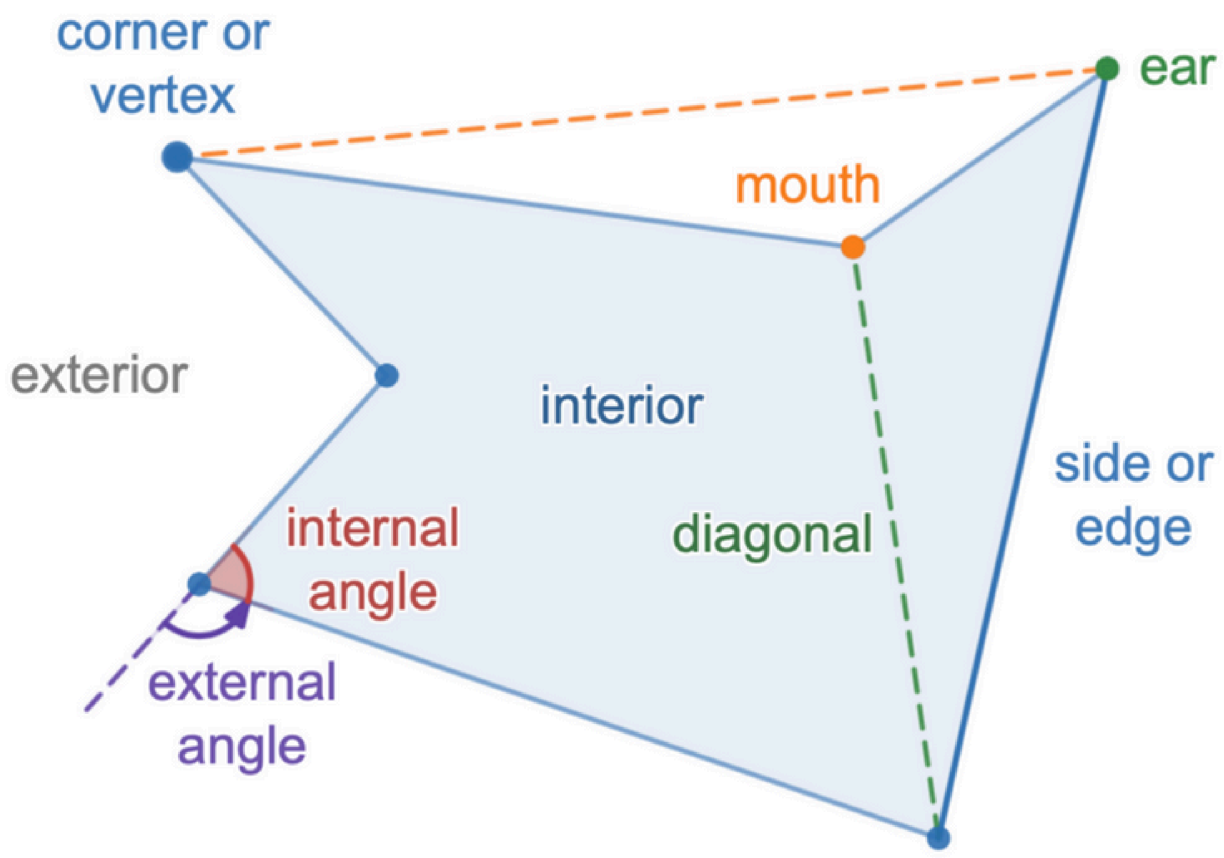
PKW

JANUARY 23, 2025

COMPUTATIONAL GEOMETRY (POLYGON)

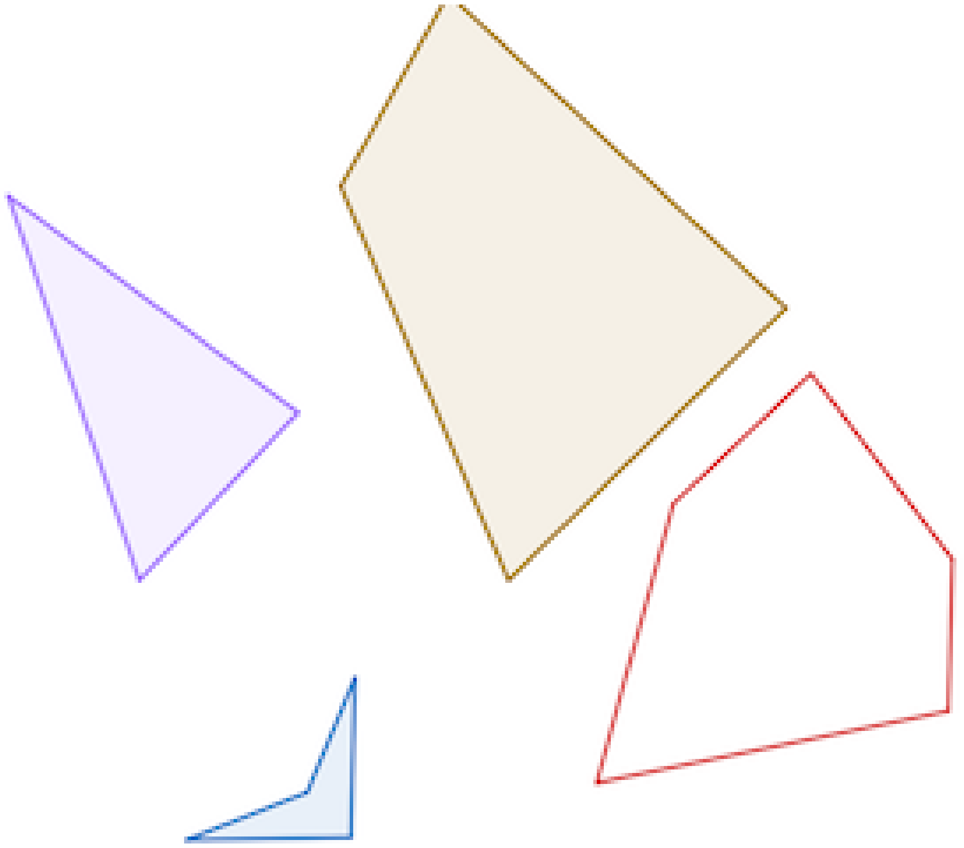
เรขาคณิตเชิงคำนวณ (รูปหลายเหลี่ยม)



POLYGON

IN GEOMETRY, A SIMPLE POLYGON IS A POLYGON THAT DOES NOT INTERSECT ITSELF AND HAS NO HOLES.

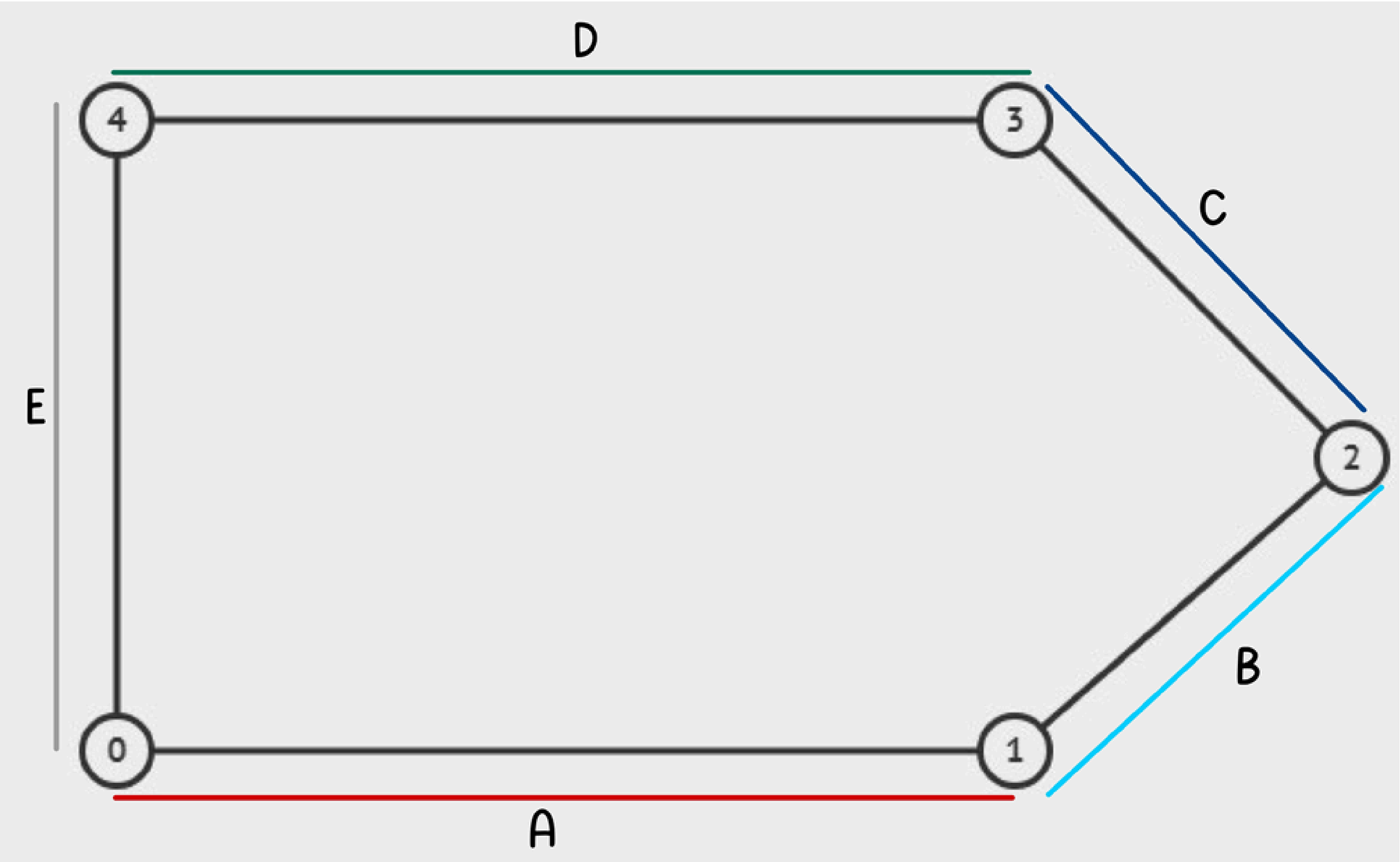
(รูปหลายเหลี่ยมธรรมดาคือรูปหลายเหลี่ยมที่ไม่ตัดกันและไม่มีรู)



1. PERIMETER (P) ★★
2. AREA (P) ★★★★★
3. ISCONVEX (P) ★★★★★
4. ISPOLYGON (PT,P) ★★★★★
5. CUTPOLYGON (LN,P) ★★★★★

PERIMETER (P)

- เป็นการวัดเส้นรอบรูปของ POLYGON โดยวัดในแต่ละจุด จากจุดหนึ่งไปอีกจุดหนึ่ง และนำความยาวของเส้นทั้งหมดมารวมกัน เพื่อให้ได้เป็นเส้นรอบรูป



หลักการคิดในส่วนนี้คือการนำความยาวของแต่ละเส้นมาบวกกัน

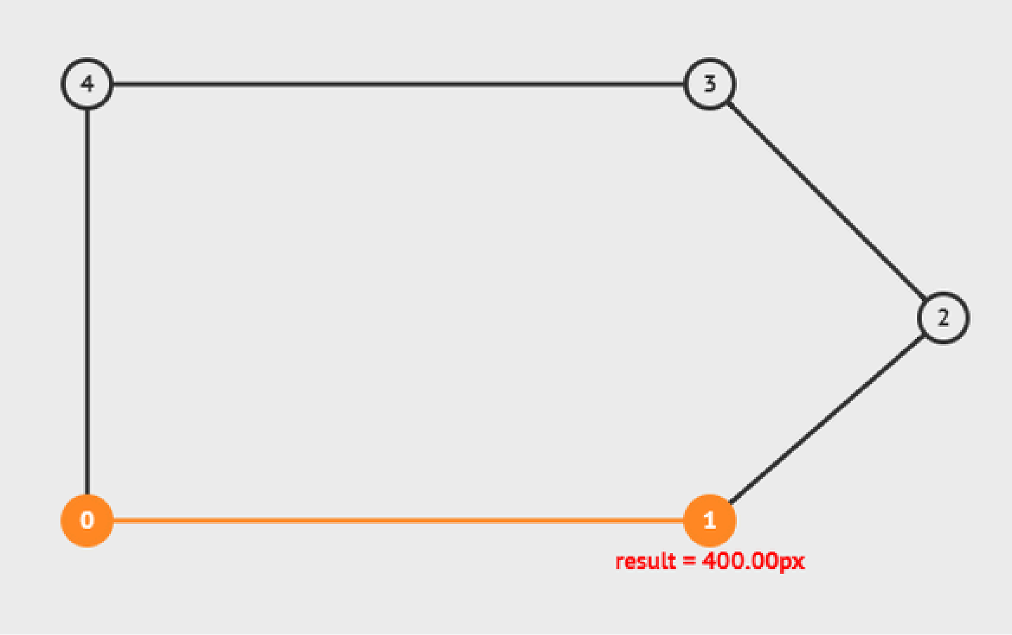
A + B +C + D + E = perimeter(P)

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

PERIMETER (P)

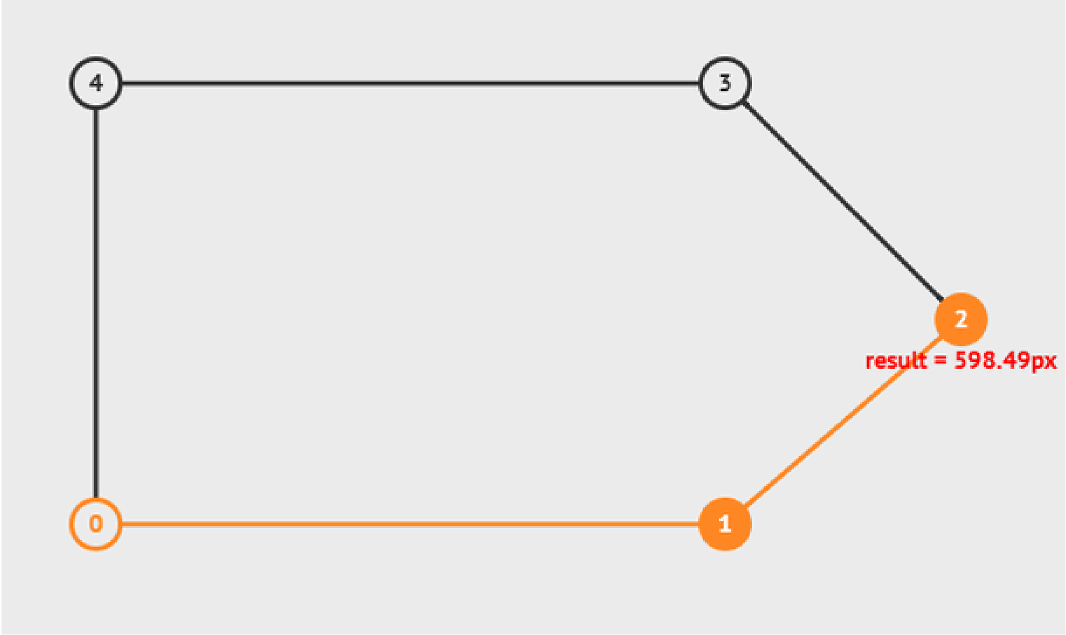
หลักการทำงานของสูตรกับการเขียนโปรแกรม

1.



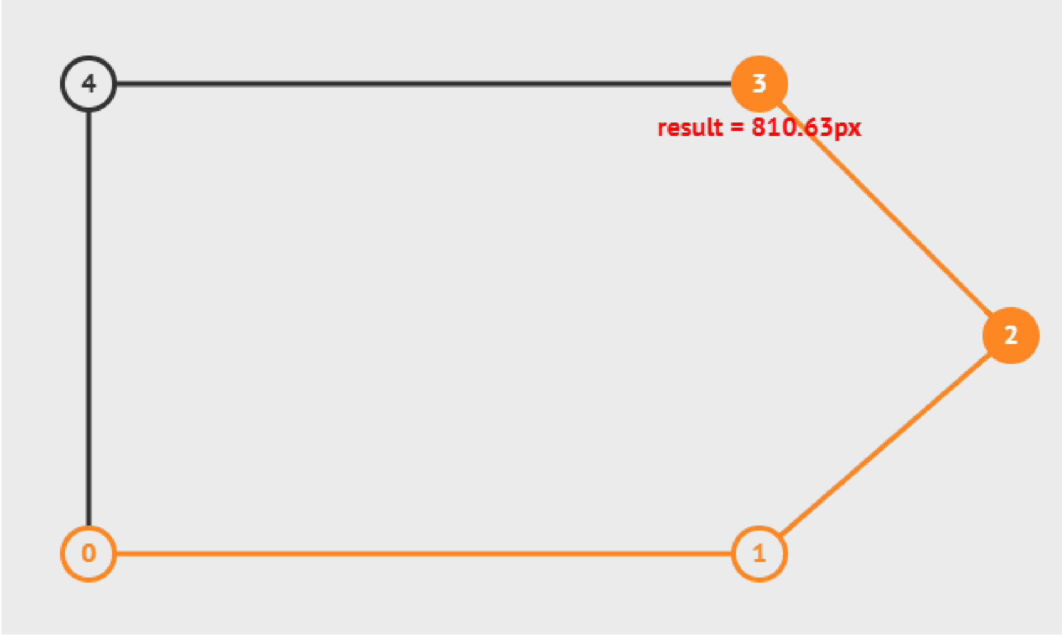
A

2.



A + B

3.

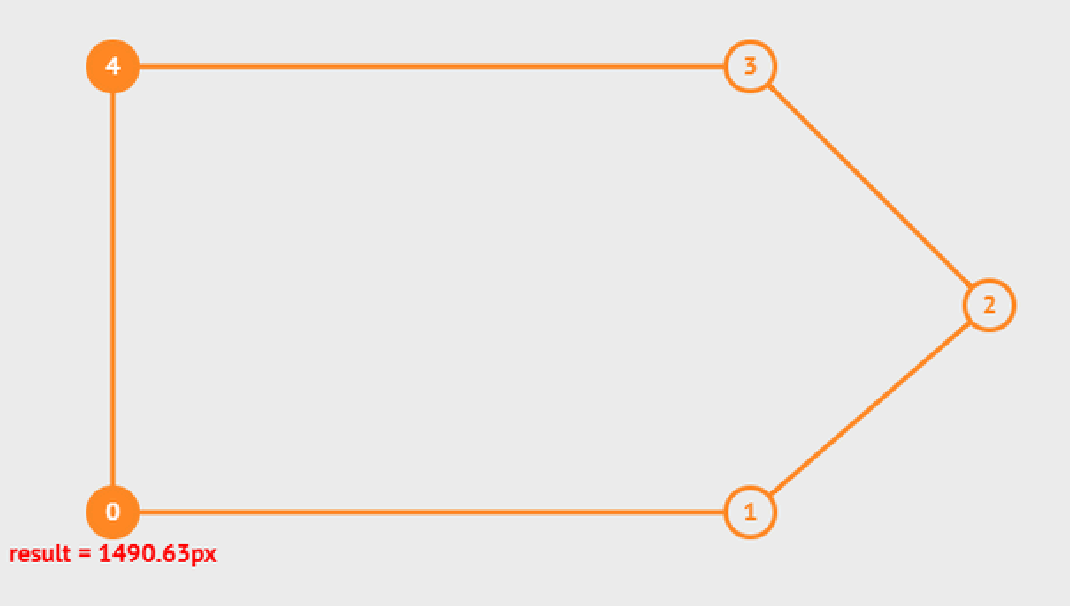


A + B +C

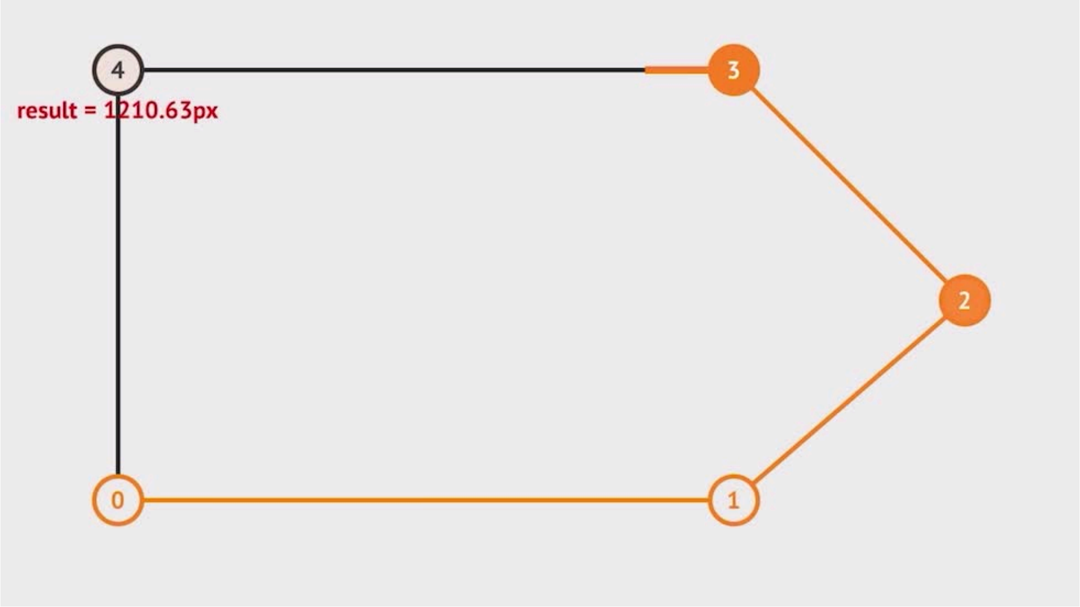
4.

A + B +C + D

5.



A + B +C + D + E



A + B +C + D + E = perimeter(P)

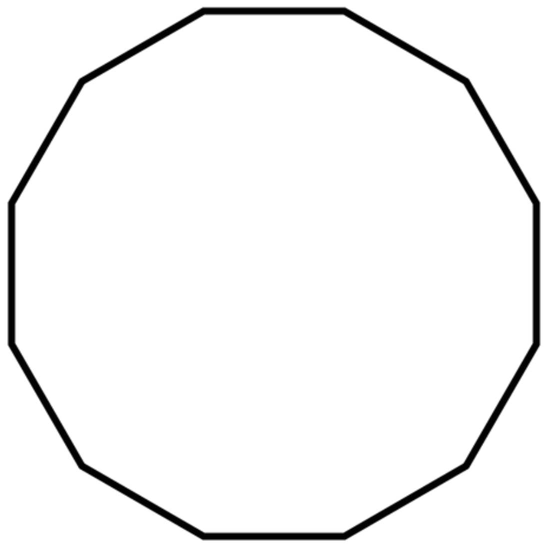
AREA (P)

- ทำการกำหนดจุดยอดของรูปหลายเหลี่ยมในลักษณะเป็นวงกลม (ตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา) เราสามารถคำนวณพื้นที่โดยใช้ สูตรเชือกผูกรองเท้า (SHOELACE FORMULA)

  สูตรเชือกผูกรองเท้า (SHOELACE FORMULA)
- ใช้คำนวณพื้นที่รูปหลายเหลี่ยมจากพิกัดจุดยอด
- อาศัยผลคูณเชิงเวกเตอร์ (CROSS PRODUCT) ของจุดปลายแต่ละด้าน
- พื้นที่ = ครึ่งหนึ่งของผลรวมทั้งหมด และใช้ค่าสัมบูรณ์

การกำหนดจุดในลักษณะวงกลม คือการนับจากจุดเริ่ม 0 จนถึง 0

$$(0) \rightarrow (1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (0)$$



AREA (P)

ADD :

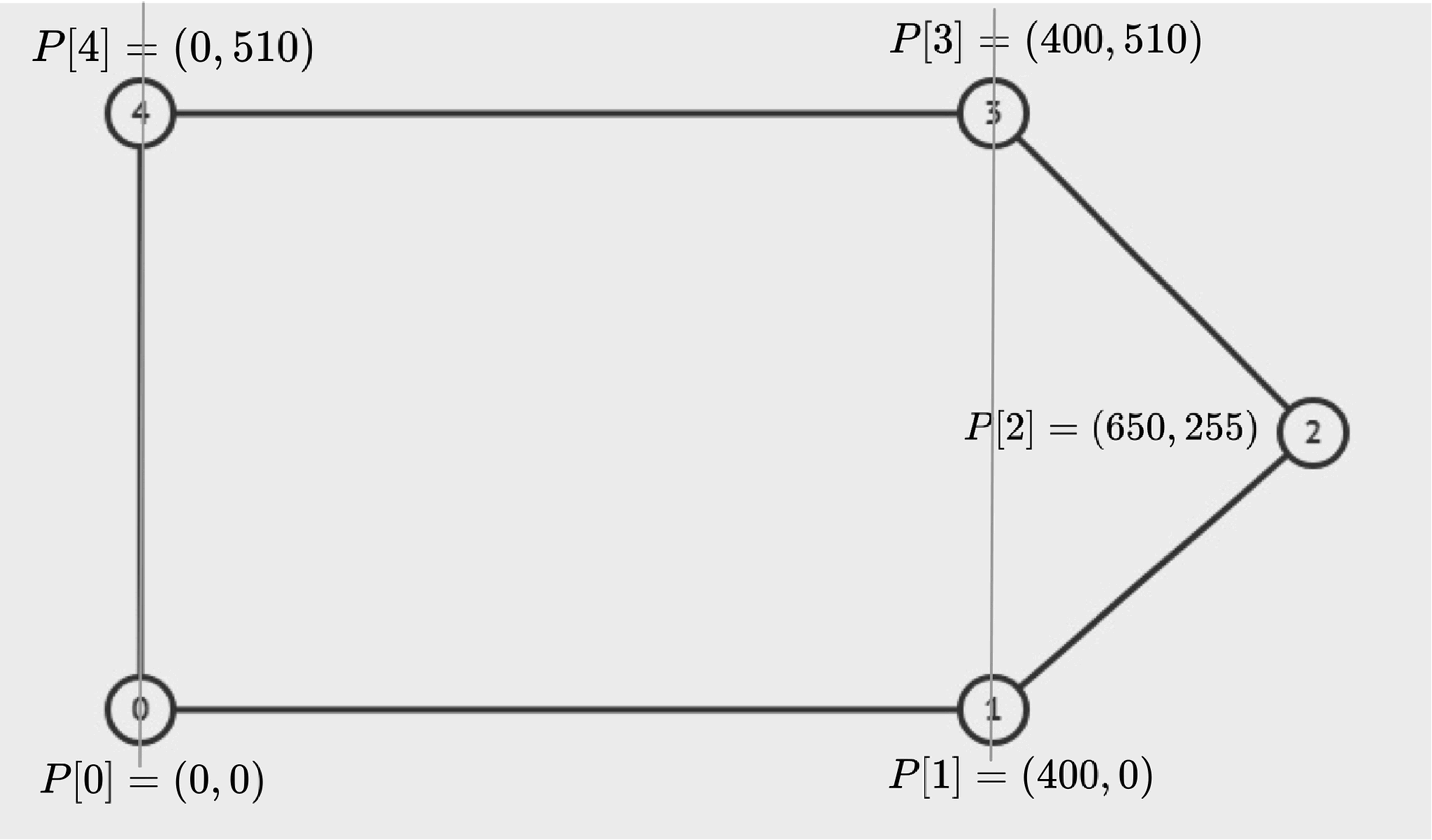
ตามเข็มนาฬิกา และ ทวนเข็มนาฬิกา = (cw (−) , ccw(+))

การเรียงลำดับจุดยอดมีผลต่อการกำหนด "ทิศทางการวน" (cw หรือ ccw) ซึ่งจะกำหนดเครื่องหมายของพื้นที่คำนวณ (บวกหรือลบ) การใส่ค่าสัมบูรณ์ในสูตรจะช่วยให้ได้ขนาดพื้นที่ที่ถูกต้องเสมอ ไม่ว่าจะเรียงแบบไหนก็ตาม

สูตรของ SHOELACE FORMULA (เชือกผูกรองเท้า)

$$Area = \frac{1}{2}\left|\sum(x_i \cdot y_{i+1} - y_i \cdot x_{i+1})\right|$$

ตัวอย่าง



ตัวอย่างนี้พยายามทำขึ้นมาเพื่อให้ได้ค่าที่ใกล้เคียงกับภายในเว็ปมากที่สุดครับ

ตัวอย่าง

กรณียกตัวอย่างในการคำนวณ

(X, Y)

$$P[0] = (0, 0)$$

$$P[1] = (400, 0)$$

$$P[2] = (650, 255)$$

$$P[3] = (400, 510)$$

$$P[4] = (0, 510)$$

นำสูตรมาใช้

$$Area = \frac{1}{2} \left| \sum_{i=0}^{n+1} (x_i \cdot y_{i+1} - y_i \cdot x_{i+1}) \right|$$

n = จำนวนจุดยอด (4)

Xi,Yi =พิกัดของจุดที่ i

Xi+1,Yi+1 คือการคิดพิกัดของจุดถัดไป และวนกลับไป 0 หรือจุดเริ่มต้น

ตัวอย่าง

$x_i \cdot y_{i+1}$

$$x_0y_1 + x_1y_2 + x_2y_3 + x_3y_4 + x_4y_0$$

$$(0 \cdot 0) + (400 \cdot 255) + (650 \cdot 510) + (400 \cdot 510) + (0 \cdot 0)$$

$$0 + 102000 + 331500 + 204000 + 0$$

$$= 637500$$

$y_i \cdot x_{i+1}$

$$y_0x_1 + y_1x_2 + y_2x_3 + y_3x_4 + y_4x_0$$

$$(0 \cdot 400) + (0 \cdot 650) + (255 \cdot 400) + (510 \cdot 0) + (510 \cdot 0)$$

$$0 + 0 + 102000 + 0 + 0$$

$$= 102000$$

ตัวอย่าง

$$\sum(x_i \cdot y_{i+1}) - \sum(y_i \cdot x_{i+1})$$

$$637500 - 102000$$

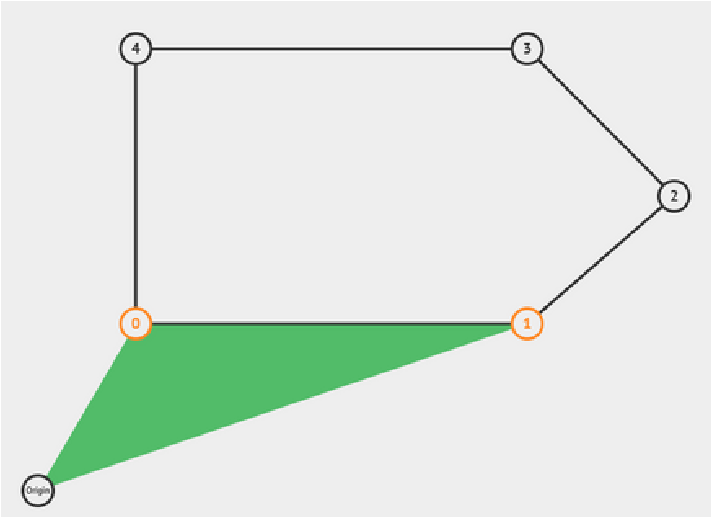
$$= 535500$$

$$Area = \frac{1}{2} \cdot |535500|$$

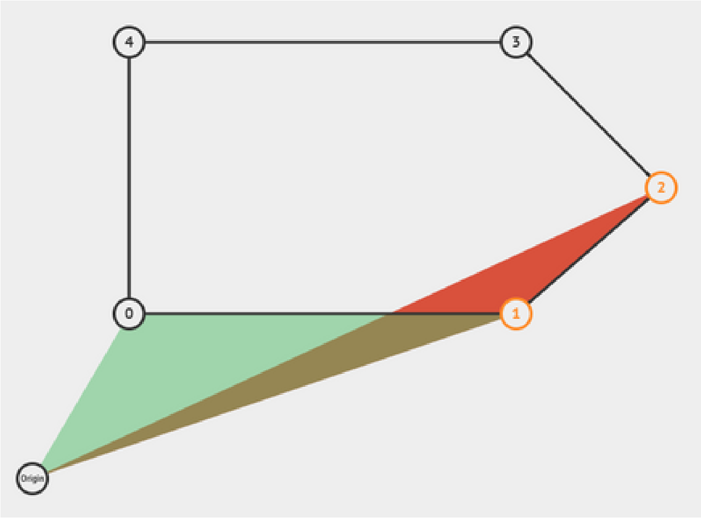
$$Area = 267750px^2$$

และนี่คือตัวอย่างที่ทำการกำหนดจุดขึ้นมาเอง เพื่อนำมาพิสูจน์กับแนวคิด
ของSHOELACE FORMULA (เชือกผูกรองเท้า)

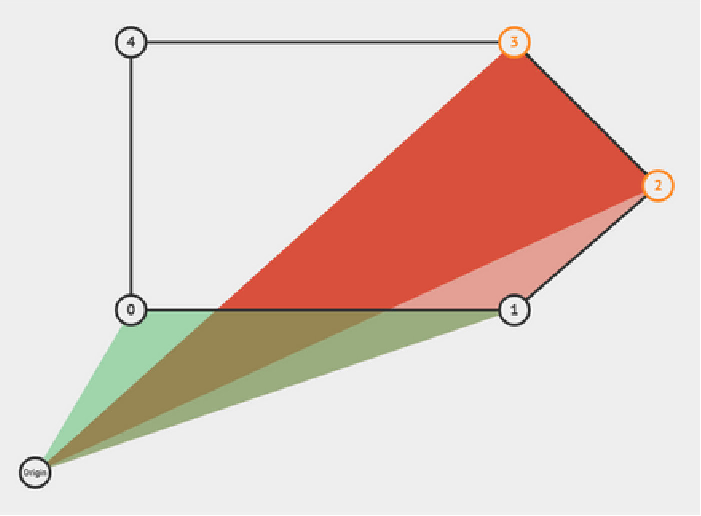
ตัวอย่าง
หลักการทำงานของสูตรกับการเขียนโปรแกรม



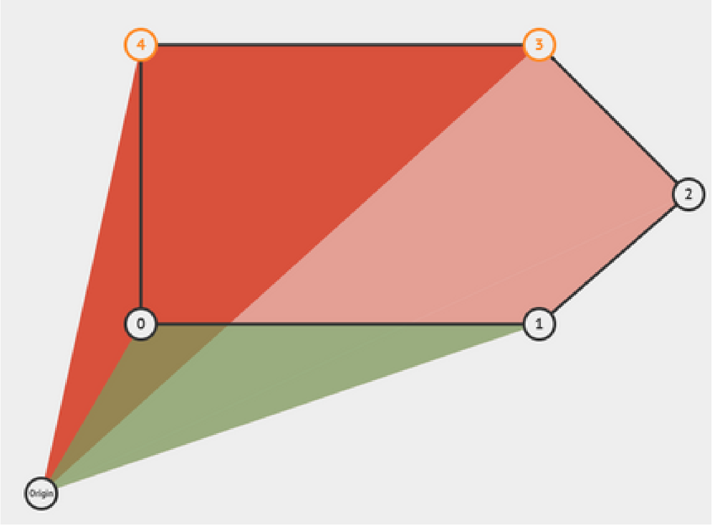
$$\frac{(x_0y_1)-(y_0x_1)}{2}$$



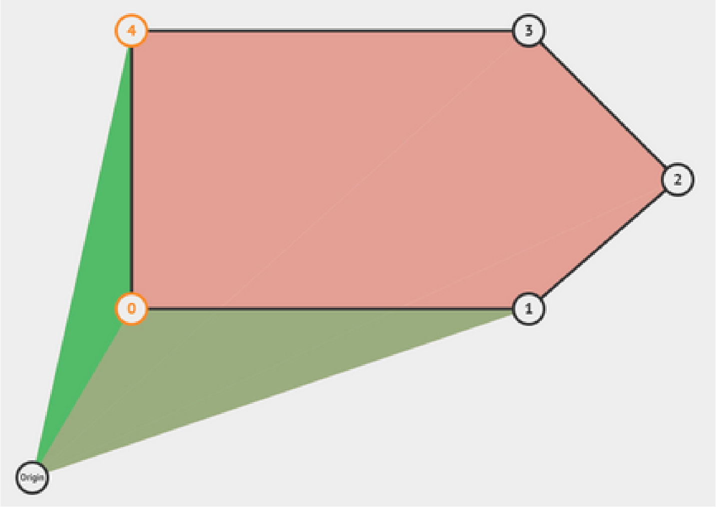
$$\frac{(x_0y_1)-(y_0x_1)}{2}+\frac{(x_1y_2)-(y_1x_2)}{2}$$



$$\frac{(x_0y_1)-(y_0x_1)}{2}+\frac{(x_1y_2)-(y_1x_2)}{2}+\frac{(x_2y_3)-(y_2x_3)}{2}$$

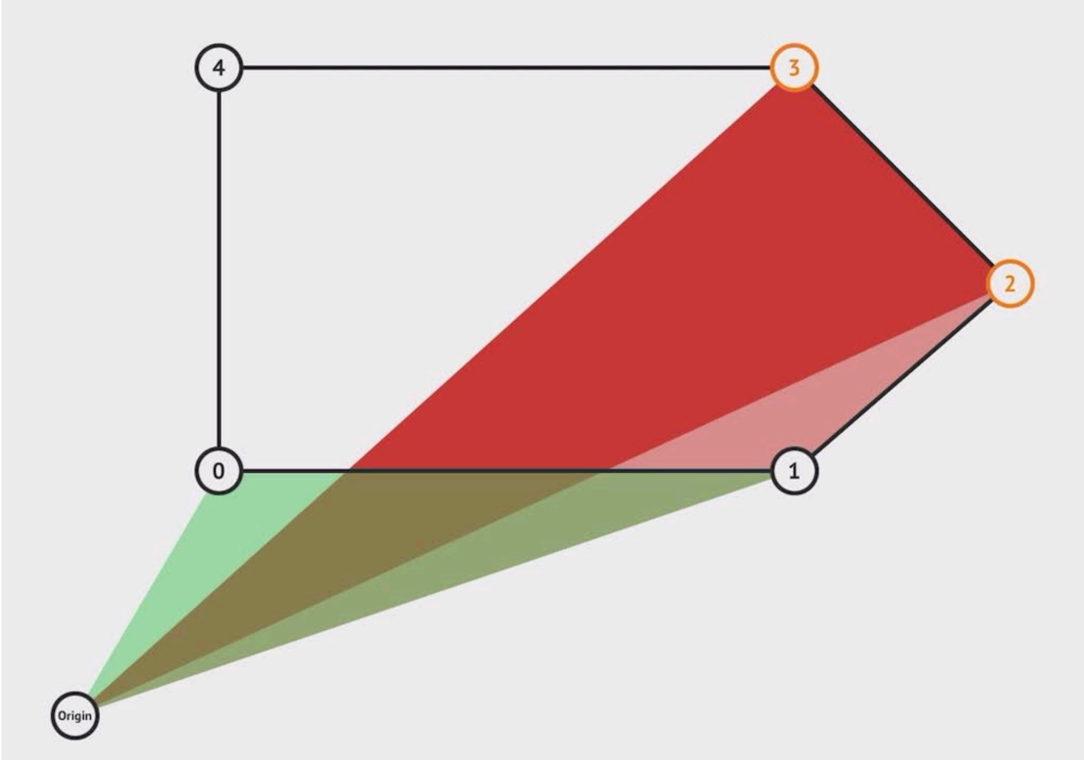


$$\frac{(x_0y_1)-(y_0x_1)}{2}+\frac{(x_1y_2)-(y_1x_2)}{2}+\frac{(x_2y_3)-(y_2x_3)}{2}+\frac{(x_3y_4)-(y_3x_4)}{2}$$



$$\frac{(x_0y_1)-(y_0x_1)}{2}+\frac{(x_1y_2)-(y_1x_2)}{2}+\frac{(x_2y_3)-(y_2x_3)}{2}+\frac{(x_3y_4)-(y_3x_4)}{2}+\frac{(x_4y_0)-(y_4x_0)}{2}$$

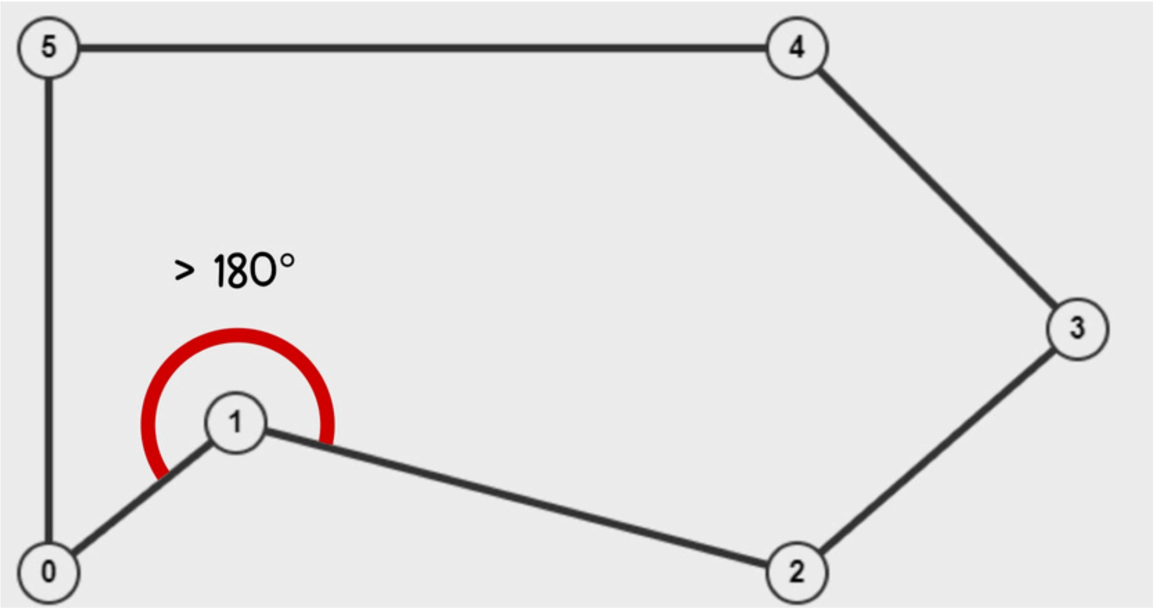
ตัวอย่าง



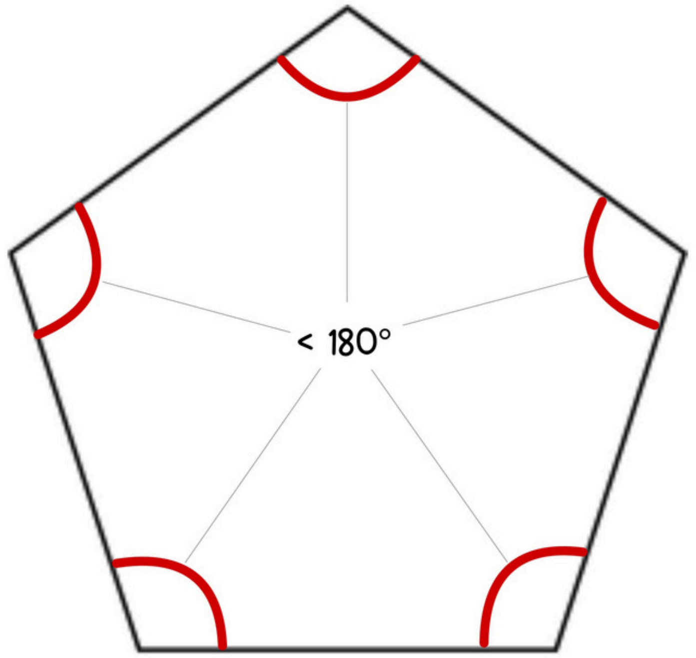
$$\frac{(x_0y_1)-(y_0x_1)}{2}+\frac{(x_1y_2)-(y_1x_2)}{2}+\frac{(x_2y_3)-(y_2x_3)}{2}+\frac{(x_3y_4)-(y_3x_4)}{2}+\frac{(x_4y_0)-(y_4x_0)}{2}=Area$$

ISCONVEX (P) (รูปหลายเหลี่ยมนูน)

- มุมภายในทุกมุมมีขนาดน้อยกว่า 180 องศา
- ส่วนของเส้นตรงทุกเส้นที่เชื่อมระหว่างจุดยอดสองจุดใด ๆ จะวางตัวอยู่ภายในขอบเขตของรูปหลายเหลี่ยมเสมอ สะท้อนถึงความเป็นรูปทรงปิดที่สมบูรณ์และไม่มีการตัดกันของเส้นขอบ



NOT CONVEX

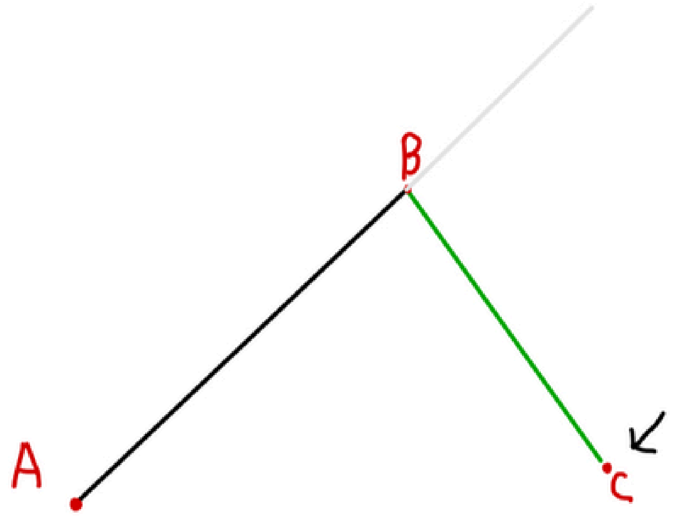


CONVEX

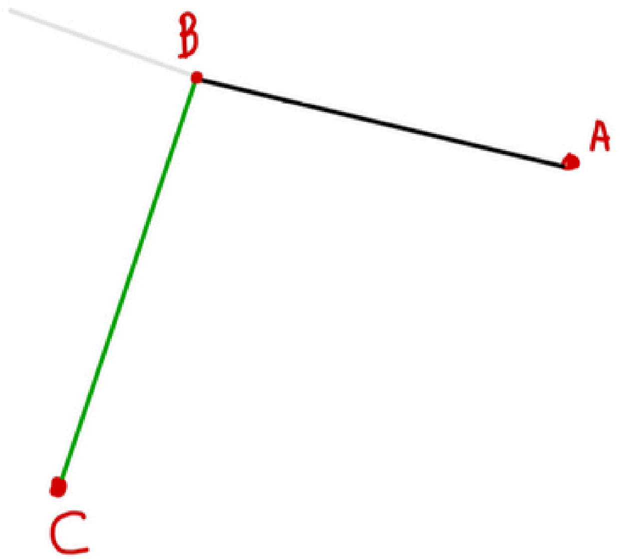
FUNCTION CCW

CCW !=

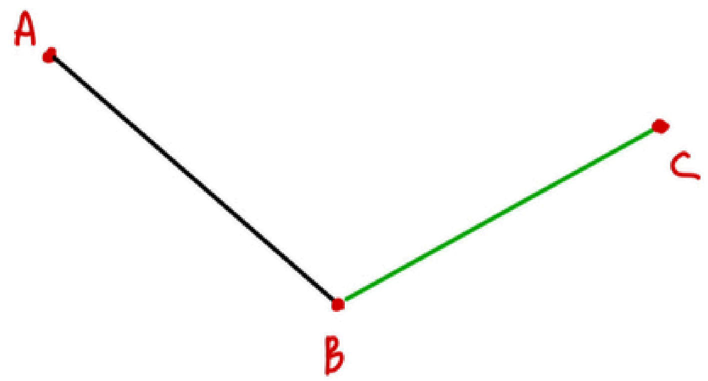
FUNCTION CCW เป็นฟังก์ชั่นตรวจสอบว่าจุด C อยู่ทาง ซ้ายมือ หรือ ขวามือ ของเส้นตรง AB



C = ขวา



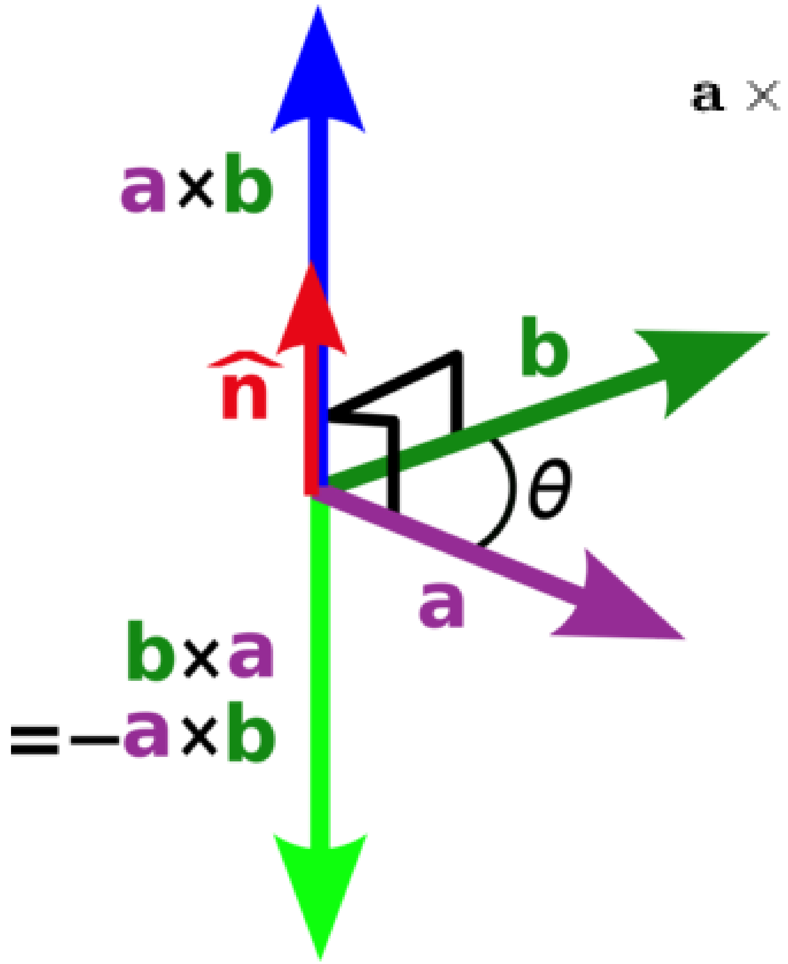
C = ซ้าย



C = ซ้าย

CROSS PRODUCT

เมื่อ θ คือขนาดของ มุม (ที่ไม่ใช่มุมป้าน) ระหว่าง A กับ B ($0^\circ \leqslant \theta \leqslant 180^\circ$) A กับ B ในสูตรคือ ขนาด ของเวกเตอร์ A และ B ตามลำดับ และ N^ คือ เวกเตอร์หน่วย ที่ตั้งฉากกับเวกเตอร์ A และ B ถ้าหากทั้งสองเวกเตอร์นั้นร่วมเส้นตรงกัน (คือมีมุมระหว่างเวกเตอร์เป็น 0° หรือ 180°) ผลคูณไขว้จะได้ผลลัพธ์เป็น เวกเตอร์ศูนย์ 0

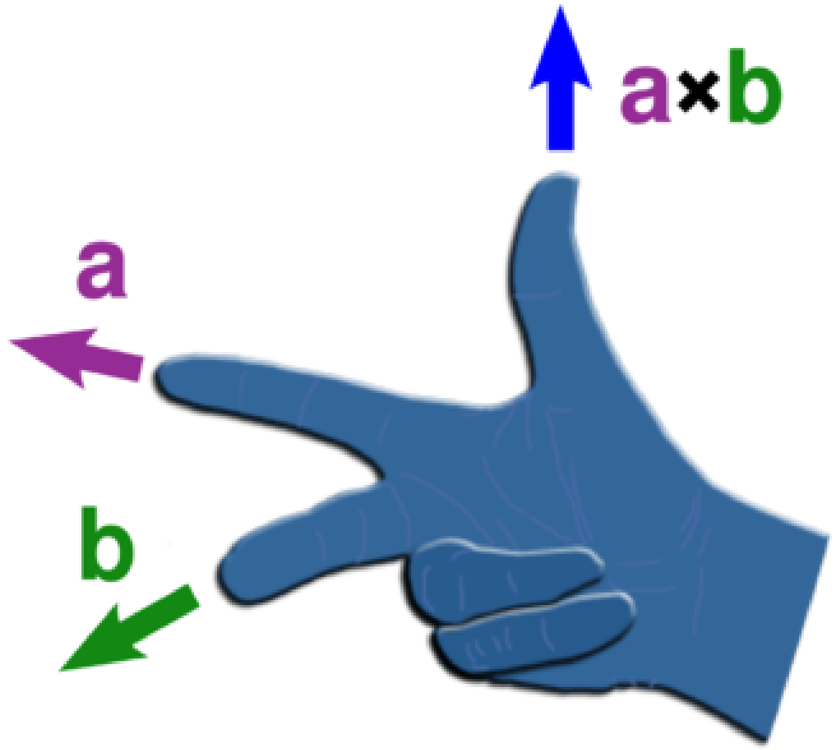


$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \mathbf{i}(a_2b_3) + \mathbf{j}(a_3b_1) + \mathbf{k}(a_1b_2) - \mathbf{i}(a_3b_2) - \mathbf{j}(a_1b_3) - \mathbf{k}(a_2b_1)$$

A = (0, 0) B = (4, 0) C = (2, 2)

Z = (B[0] − A[0]) * (C[1] − A[1]) − (B[1] − A[1]) * (C[0] − A[0])

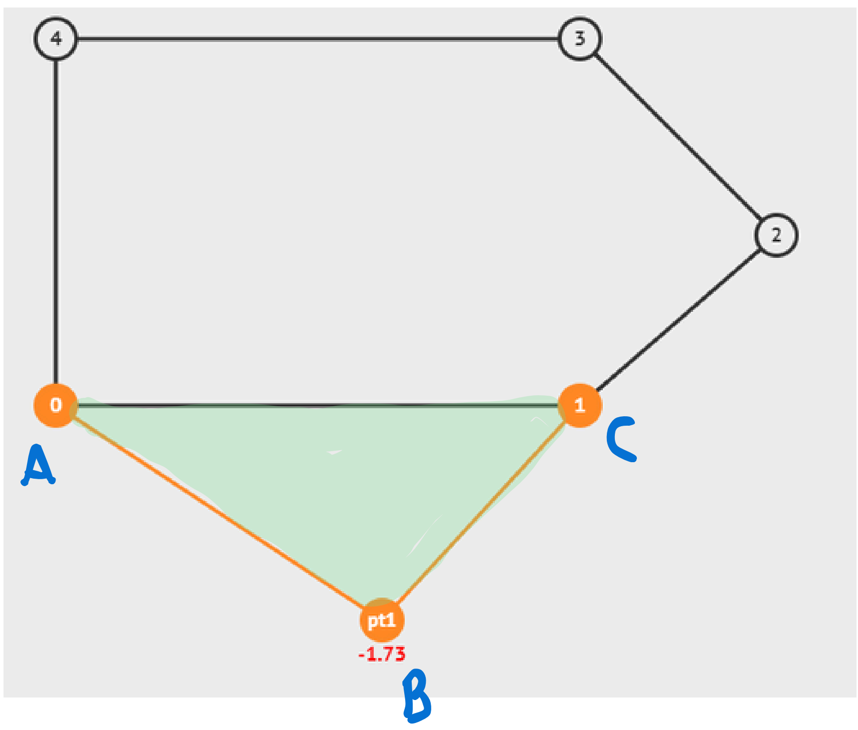
- AB = (4−0,0−0)=(4,0)
- AC = (2−0,2−0)=(2,2)
- Z=(4 · 2)−(0 · 2)=8



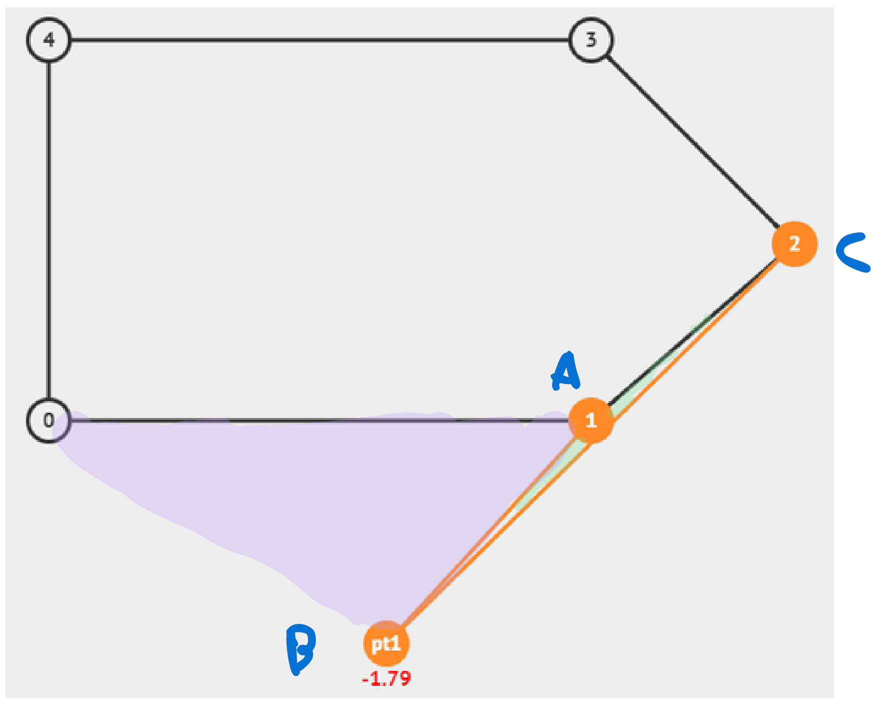
+1: จุด C อยู่ซ้ายของเส้น , −1: จุด C อยู่ขวาของเส้น, 0: จุด C อยู่บนเส้นเดียวกับเส้นตรง

ISPOLYGON (P,PT,P+1)

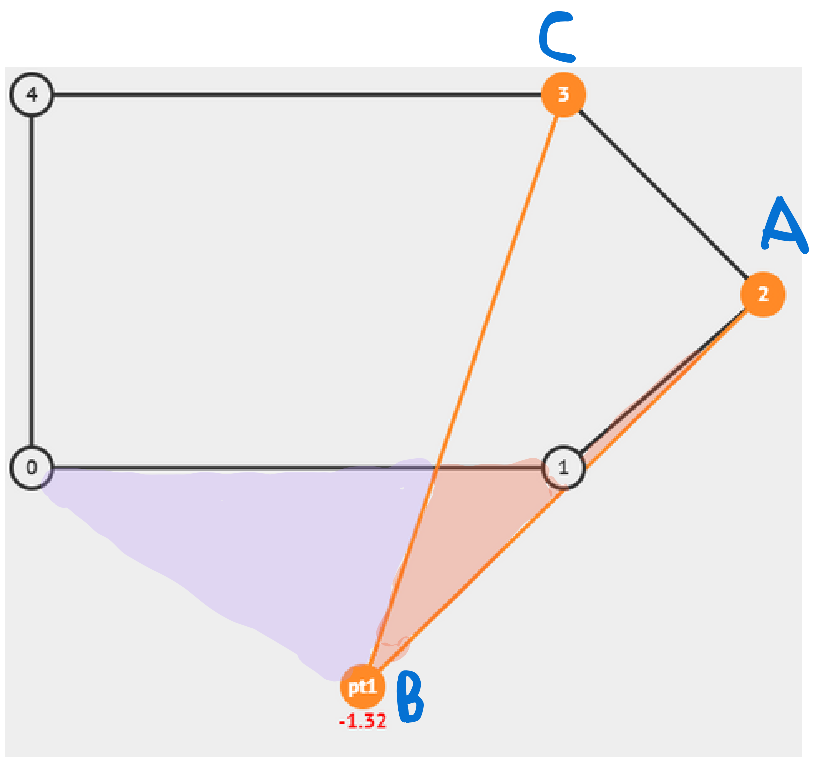
1.



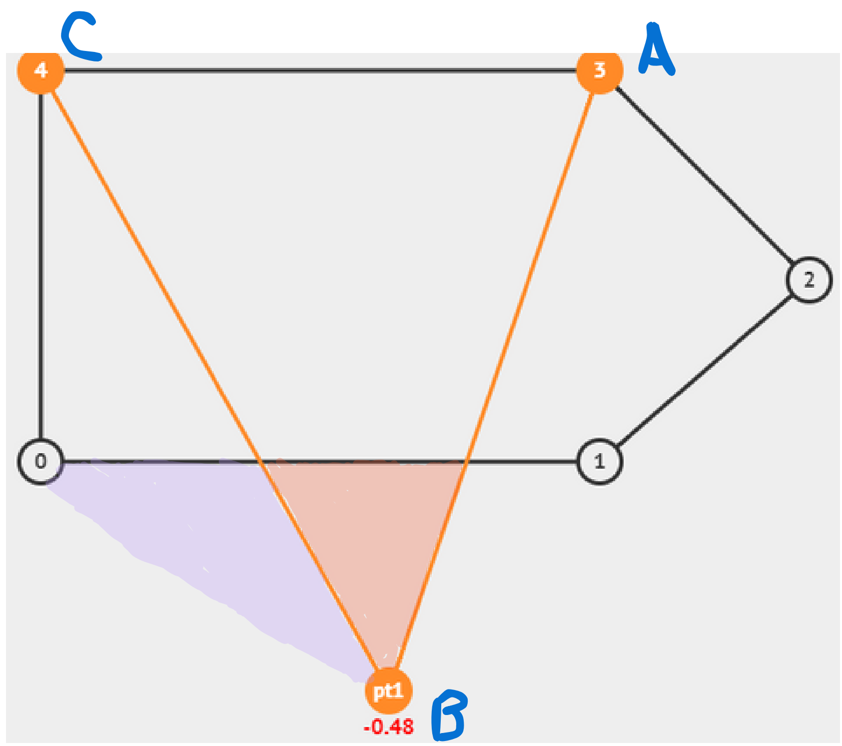
2.



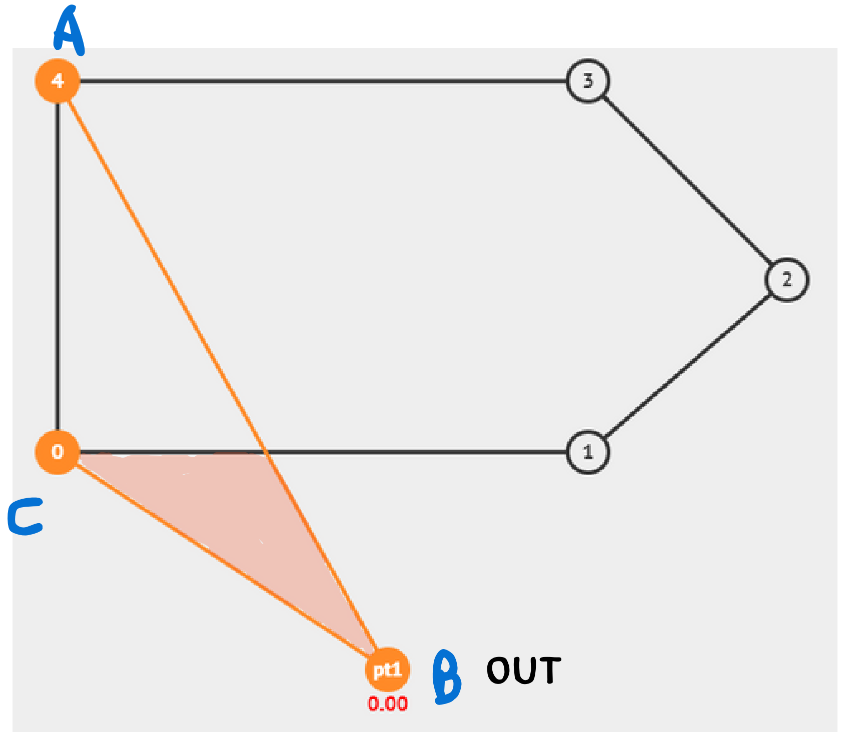
3.



4.

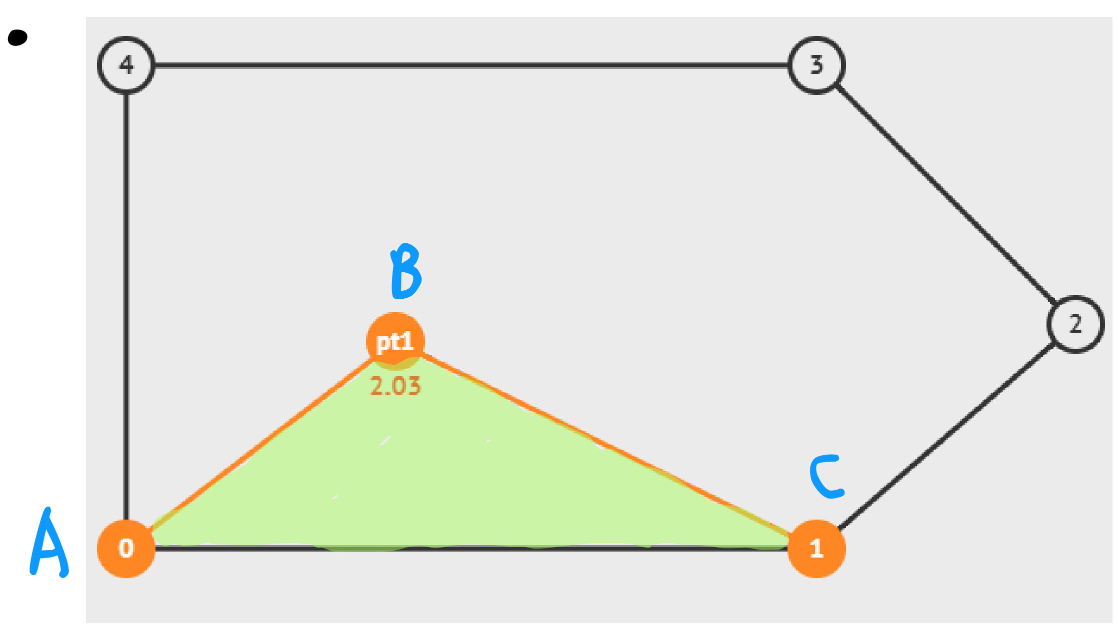


5.

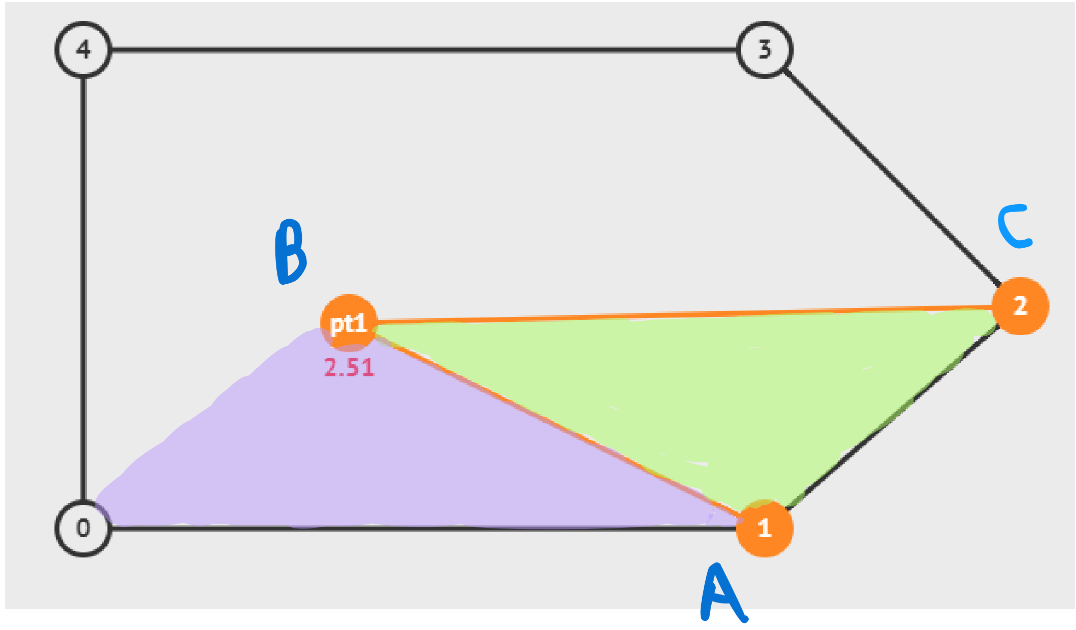


ISPOLYGON ((P,PT,P+1))

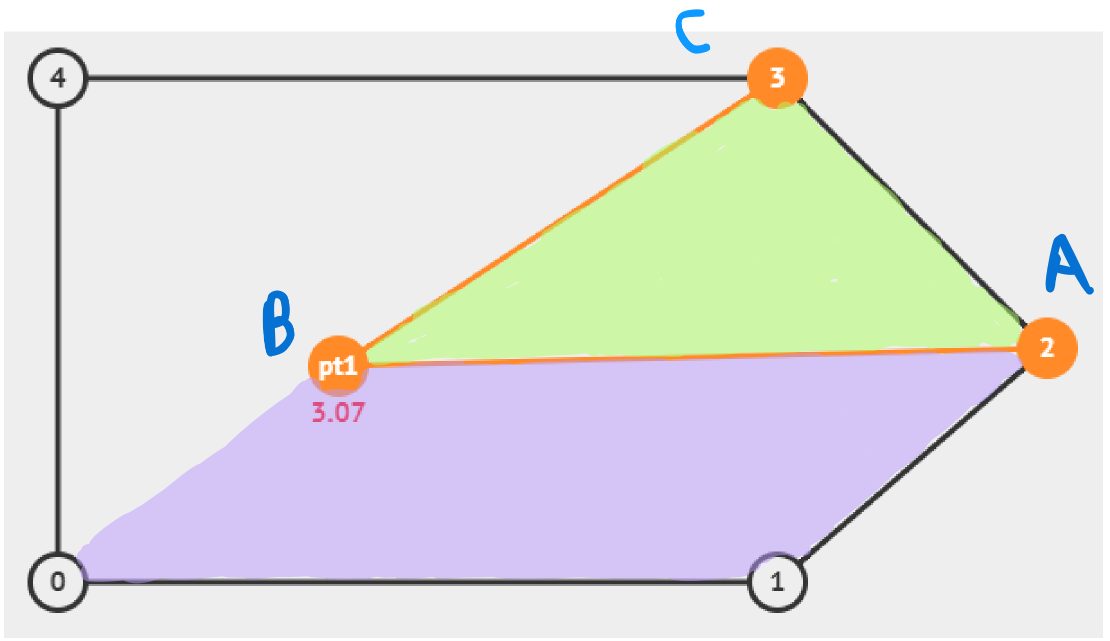
1.



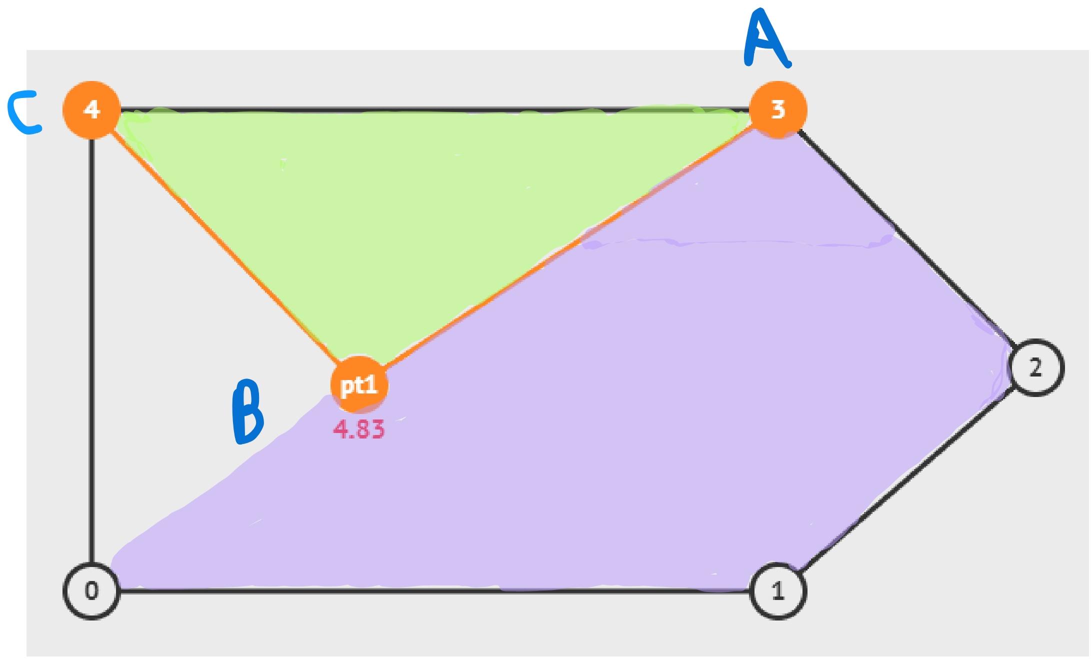
2.



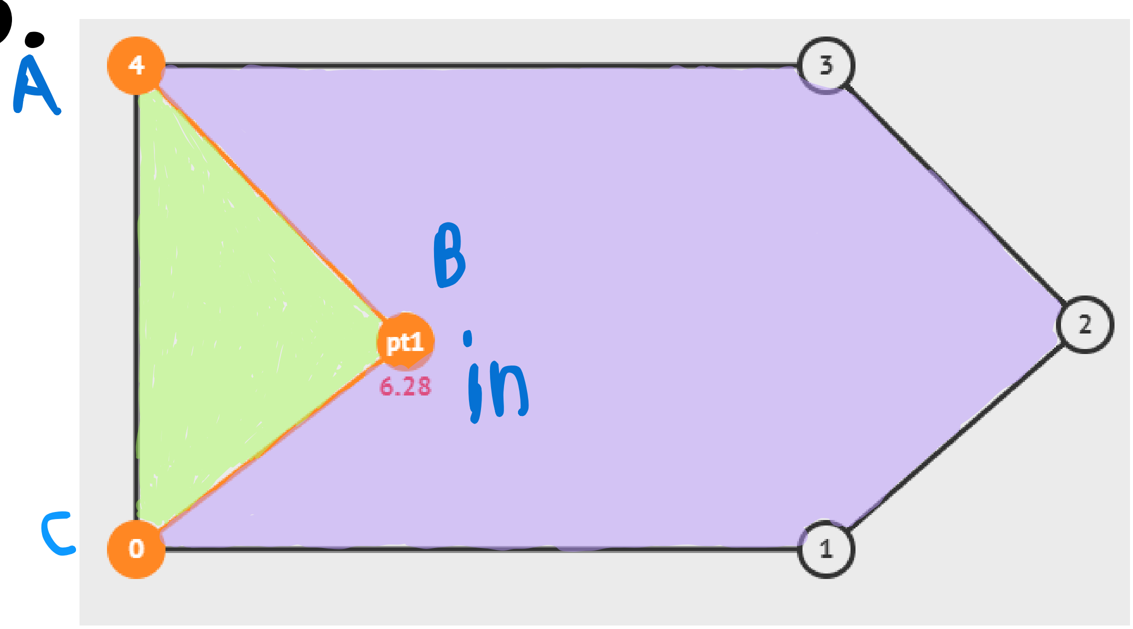
3.



4.

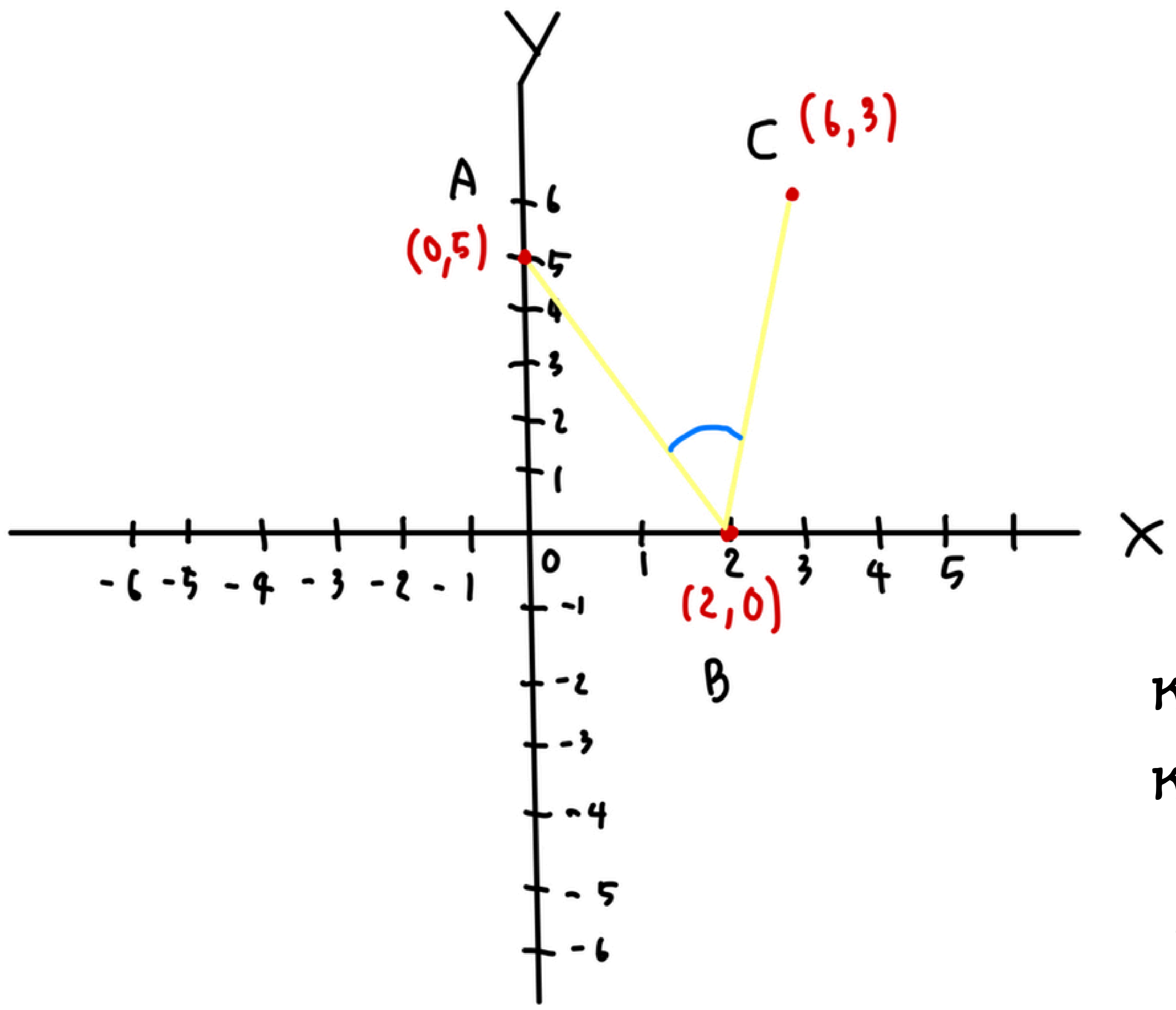


5.



FUNCTION ANGLE

- DOT PRODUCT และ NORM เพื่อคำนวณมุมระหว่างเวกเตอร์
- คืนค่ามุมระหว่างเวกเตอร์ในรูป เรเดียน



- A=(0,5)
- B=(2,0)
- C=(6,3)

หาเวกเตอร์ BA (AX − BX, AY − BY) = 0−2, 5−0 = −2,5

หาเวกเตอร์ BC (CX− BX, CY − BY) = 6−2, 3−0 = 4,3

DOT PRODUCT = (BAX•BCX) + (BAY•BAY) = (−2•4) + (5•3) = (−8)+15 = 7

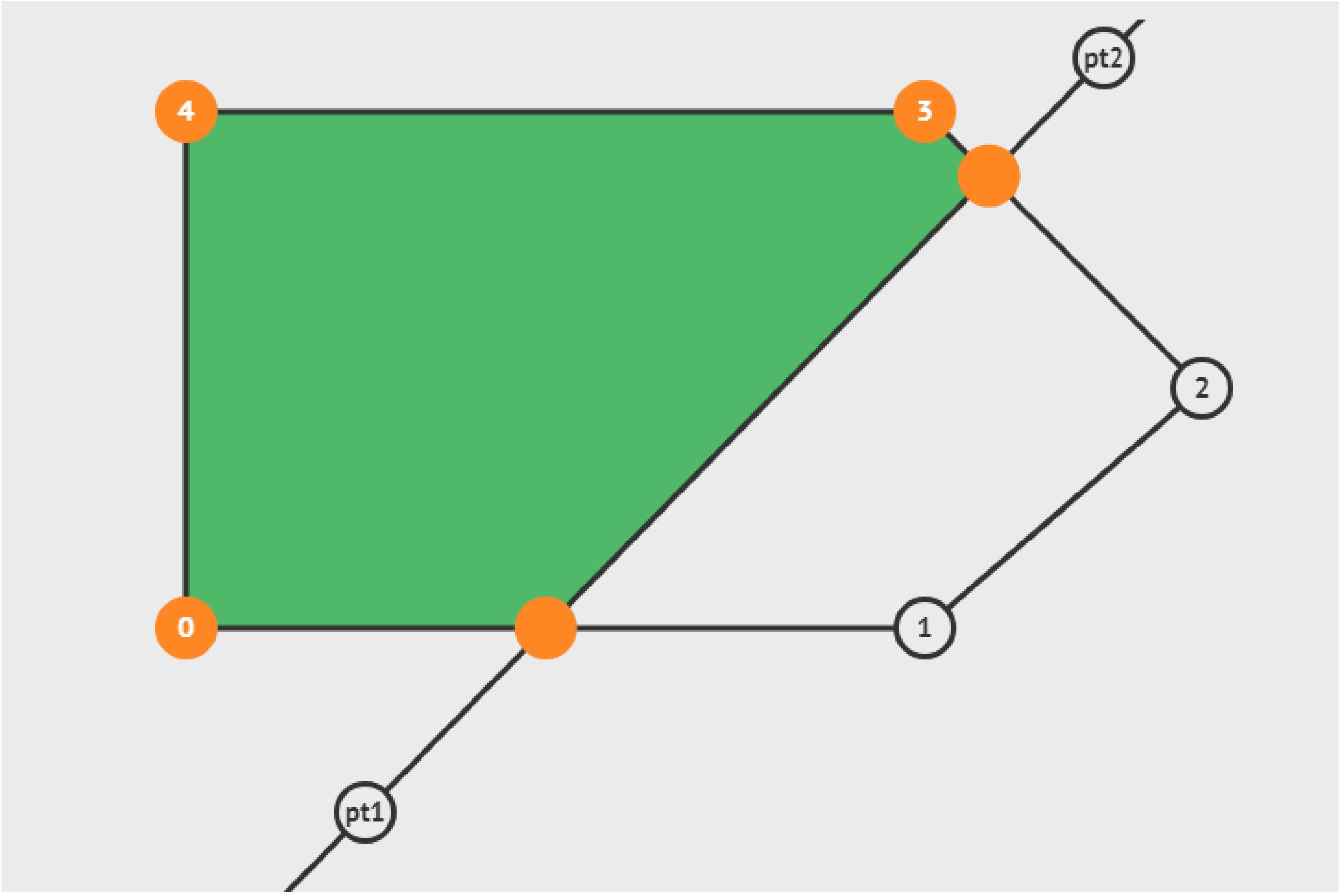
หาขนาดเวกเตอร์ $|BA| = \sqrt{(BAX)^2 + (BAY)^2} = \sqrt{-(2)^2+(5)^2} = \sqrt{4+25} = \sqrt{29}$

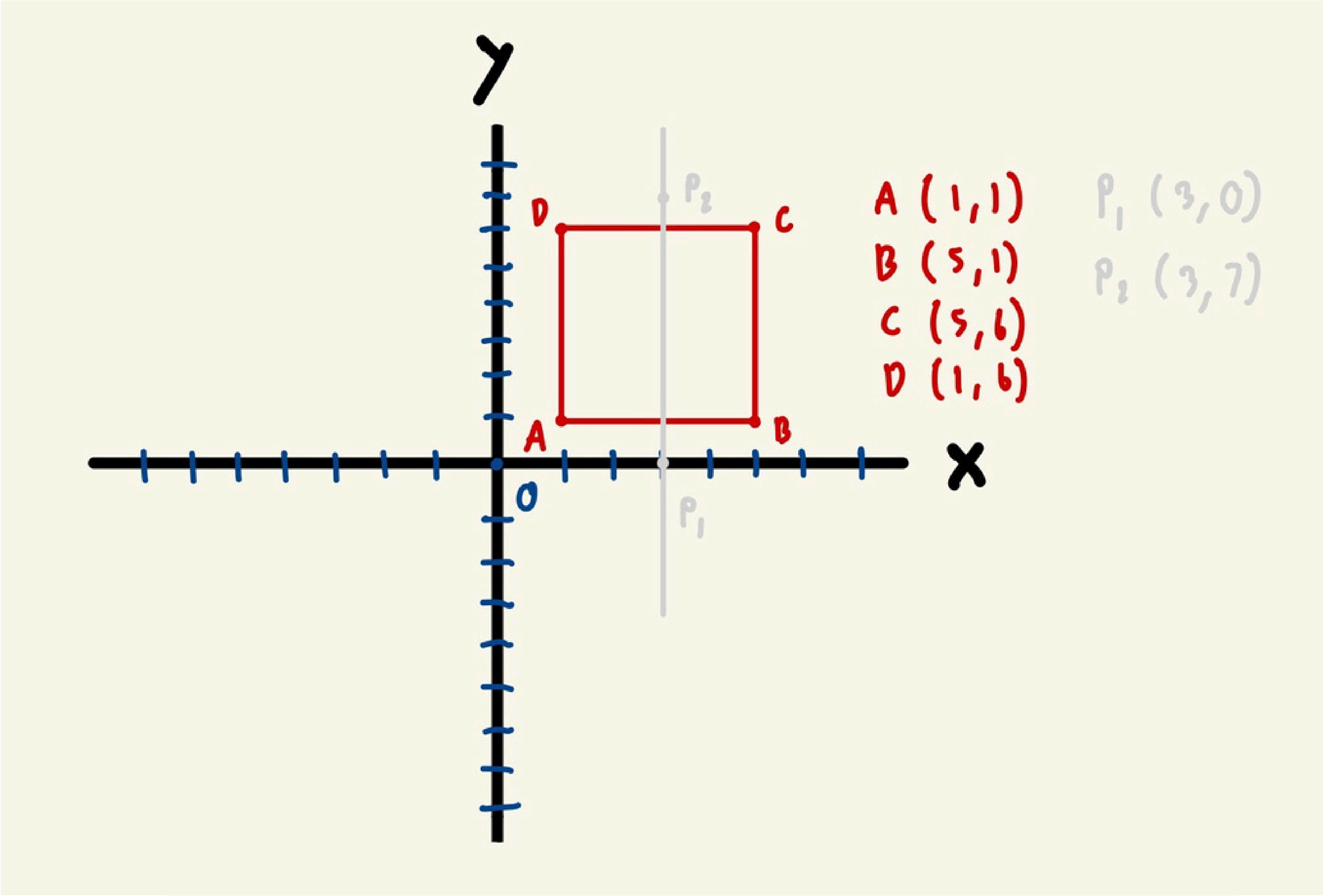
หาขนาดเวกเตอร์ $|BC| = \sqrt{(BCX)^2 + (BCY)^2} = \sqrt{-(4)^2+(3)^2} = \sqrt{16+9} = \sqrt{25}$

$$\cos(\theta) = \frac{\text{DOT PRODUCT}}{|BA|*|BC|} = \frac{7}{5\sqrt{29}} = \cos^{-1}(0.2602) \approx 74.93^\circ$$

CUTPOLYGON (LN,P)

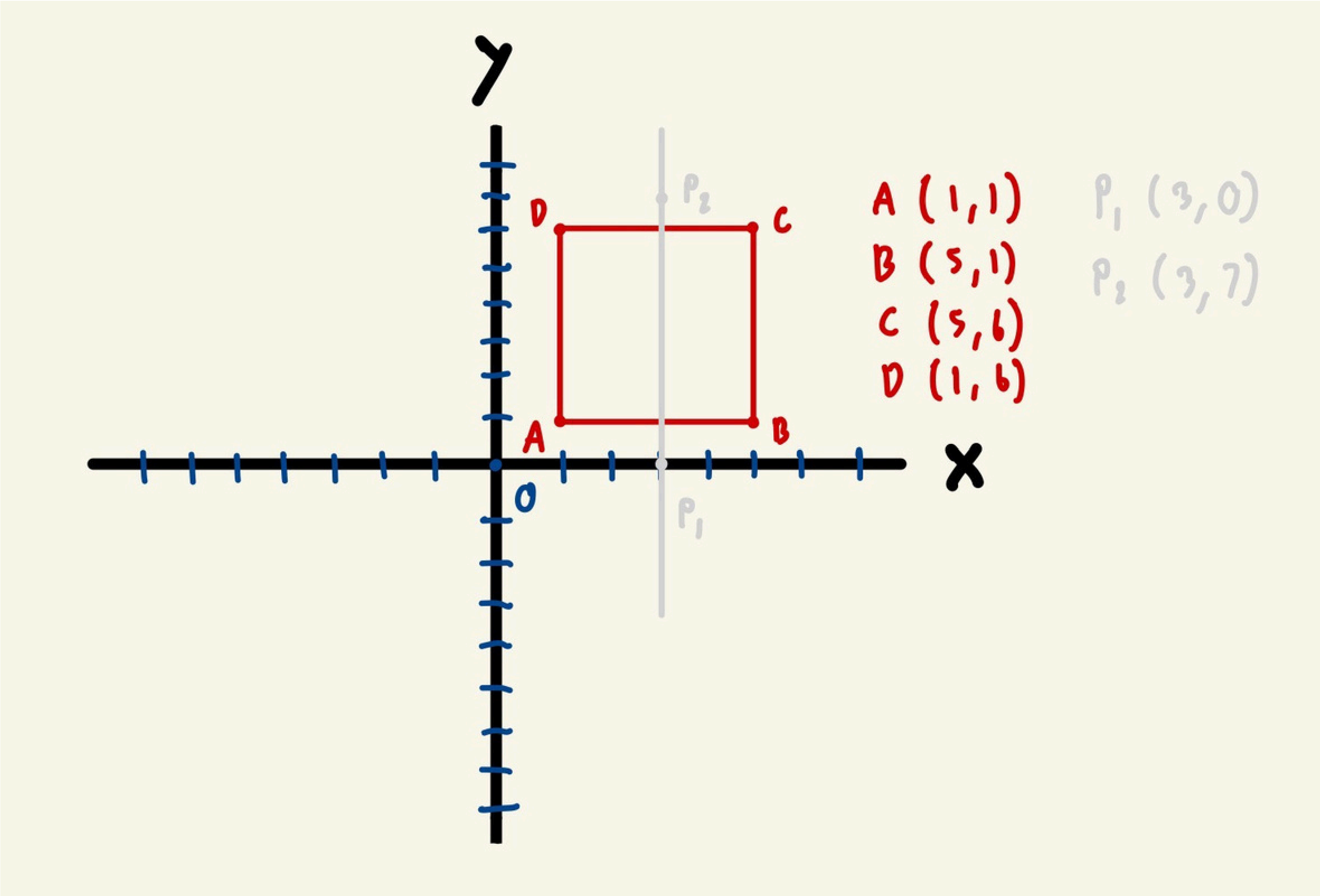
- อัลกอริทึมนี้ใช้สำหรับการตัดรูปหลายเหลี่ยมด้วยเส้นตรง จากจุดที่เรากำหนด ซึ่งจะได้ผลลัพธ์เป็นรูปหลายเหลี่ยมใหม่





SOLUTION

$$\text{Cross}(A, B, P) = (B_x - A_x)(P_y - A_y) - (B_y - A_y)(P_x - A_x)$$



- ใช้ CROSS PRODUCT ในการตรวจสอบจุดแต่ละจุดในรูปนี้ว่าจุดไหนอยู่ทางซ้าย–ขวา หรือตรงเส้นตัด

A (1,1) :

$$\text{Cross}(P_1, P_2, A) = (3-3)(1-0) - (7-0)(1-3) = 14$$

B (5,1) : -14

C (5,6) : -14

D (1,6) : 14

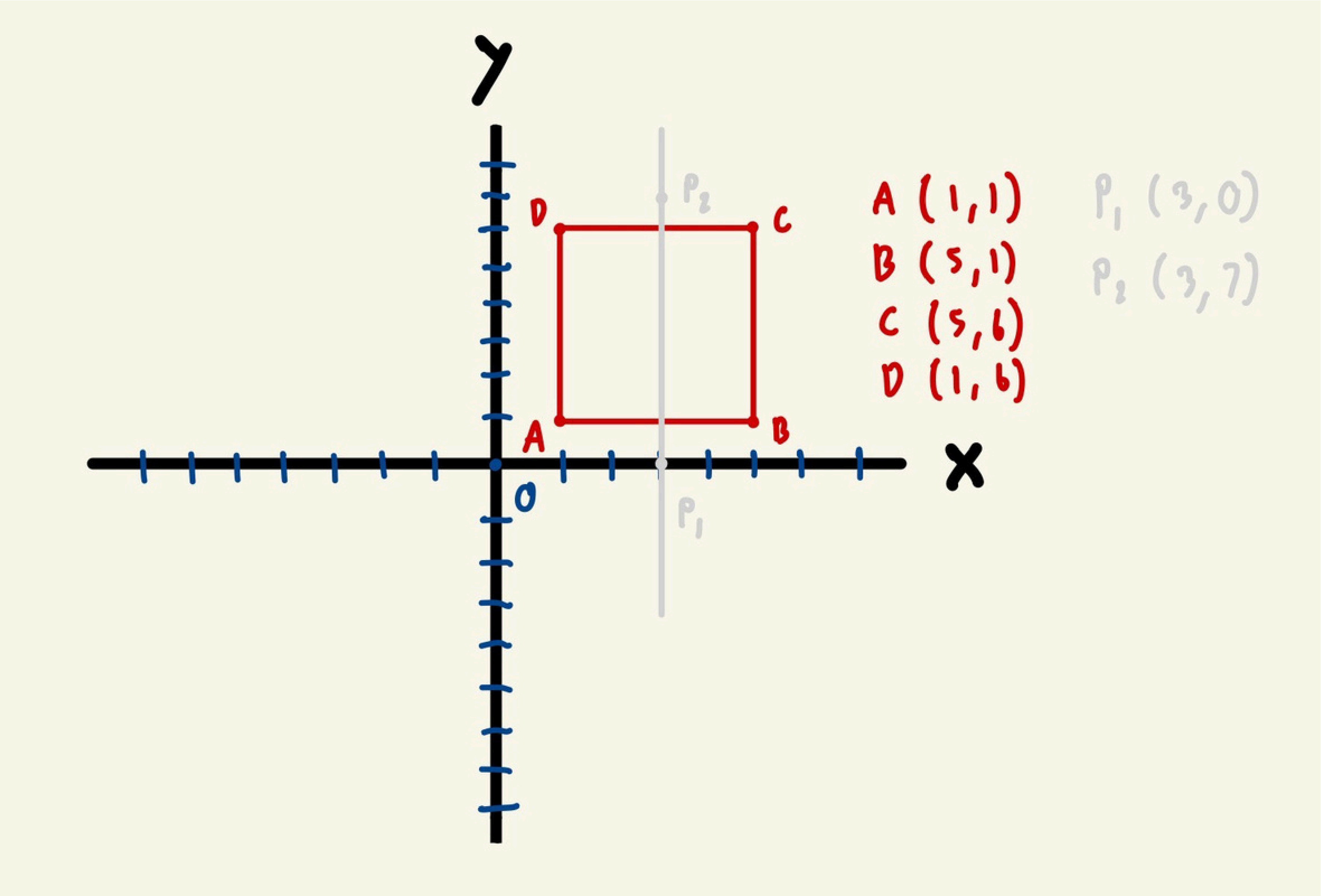
ถ้าค่าที่ได้ออกมาเป็น

− : RIGHT

+ : LEFT

SOLUTION

- ใช้หลักการ LINEAR INTERPOLATION ในการคำนวณหาจุดระหว่างเส้นตัดและรูปสี่เหลี่ยม



A(1,1) : 14 LEFT

B(5,1) : −14 RIGHT

C(5,6) : −14 RIGHT

D(1,6) : 14 LEFT

คู่ที่ 1 A,B = L,R

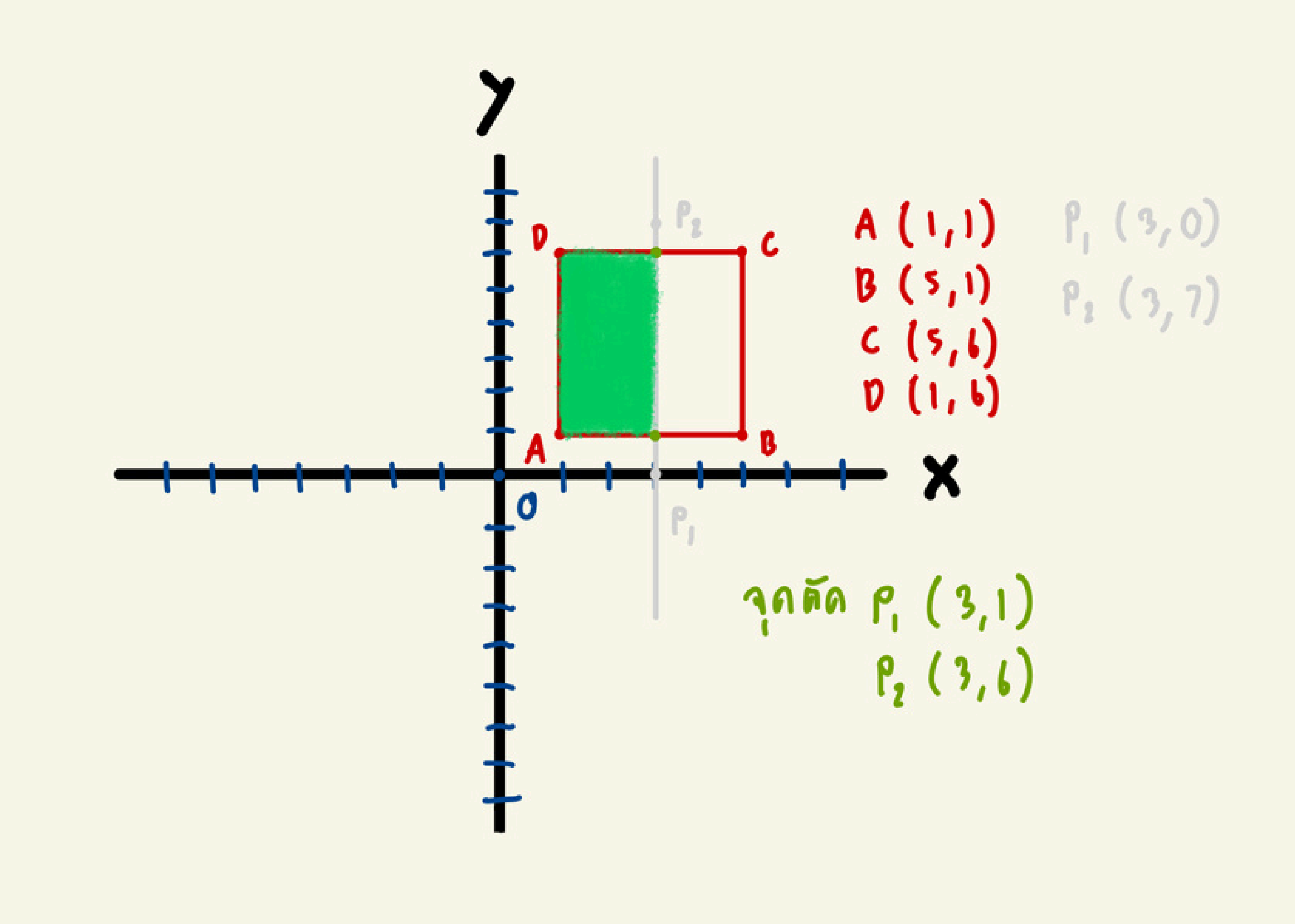
คู่ที่ 2 B,C = R,R

คู่ที่ 3 C,D = R,L

คู่ที่ 4 D,A = L,L

SOLUTION

- การคำนวณหาจุดตัดโดยใช้สมการ LINEAR INTERPOLATION



$$\boxed{P = A + t(B - A)}$$

$$P = (1,1) + t\big((5,1) - (1,1)\big) \longrightarrow t = \frac{\mathrm{cross}(P_1, P_2, A)}{\mathrm{cross}(P_1, P_2, A) - \mathrm{cross}(P_1, P_2, B)}$$

$$= \frac{14}{14 - (-14)}$$

$$= \frac{14}{28} = 0.5$$

$$P = (1,1) + 0.5\big((5,1) - (1,1)\big)$$

$$P = (1,1) + 0.5(4,0)$$

$$P = (1,1) + (2,0)$$

$$P = (3,1)$$

$\therefore$ จุดตัดระหว่าง A, B คือ (3,1)

ค่าที่ออกมาจากโปรแกรม คือ [[1, 1], [3.0, 1.0], [3.0, 6.0], [1, 6]]

จุดตัด A, B : (3,1)
C, D : (3,6)

สรุป

COMPUTATIONAL GEOMETRY (POLYGON)

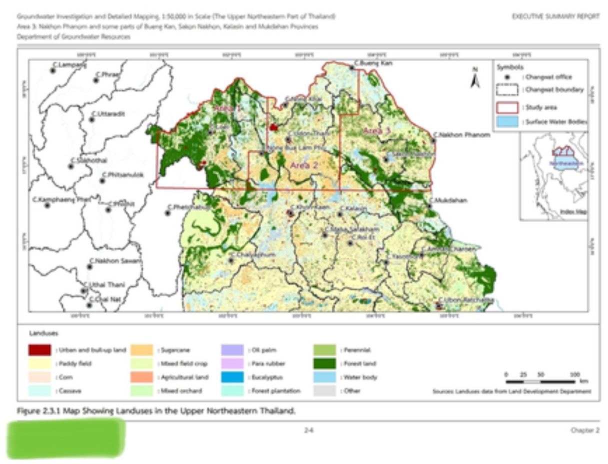
รูปหลายเหลี่ยมสามารถใช้ในงานการคำนวณเชิงเรขาคณิต เช่น คำนวณเส้นรอบรูป พื้นที่ ตรวจสอบความนูน–เว้า และตัดรูปหลายเหลี่ยม อัลกอริทึมที่ใช้งานมีประสิทธิภาพสูง (O(N)) เช่น SHOELACE FORMULA, WINDING NUMBER ALGORITHM และการตัดด้วยเส้นตรง เหมาะสำหรับการประยุกต์ใช้งาน เช่น การเขียนโปรแกรมคอมพิวเตอร์ (UVA, KATTIS) และการแก้ปัญหาเรขาคณิตขั้นสูง

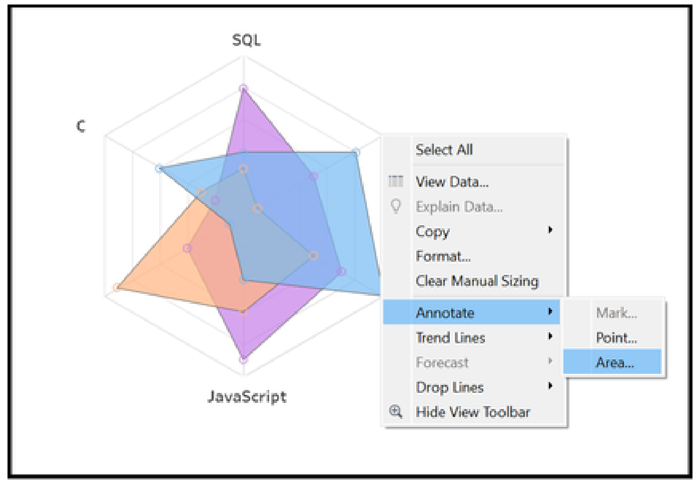
การทำงาน	Best Case	Worst Case	คำอธิบาย
คำนวณเส้นรอบรูป (Perimeter)	$O(n)$	$O(n)$	ต้องคำนวณความยาวเส้นรอบรูปทุกกรณีโดยวนผ่านจุด n จุด
คำนวณพื้นที่ (Area)	$O(n)$	$O(n)$	ใช้ Shoelace Formula ซึ่งต้องวนลูปผ่านจุด n จุดทุกกรณี
ตรวจสอบจุดใน Polygon (inPolygon)	$O(n)$	$O(n)$	ต้องตรวจสอบเส้นขอบ n เส้นเสมอใน Ray-Casting Algorithm
ตรวจสอบว่าเป็น Convex หรือไม่ (isConvex)	$O(1)$	$O(n)$	Best Case $O(1)$: ตรวจพบว่าไม่เป็น Convex ทันที หรือ $n = 3$; Worst Case ต้องตรวจสอบทุกจุด
การตัด Polygon (cutPolygon)	$O(n)$	$O(n)$	ต้องตรวจสอบจุดและคำนวณจุดตัดสำหรับทุกเส้นขอบของ Polygon

สรุป

การนำไปใช้ในชีวิตจริงของ POLYGON

อัลกอริทึมบนรูปหลายเหลี่ยมมีการนำไปใช้จริงในหลายอุตสาหกรรม ตั้งแต่การทำแผนที่ การออกแบบวิศวกรรม การพัฒนาเกมไปจนถึงการวางแผนทรัพยากรและการเดินทาง โดยมีบทบาทสำคัญในงานที่ต้องการการคำนวณเชิงเรขาคณิตและความแม่นยำสูง





การเก็บข้อมูลCOMPUTATIONAL GEOMETRY (POLYGON)

- LIST / ARRAY: ใช้ในงานพื้นฐาน เช่น การวาดกราฟหรือคำนวณพื้นที่
- LINKED LIST: ใช้ในงานที่ต้องการแทรกหรือปรับจุดใน POLYGON บ่อยครั้ง
- PANDAS DATAFRAME: ใช้ในงานวิเคราะห์ข้อมูล
- GEOJSON: ใช้ในงาน GIS หรือการเก็บข้อมูล POLYGON ที่ต้องการ METADATA

สรุป

เป้าหมายหลักของ ALGORITHM

Algorithm	เป้าหมาย
`perimeter`	คำนวณความยาวเส้นรอบรูปของ Polygon
`area`	คำนวณพื้นที่ของ Polygon
`isConvex`	ตรวจสอบว่า Polygon เป็น Convex หรือไม่
`inPolygon`	ตรวจสอบว่าจุดอยู่ภายในหรือภายนอก Polygon
`cutPolygon`	ตัด Polygon ด้วยเส้นตรงและสร้างรูปหลายเหลี่ยมใหม่ที่เหลือหลังการตัด

Thank You