



Hình học tính toán

Geometry

Nội dung

- Biểu diễn các đối tượng hình học cơ bản
- Tập điểm phủ các đoạn thẳng
- Tính diện tích đa giác không tự cắt, Xác định chiều của đa giác
- Kiểm tra đa giác có phải là đa giác lồi
- Tìm bao lồi tập điểm
- Kiểm tra điểm nằm trong đa giác lồi
- Tìm cặp điểm gần nhất
- Xác định hình tròn nhỏ nhất bao phủ tập điểm
- Các tâm của tam giác
- Tìm trọng tâm của đa giác lồi

Các đối tượng hình học

- ➡ Điểm
- ➡ Đoạn thẳng
- ➡ Vector
- ➡ Các hình cơ sở: hình tròn, hình vuông, đa giác đều
- ➡ Đa giác không tự cắt

Thuật toán CCW – check clockwise

```
1. //-1: clockwise, 1: counter-clockwise, 0: collinear
2. int ccw (Point P0, Point P1, Point P2) {
3.     dx1 = P1.x - P0.x;
4.     dx2 = P2.x - P0.x;
5.     dy1 = P1.y - P0.y;
6.     dy2 = P2.y - P0.y;
7.     if (dy1 * dx2 > dy2 * dx1) return -1;
8.     if (dx1 * dy2 > dy1 * dx2) return 1;
9.     if ((dx1 * dx2 < 0) || (dy1 * dy2 < 0)) return 1;
10.    if ((dx1 * dx1 + dy1 * dy1) < (dx2 * dx2 + dy2 * dy2))
11.        return -1;
12.    return 0;
13.}
```

Kiểm tra 2 đoạn thẳng cắt nhau

```
1. bool intersect (Vector2D l1, Vector2D l2)
2. {
3.     return (((ccw(l1.start, l1.end, l2.start) *
4.         ccw(l1.start, l1.end, l2.end)) <= 0) &&
5.         ((ccw(l2.start, l2.end, l1.start) *
6.         ccw(l2.start, l2.end, l1.end)) <= 0));
7. }
```

Tập điểm phủ các đoạn thẳng

Cho n đoạn thẳng trên trục hoành, đoạn thứ i có tọa độ các điểm đầu và cuối là a_i, b_i . Hãy xác định số lượng ít nhất các điểm cần chọn để mỗi đoạn thẳng đã cho chứa ít nhất một điểm trong số đã chọn.

Tổ chức dữ liệu

1. Vector x lưu trữ các điểm đã cho, mỗi điểm được ghi nhận là 1 bộ 3 số: tọa độ (a_i/b_i) , là điểm cuối $(1/0)$, số hiệu của đoạn
2. Mảng $flag[]$ để đánh dấu, $flag[i] = \text{true}$ nếu đoạn i đã có điểm đại diện, **false** trong trường hợp ngược lại
3. Mảng v để ghi nhận số hiệu các đoạn đang được xem xét

Giải thuật:

1. Sắp xếp x tăng dần tăng dần theo tọa độ,
2. Duyệt x từ đầu đến cuối:
 - Gặp điểm đầu: nạp số thứ tự của đoạn tương ứng vào v ,
 - Gặp điểm cuối: kiểm tra nếu đoạn tương ứng chưa có đại diện thì thêm 1 đại diện, đánh dấu có đại diện cho tất cả các đoạn đang được lưu trữ trong v , xóa v

Độ phức tạp: $O(n \log n)$

Tính diện tích đa giác không tự cắt

Cho đường gấp khúc khép kín n đỉnh không tự cắt, đỉnh thứ i có tọa độ thực (x_i, y_i) ($|x_i|, |y_i| \leq 10^6, i=1..n$). Các đỉnh được liệt kê theo đúng trình tự xuất hiện. Hãy tính diện tích của đa giác tạo bởi đường gấp khúc này.

Công thức:
$$S = \frac{1}{2} \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i)$$

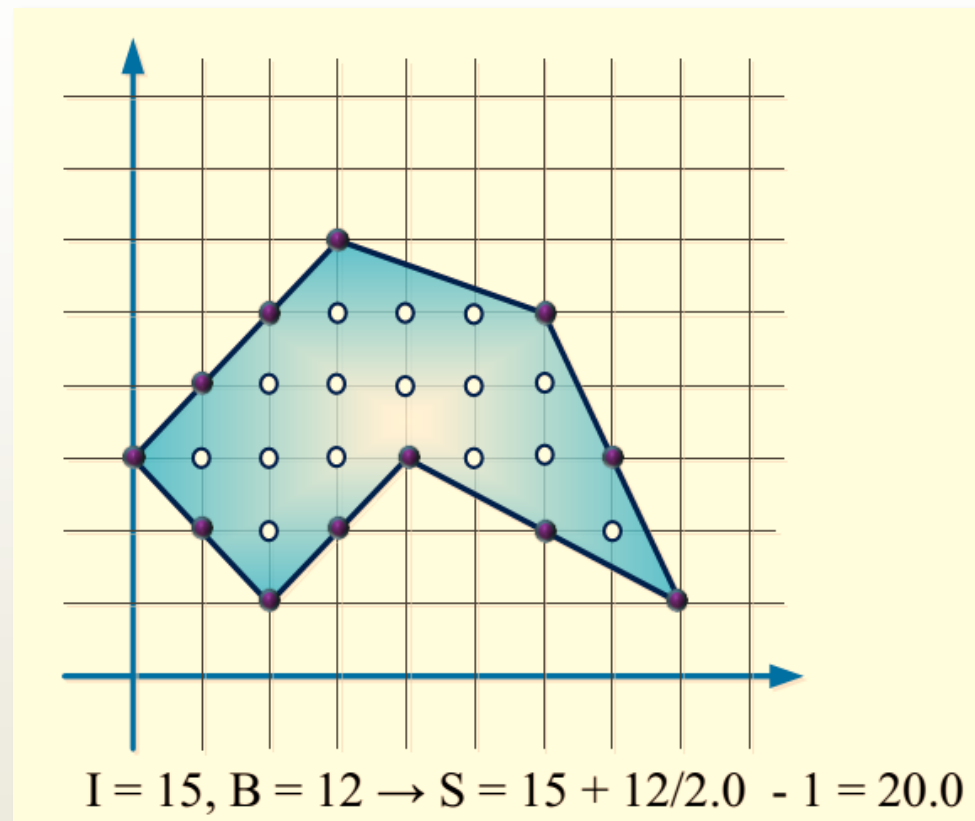
$S > 0$ ngược chiều kim đồng hồ, < 0 : cùng chiều, $= 0$: thẳng hàng

Định lý Pick

Xét đa giác diện tích khác 0, cạnh không tự cắt và trong mặt phẳng với hệ tọa độ Đềcác, tọa độ các đỉnh đều nguyên.

Gọi S là diện tích của đa giác, B – số điểm có tọa độ nguyên nằm trên cạnh của đa giác, I – số điểm trong có tọa độ nguyên

$$\text{Khi đó: } S = I + \frac{B}{2} - 1$$



Tìm bao lồi của tập điểm

Bài toán:

Cho n điểm trên mặt phẳng, điểm thứ i có tọa độ (x_i, y_i) . Hãy xác định bao lồi (hay nói cách khác – xác định đa giác lồi nhỏ nhất) chứa các điểm đã cho.

input	
9	
3	3
7	2
5	3
4	0
3	5
5	2
1	1
6	5
5	1

Output	
5	
1.00	1.00
3.00	5.00
6.00	5.00
7.00	2.00
4.00	0.00

Giải thuật Graham

Ý tưởng:

1. Xác định điểm THẤP nhất : P_0 (nếu có nhiều điểm thấp nhất, chọn điểm trái nhất)
2. Sắp xếp các điểm P_i còn lại theo trình tự tăng dần của góc tạo bởi tia (P_0, P_i) và Ox (góc tọa độ cực)
3. Trên danh sách đã sắp xếp, duyệt P_i từ P_2 đến cuối, nếu (P_{i-2}, P_{i-1}, P_i) không ngược chiều kim đồng hồ thì loại bỏ p_{i-1} khỏi danh sách

Độ phức tạp: $O(n \log n)$

Điểm nằm trong đa giác lồi

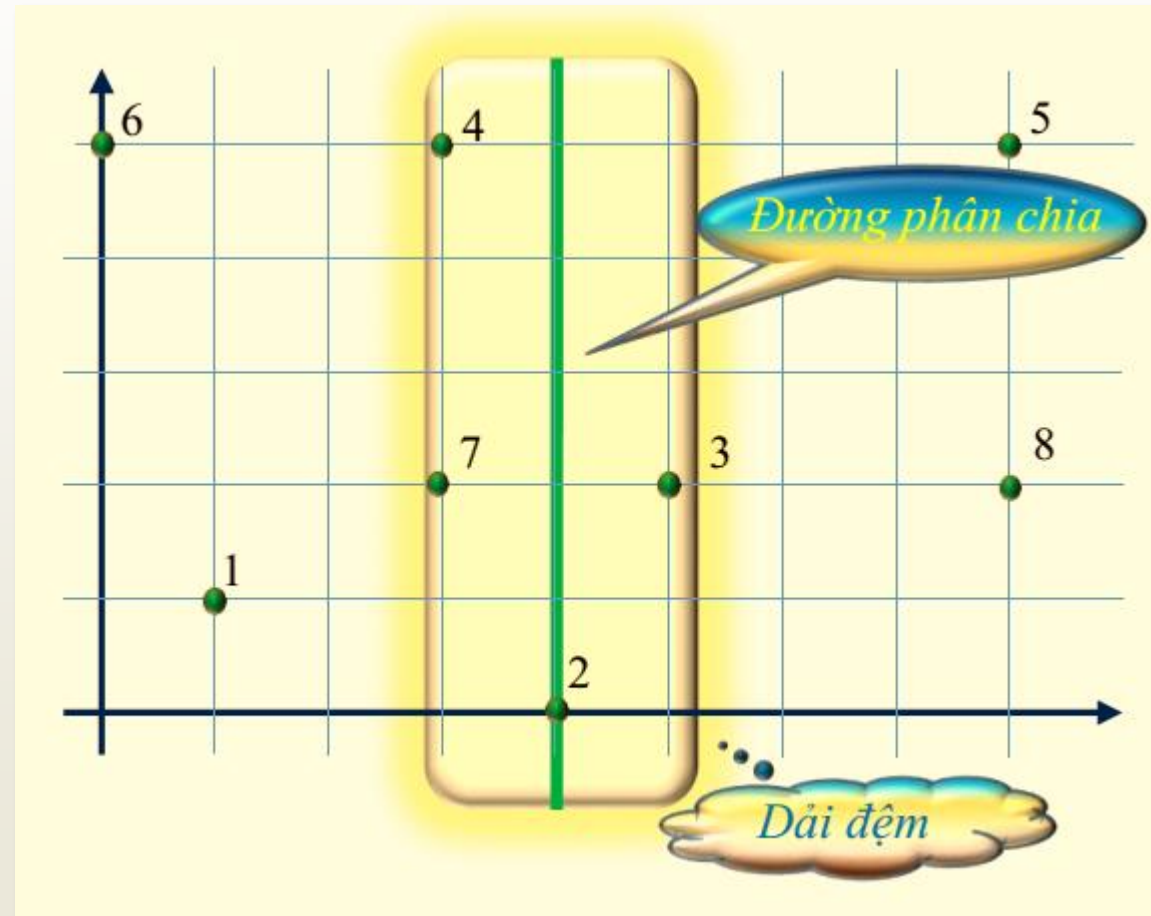
Ý tưởng:

1. Xác định điểm thấp nhất : P_0 (nếu có nhiều điểm thấp nhất, chọn điểm trái nhất)
2. Tìm nhị phân để xác định được 2 đỉnh P_i, P_{i+1} có góc tọa độ cực “kẹp” góc tọa độ cực của điểm p
3. Nếu xét chiều (P_i, p, P_{i+1}) hết luận \rightarrow nằm trong, nằm ngoài, thẳng hàng

Độ phức tạp: $O(\log n)$

Tìm cặp điểm gần nhất

➡ Giải thuật: chia để trị, kỹ thuật: đệ qui, độ phức tạp: $O(n \log n)$



Bài tập

1. [SPOJ: GEOPROB - One Geometry Problem](#)
2. [SPOJ: ADAPICK – Ada and Cucumber](#)
3. [SPOJ: ANTTT – The Ant](#)
2. [SPOJ: CLOPPAIR – Closest Point Pair](#)
4. [SPOJ: INOROUT – Inside or outside](#)

Tài liệu tham khảo

1. Robert Sedgewick – *Algorithm* – Addison Wesley 1984 (1946)
2. Steven & Felix Halim – *Competitive programming* – Lulu 2013
3. Nguyễn Thanh Tùng – *Kỹ thuật Lập trình V09* – 2015