

CHUYÊN ĐỀ HÌNH HỌC

A. PHẦN MỞ ĐẦU

I. Lý do chọn đề tài

Trong quá trình giảng dạy học sinh đội tuyển tham gia thi chọn học sinh giỏi các cấp. Tôi cũng như các học trò của mình vấp phải rất nhiều khó khăn. Do nguồn tài liệu trên mạng thì rất nhiều nhưng đa phần là tiếng anh, những tài liệu tiếng Việt thì thiếu tính hệ thống. Vì vậy tôi chọn viết đề tài này với mục tiêu có thể cung cấp đến người đọc những bài toán hình học cơ bản trong tin học cùng với phần cài đặt bằng ngôn ngữ C++.

Trong nội dung đề tài này tôi sẽ trao đổi với các thầy cô về những tính toán hình học cơ bản trong tin. Ứng dụng giải quyết một số bài toán hình học tiêu biểu. Các bài toán đều có đề bài, hướng dẫn giải, code và test.

II. Mục đích của đề tài

Nghiên cứu đề tài: “Một số bài toán hình học cơ bản trong Tin học và ứng dụng” nhằm giúp học sinh tiếp cận, làm quen và có thể giải các bài toán hình học trong lập trình.

Vận dụng giải các bài toán tin học có sử dụng kiến thức hình học, phát triển tư duy logic cho học sinh đồng thời nâng cao khả năng tự học của học sinh, tạo được hứng thú học tập môn Tin học.

III. Thời gian và địa điểm: suốt cả năm học, địa điểm: Trường THPT Chuyên Hạ Long

IV. Đóng góp mới về mặt thực tiễn: Đề tài có nhiều bài tập hay, có đủ cả test và solution vì vậy sẽ là một tài liệu phục vụ thiết thực cho việc giảng dạy đối tuyển.

B. PHẦN NỘI DUNG

I. LÝ THUYẾT

Thuật giải cho các bài toán hình học thường không khó, nó yêu cầu học sinh phải làm việc cẩn trọng, tỉ mỉ và tập trung vì nếu không rất có thể dẫn đến những sai lầm. Mặc dù kiến thức hình học các em được học trong bộ môn Toán. Nhưng trước tiên ta đi nhắc lại một số khái niệm cơ bản:

1. Biểu diễn hình học trên máy tính.

Trong chuyên đề này, tôi thống nhất cách biểu diễn những khái niệm cơ bản như điểm, đường thẳng, đa giác như sau:

- Điểm

```
struct Point
```

```
{
```

```
    int x;
```

```
    int y;
```

```
};
```

- Đường thẳng:

```
struct LineSegment
```

```
{
```

```
    P1, P2: Point;
```

```
};
```

- Véc tơ

```
Struct Vector
```

```
{
    P1, P2: Point;
};
```

- Đa giác:

```
Point Polygon[n];
```

Để thuận lợi thì khi biểu diễn đa giác ta nên thêm hai đỉnh ở đầu và cuối: đỉnh 0 bằng đỉnh n và đỉnh n + 1 bằng đỉnh 1.

2. Điểm, đoạn thẳng - đường thẳng, diện tích đa giác.

2.1. Quan hệ giữa các điểm - hàm CCW.

a) Khoảng cách giữa hai điểm:

```
Double Dist(Point P1, Point P2){
    return sqrt(abs(P1.x - P2.x)*(P1.x-P2.x) + (P1.y-P2.y)*(P1.y-P2.y)); //bỏ abs
}

int CCW(Point p, Point q, Point r)
{
    int val = (q.y - p.y) * (r.x - q.x) - (q.x - p.x) * (r.y - q.y);
    if (val == 0) return 0;
    return (val > 0)? 1: 2;
}
```

b) Hệ số góc của đường thẳng đi qua hai điểm:

Cho hai điểm . Nếu AB không song song với trục Oy thì hệ số góc của đường thẳng AB là:

c) Kiểm tra ba điểm có thẳng hàng hay không?

Giả sử chúng ta được cho trước ba điểm

- Viết phương trình đường thẳng AB, AC
- Giả sử ;
là các phương trình đã tối giản các hệ số và .
- Khi đó, A, B và C thẳng hàng khi và chỉ khi

d) Kiểm tra hai đoạn thẳng giao nhau hay không

```
bool onSegment(Point p, Point q, Point r)
{
    if (q.x <= max(p.x, r.x) && q.x >= min(p.x, r.x) &&
        q.y <= max(p.y, r.y) && q.y >= min(p.y, r.y))
        return true;
    return false;
}
```

```

}

bool doIntersect(LineSegment LS1, LineSegment LS2)
{
    Point p1, Point q1, Point p2, Point q2;
    p1 = LS1.p1; q1 = LS1.p2;
    p3 = LS2.p1; q2 = LS2.p2;
    int o1 = CCW (p1, q1, p2);
    int o2 = CCW (p1, q1, q2);
    int o3 = CCW (p2, q2, p1);
    int o4 = CCW (p2, q2, q1);
    if (o1 != o2 && o3 != o4)
        return true;
    if (o1 == 0 && onSegment(p1, p2, q1)) return true;
    if (o2 == 0 && onSegment(p1, q2, q1)) return true;
    if (o3 == 0 && onSegment(p2, p1, q2)) return true;
    if (o4 == 0 && onSegment(p2, q1, q2)) return true;
    return false;
}

```

e) Diện tích của đa giác :

```

int Area (Point p[n]){
    int s;
    p[n] = p[0];
    for(int i = 0 ; i <= n ; i++)
        s=s+(p[i].x*p[i+1].y-p[i].y*p[i+1].x) ;
    return abs(s/2) ;
}

```

2.2. Điểm trong đa giác:

a) Bài toán : Cho một đa giác không tự cắt, hãy kiểm tra xem một điểm có nằm trong đa giác hay không?

Tư tưởng cho bài toán này nói qua thì rất đơn giản và dễ hiểu: Từ điểm cần kiểm tra ta kẻ một tia bất kỳ, nếu tia đó giao với đa giác một số chẵn lần thì có nghĩa là nó nằm ngoài đa giác, một số lẻ lần thì nó nằm trong đa giác.

b) Cài đặt:

```

bool isInside(Point polygon[], int n, Point p)
{

```

```

if (n < 3) return false;
Point extreme = {INF, p.y};
int count = 0, i = 0;
do
{
    int next = (i+1)%n;
    if (doIntersect(polygon[i], polygon[next], p, extreme))
    {
        if (orientation(polygon[i], p, polygon[next]) == 0)
            return onSegment(polygon[i], p, polygon[next]);
        count++;
    }
    i = next;
} while (i != 0);
return count&1;
}

```

2.3. Bao lồi:

- a) bài toán: Cho một tập các điểm trên mặt phẳng. Hãy xác định bao lồi của tập điểm.
- b) Hướng dẫn thuật toán : Tham khảo chi tiết các thuật toán bao lồi tại <http://vnoi.info/wiki/translate/wcipeg/Convex-Hull>
- b) Cài đặt : <https://www.dropbox.com/s/n6p1o9gcfzmk756/convex%20hull.rar?dl=0>

II. BÀI TẬP LUYỆN TẬP

2.1. BÀI TẬP ỨNG DỤNG HỆ SỐ GÓC CỦA ĐƯỜNG THẲNG

Trong phần này, Tôi sẽ trình bày một số bài toán ứng dụng hệ số góc để giải bài toán hình học rất hiệu quả.

Bài 1. QBPOINT – Bộ ba điểm thẳng hàng

Trong các cuộc thi tin học, sự xuất hiện của những bài toán hình học làm đội tuyển CBQ khá lúng túng. Do đó thầy Thạch quyết định cho đội tuyển luyện tập các bài toán hình học. Bắt đầu từ điểm, thầy đưa ra bài toán.

Cho n điểm phân biệt trong mặt phẳng Oxy, hãy đếm số bộ 3 điểm thẳng hàng.

Dữ liệu đầu vào:

- Dòng thứ nhất ghi là số điểm trên mặt phẳng.
- dòng tiếp theo, mỗi dòng ghi tọa điểm của một điểm.

Kết quả:

- Ghi ra một số duy nhất là số bộ ba điểm thẳng hàng.

Ví dụ:

QBPOINT . INP	QBPOINT . OUT
6	3
0 0	
0 1	
0 2	
1 1	
2 0	
2 2	

Giới hạn: $1 \leq N \leq 2000$, Tọa độ các điểm có trị tuyệt đối không quá 10000.

Nguồn: <https://vn.spoj.com/problems/QBPOINT>

Hướng dẫn thuật toán:

- Cố định một điểm (giả sử điểm A) rồi tính hệ số góc của đường thẳng đi qua điểm A và tất cả các điểm còn lại.
- Sắp xếp lại mảng hệ số góc
- Đếm số lượng số hệ số góc bằng nhau, nếu có số hệ số góc bằng nhau thì tăng kết quả lên .
- Giả code:

Tính mảng hệ số góc slops[];

Sắp xếp lại mảng slops[];

Đếm trên mảng slops số lượng số hệ số góc giống nhau

Cnt += .

Code đầy đủ: <https://www.dropbox.com/s/j4qu3mo3hreh5jr/BQPOINT.rar?dl=0>

Cảm nhận: Bài toán kiểm tra ba điểm thẳng hàng là một bài toán cơ bản trong toán học và có nhiều ứng dụng trong tin học. Sử dụng hệ số góc để kiểm tra ba điểm có thẳng hàng hay không phải là một phương pháp phổ biến trong toán học, tuy nhiên trong các bài toán Tin học thì

Bài 2. BALLGMVN – VOI 2014 – Trò chơi với những viên bi

Trong một hội thi Ballgame, ban tổ chức chuẩn bị một bàn lớn. Trên mặt bàn có n bi xanh đánh số từ 1 đến n và n bi đỏ đánh số từ n + 1 đến 2n. Mỗi trận đấu, các vận động viên sẽ chơi luân phiên nhau. Đến lượt chơi của mình, Hùng cần tìm 3 bi mà vị trí của chúng là thẳng hàng hanu và sao cho trong số đó có hai bi đỏ và 1 bi xanh (khi đó ăn được một bi đỏ), hoặc là có hai bi xanh và 1 bi đỏ (khi đó được ăn 1 bi xanh).

Yêu cầu: Cho biết tọa độ trên mặt phẳng tọa độ Đề-các của vị trí và màu của các bi hiện tại trên bàn, bạn hãy giúp Hùng chọn 3 bi để chơi.

Dữ liệu đầu vào:

- Dòng đầu ghi số nguyên dương n.

- Dòng thứ i trong số n dòng tiếp theo ghi hai số nguyên là hoành độ và tung độ trên mặt phẳng tọa độ Đề-các của vị trí đặt bi xanh với chỉ số
- Dòng thứ i trong số n dòng cuối cùng ghi hai số nguyên là hoành độ và tung độ trên mặt phẳng tọa độ Đề-các của vị trí đặt bi đỏ với chỉ số $n + i$.

Hoành độ và tung độ không vượt quá , vị trí các bi là đôi một phân biệt.

Kết quả:

- Ghi ra 3 chỉ số của các viên bi mà Hùng cần chọn, nếu không thể chọn được 3 bi nào, ghi ra -1. Nếu có nhiều đáp án, ghi ra một đáp án bất kì.

Ví dụ:

BALLGMVN . INP	BALLGMVN . OUT
3	1 2 4
1 1	
2 2	
4 9	
3 3	
6 20	
8 100	

Giới hạn

- 30% số test có $n \leq 2$;
- 30% số test khác có $n \leq 100$.
- 40% số test còn lại có $n \leq 1000$.

Nguồn: <https://vn.spoj.com/problems/BALLGMVN/>

Hướng dẫn thuật toán:

- Chia bài toán thành hai trường hợp, trường hợp 1 xét những đường thẳng đi qua hai bi xanh một bi đỏ, trường hợp hai xét đường thẳng đi qua hai bi đỏ một bi xanh.
- Trường hợp: Xét đường thẳng đi qua hai xanh một bi đỏ:
- Cố định bi đỏ (giả sử là A)
- Xác định mảng hệ số góc của đường thẳng đi qua bi đỏ A và mọi bi xanh.
- Sắp xếp lại mảng hệ số góc, từ đó kiểm tra trên mảng hệ số góc có giá trị nào xuất hiện ít nhất hai lần thì bài toán tồn tại và ghi ra chỉ số các điểm, nếu không thì kiểm tra trường hợp hai.

Độ phức tạp của thuật toán là .

Chi tiết code tham khảo tại: <https://www.dropbox.com/s/wt5z1vf91cwxsnd/BALLGMVN.cpp?dl=0>

Bài 3: SWAGE

Phú ông có khu vườn trồng cây ăn trái, cây thứ i được trồng tại vị trí có tọa độ (x_i, y_i) . Phú ông thuê Bờm quản lý và chăm sóc vườn cây đối lại Bờm được thu hoạch một số cây trong vườn cho riêng mình.

Phú ông biết Bờm không được học nhiều nên ra điều kiện làm khó là Bờm chỉ được thu hoạch những cây nằm trên cùng một đường thẳng. Bờm không biết làm thế nào để có thể thu hoạch được nhiều cây nhất mà vẫn thỏa điều kiện của Phú ông đưa ra.

Yêu cầu: Bạn hãy giúp Bờm xác định số cây nhiều nhất mà Bờm được thu hoạch.

Input: SWAGE.INP

- Dòng thứ nhất ghi số N là số cây trong vườn.
- Dòng thứ i trong N dòng tiếp theo ghi tọa độ của cây thứ i.

Output: SWAGE.OUT

- Một số duy nhất là số cây nhiều nhất mà Bờm được thu hoạch.

Giới hạn: $1 \leq N \leq 2000$. Tọa độ các điểm có trị tuyệt đối không quá 15000.

Ví dụ:

SWAGE . INP	SWAGE . OUT
6	3
0 0	
0 1	
0 2	
1 1	
2 0	
2 2	

Ràng buộc:

Subtask 1 (50% điểm): Giả thiết $N \leq 100$

Subtask 2 (50% điểm): Giả thiết $N \leq 2000$

Hướng dẫn thuật toán:

{

Tính mảng hệ số góc slopes của đường thẳng đi qua điểm i và j.

Sắp xếp lại mảng slopes

Tìm giá trị xuất hiện nhiều nhất trên mảng slopes

Cập nhật lại max

}

Độ phức tạp của thuật toán $O()$

Chi tiết code và test tham khảo tại:

<https://www.dropbox.com/s/23vepv2pseni92r/SWAGE.rar?dl=0>

Bài 4: LINES - Game of Lines (Trò chơi của những đường thẳng)

John đã thách thức Bessie một trò chơi như sau: John có một con cái bảng với các điểm được đánh dấu tại N vị trí phân biệt. Điểm thứ i có tọa độ nguyên

Bessie có thể ghi một điểm bằng cách chọn hai điểm và vẽ một đường thẳng giữa hai điểm đó; tuy nhiên, Bessie không được phép vẽ một đường thẳng nếu cô ấy đã từng vẽ một đường thẳng song song với nó trước đó.

Yêu cầu: Hãy lập trình tính giúp Bessie số điểm lớn nhất mà cô ấy có thể rành được.

Dữ liệu:

- Sẽ có nhiều test. Với mỗi test, dòng đầu tiên ghi số N, và mỗi dòng tiếp theo của N dòng ghi một cặp số nguyên. File kết thúc với trường hợp

Kết quả:

- Mỗi dòng ghi kết quả của mỗi trường hợp.

Ví dụ:

Input:	Output:
4 -1 1 -2 0 0 0 1 1 0	4

Bessie có thể vẽ các đường thẳng với hệ số góc: -1, 0, $\frac{1}{2}$, and 1.

Nguồn: <https://www.spoj.com/problems/LINES/>

Hướng dẫn thuật toán:

- Duyệt đến mọi đường thẳng được tạo bởi hai điểm bất kì
- Tính hệ số góc của đường thẳng đi qua hai điểm đó
- Dùng map để lưu hệ số góc.
- Kết quả của bài toán bằng kích thước của map.

Tham khảo code chi tiết tại: <http://spojolutionsall.blogspot.com/>

Bài 5. Line (Đường thẳng)

Trong tọa độ đề các, cho N đoạn thẳng được xác định bởi hai điểm, N đoạn thẳng này tạo thành N đường thẳng. Đếm số lượng tất cả các cặp gồm 2 đường thẳng được tạo thành chúng vuông góc và cặp đoạn thẳng đó có độ dài bằng nhau.

Dữ liệu vào:

- Dòng đầu là số nguyên N
- N dòng tiếp theo, mỗi dòng gồm 4 số nguyên x1, y1, x2, y2 lần lượt là tọa độ của 2 đầu mút

Kết quả:

- Một dòng duy nhất là số cặp tìm được thỏa mãn yêu cầu.

Line.inp	Line.out
----------	----------

6	6
1 1 0 2	
2 1 2 0	
1 1 1 2	
1 1 2 1	
2 2 3 2	
-1 -1 0 -1	

Ràng buộc:

- 50% test : $N < 10^3$
- 100% test: $N < 10^5$

Hướng dẫn thuật toán:

- Với mỗi đường thẳng được tạo bởi đoạn thẳng AB ta tính độ dài đoạn thẳng AB và véc tơ (dùng pair<int, pair<int, int>> để lưu dữ liệu) lưu vào mảng vt.
- Sắp xếp lại mảng vt.
- Với mỗi phần tử trong mảng vt có dạng (d, (a, b))
- Dùng tìm kiếm nhị phân để đếm số lượng phần tử của mảng vt có dạng (d, (-b, a)) => tăng kết quả của bài toán.

Tham khảo chi tiết code và test của bài toán tại:

<https://www.dropbox.com/s/5we9xw4jdt078cf/Lines.rar?dl=0>

Bài 6. Góc nhìn (viewangle) (nguồn: codeforces.com)

Gần đây ở Flatland đã giới thiệu một kiểu kiểm tra mắt mới cho các tài xế tham gia thi lấy giấy phép lái xe. Việc kiểm tra này như sau: Trong mặt phẳng có các hình nộm đứng trên đó. Bạn cần phải nói giá trị của góc nhìn nhỏ nhất với đỉnh tại gốc tọa độ và tất cả các hình nộm đứng bên trong hoặc trên cạnh của góc này.

Cho trước tọa độ của các hình nộm trong mặt phẳng, bạn hãy viết chương trình tính giá trị nhỏ nhất của góc nhìn này.

Dữ liệu:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) – số lượng hình nộm.
- n dòng sau, mỗi dòng chứa 2 số nguyên x_i, y_i ngăn cách nhau bởi một dấu cách là tọa độ hình nhân thứ i ($-10^9 \leq x_i, y_i \leq 10^9$). Dữ liệu đảm bảo không có hình nộm nào đặt tại gốc tọa độ nhưng có thể có nhiều hơn một hình nộm ở cùng một vị trí.

Kết quả:

- Ghi ra một số thực với độ chính xác 6 chữ số thập phân sau dấu phẩy là kết quả của bài toán.

Ví dụ:

Viewangle.inp	Viewangle.out
2 2 0 0 2	90.000000
3 2 0 0 2 -2 2	135.000000

4 2 0 0 2 -2 0 0 -2	270.000000
2 2 1 1 2	36.869898

Hướng dẫn thuật toán:

- Với mỗi điểm $P[i]$ chúng ta tính được $\tan[i]$ là tan của góc tạo bởi tia $OP[i]$ và chiều dương của trục Ox.
- Sắp xếp lại dãy $\{\tan[i]\}$ theo chiều tăng dần.
- Kết quả là:

$\text{double res} = (g[n] - g[1]) / M_PI * 180.0;$

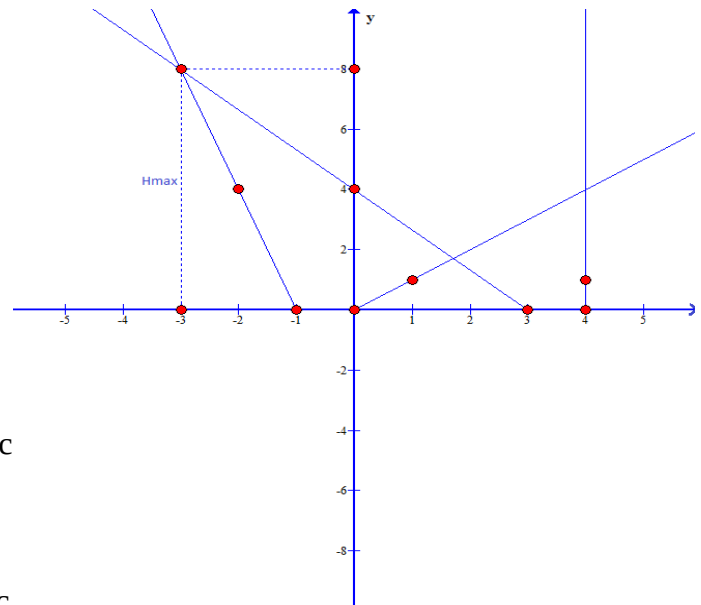
$\text{for}(i, 2, n) \text{ res} = \min(\text{res}, 360.0 - (g[i] - g[i-1]) / M_PI * 180.0);$

Test + code: <https://www.dropbox.com/s/xks5ycffu7guwdh/viewangle.rar?dl=0>

Bài 7. Giao điểm cao nhất

Trong lễ hội bắn pháo hoa năm nay. Tiết mục trình diễn ánh sáng trong lễ khai mạc của chủ nhà Đà Nẵng, có N tia laser được chiếu lên trời nhờ vào các đèn chiếu có công suất rất lớn, vì vậy các tia laser này có thể đi rất xa. Các tia laser được chiếu lên nằm trên cùng một mặt phẳng thẳng đứng nên nếu không có 2 tia laser nào song song với nhau thì 2 tia laser hoặc là cắt nhau hoặc là không cắt nhau. Các tia laser được biểu diễn bởi 3 số nguyên:

Một số là tọa độ X ở dưới đất của ngọn đèn chiếu, 2 số còn lại là tọa độ của một điểm nào đó thuộc tia laser này. Biết rằng không có 2 tia laser nào song song với nhau và cũng không có tia Laser nào trùng với mặt đất (mặt đất được coi như là đường thẳng $y = 0$). Không có đèn chiếu nào đặt cùng một vị trí trên trục tọa độ.



Yêu cầu: Các nhà tổ chức buổi trình diễn muốn bạn cho biết với các tia laser sẽ được chiếu lên trời như trong kế hoạch thì 2 tia laser nào cắt nhau tại điểm cao nhất.

Dữ liệu: Từ tệp văn bản 'H_MAX.INP' gồm:

- Dòng 1 ghi số nguyên N là số tia laser
- N dòng, mỗi dòng ghi 3 số nguyên với ý nghĩa là tia laser thứ i đi qua 2 điểm và .

Kết quả: Ghi ra tệp văn bản 'H_MAX.OUT' như sau:

- Trong trường hợp không có 2 tia laser nào cắt nhau thì ghi ra duy nhất một số - 1.
- Nếu tồn tại 2 tia Laser cắt nhau thì ghi số H_{\max} là độ cao lớn nhất giao điểm của 2 tia laser. (h_{\max} được ghi với độ chính xác 3 chữ số sau dấu phẩy).

Ví dụ:

H_MAX.INP	H_MAX.OUT
2 -1 -2 4	-1
4 -1 -2 4 0 1 1 3 0 4 4 4 1	8.000

Thuật toán:

- **Thuật toán 1:**

- Duyệt chọn ra 2 tia, tìm điểm giao và tính chiều cao. Trong các chiều cao đó chọn ra cặp 2 tia có chiều cao lớn nhất. Độ phức tạp thuật toán $\sim O(N^2)$. Làm cách này thí sinh có không quá 40% số điểm của bài thi.

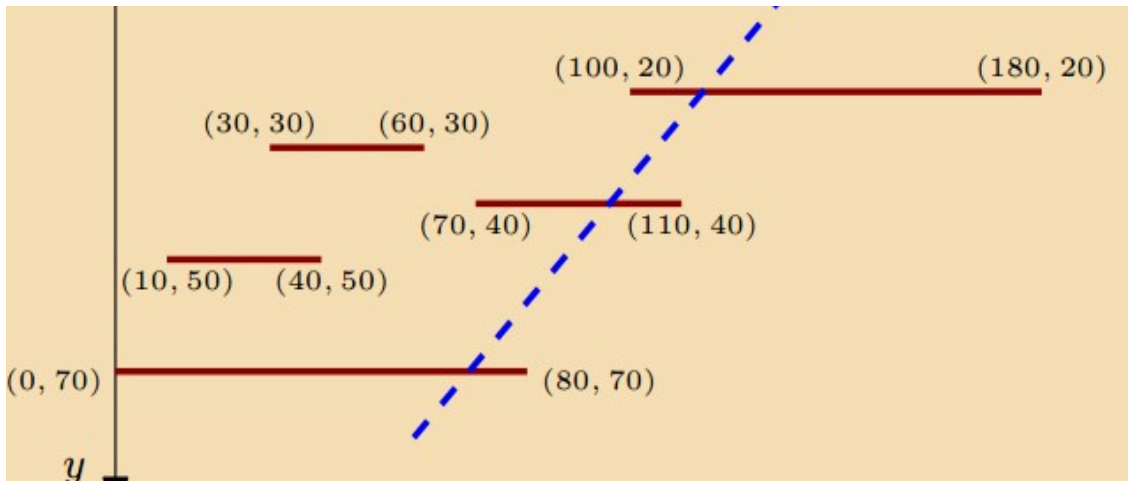
- **Thuật toán 2:**

- Bạn nên chú ý tới giá trị N – số tia laser, nó rất lớn ($N \leq 10^5$). Ta gọi góc tạo bởi một tia laser với trục nằm ngang (chiều dương là chiều ngược chiều kim đồng hồ) là *góc của tia đó*.
- Nếu tia i và tia j giao nhau (với điều kiện góc của tia j là góc có giá trị gần với góc của tia i nhất) thì độ cao giao điểm của hai tia đó sẽ lớn hơn độ cao giao điểm của tia i với các tia khác tia j . Từ đó ta có thuật toán:
- Sắp xếp các tia theo thứ tự tăng dần (hoặc giảm dần) của góc tạo bởi các tia.
- Duyệt 1 lần theo thứ tự và xét chiều cao giao điểm của hai tia cạnh nhau theo thứ tự mảng đó.
- Lưu lại giá trị lớn nhất tìm được.
- Độ phức tạp thuật toán chủ yếu nằm ở bước sắp xếp các tia $\sim O(N \log N)$.

Code+test: https://www.dropbox.com/s/fr2aj1y7ci5ut7i/H_MAX.rar?dl=0

Bài 8. KHOANGDAU- ACM/ICPC 2016-PROBLEM G:

Công ty khai thác dầu SÁNG TẠO đang cố gắng tìm cách đổi mới phương pháp khai thác dầu để được lợi nhuận tối đa. Quá trình khảo sát địa chất cho thấy các mỏ dầu nằm thành từng lớp với độ dài khác nhau trên bề mặt trái đất. Công ty muốn tạo một đường khoang (đường đứt khoảng trên hình vẽ) là đường thẳng sao cho xuất phát từ bề mặt trái đất đi qua được nhiều mỏ dầu nhất.



Hình 1. Hình mô phỏng từ ví dụ 1

Hiệu quả của đường khoang là tổng số dầu thu được từ những mỏ dầu mà đường khoang đi qua. Đường khoang vẫn có thể thu được dầu khi đi qua điểm đầu hoặc điểm cuối của mỏ dầu. Sản lượng của mỏ dầu chính bằng độ dài của mỏ dầu.

Hãy viết chương trình giúp công ty SÁNG TẠO xác định lượng dầu tối đa mà công ty thu được với 1 đường khoang.

Dữ liệu:

- Dữ liệu vào từ file KHOANGDAU.INP gồm
- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 2000$), là số lượng mỏ dầu.
- N dòng tiếp theo mỗi dòng mô tả cho một mỏ dầu, gồm 3 số nguyên x_0 , x_1 , y xác định tọa độ điểm đầu và cuối của mỏ dầu là (x_0, y) và (x_1, y) .

Kết quả:

- Ghi ra file KHOANGDAU.OUT một số nguyên duy nhất là sản lượng dầu tối đa mà đường khoang thu được.

VÍ DỤ:

KHOANGDAU.INP	KHOANGDAU.OUT
5 100 180 20 30 60 30 70 110 40 10 40 50 0 80 70	200
3 50 60 10 -42 -42 20	25

Hướng dẫn thuật toán (<https://icpc.baylor.edu/worldfinals/problems/icpc2016.pdf>):

Đây là bài cơ bản về tập hợp trong hình học.

Nếu tất cả các mỏ dầu đều nằm trên 1 đường ngang thì đường cắt chỉ lấy được dầu từ 1 mỏ dầu và chắc chắn phương án tối ưu là lấy từ mỏ dầu có trữ lượng lớn nhất. Ngược lại, thì tối thiểu đường cắt chúng ta có thể đi qua tối thiểu là 2 mỏ dầu bằng cách di chuyển đường cắt (sang trái, sang phải, xoay).

Duyệt toàn bộ với độ phức tạp $O(n^3)$. Trường hợp mỗi cặp điểm không nằm trên 1 hàng thì đường khoan sẽ đi qua hai điểm, kiểm tra tất cả các mỏ dầu mà đường khoan đi qua. Tuy nhiên cách này khá chậm.

Để cải tiến tốc độ ta sử dụng một đường quét xoay vòng, chọn điểm P bất kì mà đường cắt sẽ đi qua (chúng ta sẽ thực hiện lặp lại với tất cả các điểm P được chọn). các điểm không nằm trên cùng một mỏ dầu chứa P và đường thẳng đi qua P sẽ tạo thành 1 góc, sắp xếp theo độ dốc của các góc tạo bởi điểm P và các điểm khác với đường cắt. Nếu gặp điểm đầu của mỏ dầu, ta thêm giá trị trữ lượng dầu vào kết quả hiện tại. Nếu ta gặp điểm thứ hai, ta trừ đi trữ lượng dầu trong kết quả hiện tại. Giải thuật này chạy trong $O(n^2 \log n)$.

Khi cài đặt, cần cẩn thận trong việc sắp xếp dữ liệu theo độ dốc, xử lý đường quét, xây dựng đúng thủ tục thêm mỏ dầu và bớt mỏ dầu ra khỏi kết quả hiện tại. Đặc biệt cần lưu trữ độ dốc là những cặp số nguyên, nó khác với số thực.

Tham khảo code chi tiết để hiểu thuật toán:

Nguồn: <https://www.spoj.com/problems/FN16OIL/>

Kết luận:

Hệ số góc của đường thẳng kiểm tra ba điểm thẳng hàng là một phương pháp rất hiệu quả. Giúp chúng ta giải được nhiều bài toán hình học trong tin học một cách hiệu quả.

2.2. BÀI TẬP ỨNG DỤNG BAO LỒI CỦA TẬP ĐIỂM

Bài 1. THỪA ĐẤT LỚN NHẤT (Nguồn: Thầy Lê Minh Hoàng)

Bờm lại thắng Phú ông trong một cuộc đánh cược và theo thỏa thuận từ trước, Phú ông buộc phải cho Bờm một thửa đất trong phần đất đai rộng lớn của mình. Bản đồ phần đất của Phú ông có thể coi là một mặt phẳng với hệ trục tọa độ Descartes vuông góc Oxy trên đó đánh dấu n ($n \geq 3$) cột mốc hoàn toàn phân biệt và không đồng thời thẳng hàng, cột mốc thứ i có tọa độ (x_i, y_i) . Bờm được chọn ba cột mốc trong số đó để nhận thửa đất có dạng hình tam giác có ba đỉnh là vị trí ba cột mốc được chọn.

Yêu cầu: Hãy giúp Bờm chọn ba cột mốc để nhận được thửa đất có diện tích lớn nhất.

Dữ liệu: Vào từ file văn bản TRILAND.INP

- Dòng 1 chứa số nguyên dương n ($3 \leq n \leq 3000$)
- n dòng tiếp theo, dòng thứ i chứa hai số nguyên x_i, y_i ($\forall: |x_i|, |y_i| \leq 10^9$) cách nhau bởi dấu cách

Kết quả: Ghi ra file văn bản TRILAND.OUT diện tích của thửa đất Bờm sẽ nhận theo phương án tìm được. Diện tích này phải ghi dưới dạng số thực với đúng 1 chữ số sau dấu chấm thập phân.

Ví dụ:

TRILAND . INP	TRILAND . OUT
8 1 1 1 2 1 5 2 2 3 1 3 3 4 1 6 6	11.5

Hướng dẫn:

- Dễ dàng nhận thấy để có được diện tích tam giác lớn nhất thì 3 đỉnh phải thuộc tập các đỉnh là bao lồi
- Giả sử ta gọi các đỉnh được chọn tạo thành tam giác theo thứ tự là 1,2,3
- Sau khi có được tập đỉnh bao lồi nhận thấy nếu cố định 2 đỉnh (1 và 3) và đỉnh còn lại (đỉnh 2) được duyệt từ trái -> phải thì diện tích tam giác sẽ tăng đến cực đại và giảm dần đến cuối tập đỉnh
- Sử dụng công thức “dây giầy” để tính diện tích tam giác.
- Vậy từ đó ta chỉ cần cố định 2 đỉnh đầu và cuối và đỉnh ở giữa sẽ xét trong khoảng giữa 2 đỉnh trên . Khi diện tích đạt giới hạn ta ngừng xét đỉnh giữa và thay đỉnh số 3 thành đỉnh số 3 hiện tại +1 và tiếp tục xét đỉnh giữa . Cứ thế và lấy MAX.

Code + test: <https://www.dropbox.com/s/o6v41t0misn95p5/Triland.rar?dl=0>

Bài 2. METERAIN - Mưa thiên thạch (Nguồn: SPOJ)

Phú ông nhận được thông tin về một trận mưa thiên thạch sắp ập xuống trái đất. Không những thế, Phú ông còn biết tọa độ của vị trí điểm rơi của mỗi một thiên thạch. Phú ông nhờ Cuội xác định xem có bao nhiêu thiên thạch có thể rơi xuống cánh đồng của ông ta. Cánh đồng của Phú ông có dạng một hình đa giác lồi được xác định bởi danh sách các đỉnh được liệt kê theo thứ tự ngược chiều kim đồng hồ.

Yêu cầu: Xác định xem trong tập cho trước các điểm rơi của thiên thạch, có bao nhiêu điểm nằm trong cánh đồng của Phú ông. Các điểm nằm trên biên của cánh đồng không được tính là điểm nằm trong cánh đồng.

Dữ liệu:

- Dòng đầu tiên là số nguyên n ($3 \leq n \leq 5000$) là số đỉnh của đa giác lồi mô tả cánh đồng của Phú ông.
- Mỗi dòng trong n dòng tiếp theo chứa cặp tọa độ của một đỉnh của đa giác lồi.
- Dòng tiếp theo là số nguyên m ($2 \leq m \leq 5000$) - số thiên thạch rơi xuống.

- Mỗi dòng trong số m dòng cuối cùng chứa 2 số là tọa độ điểm rơi của một thiên thạch. Các tọa độ là các số nguyên có trị tuyệt đối không quá 10^6 .

Kết quả:

- Ghi ra m dòng, mỗi dòng tương ứng với 1 điểm rơi của thiên thạch. Ghi "YES" nếu điểm rơi của thiên thạch nằm trong cánh đồng và ghi "NO" nếu trái lại.

METERAIN.INP	METERAIN.OUT
4	NO
2 4	NO
8 4	YES
6 8	YES
4 6	
4	
3 5	
4 7	
5 5	
6 7	

Hướng dẫn:

- Nhận thấy để kiểm tra thiên thạch có nằm trong mảnh ruộng không ta chỉ cần thực hiện phép quay giữa 2 đỉnh liên tiếp với tọa độ của thiên thạch
- Do tọa độ của mảnh ruộng được liệt kê theo ngược chiều kim đồng hồ nên ta phải đảo ngược lại với các phép quay điều hướng.

Test: <https://vnoi.info/problems/METERAIN/>

Bài 3: ĐA GIÁC KHÔNG TỰ CẮT (Nguồn: Thầy Lê Minh Hoàng)

- Cho n điểm trên mặt phẳng, trong đó có ít nhất 3 điểm không thẳng hàng. Từ các điểm trong n điểm trên ta có thể dựng được rất nhiều đa giác không tự cắt. Trong bài toán này sẽ quan tâm đến các đa giác không tự cắt và diện tích của chúng.

Yêu cầu: Cho n điểm và số nguyên k , hãy tìm ít nhất ba điểm và không quá k điểm trong n điểm trên, sau đó dựng một đa giác không tự cắt từ các điểm được chọn để được đa giác có diện tích là lớn nhất.

Dữ liệu: Vào từ file văn bản POLY.INP có dạng:

- Dòng đầu chứa hai số nguyên dương n, k ($3 \leq n \leq k \leq 200$)
- k dòng sau, dòng thứ i gồm 2 số nguyên x_i, y_i ($|x_i|, |y_i| \leq 10^6$) là tọa độ điểm thứ i

Kết quả: Đưa ra file văn bản POLY.OUT một số thực với 2 chữ số sau dấu chấm là diện tích đa giác lớn nhất dựng được

POLY . INP	POLY . OUT
4 3 0 0 2 0 0 3	3 . 00

2 2	
-----	--

Hướng dẫn:

- Dễ dàng nhận thấy với $k=3$ thì chúng ta phải tìm diện tích tam giác lớn nhất tạo bởi 3 điểm
- Vẫn công việc đầu tiên là tìm tập các đỉnh bao lồi rồi sau đó tuần tự xét tập 3 đỉnh và dùng công thức tính diện tích dây giầy để tính
- Với bài toán $k=4$ ta nhận thấy giả sử có 3 đỉnh 1,2,3 tạo thành diện tích tam giác lớn nhất trong n đỉnh và khi thêm đỉnh số 4 vào sao cho 1, 3, 4 là diện tích tam giác lớn thứ 2 ta tìm được thì diện tích đa giác lớn nhất của bài toán sẽ là 1,2,3,4.
- Tương tự với $k=5$ thì diện tích lớn nhất sẽ là kết quả của $k=4$ và thêm 1 đỉnh.
- Vậy nhận thấy bài toán sẽ trở thành bài toán QHĐ cơ bản trên tập đỉnh bao lồi.

Lưu ý: khi sử dụng công thức dây giầy để tính diện tích đa giác ta cần phải lưu ý khi thêm đỉnh mới vào sẽ phát sinh một số tiểu tiết trong công thức phải xử lí.

Code + test: <https://www.dropbox.com/s/i5jln9bet8a45y3/Poly-Converhull.rar?dl=0>

Bài 4: RAOVUON - Rào Vườn (nguồn: SPOJ)

Bạn Minh Đức là chủ một vườn cây ăn quả lớn ở miền Nam. Nếu nhìn từ trên cao xuống, các gốc cây giống như các điểm trên mặt phẳng tọa độ là mặt đất. Đã nhiều năm rồi, bạn Minh Đức không được bội thu do nạn đạo tặc. Do vậy, năm ngoái bạn Minh Đức quyết tâm rào khu vườn của mình lại. Để làm được điều này, bạn Minh Đức chẳng đường rào theo các gốc cây để tạo thành một đa giác bao kín vườn cây. Do tính keo kiệt và chi phí của đường rào là rất đắt, bạn Minh Đức đã tính toán chi li để đường rào có chu vi nhỏ nhất có thể. Chắc các bạn cũng biết đây là bài toán tin cơ bản : tìm bao lồi nhỏ nhất của một tập điểm. Tuy nhiên, năm nay bạn Minh Đức cũng không thu hoạch được thêm nhiều. Lý do là có một số cây ở đường biên của hàng rào vẫn không thoát khỏi bàn tay của đạo tặc. Năm nay bạn Minh Đức quyết xây dựng lại hàng rào để cho không còn cây nào nằm ở đường biên nữa. Để làm được điều này, thay vì chẳng đường rào theo các gốc cây, bạn Minh Đức sẽ chẳng đường rào theo các cột sắt có sẵn trong vườn. Vị trí của các cây và cột sắt đã rõ, nhưng xây dựng làm sao để hàng rào có chu vi nhỏ nhất vẫn là vấn đề nan giải. Bạn hãy giúp bạn Minh Đức giải quyết bài toán khó trên và cùng chia sẻ một vụ mùa bội thu.

Dữ liệu:

- Dòng đầu là số N ($N \leq 100$). Là số cây trong vườn.
- N dòng sau, mỗi dòng ghi 2 số là tọa độ của một cây trong vườn.
- Dòng tiếp theo là số M ($M \leq 100$). Là số cột sắt trong vườn.
- M dòng sau, mỗi dòng ghi 2 số là tọa độ của một cột sắt. Các tọa độ đều là số nguyên trong khoảng $-10000..10000$.

Kết quả:

- In ra một số duy nhất là độ dài nhỏ nhất của hàng rào với đúng 2 chữ số sau dấu chấm thập phân (có làm tròn). Dữ liệu luôn đảm bảo có ít nhất 1 cách xây hàng rào thỏa mãn.

Ví dụ:

Input	Output
1 0 2 3 -2 0 2 0 0 4	12.94

Hướng dẫn thuật toán:

- Trước hết, ta sort lại các cột sắt theo chiều tọa độ x tăng dần.
 - Sau đó, duyệt cột đầu tiên được chọn (giả sử là điểm $R(x_0, y_0)$).
 - Khi đó, tất cả các cây đều phải có tọa độ $x \geq x_0$, nếu không thì sẽ không thể bao hết được.
 - Xét một đường bao bất kì chứa được tất cả các cây:
 - Thứ nhất, đường bao đó phải là một đa giác lồi với các đỉnh là các cột sắt, trong đó cột ở (x_0, y_0) chắc chắn được chọn.
 - Thứ hai, giả sử ta có các cạnh của đa giác lồi theo chiều ngược kim đồng hồ. Vì các cây đều nằm trong đa giác nên mỗi cây đều nhìn các cạnh của đa giác theo chiều ngược kim đồng hồ. Nói cách khác, nếu cây đang xét ở điểm O, thì với mỗi cạnh AB của đa giác, ta có $OA \times OB > 0$ (OA và OB ở đây là 2 vector).
 - Như vậy, ta có thể tính cách dựng đa giác tối ưu như sau:
 - Sort các cột sắt theo thứ tự ngược chiều kim đồng hồ so với điểm $R(x_0, y_0)$. Nói cách khác, điểm A đứng trước điểm B nếu $RA \times RB > 0$. Lưu chúng vào mảng A, sau đó thêm điểm R vào cuối mảng A.
 - Quy hoạch động $dp[i]$ có nghĩa là chi phí tối thiểu để xây một bao lồi sao cho:
 - Bao lồi bắt đầu ở R, kết thúc ở điểm $A[i]$ (bao lồi bị "hở", nếu $A[i] = R$ thì bao lồi kín và đây chính là đáp án).
 - Mọi cây đều nhìn các cạnh của bao lồi theo chiều ngược kim đồng hồ.
- Ta xét điểm $A[j]$, sao cho $j < i$. Nếu tất cả các cây đều nhìn cạnh $(A[j], A[i])$ theo chiều ngược kim đồng hồ (có nghĩa là $OA[j] \times OA[i] > 0$ với mọi cây O) thì ta có thể tính $dp[i] = \min(dp[i], dp[j] + \text{dist}(A[i], A[j]))$ trong đó $\text{dist}(A, B)$ là khoảng cách giữa 2 điểm A và B.
- Đáp án chính là phần tử cuối cùng của mảng dp (khi $A[i] = R$).

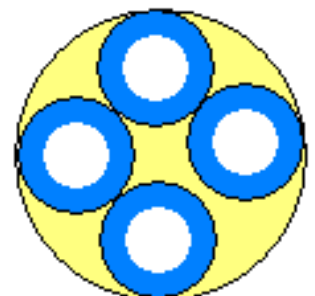
Code: <https://www.dropbox.com/s/xvbbbl5quajn8xa/Raovuon.rar?dl=0>

Test: <http://vn.spoj.com/problems/RAOVUON/>

2.3. BÀI TẬP TỔNG HỢP

Bài 1. Bàn tiệc mùa xuân (table) (nguồn: codeforces.com)

Gnouc đang giúp mẹ chuẩn bị mâm cơm đêm giao thừa, mâm cơm có dạng hình tròn với bán kính là R . Gia đình Gnouc chuẩn bị n đĩa thức ăn cũng có dạng hình tròn và các đĩa thức ăn đều có bán kính là r . Gnouc đang băn khoăn liệu mình có thể sắp xếp tất cả các đĩa thức ăn lên mâm cơm sao cho tất cả các đĩa thức ăn phải nằm trọn vẹn trong mâm và phải chạm vào thành



mâm. Tất nhiên các đĩa không được xếp chồng lên nhau nhưng chúng có thể tiếp xúc với nhau. Ví dụ với $n = 4, R = 10, r = 4$ ta có cách xếp các đĩa thức ăn như sau:

Dữ liệu:

- Gồm một dòng duy nhất chứa ba số nguyên n, R, r ($1 \leq n \leq 100, 1 \leq r, R \leq 1000$)

Kết quả:

- Ghi ra thông báo “YES” nếu có thể xếp các đĩa thức ăn theo yêu cầu đề bài, ghi ra thông báo “NO” trong trường hợp ngược lại

Ví dụ:

Input	Output
4 10 4	YES
5 10 4	NO
1 10 10	YES

Hướng dẫn thuật toán + code + test: <https://www.dropbox.com/s/50y8qfmruterqy/table.rar?dl=0>

Bài 2. Tứ giác lớn nhất (nguồn: codeforces.com)

Cho một tập gồm n điểm trong mặt phẳng tọa độ Decartes trong đó không có 3 điểm nào thẳng hàng và 2 điểm nào trùng nhau. Tìm tứ giác có diện tích lớn nhất được tạo nên bởi 4 trong số n điểm trong tập điểm nói trên. Lưu ý rằng tứ giác không nhất thiết phải lồi.

Dữ liệu:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 300$).
- n dòng tiếp theo mỗi dòng chứa 2 số nguyên x_i, y_i ($-1000 \leq x_i, y_i \leq 1000$).

Kết quả:

- Đưa ra một số thực với độ chính xác 6 chữ số sau dấu phẩy là kết quả của bài toán.

Ví dụ:

Input	Output
5 0 0 0 4 4 0 4 4 2 3	16.000000

Hướng dẫn thuật toán :

- Ta xét mỗi bộ 2 điểm, coi như đó là đường chéo của tứ giác, đường chéo này phải chia tứ giác thành 2 mặt phẳng, mỗi điểm trong 2 điểm còn lại nằm trong 1 mặt phẳng, với tứ giác lồi cả 2 đường chéo đều thỏa mãn điều trên, với tứ giác lõm thì chỉ có một đường chéo như vậy.
- Với mỗi bộ 2 điểm như vậy ta xét tất cả các đỉnh còn lại và phân hoạch nó vào 2 phần, 1 phần nằm “bên trái” (rẽ trái), 1 phần nằm “bên phải” (rẽ phải) (dùng hàm CCW). Như

vậy với mỗi phần ta lưu diện tích lớn nhất của tam giác tạo bởi 2 điểm đang xét và một điểm nào đó nằm trong phần này, kết quả là tổng 2 diện tích lớn nhất trên 2 phần.

Test+Code: <https://www.dropbox.com/s/21u5h4xt2klofvv/maxquad.rar?dl=0>

Bài 3: Mặt trời chiếu sáng (SUNSHINE) (Nguồn: *Kì thi talipan của chuyên KHTN*)

Cuộc sống ở xứ sở Thiên đường XHCN cực kỳ tuyệt diệu: thức ăn ngon, âm nhạc,... Nhưng chỉ có một vấn đề: Mây. Tuy nhiên, chính quyền Thiên đường đã rất quan tâm đến vấn đề này: các tòa nhà cao tầng sẽ được xây dựng không quá gần nhau để nóc các tòa nhà nhận được ánh nắng và tiếng mỗi ngày. Để đơn giản, coi Thiên đường là một đường thẳng chạy từ đông sang tây. Mỗi ngày Thiên đường có 12 giờ và nhờ vào thành tựu xây dựng phi thường, các tòa nhà chỉ có đúng chiều cao. Giáo sư Natsu đi du lịch đến đây và đi trên một con phố có N tòa nhà, tòa nhà thứ i cách mốc đầu đường wivà có độ cao hi. Giáo sư muốn biết số lượng tòa nhà được chiếu sáng không ít hơn 6h là bao nhiêu (làm tròn đến giây). Số giờ nóc mỗi tòa nhà được chiếu sáng tỷ lệ với góc ánh nắng mặt trời tới tòa nhà (\geq là 12h, là 6h, \leq là 0h,...) – làm tròn đến chữ số thứ 2 sau dấu chấm thập phân. (Xem hình vẽ).

Dữ liệu:

- Dòng đầu ghi số . Sau đó là N dòng, mỗi dòng ghi 2 số nguyên và . Chú ý nếu $i < j$ thì . Các tòa nhà được liệt kê lần lượt theo chiều từ trái qua phải .

Kết quả:

- In ra số tòa nhà thỏa mãn.

Ví dụ:

Sunshine.inp	Sunshine.out
5	3
0 8	
4 6	
6 4	
7 1	
11 20	

Hướng dẫn thuật toán:

Coi đỉnh mỗi ngôi nhà là một điểm, thì tầm nhìn của một đỉnh A bị giới hạn bởi 2 đỉnh B và C, một đỉnh bên trái và một đỉnh bên phải, sao cho góc BAC là nhỏ nhất có thể. Nói cách khác, B là điểm bên trái A sao cho góc tạo bởi BA với trục Oy là nhỏ nhất, tức là đường thẳng BA nằm trên tất cả các điểm khác mà nằm bên trái A.

- Để tìm điểm B với mỗi điểm A, ta sử dụng tư tưởng ngăn xếp tương tự khi cài bao lồi: Duy trì một ngăn xếp (stack) chứa những đỉnh chưa nằm dưới đường thẳng nào. Trước khi đưa một đỉnh A vào stack, ta kiểm tra đỉnh C ở đầu stack có nằm dưới đường thẳng

AB, với B là đỉnh thứ hai trong stack, hay không. Với mỗi đỉnh A, đỉnh B cần tìm chính là đỉnh ở đầu stack trước khi xếp A vào stack (nếu stack rỗng thì A không bị chặn bởi điểm nào cả). Cuối cùng, ta chỉ cần đếm số đỉnh A sao cho A không bị chặn ở 1 trong 2 hướng hoặc góc BAC tù.

Độ phức tạp:

Test + code: <https://www.dropbox.com/s/9gsoz9yzfv6hvju/Sunshine.rar?dl=0>

Bài 4: Cắt hình vuông (squarepart) (Nguồn: Thầy Hồ Đắc Phương Chuyên KHTN)

Xét một lưới điểm nguyên. Xét các điểm nguyên nằm trên cạnh hình vuông có các cạnh song song với trục tọa độ. Các điểm này đánh số từ 1, từ góc dưới cùng bên trái và theo chiều ngược chiều kim đồng hồ (xem hình minh họa). Sau đó hình vuông này được cắt bằng một số đoạn thẳng nối các điểm nguyên. Đếm xem các đoạn thẳng này chia hình vuông thành bao nhiêu phần.

Dữ liệu:

- Dòng đầu ghi 2 số n và k, trong đó n là độ dài cạnh hình vuông, k là số đoạn thẳng dùng để cắt ()
- Sau đó là k dòng, mỗi dòng ghi 2 số là số thứ tự 2 điểm đầu mút của đoạn thẳng (tất cả các đoạn thẳng không nằm trên cạnh của hình vuông).

Kết quả:

- In ra số mảnh hình vuông bị cắt ra.

Ví dụ:

Squarepart.inp	Squarepart.out
8 6 23 15 19 28 1 18 30 14 9 21 17 1	16

Hướng dẫn thuật toán:

- Ta dễ dàng nhận thấy bài toán trên tương đương với bài toán tìm số miền của một đồ thị phẳng. Với đỉnh của đồ thị là các điểm và giao điểm, các cạnh là các đoạn thẳng nối 2 điểm mà không đi qua điểm nào khác. (Tìm hiểu về đồ thị phẳng ở đây: https://en.wikipedia.org/wiki/Planar_graph)
- Ta có một tính chất của đồ thị phẳng là : $V - E + F = 2$. Trong đó V là số đỉnh, E là số cạnh, F là số miền.
- Trở lại với bài toán ta thấy rằng kết quả của bài toán này chính là $F - 1$ (ko tính phần ngoài cùng). Vậy giờ chúng ta chỉ cần đếm số đỉnh và số cạnh. Ta sẽ duyệt lần lượt các đoạn được thêm vào, với mỗi đoạn ta kiểm tra xem nó với các đoạn phía trước xem có bao nhiêu giao điểm mới được tạo thành (chính là lượng đỉnh tăng lên) và xem đoạn mới thêm vào này bị chia thành bao nhiêu phần (chính là lượng đỉnh tăng lên). Các thao tác này có thể quản lý bằng set.

Test+code: <https://www.dropbox.com/s/lwv6iqqmulgkxn/squarepart.rar?dl=0>

Bài 5: Đa giác (Polygon) (nguồn: codeforces.com)

Sơn muốn sử dụng các tấm gỗ có sẵn để ghép thành hàng rào xung quanh vườn. Hàng rào có dạng hình đa giác lồi (diện tích khác 0) được ghép từ các cạnh là các tấm gỗ đặt nằm ngang. Sơn không muốn cưa hay phá hỏng các tấm gỗ để làm hàng rào. Rõ ràng không phải từ các tấm gỗ bất kì nào cũng có thể ghép thành hàng rào đa giác lồi. Ví dụ từ 3 tấm gỗ có độ dài có thể ghép thành hàng rào tam giác, tuy nhiên từ 4 tấm gỗ có độ dài thì không thể ghép thành hàng rào tứ giác vì tấm gỗ dài lớn hơn tổng độ dài của 3 tấm gỗ còn lại.

Sơn có tấm gỗ. Sơn tự hỏi có bao nhiêu cách chọn ra các tấm gỗ từ các tấm đã có để từ chúng có thể ghép thành hàng rào đa giác lồi. Hai cách chọn được gọi là khác nhau nếu như có một tấm gỗ thuộc cách chọn này nhưng không thuộc cách chọn kia.

Yêu cầu: Bạn hãy giúp Sơn trả lời câu hỏi này. Do số lượng cách chọn có thể rất lớn nên bạn cần đưa ra số dư của số cách chọn trong phép chia cho

Dữ liệu:

- Dòng đầu tiên chứa số nguyên là số lượng tấm gỗ.
- Dòng thứ hai chứa số nguyên là độ dài các tấm gỗ.

Kết quả:

- Ghi ra một dòng duy nhất là kết quả của bài toán.

Ví dụ:

polygon.inp	polygon.out
5	7
1 2 3 4 6	
4	5
5 5 5 5	
5	0
10 1 2 3 50	
6	40
12 16 14 8 17 7	

Giải thích:

Ví dụ 1: Ta có các đa giác với độ dài các cạnh tương ứng là:

Ví dụ 2: Chọn ba cạnh bất kì trong 4 cạnh luôn được một tam giác đều, do đó ta có tam giác đều với độ dài các cạnh tương ứng là: và . Chọn cạnh ta được thêm một tứ giác . Vì vậy kết quả của bài toán là 5.

Hướng dẫn thuật toán:

- Nhận xét: Một tập hợp các thanh gỗ ghép được thành một đa giác lồi khi và chỉ khi độ dài thanh dài nhất nhỏ hơn tổng độ dài các thanh còn lại.
- Trước tiên ta sắp xếp lại độ dài của các thanh gỗ theo thứ tự tăng dần. Khi đó xét mỗi thanh gỗ, ta sẽ tìm số cách chọn các thanh gỗ từ đến sao cho tổng của các thanh gỗ đó lớn hơn. Ta sẽ đếm bằng cách đếm phần bù, đếm số cách chọn các thanh gỗ sao cho tổng của các thanh gỗ là. Đến đây ta sử dụng quy hoạch động, đây là một bài toán quy hoạch động rất cơ bản, gọi là số cách chọn các thanh gỗ từ đến sao cho tổng các thanh gỗ là. Từ đây ta dễ dàng tìm được kết quả bài toán.

Test+code: <https://www.dropbox.com/s/sqz9w5iizogkyrd/Polygon.rar?dl=0>

Bài 6: Nối dây (Lines) (nguồn: sưu tầm)

Cho hai đường thẳng song song nằm ngang a và b . Trên mỗi đường thẳng, người ta chọn lấy n điểm phân biệt và gán cho mỗi điểm một số nguyên dương là nhãn của điểm đó:

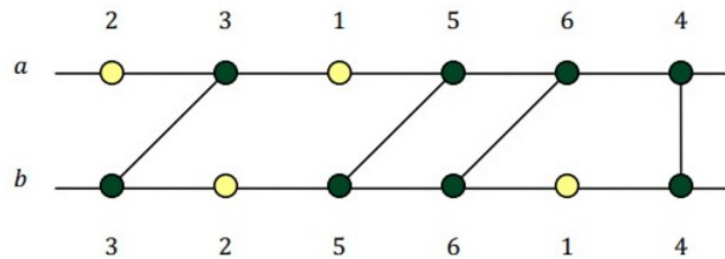
- Trên đường thẳng a , điểm thứ i (theo thứ tự từ trái qua phải) được gán nhãn là a_i .
- Trên đường thẳng b , điểm thứ j (Theo thứ tự từ trái qua phải) được gán nhãn là b_j .

Ở đây (a_1, a_2, \dots, a_n) và (b_1, b_2, \dots, b_n) là những hoán vị của dãy số $(1, 2, \dots, n)$

Yêu cầu: Hãy chỉ ra một số tối đa các đoạn thẳng thỏa mãn:

- Mỗi đoạn thẳng phải nối hai điểm có cùng một nhãn: Một điểm trên đường thẳng a và một điểm trên đường thẳng b .

- Các đoạn thẳng đôi một không có điểm chung



Dữ liệu: Vào từ tệp văn bản LINES.INP

- Dòng 1: chứa số nguyên dương $n \leq 10^5$
- Dòng 2: chứa n số theo thứ tự là a_1, a_2, \dots, a_n
- Dòng 3: Chứa n số theo thứ tự là b_1, b_2, \dots, b_n

Kết quả: Ghi ra tệp văn bản LINES.OUT

- Ghi số k là số đoạn thẳng nối được.

Ví dụ:

LINES.INP	LINES.OUT
6	4
2 3 1 5 6 4	
3 2 5 6 1 4	

Hướng dẫn thuật toán: Thuật toán QHD + điều kiện hai đoạn thẳng không cắt nhau

Test+code: <https://www.dropbox.com/s/5we9xw4jdt078cf/Lines.rar?dl=0>

Bài 7. Dịch chuyển điểm (Paralelogrami) (nguồn: COCI 2016/2017)

Gần đây, một trò chơi máy tính mới đã xuất hiện, được gọi là “Parallelograms”. Khi trò chơi bắt đầu, máy tính sẽ vẽ ra N điểm trên màn hình tọa độ của các điểm là các số nguyên nằm trong phạm vi -10 đến 10.

Chỉ có một di chuyển trong game được thực hiện là lấy ba điểm không thẳng hàng A, B và C, và rồi thay điểm C bằng điểm D sao cho ABCD là hình bình hành đường chéo là AB. Chú ý rằng điểm D luôn tồn tại và duy nhất.

Khi bắt đầu trò chơi, tất cả các điểm đều khác nhau, nhưng trong quá trình chơi thì được phép xảy ra hai hay nhiều hơn hay nhiều hơn hai điểm có thể có tọa độ trùng nhau. Thêm nữa các điểm mới được tạo ra thì tọa độ phải thỏa mãn có giá trị tuyệt đối nhỏ hơn .

Mục tiêu của trò chơi là dùng một dãy các di chuyển, di chuyển tất cả các điểm vào góc phần tư thứ nhất. Một cách chi tiết hơn, kết thúc trò chơi, tất cả các điểm phải có tọa độ không âm.

Tìm một dãy các di chuyển (không được vượt quá 2500 di chuyển), sao cho di chuyển được tất cả các điểm vào góc phần tư thứ nhất, hoặc xác định dãy các di chuyển là không tồn tại.

Dữ liệu:

- Dòng đầu tiên ghi số nguyên dương N , là điểm được cho trước.
- N dòng tiếp theo mỗi dòng ghi tọa độ của điểm thứ i , Sao cho, hai điểm bất kì là không trùng nhau.

Kết quả:

- Nếu không tồn tại thì ghi ra -1
- Nếu tồn tại dãy di chuyển thì dòng đầu tiên ghi ra số nguyên M là số các bước di chuyển
- Mỗi dòng trong M dòng phải ghi ba nguyên khác nhau A, B, C () là chỉ số của các điểm liên quan trong phép di chuyển. Trong đó C là chỉ số của điểm được thay đổi, và các điểm với chỉ số A, B là không thay đổi.

Ví dụ:

Input 3 0 0 4 0 3 -1	Input 4 5 0 0 5 -2 -2 -3 2	Input 3 -1 -1 -2 -2 -3 -3
Output 1 1 2 3	Output 2 1 2 3 1 2 4	Output -1

Hướng dẫn thuật toán:

- Nếu tất cả các điểm là thẳng hàng, thì không thể tồn tại một phép di chuyển nào, vì vậy không tồn tại phép di chuyển nào. Kết quả đưa ra là -1.
- Trước tiên, Chú ý rằng phép toán thay thế điểm C thành điểm D sao cho $ABCD$ là hình bình hành đường chéo AB có thể thay thế bởi phép toán đại số sau:
 - Giả sử rằng có ba điểm không thẳng hàng. Chúng ta sẽ chứng minh rằng chúng ta có thể di chuyển chúng vào một phần của mặt phẳng mà đảm bảo tọa độ nhỏ hơn 5. Và rồi mỗi điểm trong $N-3$ điểm còn lại có thể di chuyển vào góc phần tư thứ nhất bằng chính xác một phép di chuyển.
 - Nếu ba điểm không thẳng hàng, hãy chứng minh rằng chúng ta có thể di chuyển chúng vào phần nói trên của mặt phẳng. Cho ba điểm A, B và C và cho D là điểm được tạo ra bằng một phép ánh xạ trên. Bây giờ chúng ta có hình bình hành $ABCD$. Chú ý rằng, bằng cách dùng hình bình hành này, chúng ta có thể lát mặt phẳng theo cách được chỉ ra như trên hình vẽ.
 - Cũng chú ý rằng, với mỗi hình bình hành được sử dụng để lát mặt phẳng, có một dãy các phép toán biến ba điểm thành ba đỉnh của các hình bình hành đó.
 - Vì các hình bình hành lát toàn bộ mặt phẳng, vì vậy nó cũng lát góc phần tư mà đảm bảo tọa độ nhỏ hơn 5.
 - Do đó, sẽ phải tồn tại một hình bình hành mà nó được đặt chọn vẹn trong góc phần tư đó và một dãy các phép toán sao cho biến các điểm của chúng ta thành ba điểm của góc phần tư đó.

- Câu hỏi còn lại là có bao nhiêu phép di chuyển được thực hiện.
- Chúng ta thực hiện thuật toán BFS với trạng thái là ba điểm được xét đến, và phép biến đổi là áp dụng một trong ba phép toán có thể của mỗi trạng thái. Chúng ta hạn chế BFS không loang vào tam giác mà không giao với phần mặt phẳng, vì chúng ta biết rằng lời giải tồn tại khi chúng ta không loang vào những tam giác như vậy. Các tọa độ là nhỏ vì vậy chúng ta có thể dùng vét cạn.
- Chúng ta kí hiệu A, B, C là ba đỉnh của tam giác. Chú ý rằng tồn tại một đỉnh của tam giác, không mất tính tổng quát, giả sử đỉnh đó là C, sao cho phần mặt phẳng được giới hạn bởi CA và CB là phần mặt phẳng chúng ta muốn di chuyển ba điểm vào đó. Chúng ta ánh xạ điểm C qua đường thẳng AB.

Test+Code: <https://www.dropbox.com/s/3y4xw7orcnd1mzi/Paralelogrami.rar?dl=0>

Bài 8: Cặp điểm gần nhất (NEAREST) (nguồn: SPOJ)

- Cho n ($2 \leq n \leq 100,000$) điểm trên mặt phẳng, hãy tìm cặp điểm có khoảng cách nhỏ nhất.

Dữ liệu:

- Dòng đầu tiên chứa số n .
- dòng tiếp theo mỗi dòng chứa một cặp số thực (giá trị tuyệt đối không lớn hơn 10^7) biểu diễn tọa độ một điểm.

Kết quả:

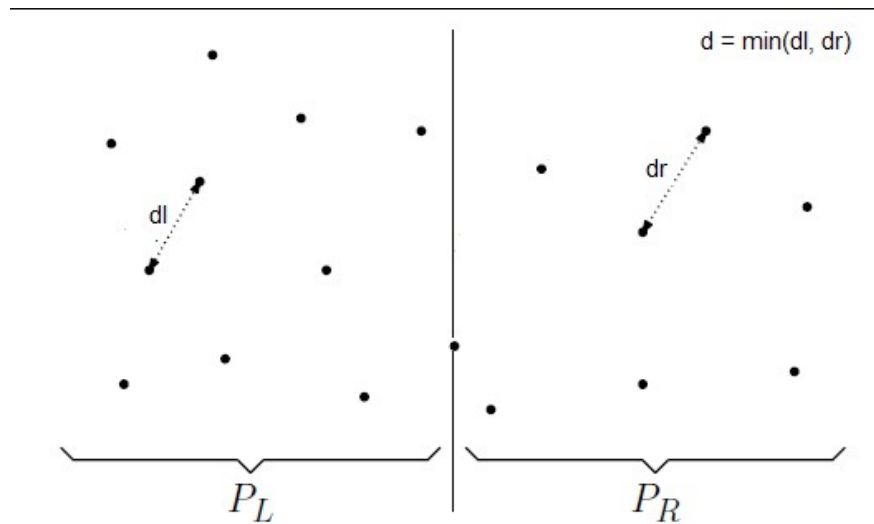
- Một số duy nhất (ghi chính xác đến 3 chữ số thập phân sau dấu phẩy) là khoảng cách nhỏ nhất tìm được.

Ví dụ:

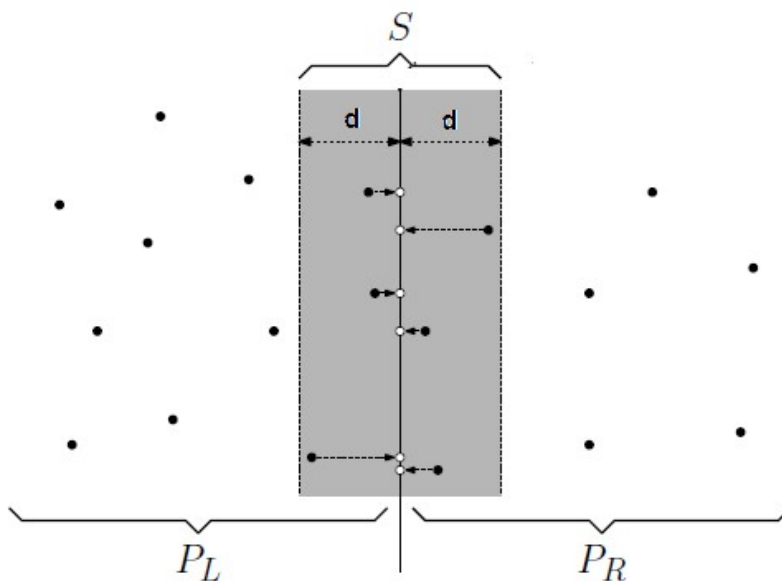
Input	Output :
5 1 1 2 2 3 3 4 4 5 5	1.414

Hướng dẫn thuật toán:

- Đây là một bài toán khá cơ bản và nổi tiếng trong lĩnh vực hình học. Trong chuyên đề này chúng ta thảo luận một giải pháp chia để trị cho bài toán này.
- Giải thuật độ phức tạp $O(n \log^2 n)$
- Chúng ta sắp xếp lại tọa độ các điểm theo tọa độ x .
- Chỉ tập các điểm thành hai dãy con, dãy thứ nhất chứa các điểm từ $P[0]$ đến $P[n/2]$, dãy con thứ hai chứa các điểm từ $P[n/2+1]$ đến $P[n]$.
- Chúng ta thực hiện thuật toán đệ quy để tìm khoảng cách nhỏ nhất trên cả hai tập con.
- Lấy minimum của hai khoảng cách nhỏ nhất. kí hiệu là d (xem hình vẽ)



- Từ bước trên chúng ta có một chặn trên d của khoảng cách nhỏ nhất. Bây giờ chúng ta cần xét các cặp điểm sao cho một điểm trong cặp là ở bên trái điểm kia là ở bên phải. Chúng ta xét một đường thẳng vuông góc với trục hoành và đi qua điểm $P[n/2]$ và tìm tất cả các điểm có khoảng cách đến đường thẳng nhỏ hơn d . Gọi $strip[]$ là mảng lưu tất cả các điểm như vậy.
- Sắp xếp mảng $strip[]$ theo tọa độ y . Bước này mất độ phức tạp $O(n \log n)$ có thể thực hiện trong $O(n)$ bằng cách gọi đệ quy $sort$ và $merge$.



- Tìm khoảng cách nhỏ nhất trên $strip[]$. Dường như mất $O(n^2)$ nhưng thực chất chỉ mất $O(n)$.
- Cuối cùng trả về minimum của d và khoảng cách nhỏ nhất được tìm thấy trên $strip[]$.
- Giải thuật độ phức tạp $O(n \log n)$:
- Chúng ta cài thiện thuật toán trên với bước tìm khoảng cách nhỏ nhất trên tập $strip[]$.
- Chúng ta sắp xếp lại các điểm theo tọa độ y và lưu vào mảng $Py[]$.
- Khi đó bước tìm minimum trên mảng $strip$ cũng được thực hiện bằng phương pháp chia để trị.

Code: <https://www.dropbox.com/s/6fnl8cikeejz8tx/Nearest.rar?dl=0>

Test: <http://vn.spoj.com/problems/NEAREST/>

Bài 9: Chèo Thuyền (ROWING) (nguồn: Thầy Lê Minh Hoàng)

Người dân nước GeoLand say mê các môn thể thao mạo hiểm đòi hỏi tư duy hình học chuyên nghiệp. Một trong những môn thể thao đó là bơi thuyền vượt bãi đá trên sông Rect River – con sông dài nhất GeoLand. Bản đồ con sông được vẽ trên mặt phẳng tọa độ với hệ tọa độ Descartes vuông góc, hai bờ sông là hai đường thẳng song song và . Bãi đá trên sông gồm n tảng đá đánh số từ 1 tới n , tảng đá thứ i có tọa độ trên bản đồ. Mỗi vận động viên tham gia bài thi với một thuyền thúng hình tròn. Anh ta được đặt thuyền của mình ở vị trí tùy chọn nằm hoàn toàn bên trái bãi đá và cần bơi thuyền tới một vị trí tùy chọn nằm hoàn toàn bên phải bãi đá. Thuyền được di chuyển theo hướng tùy ý nhưng không được chạm vào bờ sông hay chạm vào một tảng đá nào của bãi đá (kể cả đường biên của thuyền).

Yêu cầu: Tìm số nguyên lớn nhất để mọi thuyền có đường kính đều thực hiện được bài thi.

Dữ liệu: Vào từ file văn bản ROWING.INP

- Dòng 1 chứa hai số nguyên
- n dòng tiếp theo mỗi dòng chứa hai số nguyên và

Kết quả:

- Kết quả ghi ra file văn bản ROWING.OUT một số nguyên duy nhất là số tìm được.

Ví dụ:

Rowing.inp	Rowing.out
4 8	5
1 2	
4 6	
9 2	
9 7	

Hướng dẫn thuật toán:

- Xét đường đi từ bờ dưới lên bờ trên và đi qua các tảng đá.
- Chi phí của đường đi là khoảng cách lớn nhất giữa hai tảng đá bất kì, hay giữa tảng đá và một bờ.
- Khi đó, chúng ta dễ dàng kiểm tra được kết quả của bài toán là đường đi có chi phí nhỏ nhất.
- Do vậy bài toán trở thành bài toán tìm đường đi ngắn nhất từ bờ dưới lên bờ trên đi qua các tảng đá.

Test + code: <https://www.dropbox.com/s/8i5ldi553qqe53l/Rowing.rar?dl=0>

Bài 10: Đa giác (Poly) (nguồn: sưu tầm)

Trên mặt phẳng tọa độ, xét đa giác lồi n đỉnh, các đỉnh của đa giác được liệt kê theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ.

Yêu cầu: Cho đoạn thẳng xác định bởi hai điểm có tọa độ là (x_1, y_1) và (x_2, y_2) . Trong đó x_1, y_1, x_2, y_2 là các số nguyên và có giá trị tuyệt đối không vượt quá 1000. Hãy xác định độ dài L là phần của đoạn thẳng nằm trong đa giác hay nằm trên cạnh của đa giác.

Dữ liệu:

- Dòng đầu tiên chứa số nguyên n

- Dòng thứ n trong n dòng sau chứa hai số nguyên xác định tọa độ đỉnh i của đa giác;
- Dòng cuối cùng chứa bốn số nguyên.

Kết quả:

- Ghi một số nguyên của tích

Ví dụ:

Poly.inp	Poly.out
4 0 1 1 0 0 -1 -1 0 -2 0 0 0	100

Hướng dẫn giải:

- Gọi hai điểm (x_1, y_1) và (x_2, y_2) lần lượt là A và B. Do đa giác đã cho là đa giác lồi
- Có 4 trường hợp:
 - Đường thẳng (AB) chứa một cạnh của đa giác.
 - Đường thẳng (AB) đi qua một đỉnh của đa giác.
 - Đường thẳng (AB) cắt đa giác tại hai điểm.
 - Đường thẳng (AB) không cắt đa giác.

Trường hợp 1:

Có 2 khả năng:

- Đoạn AB nằm ngoài cạnh của đa giác.
- Đoạn AB có một phần trên cạnh hoặc chung đầu mút với cạnh của đa giác.

Kiểm tra A, B có điểm nào nằm trên cạnh của đa giác.

Nếu không \Rightarrow a, trả về kết quả là 0.

Nếu có \Rightarrow b.

Xét A, B, C, D (với C, D là các đầu mút của cạnh đa giác).

Do đã biết AB có một phần trên cạnh hoặc chung đầu mút

Phần độ dài chung (hay phần CD chứa của AB) có 2 đầu mút là 2 điểm nằm giữa trong 4 điểm A, B, C, D.

Mà A, B, C, D đã cùng thuộc một đường thẳng \Rightarrow so sánh / sort theo x hoặc y, dễ dàng tìm ra 2 điểm này.

(Trường hợp hai đoạn AB, CD chỉ chung nhau một đầu mút, 2 điểm nằm giữa trùng nhau \Rightarrow kết quả vẫn bằng 0)

Trường hợp 2 & 3:

Tìm giao điểm của (AB) với các cạnh của đa giác bằng cách lập phương trình đường thẳng chứa các cạnh. Dễ dàng tìm được bằng cách giải hệ phương trình 2 ẩn.

Nếu phương trình vô nghiệm \Rightarrow (AB) không cắt cạnh.

Nếu phương trình có vô số nghiệm => Trường hợp 1.

Nếu phương trình có một nghiệm => Giao điểm của (AB) và đường thẳng chứa cạnh.

Kiểm tra cạnh có chứa giao điểm hay không.

Chú ý trường hợp (AB) cắt cạnh tại đỉnh => lấy 2 giao điểm trùng nhau. Dùng biến đánh dấu để tránh lấy lặp giao điểm.

Khi tìm được 2 giao điểm => có 2 khả năng:

AB nằm ngoài đa giác. => cần kiểm tra 2 giao điểm có nằm trên AB.

AB có phần độ dài nằm trong đa giác (Khi đó tính được L tương tự như ở trường hợp 1 với C,D là 2 giao điểm)

Trường hợp 4:

(AB) không cắt đa giác => AB không có phần nào nằm trong đa giác => $L=0$;

Chương trình:

Chuẩn bị một hàm kiểm tra điểm có nằm trên đoạn thẳng.

Chạy vòng for(1->n) xét các đường thẳng chứa cạnh với đỉnh $i, i+1$ làm đầu mút (do đề đã cho các đỉnh theo chiều kim đồng hồ nên gán đỉnh $n+1$ =đỉnh 1), tìm giao điểm

Trường hợp 1,2,3;

Nếu không tìm thấy => Trường hợp 4.

Trả về kết quả L.

Test+code: <https://www.dropbox.com/s/fxh739id7km4msg/Poly.rar?dl=0>

Bài 11: Robot (nguồn: sưu tầm)

Cho lưới nguyên Oxy. Điểm nguyên và điểm nguyên được gọi là kề với nhau nếu thỏa mãn

Một robot ban đầu đứng tại gốc tọa độ. Ở mỗi bước, robot sẽ di chuyển sang một điểm nguyên kề với vị trí hiện tại. Từ bước di chuyển thứ hai trở đi, robot có thể đi tiếp theo hướng cũ, rẽ sang trái, rẽ sang phải, hay trở về vị trí trước đó.

Ví dụ từ ô robot đi đến rẽ trái sang ô, rẽ phải sang ô, rẽ phải sang ô rẽ trái sang cuối cùng rẽ phải sang

Yêu cầu: Cho tọa độ các điểm nguyên mà robot lần lượt đã đi qua. Hãy đếm xem robot đã rẽ phải bao nhiêu lần, rẽ trái bao nhiêu lần?

Dữ liệu:

- Dòng đầu tiên chứa 1 số nguyên dương n là tổng số điểm nguyên mà robot đã đi qua (kể cả vị trí xuất phát là gốc tọa độ),
- Dòng thứ i trong n dòng tiếp theo chứa hai số nguyên và là tọa độ điểm nguyên mà robot đã đi qua
- Các số trên cùng một dòng được ghi cách nhau bởi một khoảng trắng.

Kết quả:

- Gồm một dòng chứa hai số nguyên là số lần robot đã rẽ phải và số lần robot đã rẽ trái.

Ví dụ:

ROBOT . INP	ROBOT . OUT
7 0 0 1 0 1 1 2 1 2 0 3 0 3 -1	3 2

Hướng dẫn thuật toán:

- Để giải bài tập này, trước hết ta cần phải xác định được những yếu tố để có thể xác định được xem robot rẽ trái, rẽ phải hay đi thẳng.

- Nhận thấy, với 3 vị trí liên tiếp của đường đi, chúng ta có thể dễ dàng xác định được xem hướng đi của đường đi đó là như thế nào bằng cách dùng tọa độ của 3 vị trí đó.

	$x[i] - x[i - 2] = 1$ $y[i] - y[i - 2] = 1$	$x[i] - x[i - 2] = -1$ $y[i] - y[i - 2] = 1$	$x[i] - x[i - 2] = 1$ $y[i] - y[i - 2] = -1$	$x[i] - x[i - 2] = -1$ $y[i] - y[i - 2] = -1$
$x[i] = x[i - 1]$	L	R	R	L
$y[i] = y[i - 1]$	R	L	L	R

Test+code: <https://www.dropbox.com/s/60utckb7sriqafd/Robot.rar?dl=0>

Bài 12: Land (nguồn: sưu tầm)

Một khu đất kích thước . Theo quy hoạch ban đầu, trên khu đất đó sẽ xây dựng khu vui chơi, khu vui chơi thứ sẽ là một khu vực với kích thước . Tuy nhiên, một vấn đề mới phát sinh, người ta muốn quy hoạch lại để có thêm một khu vui chơi mới có dạng hình vuông và hình vuông có diện tích càng lớn càng tốt. Các khu vui chơi được xây dựng có thể tiếp xúc với nhau nhưng không được giao nhau.

Yêu cầu: Cho và khu vui chơi, hãy tính diện tích khu đất hình vuông lớn nhất tìm được.

Dữ liệu:

- Dòng đầu ghi 3 số nguyên
- dòng sau, mỗi dòng ghi hai số nguyên .

Kết quả:

- Gồm một dòng ghi một số nguyên là diện tích khu đất hình vuông lớn nhất tìm được.

Ví dụ:

LAND.INP	LAND.OUT
----------	----------

5 5 2 1 5 1 4	16
---------------------	----

Hướng dẫn thuật toán:

Nhận xét : Xếp k hình chữ nhật $A * B$ vào hình chữ nhật kích thước $M * N$ để được khoảng trống chứa 1 hình vuông kích thước lớn nhất. Gọi hình vuông đó có kích thước là $R * R$. Do đó ta sẽ xếp hình vuông kích thước $R * R$ và k hình chữ nhật $A * B$ vào hình chữ nhật kích thước $M * N$.

Thuật toán: Sử dụng thuật toán Backtrack để xếp k + 1 hình chữ nhật vào hình chữ nhật kích thước $M * N$

Do các hình chữ nhật có thể quay 90° nên ta backtrack 0, 1 với k hình chữ nhật (0: giữ nguyên kích thước $A * B$; 1 : đổi thành kích thước $B * A$)

Với mỗi trạng thái của k hình chữ nhật, ta thêm hình vuông kích thước val * val vào đầu mảng k hình. Sử dụng backtrack để xếp k + 1 hình .

Try(int x, int val) : xếp hình thứ x trong k + 1 hình và kích thước hình vuông là val * val

Các cận để giảm thời gian: dùng tknp để tính val với left = ans + 1, right = max_val (max_val : kích thước hình vuông lớn nhất có thể thêm)

Sort k hình chữ nhật theo diện tích giảm dần.

Kết quả bài toán : ans * ans.

Test+code: <https://www.dropbox.com/s/zay4wuwwzb1rw0c/Land.rar?dl=0>

C – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN:

Kết luận: Chuyên đề đã trình bày một số bài toán hình học cơ bản trong tin học và áp dụng giải quyết những bài tập tiêu biểu liên quan. Những bài tập trong chuyên đề được tôi sưu tầm và lựa chọn ở nhiều nguồn tài liệu khác nhau. Chuyên đề đã được tôi sử dụng trong quá trình giảng dạy học sinh đội tuyển. Kết quả thu được là học sinh có cái nhìn tổng quan và nắm vững những bài toán hình học cơ bản trong tin học từ đó làm nền tảng cho việc giải quyết những bài toán khó hơn.

Hướng phát triển: Mặc dù đã rất cố gắng nhưng do thời gian có hạn nên số lượng bài tập, cùng hướng dẫn giải và test kèm theo vẫn còn hạn chế. Vì vậy trong thời gian tới Tôi sẽ tiếp tục phát triển chuyên đề này theo hướng hệ thống bài tập theo từng chủ đề liên quan đến các bài toán cơ bản. Đồng thời rất mong được sự đóng góp ý kiến của các quý Thầy Cô để chuyên đề ngày càng được hoàn thiện hơn.