2D Geometry

Contest

https://vjudge.net/contest/662497

I also have some custom problems on the slides

Geometry Problems

- Typically dealing with intersections, areas, or some other properties of a collection of a geometric primitives
- E.g. find the number of rectangles embedded within a set of points

General Tips

1. If possible, stick to integer math
2. Rely on intuition
3. Take advantage of the properties of geometric objects. You are typically dealing with squares, lines, circles, not complicated meshes
4. Think about different “parameterizations” or perspectives

Lines

Basic Definitions

A **Point** is a pair (x, y)

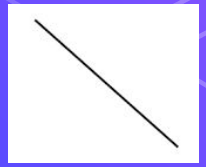
A **Line** an infinitely long straight collection of points, uniquely determined by two points on the plane

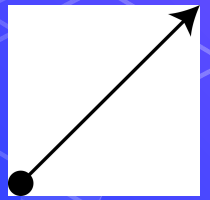
A **Line Segment** is a line bounded on both sides

- we may do a [0, 1] parameterization

A **Ray** is a line bounded on one side

A **Vector** represents differences/directions





Dot Product

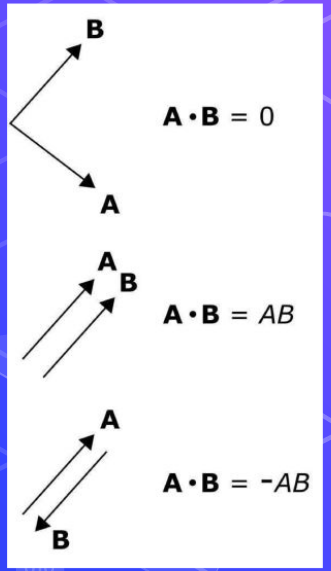
Suppose A = {ax, ay} and B = {bx, by} are vectors

dot(A, B) = A * B = axbx + ayby

If A and B point the same direction, positive

If orthogonal, zero

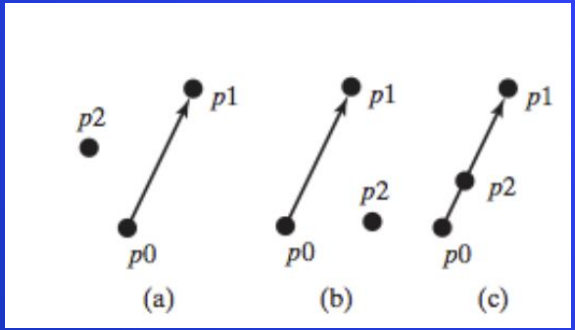
If opposite directions, negative



CSES Point Location Test

https://cses.fi/problemset/task/2189

For a point R and line PQ, see if a R is on PQ, to the left, or to the right



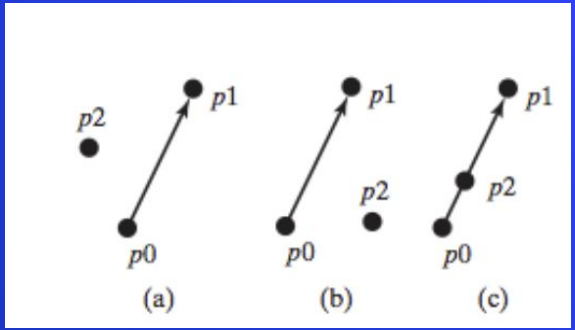
Possible Solution

Parameterize the line in slope intercept form and work from there.

Sometimes this may be the only thing we can do, but here it gets awkward for vertical lines and is subject to floating point error.

Better Solution

A better solution is to find orthogonal vector of PQ ({-dy, dx}) and take dot product.



Rectangles

Rectangles

Area: width * height

Intersection: R_1 intersects R_2 iff their x intervals intersect and y intervals intersect

General idea:

- Try to take advantage of the axis alignedness as much as possible. For axis aligned rectangles, you only need to know two corners.

Example Problem

You are given N <= 10^3 points on the 2d plane, where 1 <= x[i], y[i] <= 10^9. Please output unique the number of quadruplets i,j,k,l such that i < j < k < l and the associated points make a square (even if not axis aligned)

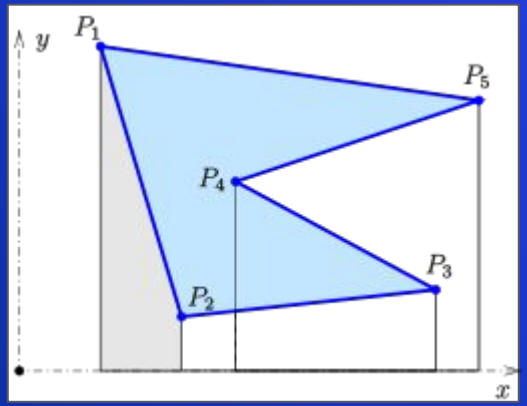
Hint: A square has 4 parameters (center (2), size (1), rotation (1)). A point has 2 parameters. If we choose two points (smartly), there are no remaining degrees of freedom.

Problem 1 (Hard)

There exists a ring of N cows ($1 <= N <= 10^5$, N even) evenly spaced on a circle. Each cow has a unique target cow, c[i]. The chords joining a cow to its target is drawn. We call a pair of cows orthogonal if the two chords intersect at right angles (inside the circle). Count the amount of orthogonal directed pair paths given N and the permutation.

Areas (just know they exist)

- Triangle (Heron's formula): s = (a + b + c) / 2; A = sqrt(s(s-a)(s-b)(s-c))
  - Many other variations
- General Polygon (Shoelace formula)



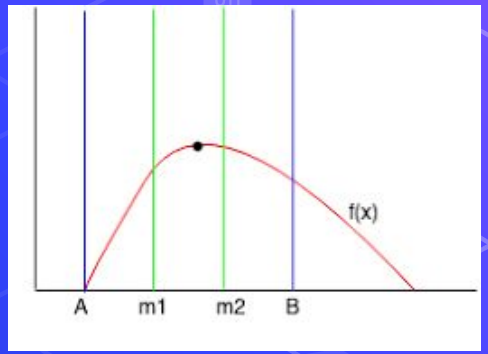
$$S = \frac{1}{2}\begin{vmatrix} x_1 & x_2 & x_3 & x_1 \\ y_1 & y_2 & y_3 & y_1 \end{vmatrix}$$

$$= \frac{1}{2}\left|x_1y_2 + x_2y_3 + x_3y_1 - x_2y_1 - x_3y_2 - x_1y_3\right|$$

Advanced Strategies

Ternary Search

- Used to find maximum or minimum of unimodal function
- Maintain search interval of maximum
- m1 = A + 1 * (B - A) / 3
- m2 = A + 2 * (B - A) / 3
- if f(m1) > f(m2) (increasing) A = m1
- else (decreasing) B = m2



Ternary Search and Pray

- If it seems like something might be unimodal in a geometry problem, there’s a good change it might be

Example Ternary Search Problem

https://www.spoj.com/problems/QUADAREA/cstart=60

Given 4 side lengths of a quadrilateral, determine maximum possible area

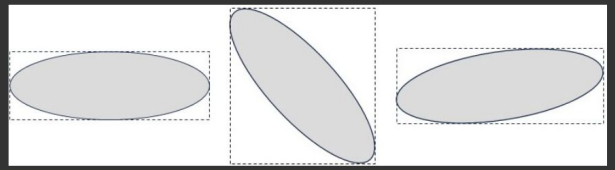
(other methods work)

Example Ternary Search Problem

NAQ problem

An *Ellipse* is a 2D geometric figure. It consists of the set of all points such that the sum of the distance from each point to two specific points is constant. The two specific points are called the *Foci* of the ellipse. The *Major Axis* of an ellipse is the diameter of the ellipse that runs through both foci. It is the longest diameter of the ellipse.

Given an ellipse's two foci and the length of its major axis, determine the coordinates of the tightest axis-aligned bounding box that can block out that ellipse.



Sweep Line

- Imagine we have a vertical line that traverses from -infinity to +infinity
- Each time we reach an “event” (i.e. point, intersection, etc), update the state (which is usually some data structure, eg. set)
- In practice, we solely visit the events, rather than checking every possible x value

Example Sweep Line

https://usaco.org/index.php?page=viewproblem2&cpid=943

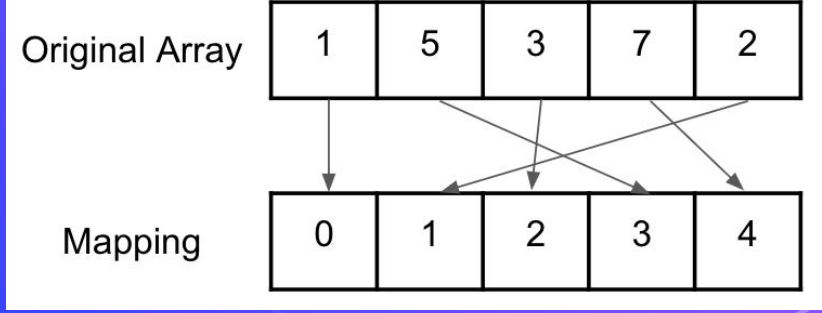
Set of line segments such that removing one will remove all intersections. Find that one

Also: https://open.kattis.com/problems/closestpair2

Coordinate Compression

- Basic idea is that if we have (relatively) few 2d points but really large coordinate points (>=10^9) we basically “compress” the points based on their relative orders
  - Point w/ smallest x-coordinate goes to 0, next one goes to 1, next goes to 2, etc.
  - The reason we do this is because >10^9 will often cause memory exceeded errors

Further Semantics



- Taking a 1d example, we basically just remap each point to it’s sorted index. This works because the relative order is preserved.
- In a 2d case, we do the same thing, but keep track of it’s x index and y index

Coordinate Compression

- Preconditions:
  - Generally speaking, all points must be distinct (in fact doubly distinct, there should be no two points that share the same x cord, or same y cord)
  - This is not strictly true, but it makes implementation much easier.

Coordinate Compression

- Preconditions:
  - Obviously, when we “map” or compress the points to the smaller range, the problem shouldn’t be changed.
    - Put differently, all that should matter is the relative order of the points
  - Bad example: compute the sum of all 2d points of a set (this depends on actual coordinates, not just relative order)
  - Good example: cows on a number line are either red or blue; find the number of pairs of cows that involve a blue cow after a red one
    - Only depends on which cows come after which

Code example

```
//assume all points are from 1 to 10^9
vector<pair<int, int>> c(n);

//remapped
vector<pair<int, int>> c_compressed(n);

//first develop the x index for each point
//the second part of the pair allows us to reverse it
vector<pair<int, int>> x_sort(n);
for (int i = 0; i < n; i++) x_sort[i] = { u1: c[i].first,  u2: i};
sort( first: x_sort.begin(),  last: x_sort.end());

//assign it back
for (int i = 0; i < n; i++) c_compressed[x_sort[i].second].first = i;

//doing y is similar
vector<pair<int, int>> y_sort(n);
for (int i = 0; i < n; i++) y_sort[i] = { u1: c[i].second,  u2: i};
sort( first: y_sort.begin(),  last: y_sort.end());

for (int i = 0; i < n; i++) c_compressed[y_sort[i].second].second = i;
```

Example Compression Problem

You are given N <= 5000 points where $1 <= x[i], y[i] <= 10^9$. All x[i] and y[i] are distinct. You are asked Q <= 10^5 queries that ask for the amount of points contained within an axis aligned rectangle, specified by $1 <= l[q], r[q], b[q], t[q] <= 10^9$. Please output Q lines, showing the number of points within each rectangle.