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

Phân tích và thiết kế thuật toán

Bài tập nhóm 2

Nhóm 13: Đỗ Quang Lực - 23520902
Nguyễn Minh Huy - 23520634

Ngày 30 tháng 11 năm 2024

Mục lục

1 Bài 1

Chi tiết cài đặt: https://github.com/doquanglucuit/CS112/blob/main/Nhom2-B1.cpp

1.1 Phân tích

Đây là bài toán tìm bao lồi của tập hợp các điểm. Sau khi có được bao lồi, ta tính chu vi của đa giác đó sẽ cho ra chiều dài ngắn nhất của sợi dây.

- Tìm bao lồi: sử dụng thuật toán Monotone Chain để tìm các điểm thuộc bao lồi
- Tính chu vi của bao lồi: Tổng các độ dài các đoạn thẳng nối các đỉnh liên tiếp.

Độ phức tạp: Gọi m, n lần lượt là số điểm trên mặt phẳng tọa độ và số điểm thuộc bao lồi:

- Sắp xếp tọa độ: $\theta(mlogm)$
- Gọi hai lần thuật toán Monotone Chain cho nửa trên và nửa dưới: $\theta(m)$
- Tính chu vi: $\theta(n)$

1.2 Các kĩ thuật sử dụng

- Tích có hướng
- Thuật toán tìm đường bao lồi

1.3 Mã giả

function **convex_hull**(A):
// Sắp xếp các điểm theo tọa độ (x, y)
Sắp xếp A tăng dần theo (x, y)
// Xây dựng đường bao lồi (nửa dưới và nửa trên)
Hull = []
for mỗi điểm $A_i \in A$:
    while (Hull có ít nhất 2 điểm **và** không tạo góc lồi):
        Loại bỏ điểm cuối trong Hull
    Thêm A_i vào Hull
for mỗi điểm $A_i \in A$ (duyệt ngược):
    Tương tự như trên để hoàn thành Hull
Loại bỏ các điểm trùng nhau ở đầu và cuối Hull
return Hull
function **calculate_perimeter**(Hull):
ChuVi = 0
for mỗi cạnh $(A_i, A_{i+1}) \in Hull$:
    ChuVi += độ dài cạnh (A_i, A_{i+1})
return ChuVi

2 Bài 2

Chi tiết cài đặt: https://github.com/doquanglucuit/CS112/blob/main/Nhom2-B2.cpp

2.1 Phân tích

Đề bài yêu cầu ta tính phần diện tích chung của hai đa giác lồi, để làm điều này ta cần xác định đa giác giao nhau, sau đó tính diện tích của đa giác.

- **Xác định đa giác**: các đỉnh thuộc đa giác nhau sẽ là các điểm giao nhau giữa hai đa giác và các đỉnh thuộc đa giác này nhưng lại nằm trong đa giác kia. Rồi sắp xếp các đỉnh lại theo thứ tự cùng chiều (hoặc ngược chiều) kim đồng hồ ta sẽ thu được đa giác.
- **Tính diện tích:** Áp dụng định lý Shoelace để tính diện tích dựa trên tọa độ các đỉnh.

Độ phức tạp: Gọi số đỉnh của hai đa giác và đa giác giao lần lượt là m, n, k:

- Xác định giao điểm giữa các cạnh của hai đa giác: $\theta(mn)$
- Sắp xếp đa giác: $\theta(klogk)$
- Tính diện tích: $\theta(k)$

2.2 Các kĩ thuật sử dụng

- Tìm giao điểm của các đoạn thẳng
- Kiểm tra điểm nằm trong đa giác
- Tạo đa giác giao
- Tính diện tích đa giác giao

2.3 Mã giả

```
function intersect_area(P, Q):
G = [] // Danh sách các điểm giao
// Tìm điểm giao giữa các cạnh của hai đa giác
for mỗi cạnh (P_i, P_{i+1}) của P:
    for mỗi cạnh (Q_j, Q_{j+1}) của Q:
        if (P_i, P_{i+1}) và (Q_j, Q_{j+1}) giao nhau:
            G.append(giao điểm)
// Thêm các đỉnh bên trong của hai đa giác
for mỗi đỉnh P_i ∈ P:
    if P_i nằm trong Q: G.append(P_i)
for mỗi đỉnh Q_j ∈ Q:
    if Q_j nằm trong P: G.append(Q_j)
// Sắp xếp các điểm giao theo thứ tự ngược chiều kim đồng hồ
centroid = tính trọng tâm của G
G = sắp_xếp_theo_góc_cực(G, centroid)
// Tính diện tích bằng công thức Shoelace
area = 0
n = số lượng điểm trong G
for x_1, y_1 và x_2, y_2 là 2 điểm liên tiếp trong đa giác
area += x_1 · y_2 − y_1 · x_2
return abs(area) / 2
```