

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

VẬN DỤNG THIẾT KẾ THUẬT TOÁN TRONG HÌNH HỌC
BÀI TẬP NHÓM 4

Sinh viên:

NGUYỄN HUY PHƯỚC - 23521234

PHAN NHẬT TÂN - 23521405

Giảng viên:

Nguyễn Thanh Sơn



UIT

TP. HỒ CHÍ MINH, năm 2024

Mục lục

1 Bài 1: Bài toán khu vườn	2
1.1 Phân tích và hướng giải:	2
1.2 Các kỹ thuật và kiến thức được áp dụng	2
1.3 Mã giả:	2
2 Bài 2: Khu vườn giao nhau	4
2.1 Phân tích và hướng giải:	4
2.2 Những kĩ thuật và kiến thức được áp dụng	4
2.3 Mã giả:	5

1 Bài 1: Bài toán khu vườn

1.1 Phân tích và hướng giải:

Bản chất của bài toán là tìm bao lồi (chu vi của đa giác lồi là chu vi của bao lồi và danh sách các điểm cần liệt kê, chính là danh sách các điểm nằm trên bao lồi này).

Ta áp dụng thuật toán có tên: monotone chain để tìm tập đỉnh bao lồi. Gồm các bước cơ bản, sắp xếp lại dãy sau đó tìm bao lồi trên và bao lồi dưới bằng cách duyệt và sử dụng tích có hướng để tìm.

Sau đó, Ta duyệt các cạnh của bao lồi thông qua tập đỉnh rút ra ở trên, ta tiến hành tính tổng khoảng cách các cạnh để rút ra chu vi.

1.2 Các kỹ thuật và kiến thức được áp dụng

Ta áp dụng kỹ thuật giảm để trị để sắp xếp lại các điểm theo chiều kim đồng hồ, chuẩn bị cho giai đoạn tìm bao lồi trên và dưới.

Ta áp dụng kiến thức về hình học (tích có hướng và chu vi đa giác) để áp dụng vào bài toán

1.3 Mã giả:

Algorithm 1 FINDCONVEXHULL

Require: *points* - Tập hợp các điểm đầu vào

Ensure: *hull* - Danh sách các đỉnh của bao lồi theo thứ tự ngược chiều kim đồng hồ

- 1: **Tìm điểm bắt đầu:** Chọn điểm *start* có *y* nhỏ nhất trong *points* (nếu trùng thì chọn *x* nhỏ nhất)
 - 2: **Sắp xếp các điểm:** Sắp xếp *points* theo góc cực tăng dần so với *start*
 - 3: **Khởi tạo:** *hull* \leftarrow []
 - 4: **for each point in points do**
 - 5: **while** *hull* có ít nhất 2 điểm và 3 điểm cuối không tạo góc quay ngược chiều kim đồng hồ **do**
 - 6: Loại bỏ điểm cuối cùng khỏi *hull*
 - 7: **end while**
 - 8: Thêm *point* vào *hull*
 - 9: **end for**
 - 10: **return** *hull*
-

Algorithm 2 CALCULATEPERIMETER

Require: *hull* - Danh sách các điểm trên bao lồi theo thứ tự ngược chiều kim đồng hồ

Ensure: *perimeter* - Chu vi của bao lồi

```
1: perimeter  $\leftarrow$  0
2: N  $\leftarrow$  Số lượng điểm trong hull
3: for i  $\leftarrow$  0 to N - 1 do
4:   P1  $\leftarrow$  hull[i]
5:   P2  $\leftarrow$  hull[(i + 1) mod N] {Điểm tiếp theo (hoặc quay lại điểm đầu nếu là điểm cuối)}
6:   perimeter  $\leftarrow$  perimeter + Khoảng cách giữa P1 và P2
7: end for
8: return perimeter
```

Algorithm 3 MAIN

Require: *points* - Danh sách các điểm đầu vào

```
1: hull  $\leftarrow$  FINDCONVEXHULL(points)
2: perimeter  $\leftarrow$  CALCULATEPERIMETER(hull)
3: In kết quả:
4: In “Các đỉnh của bao lồi:” hull
5: In “Chu vi của bao lồi:” perimeter
```

2 Bài 2: Khu vườn giao nhau

2.1 Phân tích và hướng giải:

Bản chất của bài toán là : Cho hai đa giác lồi P_1 và P_2 , mỗi đa giác được biểu diễn bằng một danh sách các đỉnh theo thứ tự ngược chiều kim đồng hồ. Yêu cầu tính diện tích vùng giao giữa hai đa giác.

1. **Tìm giao điểm giữa các cạnh của hai đa giác:** - Xét từng cặp cạnh của P_1 và P_2 .
 - Sử dụng thuật toán kiểm tra giao điểm giữa hai đoạn thẳng để xác định tất cả các điểm giao.
 - Thêm các điểm giao này vào danh sách IntersectionPoints.
2. **Kiểm tra các đỉnh bên trong:** - Kiểm tra xem các đỉnh của P_1 có nằm bên trong P_2 không:
 - Nếu có, thêm các đỉnh này vào IntersectionPoints.
 - Kiểm tra tương tự với các đỉnh của P_2 trong P_1 .
3. **Sắp xếp các điểm giao:** - Sau khi tìm được tất cả các điểm giao và các đỉnh nằm bên trong, sắp xếp chúng theo thứ tự ngược chiều kim đồng hồ để tạo thành một đa giác mới $P_{\text{intersection}}$ biểu diễn vùng giao.
4. **Tính diện tích vùng giao:** - Sử dụng công thức tính diện tích đa giác từ danh sách các đỉnh:

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

- Trong đó, $(x_{n+1}, y_{n+1}) = (x_1, y_1)$ để khép kín đa giác.

2.2 Những kĩ thuật và kiến thức được áp dụng

- Sử dụng phương pháp giảm để trị để sắp xếp các đỉnh theo chiều kim đồng hồ của đa giác giao nhau mà ta vừa tìm được
- Sử dụng phương pháp chuyển đổi để trị, bản chất công thức tính diện tích đa giác là tách thành các hình thang để tính diện tích.
- Vận dụng các kiến thức về hình học để tính giao điểm của các cặp cạnh và kiểm tra điểm có nằm trong đa giác hay không (tích có hướng).

2.3 Mã giả:

Algorithm 4 INTERSECTIONAREA

Require: *Polygon1, Polygon2* - Hai đa giác lồi được biểu diễn bằng danh sách các đỉnh theo thứ tự ngược chiều kim đồng hồ

Ensure: *Area* - Diện tích vùng giao của hai đa giác

```
1: Khởi tạo: IntersectionPoints  $\leftarrow []$ 
2: Tìm giao điểm của các cạnh:
3: for each cạnh C1 in Polygon1 do
4:   for each cạnh C2 in Polygon2 do
5:     if SEGMENTSINTERSECT(C1, C2) then
6:       Thêm giao điểm vào IntersectionPoints
7:     end if
8:   end for
9: end for
10: Thêm các đỉnh bên trong:
11: for each đỉnh P in Polygon1 do
12:   if POINTINSIDEPOLYGON(P, Polygon2) then
13:     Thêm P vào IntersectionPoints
14:   end if
15: end for
16: for each đỉnh P in Polygon2 do
17:   if POINTINSIDEPOLYGON(P, Polygon1) then
18:     Thêm P vào IntersectionPoints
19:   end if
20: end for
21: Sắp xếp các điểm giao:
22: Sắp xếp IntersectionPoints thành một đa giác mới IntersectionPolygon theo thứ tự ngược chiều kim đồng hồ
23: Tính diện tích:
24: Area  $\leftarrow$  CALCULATEPOLYGONAREA(IntersectionPolygon)
25: return Area
```

Hàm phụ trợ

Algorithm 5 SEGMENTSINTERSECT

Require: Hai đoạn thẳng $A = (x_1, y_1) \rightarrow (x_2, y_2)$, $B = (x_3, y_3) \rightarrow (x_4, y_4)$

Ensure: *True* nếu hai đoạn thẳng giao nhau, ngược lại *False*

- ```
1: Sử dụng định lý giao nhau: Kiểm tra điều kiện quay ngược chiều kim đồng hồ (CCW) giữa các điểm.
2: return Kết quả kiểm tra
```
-

---

**Algorithm 6** POINTINSIDEPOLYGON

---

**Require:** Điểm  $P = (x, y)$ , đa giác  $Polygon$

**Ensure:**  $True$  nếu  $P$  nằm bên trong  $Polygon$ , ngược lại  $False$

- 1: Sử dụng thuật toán tia bắn (ray-casting) để kiểm tra nếu số lần cắt giữa tia từ  $P$  và các cạnh của  $Polygon$  là lẻ.
  - 2: **return** Kết quả kiểm tra
- 

---

**Algorithm 7** CALCULATEPOLYGONAREA

---

**Require:**  $Polygon$  - Một đa giác được biểu diễn bằng danh sách các đỉnh

**Ensure:**  $Area$  - Diện tích của đa giác

- 1:  $Area \leftarrow 0$
  - 2:  $N \leftarrow$  Số lượng đỉnh trong đa giác
  - 3: **for**  $i = 1$  **to**  $N$  **do**
  - 4:    $x_1, y_1 \leftarrow Polygon[i]$
  - 5:    $x_2, y_2 \leftarrow Polygon[(i \bmod N) + 1]$  {Điểm tiếp theo (hoặc quay lại điểm đầu nếu là điểm cuối)}
  - 6:    $Area \leftarrow Area + (x_1 \cdot y_2 - y_1 \cdot x_2)$
  - 7: **end for**
  - 8: **return**  $\frac{|Area|}{2}$
-