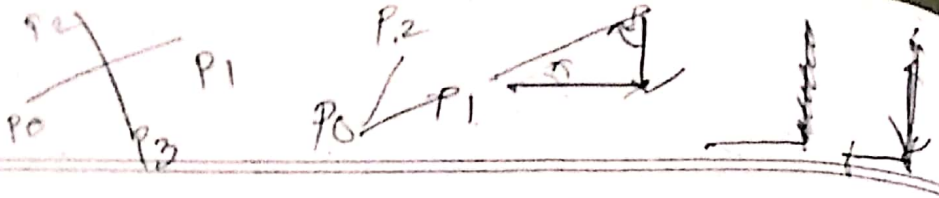
```
    return true;
}
else if (d1 == 0 && on-segment(P1, P2, P3))
    return true;
else if (d2 == 0 && on-segment(P1, P2, P4))
    return true;
else if ....
    return true
else if ....
    return true
else
    return false.
}
int direction (Pi, Pj, Pk)
{
    if(cross product ((Pj - Pi) x (Pk - Pi)) >= 0)
        return 1;
    else if( ..... < 0 )
        return -1;
    else
        return 0;
}
```

Graham scan

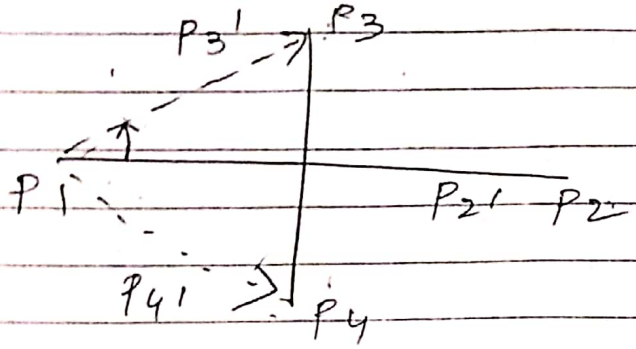


Min polar angle

fixed



Determine whether any 2 line segments intersect or not.



If one orientation is anticlockwise & other is clockwise. Hence they lie at 2 diff sides of line P_1P_2 & hence intersect.

Now with respect to P_3 P_4.

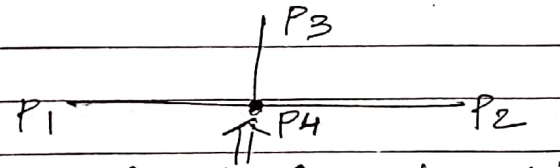
Find. $P_4 \times P_1 \Rightarrow$ Clockwise.

$P_4 \times P_2 \Rightarrow$ Anticlockwise

So P_1 & P_2 lie on diff sides of line P_3

Hence Now Surely they intersect

Exception:-



One in One direction & other on the line segment still intersect.

```
boolean Segment-intersect(P1, P2, P3
{
    d1 = direction (P1, P2, P3) → P2'
    d2 = direction (P1, P2, P4) → P2'x
    d3 = direction (P3, P4, P1) → P4'x
    d4 = direction (P3, P4, P2) → P4'x
    if((d1>0 and d2<0) or (d1<0 and d2
        and (d3>0 and d4<0) or (d3<0 and d4
```