2.2. BÀI TẬP ỨNG DỤNG BAO LỒI CỦA TẬP ĐIỂM

Bài 1. THỬA ĐẤT LỚN NHẤT

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 (, yi). 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 xi , yi ($\forall i$: $|xi|$, $|yi| \leq$ 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 giày" 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.

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 <= n <= 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 <= m <= 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 2 4 8 4 6 8 4 6 4 3 5 4 7 5 5 6 7	NO NO YES YES

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 ($3 \leq k \leq n \leq 200$)
- n dòng sau, dòng thứ i gồm 2 số nguyên xi , yi ($|xi|$, $|yi| \leq$ 10^6) là tọa độ điểm thứ i ($i = 1,2,\dots,n$)

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 2 2	3.00

Hướng dẫn:

- Dễ dàng nhận thầy 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 giày để tính
- Với bài toàn 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 giày 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.

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 chăng 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 chăng đường rào theo các gốc cây, bạn Minh Đức sẽ chăng 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 <= 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 <= 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(x0, y0)).
- Khi đó, tất cả các cây đều phải có tọa độ x >= x0, 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 (x0, y0) 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 x 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(x0, y0). Nói cách khác, điểm A đứng trước điểm B nếu RA x 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] x OA[i] > 0 với mọi cây O) thì ta có thể tính dp[i] = min(dp[i], dp[j] + dist(A[i], A[j])) trong đó dist(A, B) là khoảng cách giữa 2 điểm A và B.

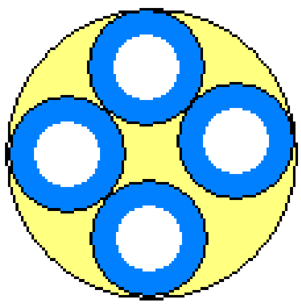
- Đáp án chính là phần tử cuối cùng của mảng dp (khi A[i] = R).

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* $(1 \le n \le 100,\ 1 \le r, R \le 1000)$

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 5 10 4 1 10 10	YES NO YES

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

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" (rẻ 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.

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 vài tiếng mỗi ngày. Để đơn giản, coi Thiên đường là một đườngthẳng 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 ($\geq 180^0$là 12h, 90^0là 6h, $\leq 0^0$là 0h,…) – làm tròn đến chữ số thứ 2 sau dấu chấm thập phân. (Xem hình vẽ).

Dữ liệu:

- Dòng đầu ghi số $(1 < N < 10^6)$. Sau đó là N dòng, mỗi dòng ghi 2 số nguyên w_i và h_i. Chú ý nếu i < j thì $w_i < w_j$. Các tòa nhà được liệt kê lần lượt theo chiều từ trái qua phải $(1 \leq w_i, h_i \leq 2000000)$.

Kết quả:

- In ra số tòa nhà thỏa mãn.

Ví dụ:

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

<u>**Hướng dẫn thuật toán:**</u>

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: $\boldsymbol{O(N)}$

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 chi hình vuông thành bao nhiêu phần.

Dữ liệu:

- Dòng đầu ghi 2 số n và k, trong đó n là độ dài cạnh hìnhvuông, k là số đoạn thẳng dùng để cắt ($1 \leq n \leq 1000, 1 \leq k \leq 200$).
- 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 du:

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 lí bằng set.

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 $10, 11, 12$ có thể ghép thành hàng rào tam giác, tuy nhiên từ 4 tấm gỗ có độ dài $100, 1, 2, 3$ thì không thể ghép thành hàng rào tứ giác vì tấm gỗ dài 100 lớn hơn tổng độ dài của 3 tấm gỗ còn lại.

Sơn có n 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 $10^9 + 7$.

Dữ liệu:

- Dòng đầu tiên chứa số nguyên n $(1 \leq n \leq 5000)$ là số lượng tấm gỗ.
- Dòng thứ hai chứa n số nguyên l_i $(1 \leq l_i \leq 5000)$ 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 1 2 3 4 6	7
4 5 5 5 5	5
5 10 1 2 3 50	0
6 12 16 14 8 17 7	40

Giải thích:

Ví dụ 1: Ta có các đa giác với độ dài các cạnh tương ứng là: $\{2;\ 3;\ 4\}; \{3;\ 4;\ 6\}; \{1;\ 2;\ 3;\ 4\}$

$\{1;\ 2;\ 4;\ 6\}; \{1;\ 3;\ 4;\ 6\}; \{2;\ 3;\ 4;\ 6\}$ và $\{1;\ 2;\ 3;\ 4;\ 6\}$

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

<u>Hướng dẫn thuật toán:</u>

- 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 $i \geq 3$, ta sẽ tìm số cách chọn các thanh gỗ từ 1 đến $i - 1$ sao cho tổng của các thanh gỗ đó lớn hơn l_i. Ta sẽ đếm bằng cách đếm phần bù, đếm số cách chọn các thanh gỗ sao cho tổng của các thanh gỗ là $1, 2, \ldots, l_i$. Đến đây ta sử dụng quy hoạch động, đây là một bài toán quy hoạch động rất cơ bản, gọi $f[i][j]$ là số cách chọn các thanh gỗ từ 1 đến i sao cho tổng các thanh gỗ là j. Từ đây ta dễ dàng tìm được kết quả bài toán.

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:

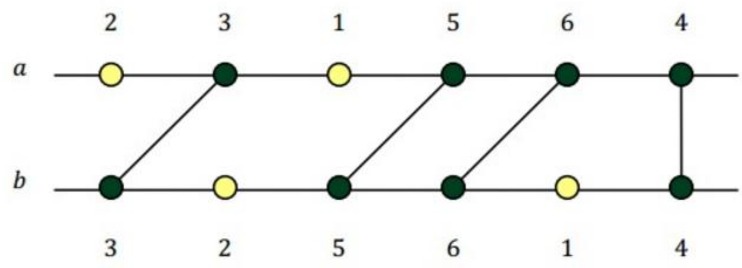
- Trên đường thẳng *a*, điểm thứ *i* (theo thứ tự từ trái qua phải) được gán nhãn là a_i.
- Trên đường thẳng *b*, điểm thứ *j* (Theo thứ tự từ trái qua phải) được gán nhãn là b_j.

Ở đây $(a_1, a_2, \ldots, a_n)$ và $(b_1, b_2, \ldots, b_n)$ là những hoán vị của dãy số (1, 2,…,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 2 3 1 5 6 4 3 2 5 6 1 4	4

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

<u>Bài 7. Dịch chuyển điểm (Paralelogrami)</u>*(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 năm trong phạm vi -10 để 10.

Chi có một di chuyên trong game được thực hiện là lấy ba điểm không thẳng hang 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 đồ trùng nhau. Thêm nữa các điểm mới được tạo ra thì tọa độ phải thỏa mãn có gia trị tuyệt đối nhỏ hơn 10^9.

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 từ thứ nhất, hoặc xác định dãy các di chuyên là không tồn tại.

Dữ liệu:

- Dòng đầu tiên ghí số nguyên dương N ($3 \leq N < 400$), là điểm được cho trước.
- N dòng tiếp theo mỗi dòng ghi tọa độ của điểm thứ i, $(x_i, y_i)(-10 \leq x_i, y_i \leq 10)$. Sao cho, hai điểm bất kì là không trung nhau.

Kết quả:

- Nếu không tồn tại thì ghi ra -1
- Nếu tồn tại dãy dị chuyển thì dòng đầu tiên ghi ra số nguyên M là số các bước di chuyển ($0 \leq M \leq 2500$).
- Mỗi dòng trong M dòng phải ghi ba nguyên khác nhau A, B, C ($1 \leq A, B, C \leq N$) là chi 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

<u>**Hướng dẫn thuật toán:**</u>

- Nếu tất cả các điểm là thẳng hàng, thì không thể tồn tài 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:

$$C(x, y) \rightarrow D(x_A + x_B - x_C, y_A + y_B - y_C)$$

- Giả sử rằng có ba điểm không thẳng hàng. Chúng ta sẽ chứng mình rằng chúng ra có thể di chuyển chúng vào một phần của mặt phẳng mà đảm bảo tọa độ x, y 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 tao ra bằng một phép ánh xạ trên. Bây giờ chúng ra 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 từ 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 phân từ đó.
- Câu hỏi còn lại là có bao nhiêu phép di chuyển được thực hiện.
- Chúng ta thực hiện thuật toán BFS với trang 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 $[-10, +\infty) \times [-10,\ +\infty)$, 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 muôn di chuyển ba điểm vào đó. Chúng ta ánh xạ điểm C qua đường thẳng AB.

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

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

Dữ liệu:

- Dòng đầu tiên chứa số n.

- 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:

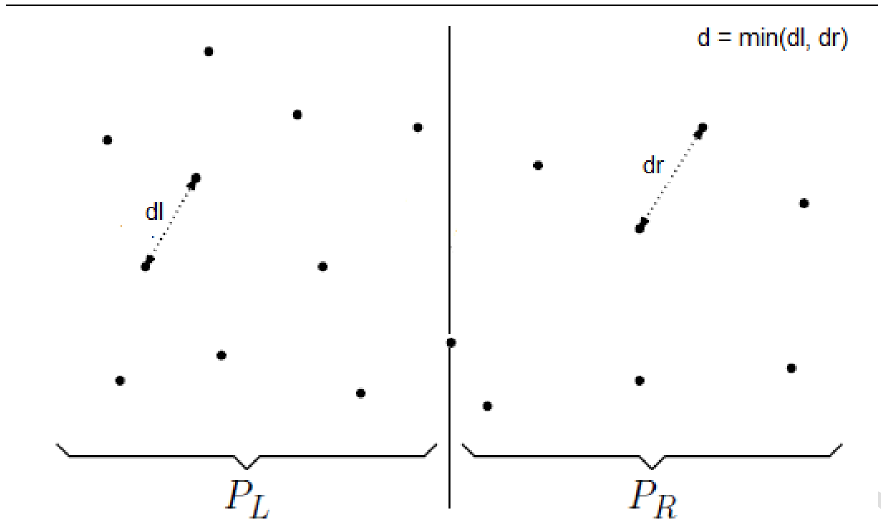
- Một số duy nhất (ghi chính xác đến 3 chữ số thập phân sau dấu phẩy) là khoảng cách nhỏ nhất tìm được.

Ví dụ:

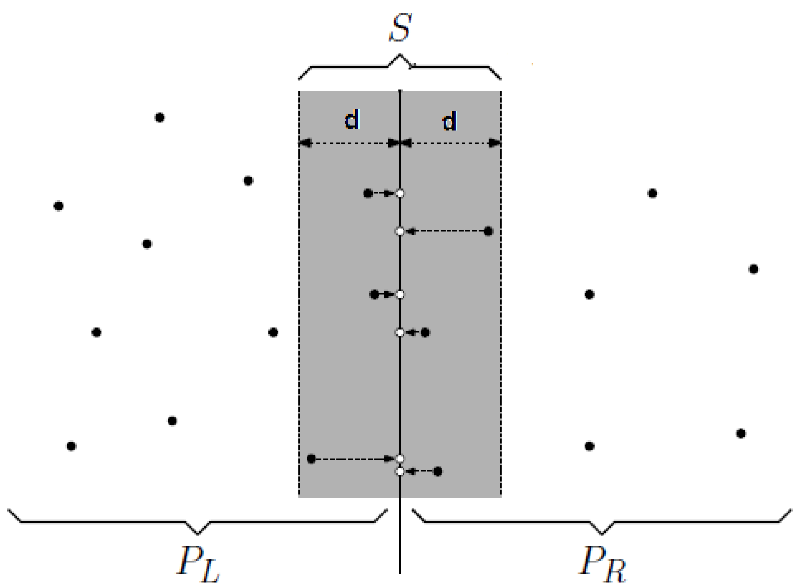
Input	Output:
5 1 1 2 2 3 3 4 4 5 5	1.414

Hướng dẫn thuật toán:

- Đây là một bài toán khả cở 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(nlog^2n)
- Chúng ta sắp xếp lại tọa độ các điểm theo tọa độ x.
- Chi tập các điểm thành hai dãy con, dãy thứ nhất chứa các điểm từ P[0] đến P[n/2], dãy con thứ hai chữa 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 hoàng 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(nlogn) có thể thực hiện trong O(n) bằng cách gọi đề quy sort và merge.



- Tìm khoảng cách nhỏ nhất trên strip[]. Dương như mất O(n^2) nhưng thực chất chỉ mật 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(nlogn):
- Chúng ta cải 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.

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

Bài 9: Chèo Thuyền (ROWING)

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 $y = 0$ và $y = h$. Bãi đá trên sông gồm n tảng đá đánh số từ 1 tới n, tảng đá thứ i có tọa độ (x_i, y_i) 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 d lớn nhất để mọi thuyền có đường kính $< d$ đều thực hiện được bài thi.

Dữ liệu: Vào từ file văn bản ROWING.INP

- Dòng 1 chứa hai số nguyên $n, h (n \leq 4000; 2 \leq h \leq 10^9)$
- n dòng tiếp theo mỗi dòng chứa hai số nguyên $x_i \leq 10^9$ và $y_i < h$.

Kết quả:

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

Ví dụ:

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

<u>Hướng dẫn thuật toán:</u>

- Xét đường đi từ bờ dưới lên bờ trên và đi qua các tảng đã.
- 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.

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

Trên mặt phẳng tọa độ, xét đa giác lồi   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 ($3 \leq n \leq 100$);
- Dòng thứ i trong n dòng sau chứa hai số nguyên xác định tọa độ đỉnh i của đa giác;
- Dòng cuối cùng chứa bốn số nguyên x_1, y_1, x_2, y_2.

Kết quả:

- Ghi một số nguyên của tích $L * 100$.

Ví dụ:

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

<u>Hướng dẫn giải:</u>

- Gọi hai điểm (x1,y1) và (x2,y2) 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 => a, trả về kết quả là 0.

Nếu có => 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 => 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 => 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 => (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.

Bài 11: Robot*(nguồn: sưu tầm)*

Cho lưới nguyên Oxy. Điểm nguyên (x_1, y_1) và điểm nguyên (x_2, y_2)được gọi là kề với nhau nếu thỏa mãn $|x_1 - x_2| + |y_1 - y_2| = 1$.

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 $(0, 0)$, robot đi đến $(1, 0)$, rẽ trái sang ô $(1, 1)$, rẽ phải sang ô $(2, 1)$, rẽ phải sang ô $(2, 0)$, rẽ trái sang $(3, 0)$ cuối cùng rẽ phải sang $(3, -1)$.

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 $(2 \leq n \leq 10^6)$ 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 $(1 \leq i \leq n)$ chứa hai số nguyên x_i và y_i 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

Bài 12: Land*(nguồn: sưu tầm)*

Một khu đất kích thước $m \times n$. Theo quy hoạch ban đầu, trên khu đất đó sẽ xây dựng k khu vui chơi, khu vui chơi thứ i sẽ là một khu vực với kích thước $a_i \times b_i$. 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 m, n và k khu vui chơi, hãy tính diện tích khu đất hình vuông lớn nhất tìm được.

Dữ liệu:

- Dòng đầu ghi 3 số nguyên m, n, k $(m, n \leq 10; k \leq 5)$;
- k dòng sau, mỗi dòng ghi hai số nguyên a_i, b_i.

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

<u>Hướng dẫn thuật toán:</u>

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.