

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ  
MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÀI TẬP NHÓM 2  
BỞI NHÓM 12

---

Giáo viên hướng dẫn: *Nguyễn Thanh Sơn*

Nhóm 12:

- *Hoàng Minh Thái - 23521414*
- *Nguyễn Trọng Tất Thành - 23521455*



## Mục lục

# GIẢI BÀI TẬP BỞI NHÓM 12

## Bài 1: Bài Toán Khu Vườn

### Đề bài

Ông Nhân có một khu vườn với  $n$  cây, mỗi cây được biểu diễn bởi tọa độ  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  trên mặt phẳng tọa độ. Ông Nhân muốn cột một sợi dây để bao quanh toàn bộ các cây sao cho độ dài dây cần dùng là ngắn nhất. Hãy tìm:

1. Độ dài ngắn nhất của sợi dây để bao quanh tất cả các cây.
2. Cách cột dây hợp lý (thứ tự các đỉnh mà dây đi qua).

### Yêu cầu

Viết chương trình nhận vào số  $n$  và tọa độ của  $n$  cây. Tính và in ra:

1. Độ dài ngắn nhất của sợi dây.
2. Danh sách các điểm (tọa độ các đỉnh) mà dây đi qua, theo thứ tự ngược chiều kim đồng hồ.

### Giải thích

Bài toán này liên quan đến việc tìm **convex hull** (vỏ bao lồi) của một tập hợp các điểm trên mặt phẳng. Một cách tiếp cận hiệu quả là sử dụng thuật toán **Graham scan** hoặc **Jarvis march**. Cả hai thuật toán này đều có thể giải quyết bài toán với độ phức tạp  $O(n \log n)$  cho Graham scan và  $O(nh)$  cho Jarvis march, trong đó  $h$  là số đỉnh của đa giác lồi.

### Mã giả:

```
1 # Thuật toán Graham scan để tìm convex hull
2 def convex_hull(points):
3     # Sắp xếp các điểm theo hoành độ và tung độ
4     points = sorted(points)
5
6     # Hàm tính tích cho ba ba điểm
7     def cross(o, a, b):
8         return (a[0] - o[0]) * (b[1] - o[1]) - (a[1] - o[1]) * (b[0] - o[0])
9
10    # Chạy qua các điểm để xây dựng phần dưới và phần trên của đa giác lồi
11    lower = []
12    for p in points:
13        while len(lower) >= 2 and cross(lower[-2], lower[-1], p) <= 0:
```



```
14     lower.pop()
15     lower.append(p)
16
17     upper = []
18     for p in reversed(points):
19         while len(upper) >= 2 and cross(upper[-2], upper[-1], p) <= 0:
20             upper.pop()
21         upper.append(p)
22
23     # Kt hp hai phn ca convex hull
24     return lower[:-1] + upper[:-1]
25
26 # Tinh din tch ca convex hull
27 def calculate_perimeter(points):
28     perimeter = 0
29     for i in range(len(points)):
30         j = (i + 1) % len(points)
31         perimeter += math.dist(points[i], points[j])
32     return perimeter
```

## Phương pháp tính diện tích của convex hull

Khi đã có các điểm của convex hull, chúng ta có thể tính diện tích của đa giác lồi này bằng công thức diện tích của một đa giác:

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

Trong đó,  $(x_{n+1}, y_{n+1})$  được coi là  $(x_1, y_1)$ .



## Bài 2: Khu Vườn Giao Nhau

### Đề bài

Ông Nhân có hai mảnh đất, mỗi mảnh đất được mô tả bởi một đa giác lồi trong mặt phẳng tọa độ. Mảnh đất thứ nhất có các đỉnh là  $A_1(x_1, y_1), A_2(x_2, y_2), \dots, A_m(x_m, y_m)$ , và mảnh đất thứ hai có các đỉnh là  $B_1(x'_1, y'_1), B_2(x'_2, y'_2), \dots, B_n(x'_n, y'_n)$ . Mỗi mảnh đất được biểu diễn theo thứ tự ngược chiều kim đồng hồ.

Nhiệm vụ của bạn là tính diện tích của vùng đất giao nhau giữa hai mảnh đất này, tức là diện tích phần đất mà cả hai mảnh đất đều bao phủ.

### Yêu cầu

Tính diện tích của vùng đất giao nhau giữa hai mảnh đất.

### Giải thích

Để giải quyết bài toán này, chúng ta cần xác định phần giao nhau giữa hai đa giác lồi. Thuật toán có thể gồm các bước sau:

1. Tìm các đoạn thẳng giao nhau giữa hai đa giác lồi.
2. Xác định các điểm giao nhau giữa các đoạn thẳng của hai đa giác.
3. Dùng các điểm giao nhau này để xây dựng đa giác lồi giao nhau.
4. Tính diện tích của đa giác giao nhau.

Thuật toán sử dụng phương pháp **liên kết các đoạn thẳng giao nhau** và áp dụng công thức diện tích cho đa giác lồi đã có các điểm giao.

### Mã giả:

```
1  # Tìm cc đ on thng giao nhau
2  def line_intersection(p1, p2, q1, q2):
3      # Hm kiểm tra giao đim gia hai đ on thng p1p2 v q1q2
4      def cross(o, a, b):
5          return (a[0] - o[0]) * (b[1] - o[1]) - (a[1] - o[1]) * (b[0] - o[0])
6
7      # Tinh giao đim ca hai đ on thng
8      d1 = cross(q1, q2, p1)
9      d2 = cross(q1, q2, p2)
10     d3 = cross(p1, p2, q1)
11     d4 = cross(p1, p2, q2)
12     if d1 * d2 < 0 and d3 * d4 < 0:
13         return (p1, p2) # Tr v đim giao
14     return None
15
16 # Tinh din tch ca đa gic giao nhau
17 def calculate_intersection_area(poly1, poly2):
18     # Tinh cc đim giao nhau
19     intersection_points = []
20     for i in range(len(poly1)):
21         for j in range(len(poly2)):
22             intersection = line_intersection(poly1[i], poly1[(i+1)%len(poly1)],
23                                             poly2[j], poly2[(j+1)%len(poly2)])
```



```
23         if intersection:
24             intersection_points.append(intersection)
25
26     # Tính diện tích của đa giác giao
27     return calculate_area(intersection_points)
```

## Phương pháp tính diện tích của đa giác giao nhau

Tương tự bài toán Bài 1, diện tích của đa giác giao nhau có thể được tính bằng công thức diện tích của đa giác:

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

Trong đó,  $(x_{n+1}, y_{n+1})$  được coi là  $(x_1, y_1)$ .