



NHÓM 2

THÁNG 11, 2024

ỨNG DỤNG HÌNH HỌC

Vinh Khánh - Thiện Nhân

BẢNG NỘI DUNG

01

Khái quát

02

Biểu diễn cơ bản

03

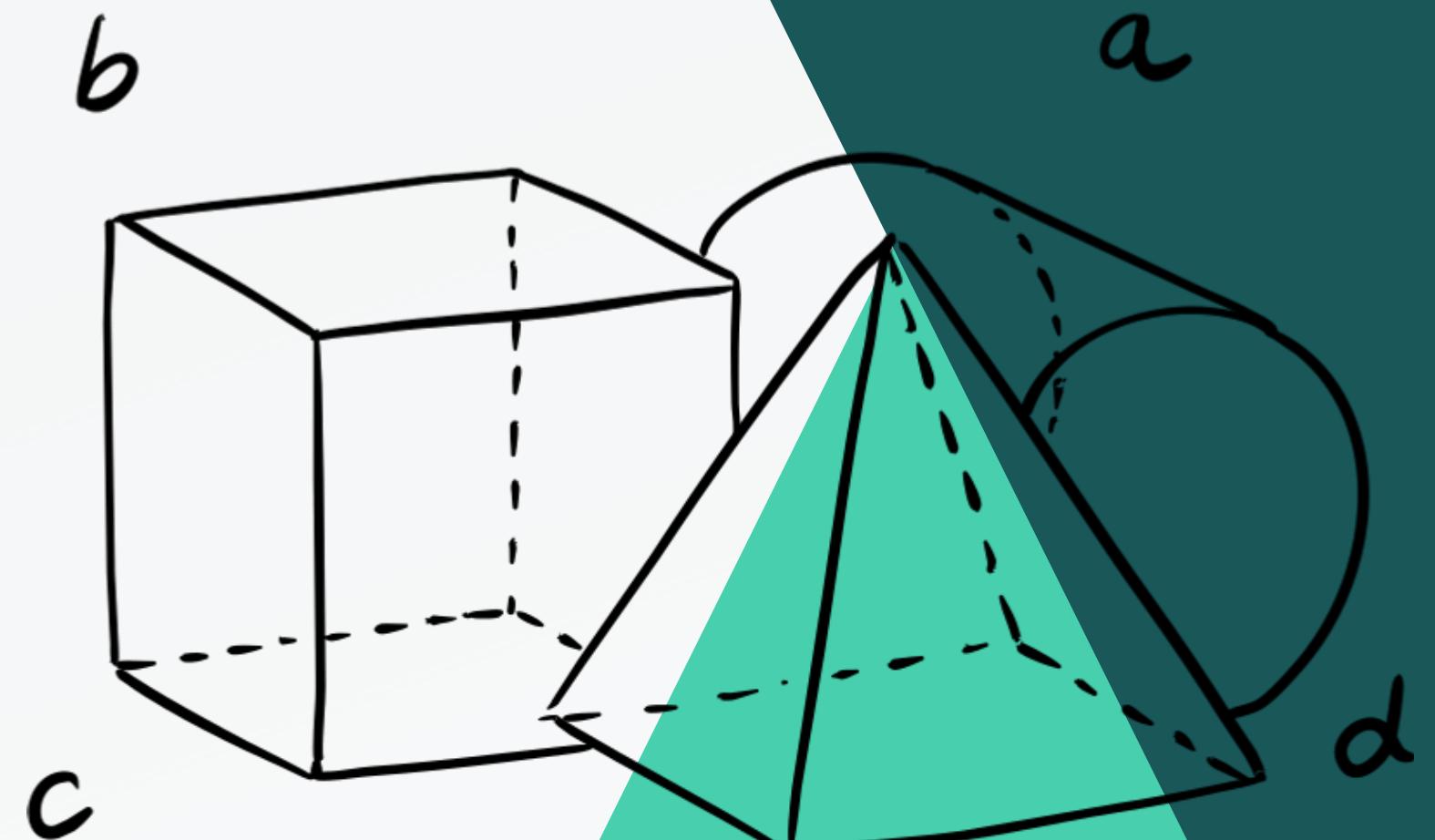
Một số công thức

04

Giới thiệu thuật toán bao lồi

GIỚI THIỆU

Hãy nêu ra một số bài toán
hoặc ứng dụng của hình
học mà bạn biết ?



ỨNG DỤNG

Thiết kế VLSI

Thị giác máy tính

Mô hình toán học

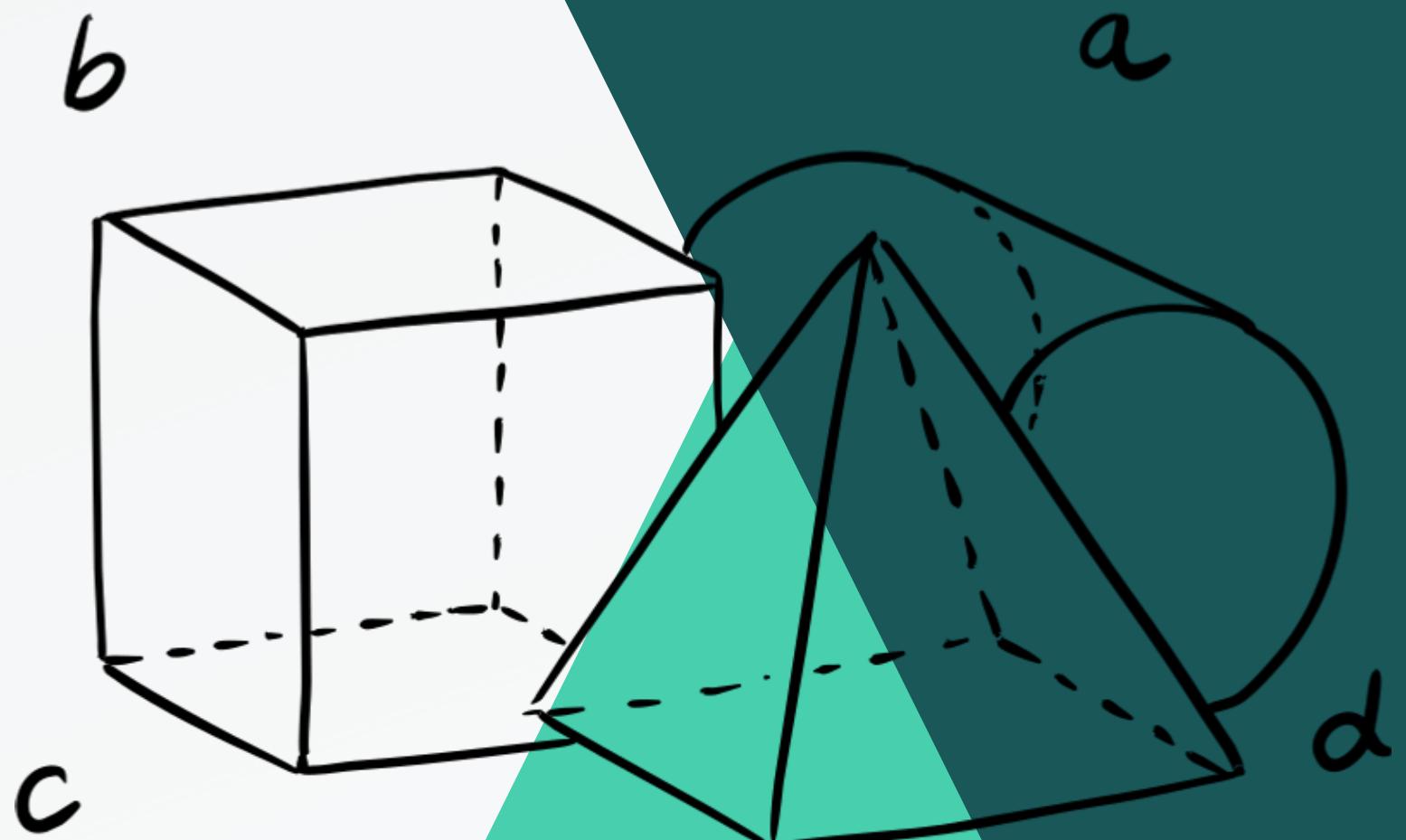
Mô phỏng thiên văn

Hệ thống thông tin địa lý

Trò chơi điện tử

THẢO LUẬN

Các bạn đã học những đối
tượng hình học nào ở cấp 3?



BIỂU DIỄN CƠ BẢN



ĐIỂM TRONG KHÔNG GIAN 2 CHIỀU

```
# Struct Lưu điểm trong mặt phẳng 2D
class Point:

    def __init__(self, x, y):
        self.x = x
        self.y = y
```

BIỂU DIỄN CƠ BẢN



ĐƯỜNG THẲNG: $AX + BY + C = 0$

Struct đường thẳng dạng $ax + by + c = 0$

```
class Line:  
    def __init__(self, a=0, b=0, c=0):  
        self.a = a  
        self.b = b  
        self.c = c
```

Khởi tạo đường thẳng đi qua hai điểm

```
@classmethod  
def from_points(cls, p1, p2):  
    a = p2.y - p1.y  
    b = p1.x - p2.x  
    c = -(a * p1.x + b * p1.y)  
    return cls(a, b, c)
```

BIỂU DIỄN CƠ BẢN



VECTOR

```
# Struct vector
class Vector:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

# Khởi tạo vector từ hai điểm
@classmethod
def from_points(cls, p1, p2):
    x = p2.x - p1.x
    y = p2.y - p1.y
    return cls(x, y)
```

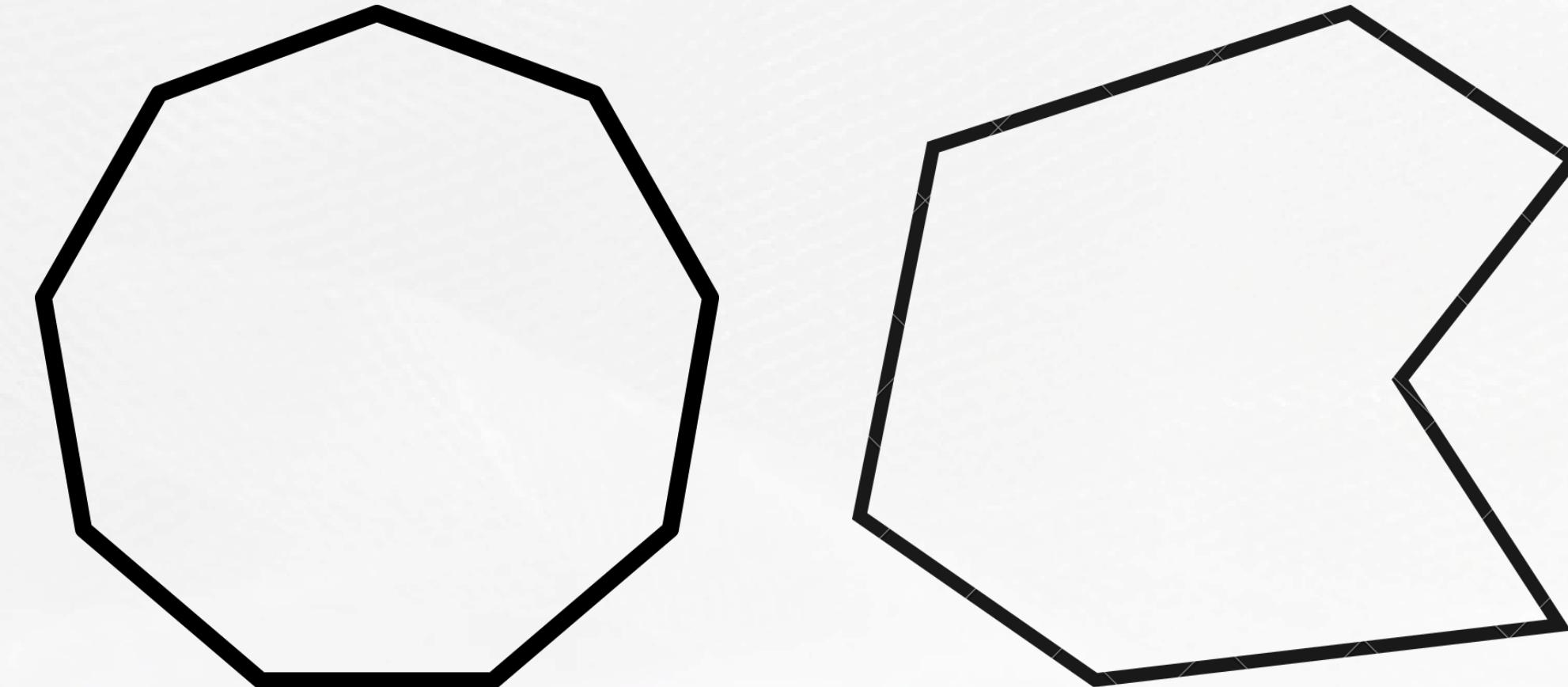
BIỂU DIỄN CƠ BẢN



ĐA GIÁC

Là tổng hợp nhiều điểm, số điểm tối thiểu là 3.

Ví dụ: Đa giác lồi, đa giác lõm, tam giác, tứ giác...



HÀM CƠ BẢN

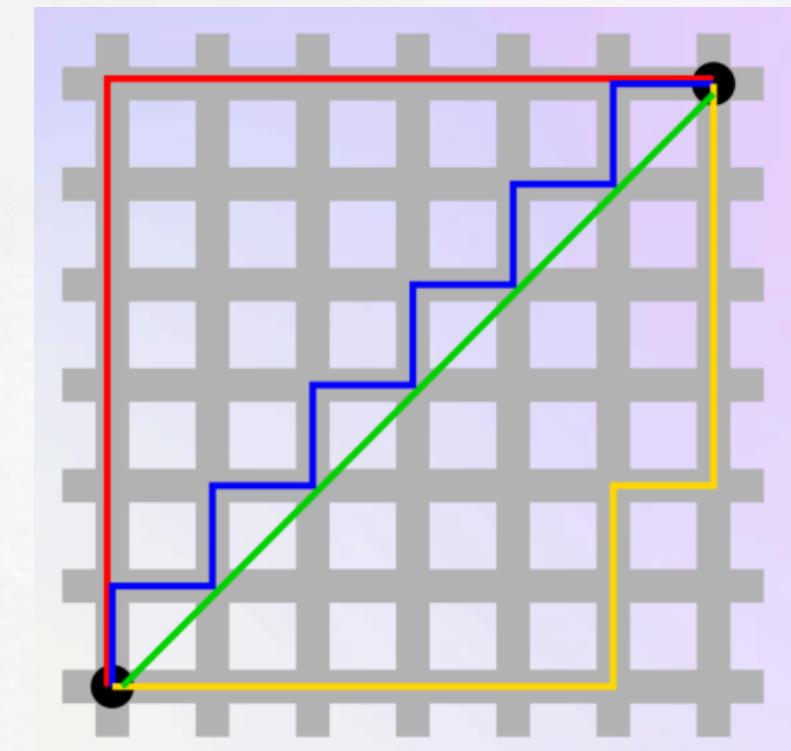


KHOẢNG CÁCH GIỮA 2 ĐIỂM



KHOẢNG CÁCH EUCLID

$$\sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2}$$



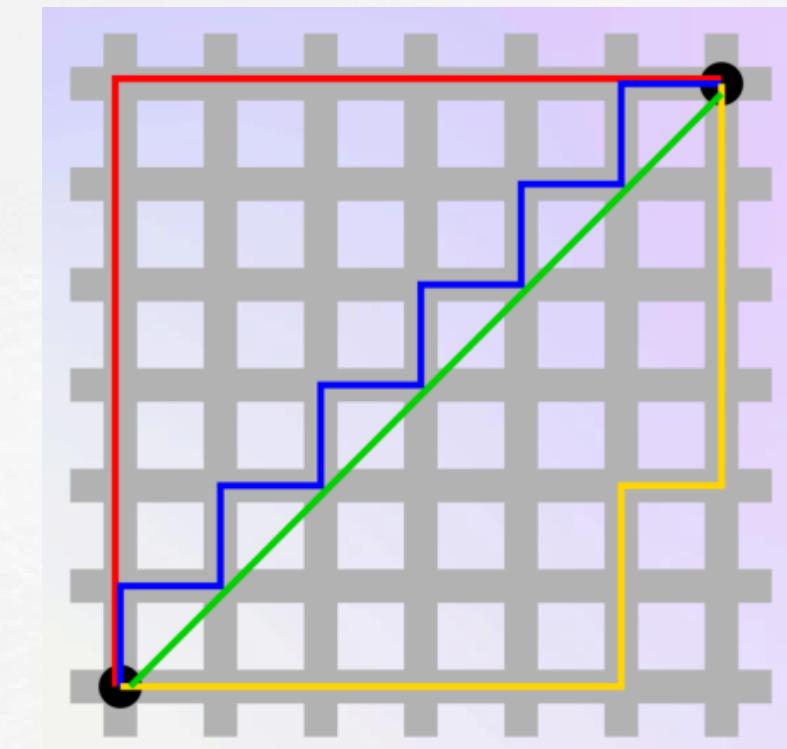
```
def euclidean_distance(p, q):  
    return math.sqrt((p[0] - q[0])**2 + (p[1] - q[1])**2)
```

KHOẢNG CÁCH GIỮA 2 ĐIỂM



KHOẢNG CÁCH MANHATTAN

$$|x_P - x_Q| + |y_P - y_Q|$$



```
def manhattan_distance(p, q):
    return abs(p[0] - q[0]) + abs(p[1] - q[1])
```



KIỂM TRA ĐIỂM NẰM TRÊN ĐOẠN THẲNG

Bước 1: Kiểm tra 3 điểm thẳng hàng.

Bước 2: Kiểm tra tọa độ.

```
def is_point_on_segment(p, seg_start, seg_end):
    def cross_product(p1, p2):
        return p1[0] * p2[1] - p2[0] * p1[1]

    # Tạo các vector
    vector_a = (seg_end[0] - seg_start[0], seg_end[1] - seg_start[1])
    vector_b = (p[0] - seg_end[0], p[1] - seg_end[1])

    # Kiểm tra xem điểm có nằm trong bounding box của đoạn thẳng
    if (seg_start[0] <= p[0] <= seg_end[0] or seg_start[0] >= p[0] >= seg_end[0]) and
       (seg_start[1] <= p[1] <= seg_end[1] or seg_start[1] >= p[1] >= seg_end[1]):
        # Kiểm tra tính cộng tuyến
        return cross_product(vector_a, vector_b) == 0
    return False
```

TÍCH GIỮA HAI VECTOR



TÍCH VÔ HƯỚNG

$$\vec{u} \cdot \vec{v} = x_1x_2 + y_1y_2$$

$$\vec{u} \cdot \vec{v} = |\vec{u}| |\vec{v}| \cos \theta$$

```
def dot_product(v1, v2):  
    return v1[0] * v2[0] + v1[1] * v2[1]
```

TÍCH GIỮA HAI VECTOR



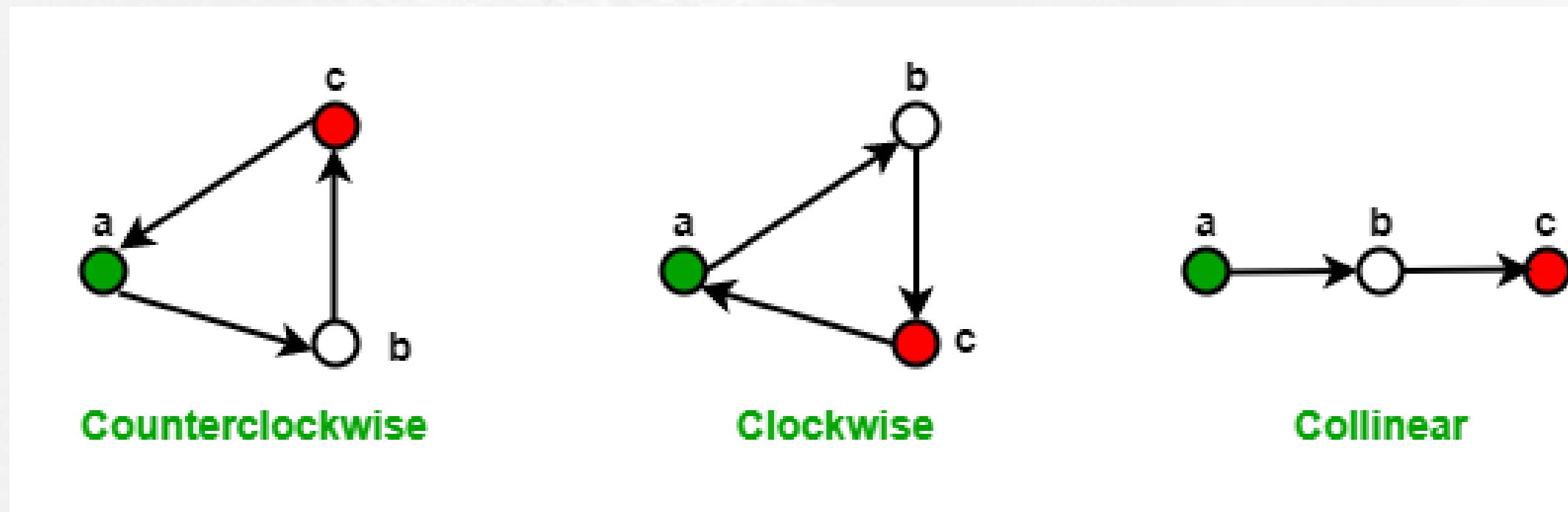
TÍCH CHÉO

$$\vec{a} \times \vec{b} = x_1y_2 - y_1x_2$$

$$\vec{a} \times \vec{b} = \vec{n}|\vec{a}||\vec{b}|\sin\theta$$

```
def cross_product(v1, v2):  
    return v1[0] * v2[1] - v1[1] * v2[0]
```

ỨNG DỤNG TÍCH CHÉO TRONG VIỆC KIỂM TRA CÙNG CHIỀU KIM ĐỒNG HỒ



Lấy tích chéo giữa vector AB và BC:
Nếu bằng 0 => 3 điểm thẳng hàng.
Nếu lớn hơn 0 => quay ngược chiều kim đồng hồ.
Nếu bé hơn 0 => quay cùng chiều kim đồng hồ.

KHOẢNG CÁCH GIỮA ĐIỂM VÀ ĐƯỜNG THẲNG

$$d = \frac{|a \cdot x_0 + b \cdot y_0 + c|}{\sqrt{a^2 + b^2}}$$

```
def distance_from_point_to_line(p, line):
    x0, y0 = p
    a, b, c = line
    return abs(a * x0 + b * y0 + c) / math.sqrt(a * a + b * b)
```

GÓC GIỮA 2 VECTOR

$$\cos \theta = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| |\vec{v}|}$$

GÓC GIỮA 2 VECTOR

```
def vector_magnitude(v):
    return math.sqrt(v[0]**2 + v[1]**2)

def angle_between_vectors(v1, v2):
    dot_product = v1[0] * v2[0] + v1[1] * v2[1]
    mag1 = vector_magnitude(v1)
    mag2 = vector_magnitude(v2)

    if mag1 == 0 or mag2 == 0:
        raise ValueError("One of the vectors has zero magnitude.")

    cos_theta = dot_product / (mag1 * mag2)
    cos_theta = max(-1.0, min(1.0, cos_theta))
    return math.acos(cos_theta)
```

CHU VI CỦA MỘT ĐA GIÁC

Áp dụng với đa giác với các đỉnh được sắp xếp theo chiều kim đồng hồ hoặc ngược lại

Chu vi đa giác = tổng độ dài các cạnh

```
import math

def calculate_perimeter(polygon):
    perimeter = 0.0
    n = len(polygon)
    for i in range(n):
        p1, p2 = polygon[i], polygon[(i + 1) % n]
        distance = math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)
        perimeter += distance
    return perimeter
```

DIỆN TÍCH CỦA ĐA GIÁC

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

```
def calculate_area(polygon):
    n = len(polygon)
    area = 0.0
    for i in range(n):
        area += polygon[i][0] * polygon[(i + 1) % n][1] - polygon[i][1] * polygon[(i + 1) % n][0]
    return abs(area) / 2.0
```

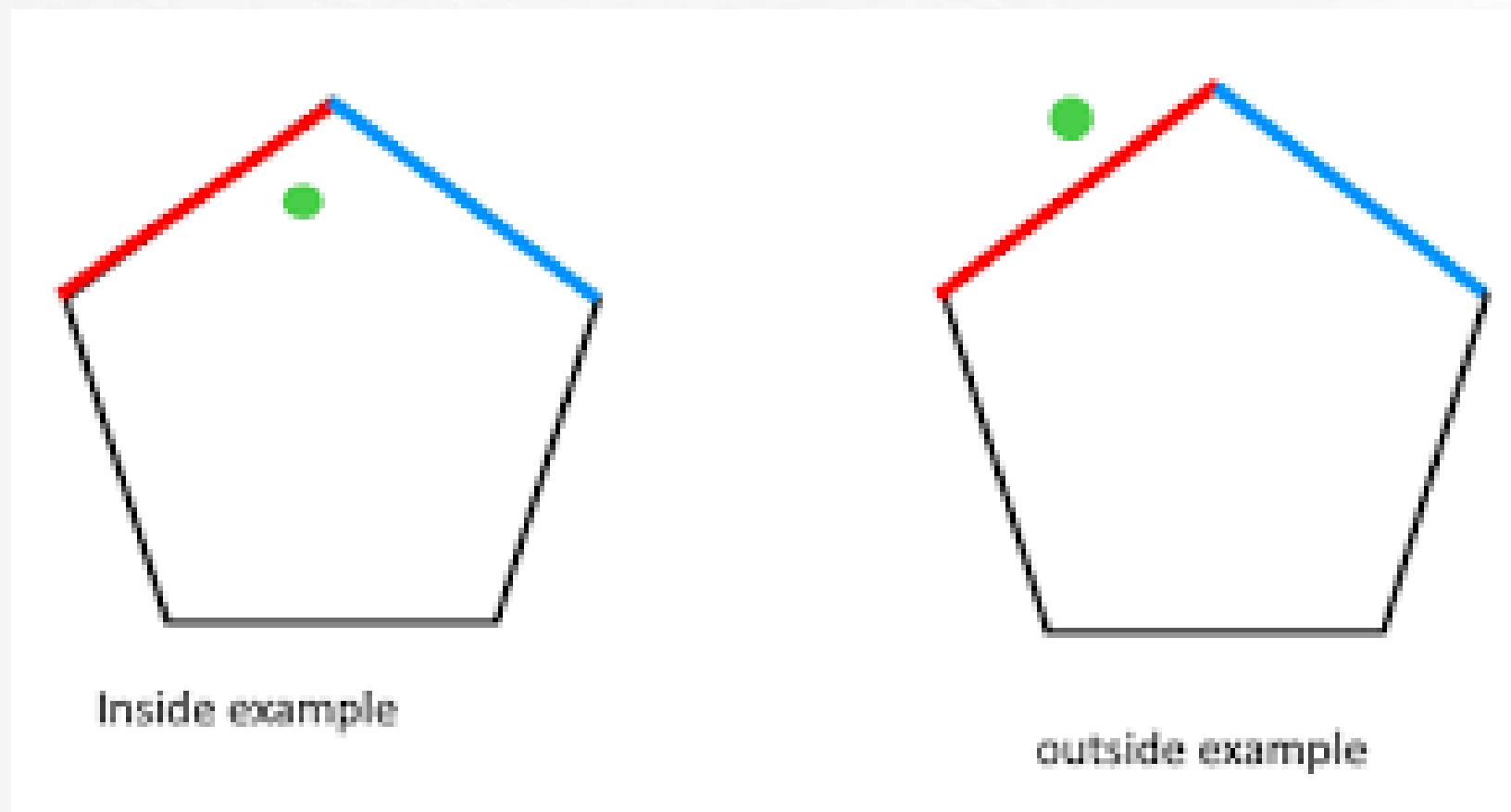


ĐỈNH NẰM TRONG ĐA GIÁC

Nếu đỉnh nằm trong đa giác:

Mọi tích chéo giữa vector cạnh $\vec{A_i A_{i+1}}$ với $\vec{A_{i+1} P}$ đều phải cùng dấu

```
def is_point_inside_polygon(p, polygon):
    n = len(polygon)
    has_positive = False
    has_negative = False
    for i in range(n):
        p1 = polygon[i]
        p2 = polygon[(i + 1) % n]
        # Tính toán tích có hướng (cross product) và kiểm tra điểm có nằm trên đoạn thẳng không
        # (Phần này giả sử bạn đã có các hàm cross_product và is_point_on_segment)
        cross = cross_product(p1, p2, p)
        if cross == 0 and is_point_on_segment(p, p1, p2):
            return True
        if cross > 0:
            has_positive = True
        if cross < 0:
            has_negative = True
        if has_positive and has_negative:
            return False
    return True|
```



Inside example

outside example

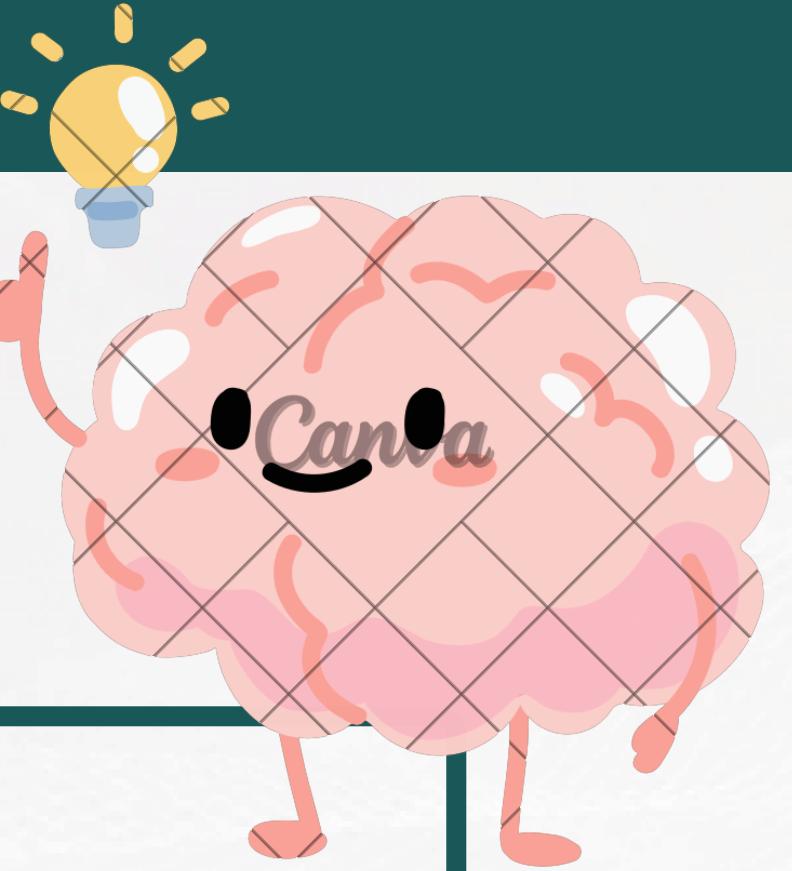
KIỂM TRA ĐA GIÁC LỒI

```
def is_convex_polygon(polygon):
    n = len(polygon)
    if n < 3:
        return False
    has_positive = False
    has_negative = False
    for i in range(n):
        p1 = polygon[i]
        p2 = polygon[(i + 1) % n]
        p3 = polygon[(i + 2) % n]
        # Tính tích có hướng (cross product)
        cross = cross_product(p1, p2, p3)
        if cross > 0:
            has_positive = True
        elif cross < 0:
            has_negative = True
        if has_positive and has_negative:
            return False
    return True
```

Tương tự với thuật toán trước thì ta sử dụng tích chéo.

Đa giác lồi khi và chỉ khi mọi tích chéo giữa 2 cạnh liên tiếp luôn cùng dấu.



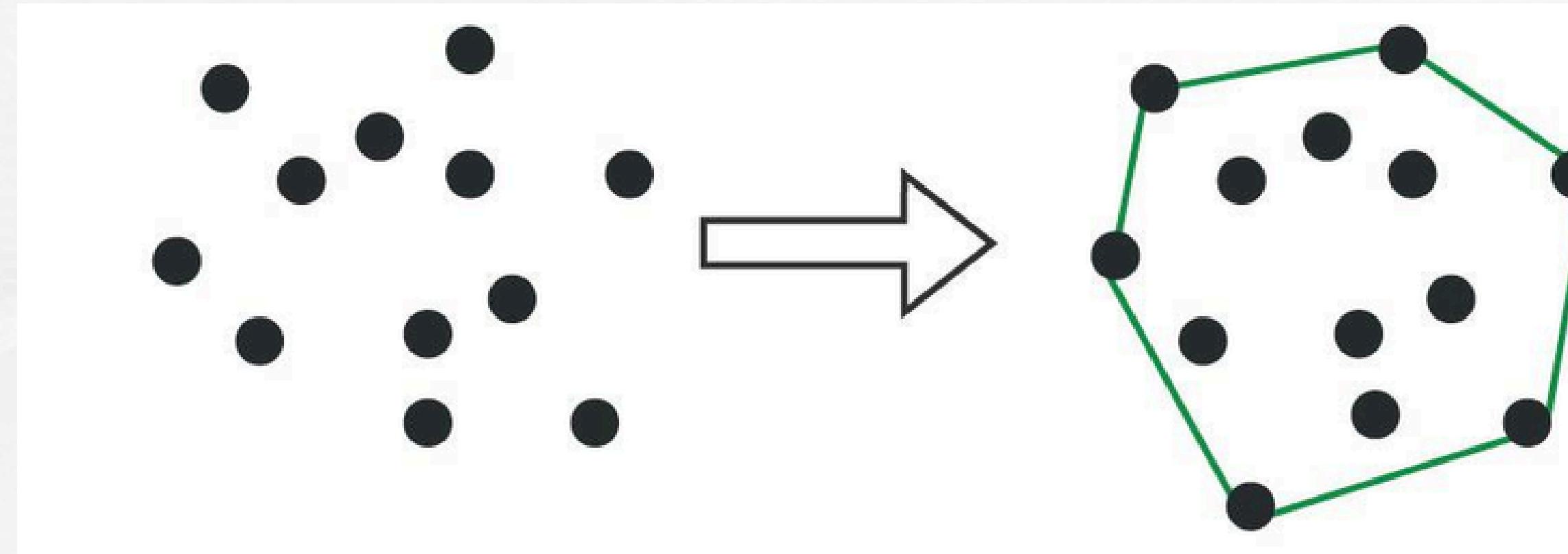


GIỚI THIỆU THUẬT TOÁN BÀO LÔI (CONVEX HULL)



BÀI TOÁN ĐẶT RA

Cho tập các điểm, chọn ra một tập hợp điểm nhỏ nhất tạo thành đa giác lồi, sao cho tất cả các điểm ban đầu đều nằm trong nó.



MỘT SỐ THUẬT TOÁN TÌM KIẾM BAO LÔI

Convex Hull Algorithm

Time Complexity

[Convex Hull using Divide and Conquer Algorithm](#)

$O(N * \log N)$

[Convex Hull using Jarvis' Algorithm or Wrapping](#)

$O(N^2)$

[Convex Hull using Graham Scan](#)

$O(N * \log N)$

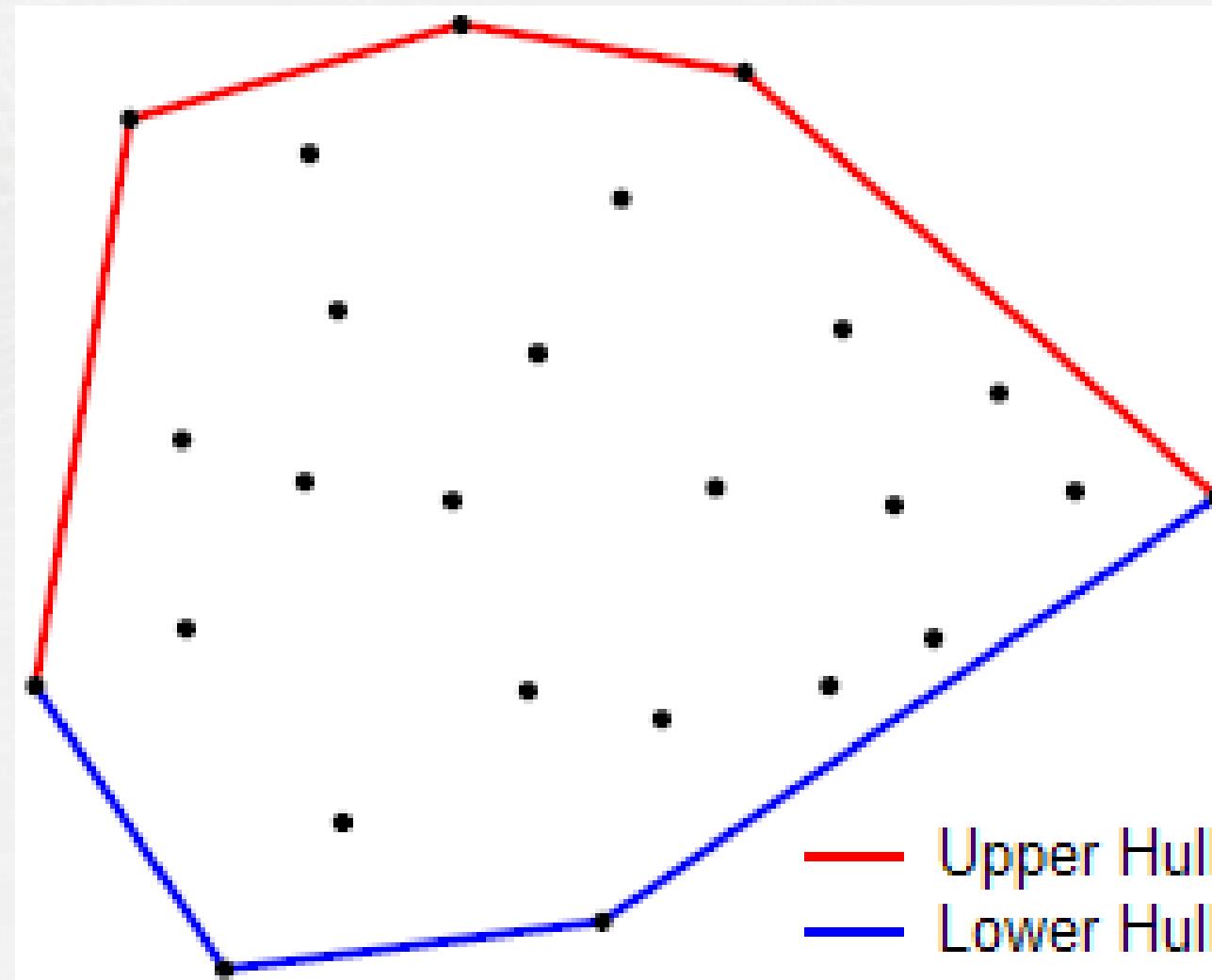
[Convex Hull using Monotone chain algorithm](#)

$O(N * \log(N))$

[Convex Hull using Quickhull Algorithm](#)

The analysis is similar to Quick Sort. On average, we get time complexity as $O(N * \log N)$, but in worst case, it can become $O(N^2)$

THUẬT TOÁN CHUỖI ĐƠN ĐIỆU



Sắp xếp các đỉnh theo hoành độ và tung độ.

Duyệt từ đầu đến cuối, thêm các đỉnh vào tập bao lồi H .

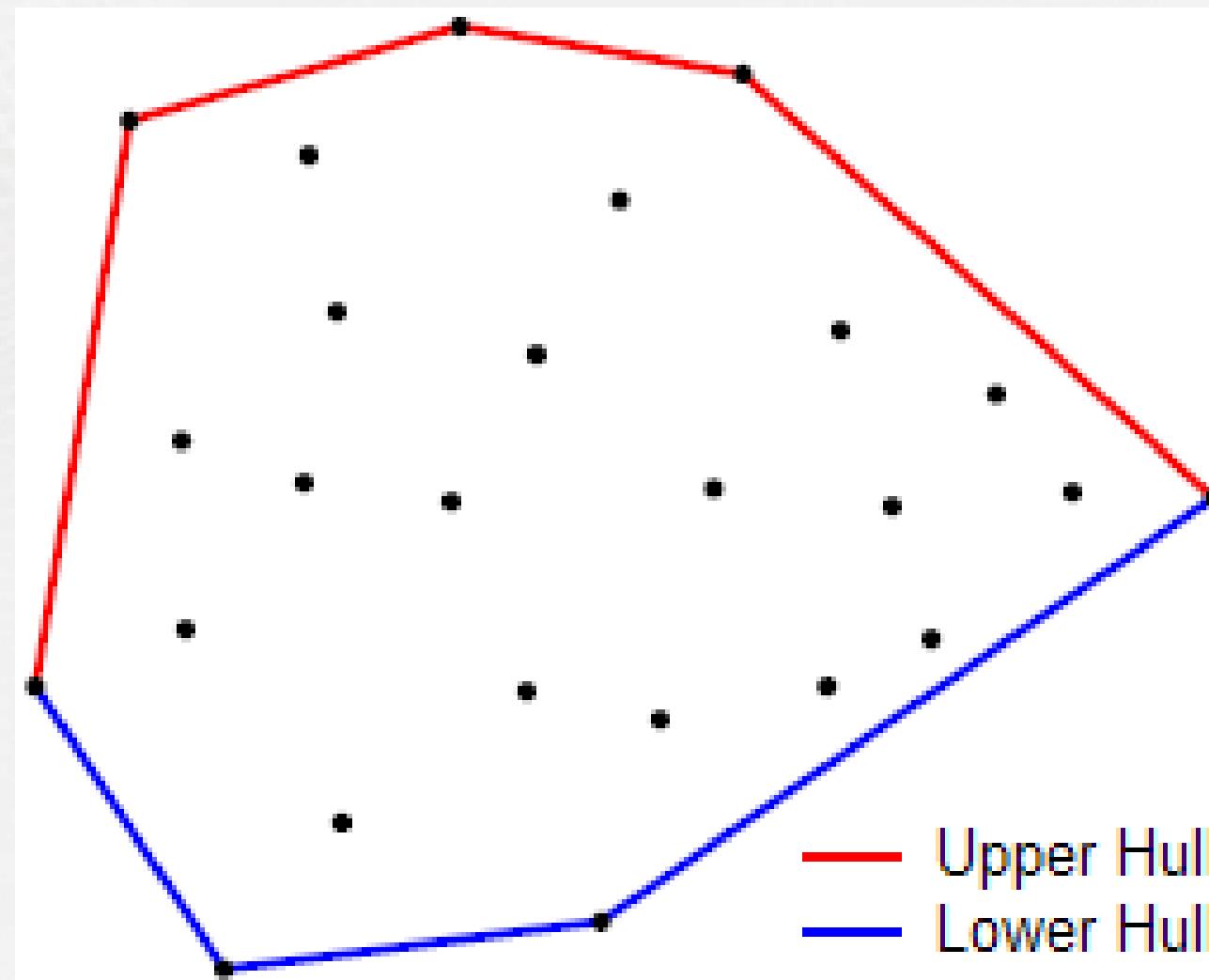
Nếu H có nhiều hơn 3 đỉnh.

Gọi $\vec{u} = \overrightarrow{H_{h-2}H_{h-1}}, \vec{v} = \overrightarrow{H_{h-1}H_h}$

Nếu $\vec{u} \times \vec{v} < 0$ thì 2 cạnh cuối cùng xoay cùng chiều kim đồng hồ, ổn.

Nếu $\vec{u} \times \vec{v} > 0$ thì 2 cạnh cuối cùng xoay ngược chiều kim đồng hồ tạo ra phần lõm trên bao. Ta sẽ xóa 2 đỉnh cuối cùng đi (xóa cạnh cuối cùng)

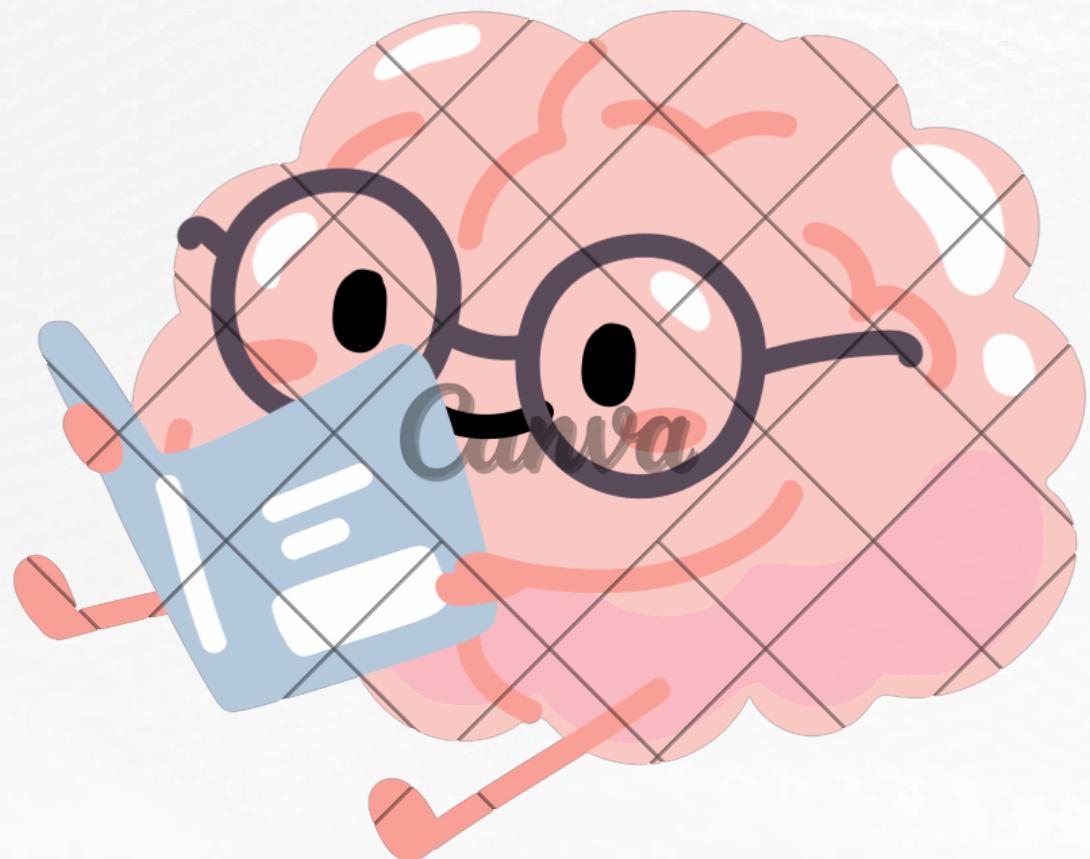
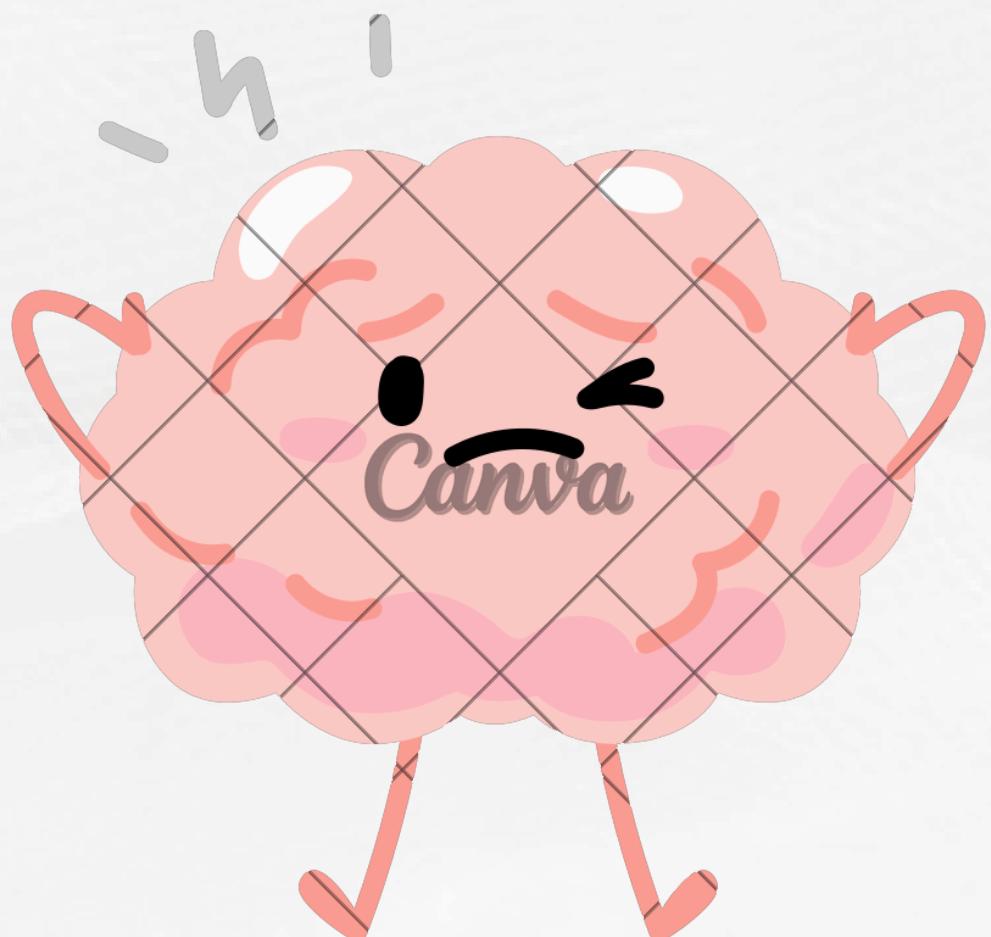
THUẬT TOÁN CHUỖI ĐƠN ĐIỆU



Sau khi thực hiện xong thì ta sẽ có phần upper hull của bao lồi. Bây giờ chỉ cần thực hiện với thứ tự ngược lại (xét lại từ phải sang trái) ta thu được phần lower hull
Ghép 2 phần trên lại ta được bao lồi hoàn chỉnh

CÂU HỎI

TỔNG KẾT BÀI



KHOẢNG CÁCH EUCLID CÓ CÔNG THỨC LÀ ?

a

$$\sqrt{(x_P - x_Q)^2 - (y_P - y_Q)^2}$$

c

$$\sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2}$$

b

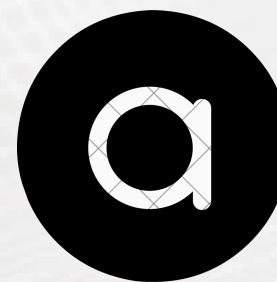
$$\sqrt{(x_P + x_Q)^2 - (y_P + y_Q)^2}$$

d

$$\sqrt{(x_P - x_Q)^2 + (y_P + y_Q)^2}$$

ĐÁP ÁN

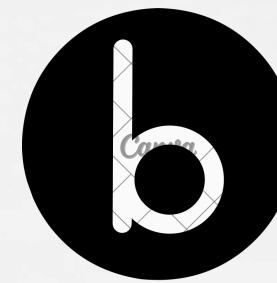
KHOẢNG CÁCH EUCLID CÓ CÔNG THỨC LÀ ?



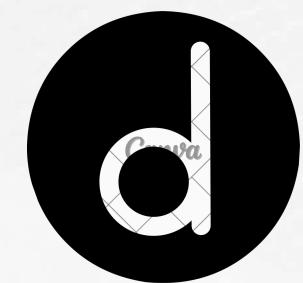
$$\sqrt{(x_P - x_Q)^2 - (y_P - y_Q)^2}$$



$$\sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2}$$



$$\sqrt{(x_P + x_Q)^2 - (y_P + y_Q)^2}$$



$$\sqrt{(x_P - x_Q)^2 + (y_P + y_Q)^2}$$

KHOẢNG CÁCH MANHATTAN CÓ CÔNG THỨC LÀ:

a

$$|x_P - x_Q| - |y_P - y_Q|$$

C

$$|x_P + x_Q| + |y_P - y_Q|$$

b

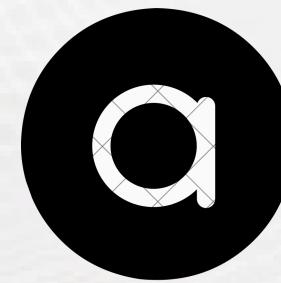
$$|x_P - x_Q| + |y_P - y_Q|$$

d

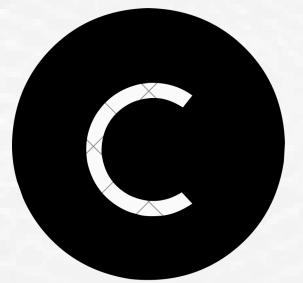
$$|x_P - x_Q| - |y_P + y_Q|$$

ĐÁP ÁN

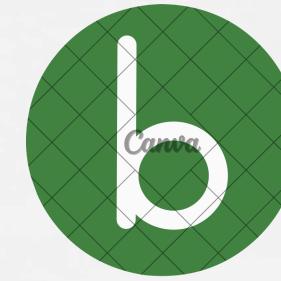
KHOẢNG CÁCH EUCLID CÓ CÔNG THỨC LÀ ?



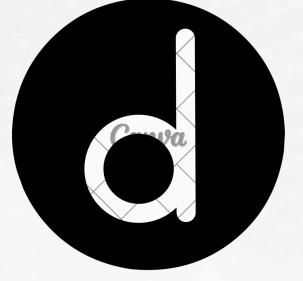
$$|x_P - x_Q| - |y_P - y_Q|$$



$$|x_P + x_Q| + |y_P - y_Q|$$



$$|x_P - x_Q| + |y_P - y_Q|$$



$$|x_P - x_Q| - |y_P + y_Q|$$

KHOẢNG CÁCH GIỮA ĐIỂM VÀ ĐƯỜNG THẲNG ĐƯỢC TÍNH THEO CÔNG THỨC?

a

$$d = \frac{|a \cdot x_0 + b \cdot y_0 + c|}{\sqrt{a^2 + b^2}}$$

b

$$d = \frac{|a \cdot x_0 - b \cdot y_0 - c|}{\sqrt{a^2 + b^2}}$$

c

$$d = \frac{|a \cdot x_0 - b \cdot y_0 + c|}{\sqrt{a^2 - b^2}}$$

d

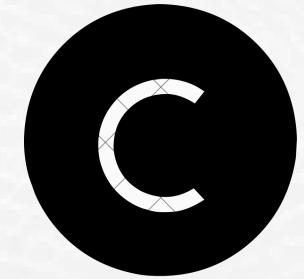
$$d = \frac{|a \cdot x_0 - b \cdot y_0 - c|}{\sqrt{a^2 - b^2}}$$

ĐÁP ÁN

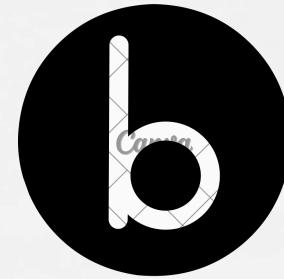
KHOẢNG CÁCH GIỮA ĐIỂM VÀ ĐƯỜNG THẲNG
ĐƯỢC TÍNH THEO CÔNG THỨC?



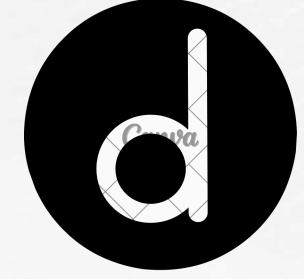
$$d = \frac{|a \cdot x_0 + b \cdot y_0 + c|}{\sqrt{a^2 + b^2}}$$



$$d = \frac{|a \cdot x_0 - b \cdot y_0 + c|}{\sqrt{a^2 - b^2}}$$



$$d = \frac{|a \cdot x_0 - b \cdot y_0 - c|}{\sqrt{a^2 + b^2}}$$



$$d = \frac{|a \cdot x_0 - b \cdot y_0 + c|}{\sqrt{a^2 - b^2}}$$

DIỆN TÍCH CỦA ĐA GIÁC ĐƯỢC TÍNH THEO CÔNG THỨC?

a

$$A = \frac{1}{4} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} + y_i \cdot x_{i-1}) \right|$$

b

$$A = \frac{1}{3} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} + y_i \cdot x_{i-1}) \right|$$

c

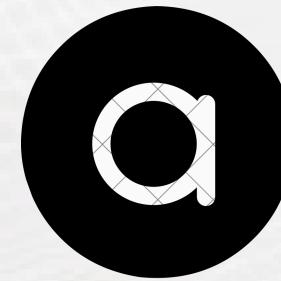
$$A = \frac{1}{3} \left| \sum_{i=1}^n (x_i \cdot y_{i-1} - y_i \cdot x_{i+1}) \right|$$

d

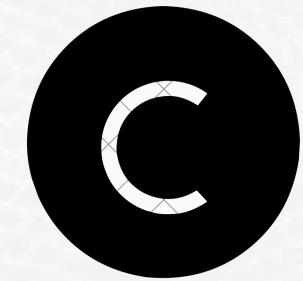
$$A = \frac{1}{2} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} - y_i \cdot x_{i+1}) \right|$$

ĐÁP ÁN

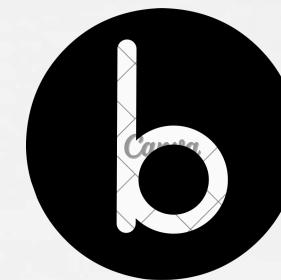
ĐÂU LÀ ĐA GIÁC LÔM?



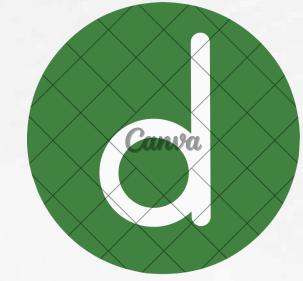
$$A = \frac{1}{4} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} + y_i \cdot x_{i-1}) \right|$$



$$A = \frac{1}{3} \left| \sum_{i=1}^n (x_i \cdot y_{i-1} - y_i \cdot x_{i+1}) \right|$$

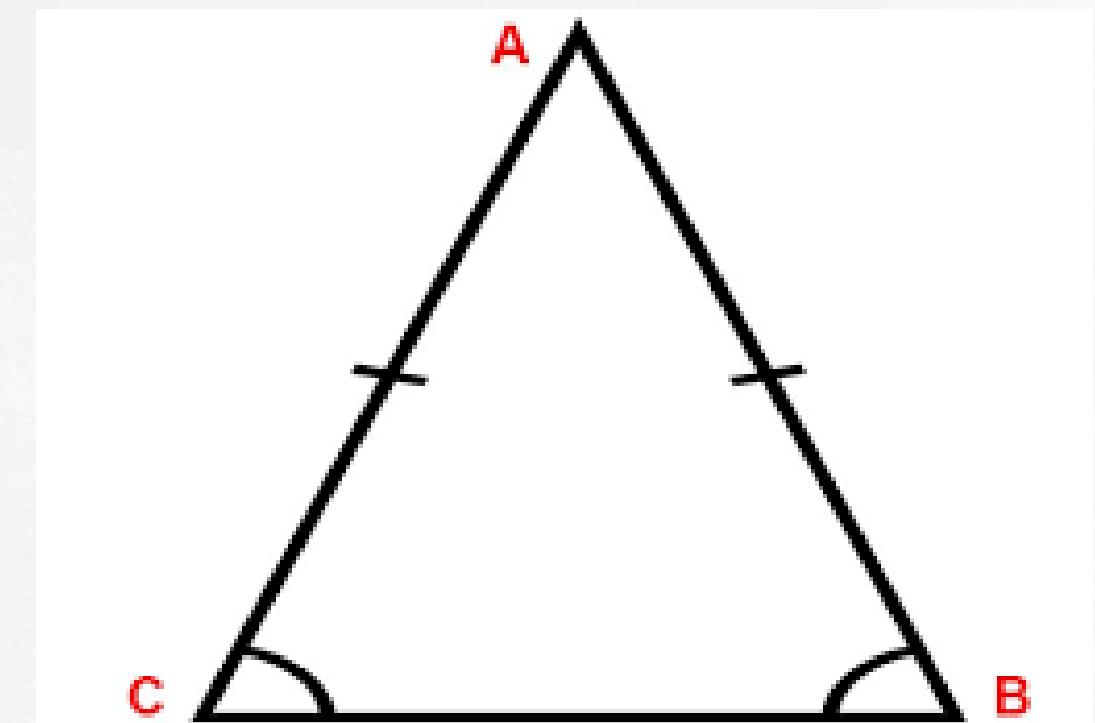
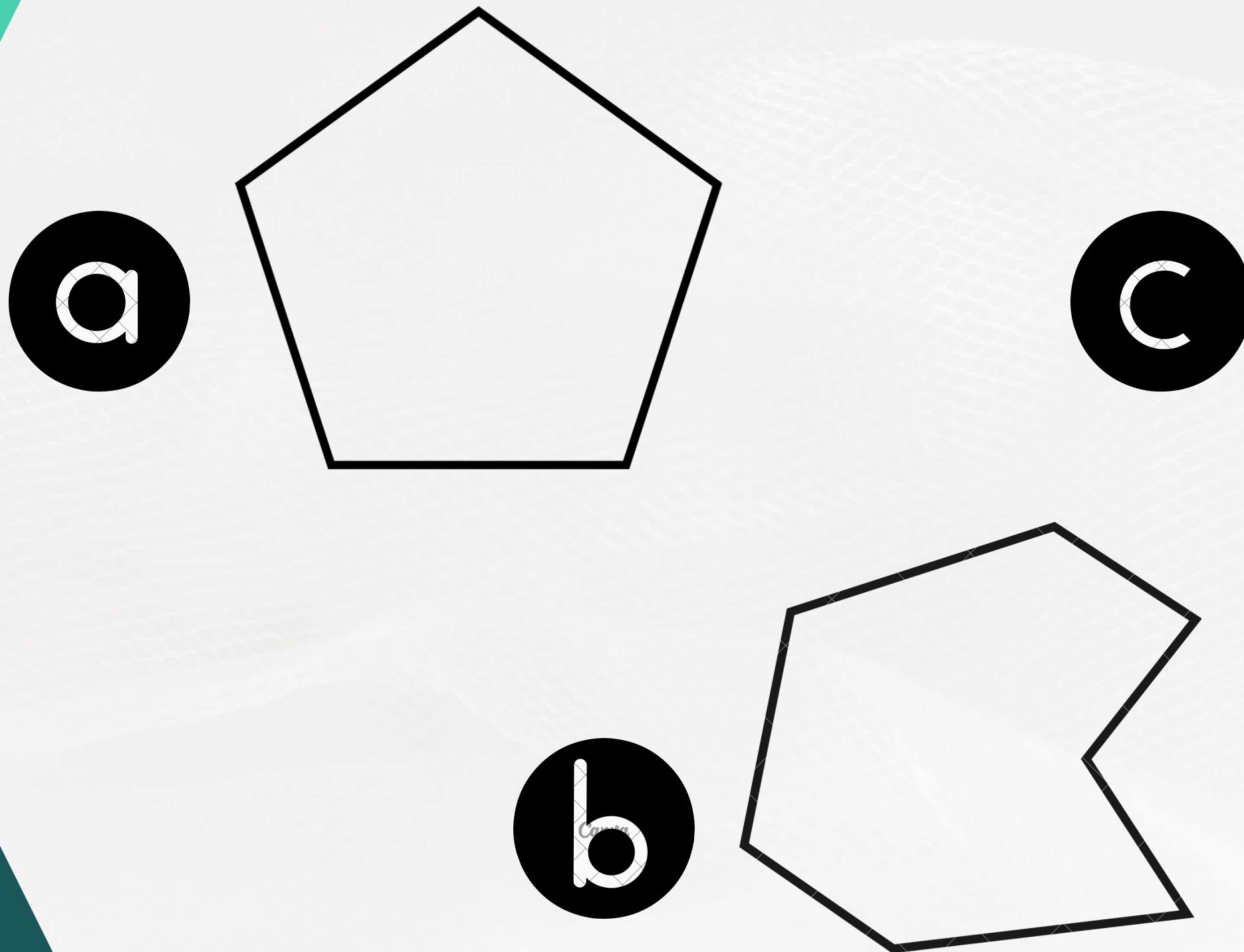


$$A = \frac{1}{3} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} + y_i \cdot x_{i-1}) \right|$$



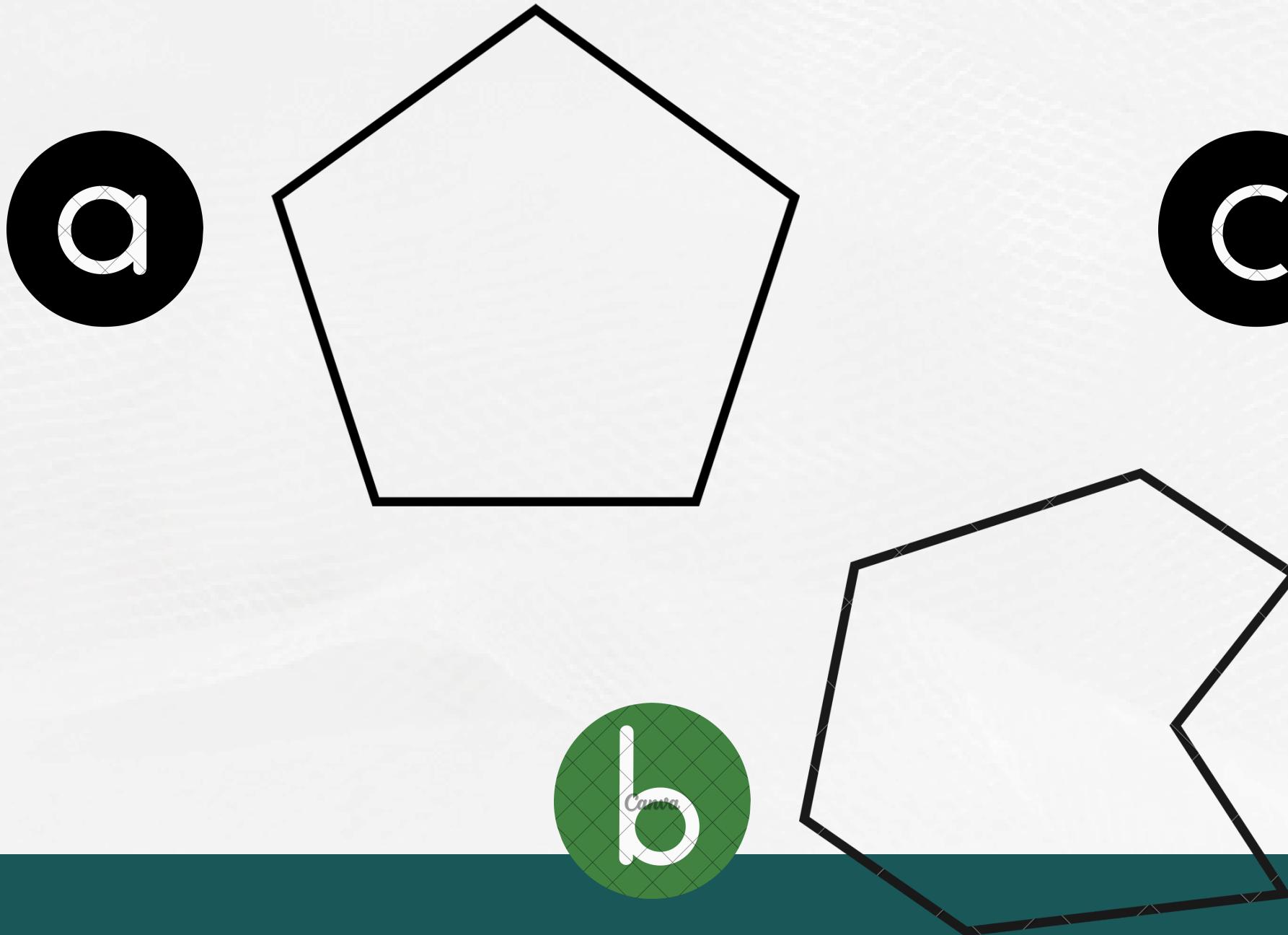
$$A = \frac{1}{2} \left| \sum_{i=1}^n (x_i \cdot y_{i+1} - y_i \cdot x_{i+1}) \right|$$

ĐÂU LÀ ĐA GIÁC LÔM?



ĐÁP ÁN

ĐÂU LÀ ĐA GIÁC LÔM?





NHÓM 2

THÁNG 11, 2024

CẢM ƠN MỌI
NGƯỜI ĐÃ
LẮNG NGHE

Vinh Khánh - Thiện Nhân

Thiên Thanh