# Тинькофф А. СПБ. Геометрия

kik0s, isaf27 ft. peltorator

11 октября 2019

## 1 Геометрические примитивы

Всегда будем считать, что на плоскости введена декартова система координат: выбрана некоторая точка (начало координат), которая будет иметь координаты $(0, 0)$. Также выбраны две перпендикулярные оси $oX$ и $oY$. Тогда каждая точка будет однозначно задаваться парой чисел $(x, y)$, своими координатами.

Все углы измеряются в радианах. 1 радиан — это такой угол, что длина дуги равна радиусу окружности. $\pi$ радиан соответствуют 180 градусам.

Очень нежелательно использовать вещественные числа, потому что у них есть проблемы с точностью. Часто можно обойтись целыми или рациональными (пара целых), лучше делать так. Но если все таки пришлось использовать вещественные числа, то сравнения надо производить, используя $\varepsilon$, который должен иметь значение в зависимости от производимых операций и необходимой точности. Чаще всего он находится в диапазоне от $10^{-9}$ до $10^{-6}$. Тогда полезно реализовать функции для сравнения вещественных чисел $eq$ (равно), $ls$ (меньше), $le$ (не больше), $gt$ (больше), $ge$ (не меньше), используя этот $\varepsilon$, а потом производить все сравнения используя именно эти функции. Также при неточных вычислениях могут появиться отрицательные числа вместо нуля, так что стоит написать свою функцию корня, которая возвращает 0, если вызвана от отрицательного числа.

### 1.1 Структура вектор

Под вектором также будем понимать пару чисел $(x, y)$, являющуюся координатами вектора. Не будем отличать точку и вектор, то есть будем считать, что точка — это просто вектор из точки $(0, 0)$ в эту точку. Самые простые операции, которые можно производить с векторами — это сложение и умножение на коэфициент. В координатах это делается по двум естесственным равенствам: $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$, $k(x, y) = (kx, ky)$. Геометрически сложение легко представлять себе, как откладывание одного вектора от конца другого, а умножение на коэфициент, как сжатие/расширение вектора и возможно отражение, если $k < 0$.

Из теоремы Пифагора легко получить, что длина вектора $v = (x, y)$ равна $|v| = \sqrt{x^2 + y^2}$.

### 1.2 Скалярное и псевдоскалярное (векторное) произведения

Пусть у нас есть два вектора $v_1 = (x_1, y_1)$ и $v_2 = (x_2, y_2)$. Определим понятие направленного угла от вектора $v_1$ к вектору $v_2$ (предполагаем, что вектора ненулевые). Отложим вектора $v_1$ и $v_2$ из точки $(0, 0)$. Пусть просто угол между этими векторами равен $\alpha \in [0, 180°]$. Посмотрим, в какую сторону нужно вращать вектор $v_1$, чтобы его направление совпало с направлением вектора $v_2$. Тогда если это направление против часовой стрелки, то будем говорить, что $\angle(v_1, v_2) = \alpha$, иначе $\angle(v_1, v_2) = -\alpha$. В случае, если $\alpha = 180°$, будем считать, что мы вращаем против часовой стрелки. Таким образом, $\angle(v_1, v_2) \in (-180°, 180°]$.

Рассмотрим еще две операции, которые очень полезны при работе с векторами:

1. **Скалярное произведение**

   Обозначим за скалярное произведение следующую величину: $(v_1, v_2) = |v_1||v_2| \cos \angle(v_1, v_2) = |v_1| \cdot pr_{v_1}(v_2)$ (тут за $pr_{v_1}(v_2)$ обозначим направленную проекцию вектора $v_2$ на вектор $v_1$). Если мы знаем координаты векторов $v_1$ и $v_2$, то найти скалярное произведение можно по следующей формуле: $(v_1, v_2) = x_1x_2 + y_1y_2$.

2. **Псевдоскалярное произведение**

   Обозначим за псевдоскалярное (программисты почему-то называют его векторным, хотя векторное произведение — это вообще вектор, а не число, и существует только в трехмерном пространстве) произведение следующую величину: $[v_1, v_2] = |v_1||v_2| \sin \angle(v_1, v_2)$. Если мы знаем координаты векторов $v_1$ и $v_2$, то найти векторное произведение можно по следующей формуле: $(v_1, v_2) = x_1y_2 - x_2y_1$.

Не будем останавливаться на том, почему существует такая связь между координатами и описанными величинами. Приведем несколько простейших свойств:

1. $(v_2, v_1) = (v_1, v_2)$, $[v_2, v_1] = -[v_1, v_2]$;
2. $v_1 || v_2 \Leftrightarrow [v_1, v_2] = 0$;
3. $v_1 \perp v_2 \Leftrightarrow (v_1, v_2) = 0$;
4. Знак $[v_1, v_2]$ это знак угла $\angle(v_1, v_2)$;
5. Если $(v_1, v_2) > 0$, угол между $v_1$ и $v_2$ острый, если $= 0$, то прямой, иначе тупой.

Удобно реализовывать эти операции через операторы (google it). Вообще, есть очень много интересных моментов в реализации, которые мы здесь не рассматриваем. Чтобы посмотреть пример кода (не очень хорошего), можно посмотреть здесь: Integer Geometry, Double Geometry.

## 1.3 Поиск угла между векторами

Если вам обязательно нужно найти угол между двумя векторами, то в языке C++ лучше всего использовать функцию $atan2$. Эта функция принимает 2 аргумента: sin и cos угла, умноженные на некоторую константу и возвращает этот угол. Поэтому, $\angle(v_1, v_2) = atan2([v_1, v_2], (v_1, v_2))$. Будьте внимательны, угол возвращается в радианах и может принимать значения из отрезка $[-\pi, \pi]$. Таким образом он может быть равен $-\pi$, что тоже самое что и $\pi$.

К примеру, $\pi = atan2(0, -1)$.

## 1.4 Поворот вектора

Пусть $v = (a, b)$. Тогда заметим, что вектор $u = (-b, a)$ — это вектор $v$, который повернули на 90 градусов против часовой стрелки, потому что $(v, u) = (-b)a + ba = 0$, так что угол между ними прямой, а также $[v, u] = a^2 + b^2 > 0$, так что поворачивать $v$ надо по часовой стрелке. Тогда $v$ и $u$ — это некоторая новая ортогональная система координат, так что если нам надо повернуть $v$ на угол $\alpha$, то этот вектор можно задать формулой $v \cdot \cos\alpha + u \cdot \sin\alpha$.

## 1.5 Площадь многоугольника

Если у нас есть треугольник, две стороны которого это вектора $v_1, v_2$, то легко видеть, что площадь этого треугольника равна $\frac{|[v_1, v_2]|}{2}$ (следует из известной формулы площади треугольника с синусом). Число $\frac{[v_1, v_2]}{2}$ при этом будет равно площади треугольника со знаком.

Теперь пусть у нас есть многоугольник $A_1 A_2 \dots A_n$ (не обязательно выпуклый) и мы хотим посчитать его площадь. Выберем некоторую точку $O$ (можно просто взять точку $(0, 0)$). Тогда заметим, что сумма
$\frac{|[\overrightarrow{OA_1}, \overrightarrow{OA_2}] + [\overrightarrow{OA_2}, \overrightarrow{OA_3}] + \dots + [\overrightarrow{OA_{n-1}}, \overrightarrow{OA_n}] + [\overrightarrow{OA_n}, \overrightarrow{OA_1}]|}{2}$ будет равна площади многоугольника.

На пальцах это можно понять так: если нарисовать картинку и просуммировать прощади треугольников $OA_iA_{i+1}$, то все части вне мноугольника сократятся, а внутри мноугольника посчитаются с коэфициентом 1 или $-1$. Не будем останавливаться на формальном доказательстве.

Если посмотреть на эту сумму без модуля, то она будет положительной, если $A_1 A_2 \dots A_n$ это обход многоугольника против часовой стрелки и отрицательной, иначе.

Альтернативный способ подсчета площади многоугольника — через трапеции. Давайте опустим из каждой вершины перпендикуляр на ось $oX$ и посчитаем сумму ориентированных площадей прямоугольных трапеций, образованных этими перпендикулярями, осью $oX$ и сторонами многоугольника. По аналогичным соображениям это будет удвоенная ориентированная площадь многоугольника.

Тогда формула для площади многоугольника будет выглядеть так:
$\frac{|(X_1 - X_0)(Y_1 + Y_0) + (X_2 - X_1)(Y_2 + Y_1) + \dots + (X_n - X_{n-1})(Y_n + Y_{n-1}) + (X_0 - X_n)(Y_0 + Y_n)|}{2}$.

## 1.6 Общее уравнение прямой. Переход от него к двум точкам и наоборот. Вектор нормали.

"Школьный" метод хранения прямых в виде $y = kx + b$ нам не подойдет, потому что мы не хотим отдельно обрабатывать вертикальные прямые, которые нельзя задать таким способом.

Как известно, любая прямая задается уравнением $ax + by + c = 0$, для некоторых вещественных $a, b, c$, при этом $a^2 + b^2 > 0$. Если $a = 0$, то прямая горизонтальная, а если $b = 0$, то вертикальная. Рассмотрим вектор $n = (a, b)$. Будем называть такой вектор **вектором нормали** этой прямой. Заметим, что это некоторый ненулевой вектор, и поскольку уравнение прямой не меняется от домножения на константу, то от вектора нормали нам будет важно только направление. Когда точка $p = (x, y)$ лежит на прямой? Заметим, что $(n, p) = ax + by = -c$. Получается, что уравнение прямой через вектор нормали записывается как $(n, p) = -c$. Поскольку $(n, p) = |n| \cdot pr_n(p)$, получаем что на прямую попадают все точки, такие что $pr_n(p) = -\frac{c}{|n|}$, то есть все точки, проеция которых на вектор $n$ попадает в определенную точку. То есть наша прямая это просто прямая, перпендикулярная вектору нормали и проходящая через некоторую фиксированную точку.

Таким образом, вектор нормали всегда перпендикулярен своей прямой. Тогда **направляющий вектор** $v = (-b, a)$, который очевидно перпендикулярен вектору нормали (так как скалярное произведение $(n, v) = a(-b) + ba = 0$), будет параллелен прямой.

Если у нас есть две различные точки $p_1 = (x_1, y_1)$ и $p_2 = (x_2, y_2)$, которые лежат на прямой, то мы можем легко построить уравнение прямой. Заметим, что вектор $p_2 - p_1 = (x_2 - x_1, y_2 - y_1)$ будет параллелен прямой, поэтому вектор $(y_2 - y_1, x_1 - x_2)$ будет вектором нормали. Тогда мы можем взять $a = y_2 - y_1, b = x_1 - x_2$, а $c = -ax_1 - by_1$.

Мы можем найти любую точку лежащую на прямой. Просто заметим, что точка с координатами $(-\frac{ac}{a^2+b^2}, -\frac{bc}{a^2+b^2})$ лежит на прямой. Чтобы найти любую другую точку, лежащую на прямой мы можем к этой точке прибавить вектор $(-b, a)$, умноженный на любую константу.

Если забыть формулу, можно рассмотреть два случая. Если $b = 0$, то прямая вертикальная, то есть имеет вид $ax+c = 0$, тогда $x = \frac{-a}{c}$, а $y$ можно выбрать любым, так что пусть $y = 0$. Если же $b \neq 0$, то для любого $x$ существует единственный соответствующий ему $y$. Тогда возьмем $x = 0$. Уравнение прямой примет вид $by + c = 0$. Тогда $y = \frac{-b}{c}$.

## 1.7 Расстояние от точки до прямой. Проекция точки на прямую

Пусть есть некоторая точка $p = (x_0, y_0)$ и прямая, задаваемая уравнением $ax + by + c = 0$. Рассмотрим вектор нормали $n = (a, b)$. Поскольку его направление перпендикулярно прямой, расстояние от точки до прямой равно $|pr_n(p) - pr_n(s)|$, где точка $s$ это любая точка прямой. Перепишем: $|pr_n(p) - pr_n(s)| = |\frac{(n,p)}{|n|} - \frac{(n,s)}{|n|}| = |\frac{ax_0+by_0}{\sqrt{a^2+b^2}} - \frac{-c}{\sqrt{a^2+b^2}}| = \frac{|ax_0+by_0+c|}{\sqrt{a^2+b^2}}$. Получаем, что расстояние от $p$ до прямой записывается следующей простой формулой: $\frac{|ax_0+by_0+c|}{\sqrt{a^2+b^2}}$.

Как найти точку, в которую попадет проекция? Найдем расстояние $h$ от точки до прямой по предыдущей формуле. Рассмотрим вектор нормали $n$. Сделаем его длину равной $h$, при этом не изменив направления, то есть рассмотрим $n' = n\frac{h}{|n|}$. Заметим, что одна из двух точек $(p - n'), (p + n')$ попадет на прямую и будет являться проекцией. Проверим каждую из них и выберем из них ту, которая лежит на прямой, то есть ту, которая при подстановке в уравнение прямой дает 0.

## 1.8 Пересечение прямых

Пусть у нас есть две прямые, задаваемые уравнениями $a_1x+b_1y+c_1 = 0$ и $a_2x+b_2y+c_2 = 0$. Как найти их пересечение? Просто решим систему из двух линейных уравнений и найдем решение.

$$\begin{cases} a_1x + b_1y = -c_1 \\ a_2x + b_2y = -c_2 \end{cases}$$

... цепочка преобразований ...

$$\begin{cases} x = \frac{c_2b_1-c_1b_2}{a_1b_2-a_2b_1} \\ y = \frac{a_2c_1-a_1c_2}{a_1b_2-a_2b_1} \end{cases}$$

При этом заметим, что если знаменатель равен 0, то $a_1b_2 - a_2b_1$, что равно векторному произведению векторов нормали прямых, то есть равносильно тому, что прямые параллельны или совпадают. В первом случае точек пересечения не существует. Во втором случае прямые совпадают, что выполнено когда числители дробей равны 0. Если знаменатель не равен 0, то точка пересечения единственная и ее можно найти по уже описанной формуле.

## 1.9 Пересечение лучей, отрезков, расстояние до луча, до отрезка

TODO

### 1.10 Пересечение прямой и окружности

Самый удобный способ хранения окружности, это хранения точки ее центра $(x_0, y_0)$ и радиуса $r$. При этом само уравнение имеет вид $dist(p,(x_0,y_0)) = r \Leftrightarrow (x-x_0)^2 + (y-y_0)^2 = r^2$.

Пусть у нас есть некоторая окружность с центром $p = (x, y)$ и радиусом $r$, а также прямая задаваемая уравнением $ax + by + c = 0$. Мы хотим найти все точки пересечения этих двух объектов.

Найдем расстояние $h$ от центра $p$ до прямой и точку проекцию $s$ точки $p$ на прямую. Заметим, что если $h > r$, то окружность и прямая не пересекаются, если $h = r$, то прямая касается окружности и точка пересечения совпадает с $s$. Пусть теперь $h < r$. Заметим, что по теореме Пифагора расстояние от $s$ до точек пересечения равно $\sqrt{r^2-h^2}$. Тогда рассмотрим вектор $v = (-b, a)$, параллельный прямой. Сделаем его длину равной $\sqrt{r^2-h^2}$, то есть рассмотрим $v' = v\frac{\sqrt{r^2-h^2}}{|v|}$. Тогда две точки $(s - v')$ и $(s + v')$ будут являться точками пересечения окружности и прямой.

### 1.11 Пересечение двух окружностей

Пусть у нас есть две окружности с центрами $(x_1, y_1)$, $(x_2, y_2)$ и радиусами $r_1$ и $r_2$. Мы хотим найти все точки их пересечения, то есть все решения $(x, y)$ системы:

$$\begin{cases}(x-x_1)^2 + (y-y_1)^2 = r_1^2\\(x-x_2)^2 + (y-y_2)^2 = r_2^2\end{cases}$$

Вычтем из нижнего уравнения верхнее:

$$\begin{cases}(x-x_1)^2 + (y-y_1)^2 = r_1^2\\x(2x_1-2x_2) + y(2y_1-2y_2) + (x_2^2 + y_2^2 - r_2^2 - x_1^2 - y_1^2 + r_1^2) = 0\end{cases}$$

Заметим, что нижнее равенство это уравнение некоторой прямой, если точки $(x_1, y_1)$ и $(x_2, y_2)$ не совпадали. Если они совпадали, то задача не имеет смысла, потому что окружности концентрические. Поэтому мы свели задачу к пересечению первой окружности и некоторой прямой, которую мы можем решить с помощью предыдущего пункта.

### 1.12 Поиск касательных к окружности

Пусть есть точка $p = (x, y)$, мы хотим найти точки, в которых касательные из $p$ к окружности с центром в $c = (x_0, y_0)$ и радиусом $r$ касаются этой окружности. Если точка $p$ внутри окружности, то ответа нету. Иначе заметим, что из теоремы Пифагора расстояние от $p$ до этих точек касания равно $\sqrt{|p-c|^2 - r^2}$. Тогда рассмотрим окружность с центром в $p$ и таким радиусом и найдем ее пересечение с изначальной окружностью с помощью предыдущего пункта. Точки пересечения и будут ответом.

## 2 Различные базовые алгоритмы

### 2.1 Проверка на принадлежность точки многоугольнику

Пусть у нас есть многоугольник $A_1A_2\dots A_n$ (не обязательно выпуклый) и некоторая точка $P$. Мы хотим проверить, лежит ли точка $P$ внутри, на границе или снаружи многоугольника. Для того, чтобы проверить, что точка $P$ лежит на границе многоугольника переберем его стороны и для каждого отрезка проверим, лежит ли точка $P$ на нем. Для того, чтобы проверить, лежит ли точка внутри или снаружи многоугольника существует 2 метода.

1. Посмотрим на следующую сумму углов со знаком: $\sum\limits_{i=1}^{n} \angle(\overrightarrow{PA_i}, \overrightarrow{PA_{i+1}})$. Утверждается, что если точка попала вовнутрь многоугольника, то эта сумма будет равна $\pm 360^\circ$ (при этом знак $-$, если обход многоугольника по часовой стрелки, иначе знак $+$). Иначе эта сумма будет равна чему-то другому. Этот метод имеет плюс в том, что тут сложно запутаться. Минус состоит в том, что вычисление углов может быть неточным, из-за чего в крайних случаях метод может работать неправильно;

2. Выпустим произвольный луч из точки $P$. Пока предположим, что ни одна вершина на него не попала. Посчитаем количество сторон, пересекающих этот луч. Заметим, что если это количество нечетное, то точка лежит внутри многоугольника, иначе снаружи. Для того, чтобы найти луч, на который не попала ни одна из точек, рассмотрим луч выходящий в направлении вектора $(p, 1)$, где $p$ это любое простое число, большее чем $C$ — максимальная разность координат. На практике обычно можно взять $p = 10^9 + 7$.

Правда, в этой ситуации нам придется пересекать лучи с отрезками. Намного удобнее пересекать горизонтальный луч и отрезок, но тогда придется правильно обрабатывать случаи, когда луч проходит через вершину, что не очень сложно, если разобраться на бумажке.

## 2.2 Поиск выпуклой оболочки точек на плоскости

Пусть у нас есть $n$ точек $A_1, A_2, \ldots, A_n$ на плоскости. Мы хотим найти их выпуклую оболочку: минимальный по площади выпуклый многоугольник, содержащий все точки. Нетрудно понять, что вершинами выпуклой оболочки будут являться некоторые из этих $n$ точек. Рассмотрим 2 алгоритма, строящие выпуклые оболочки. Оба алгоритма будут начинаться одинаково. Рассмотрим точку, имеющую минимальную $y$-координату, а среди всех таких выберем точку имеющую минимальную $x$-координату. Заметим, что эта точка обязательно будет лежать на выпуклой оболочке.

### 2.2.1 Алгоритм Джарвиса (заворачивания подарка)

Рассмотрим выбранную точку. Далее рассмотрим векторы из выбранной точки во все остальные. Среди таких векторов выберем тот, который имеет минимальный угол с предыдущей стороной (в качестве первой стороны рассмотрим вектор $(1, 0)$). Заметим, что мы перейдем к следующей точке, лежащей на выпуклой оболочке в порядке против часовой стрелки. Рассмотрим опять вектора из новой точки во все остальные и выберем из них минимальный по углу с предыдущей стороной. Будем переходить так от точки к точке пока не дойдем до точки, с которой мы стартовали алгоритм.

Кратко наш алгоритм можно описать так: мы заворачивали множество точек в выпуклый многоугольник, переходя кбщее время работы этого алгоритма $O(kn)$, где $k$ равно количеству точек в выпуклой оболочке.

## 2.3 Алгоритм Грэхэма

Рассмотрим выбранную точку. Пусть не умоляя общности это $A_1$. Рассмотрим все остальные точки $A_2, A_3, \ldots, A_n$ и отсортируем их по углу между осью $oX$ и вектором $\overrightarrow{A_1A_i}$. Пусть теперь все точки расположены именно в таком порядке. Тогда будем перебирать $i$ от 2 до $n$ и каждый раз будем поддерживать в стеке множество точек, лежащих на выпуклой оболочке точек $A_1, A_2, \ldots, A_i$ в порядке обхода начиная от $A_1$. Изначально в стеке будет лежать точка $A_1$. Осталось понять, как обновлять стек при добавлении точки $A_{i+1}$. Заметим, что нам нужно будет убрать из стека несколько последних точек и затем положить в стек $A_{i+1}$. Поэтому пока в стеке больше чем одна точка, будем проверять, должны ли мы выкинуть последнюю. Пусть последняя точка в стеке имеет номер $y$, а предпоследняя $x$. Заметим, что если направленный угол $\angle(\overrightarrow{A_xA_y}, \overrightarrow{A_xA_{i+1}}) \le 0$, то мы должны вынуть из стека точку $A_y$, а иначе нужно прекратить вынимать точки и перейти к следующему $i$.

Заметим, что часть алгоритма, после сортировки точек работает за линейное время, потому что каждая точка может быть положена и вынута из стека $\le 1$ раза.

Время работы алгоритма — $O(n \log n)$.

## 2.4 Локализация точки в выпуклом многоугольнике

Пусть у нас есть выпуклый многоугольник $A_1A_2 \ldots A_n$. Рассмотрим его триангуляцию на треугольники $\Delta A_1A_2A_3, \Delta A_1A_3A_4, \ldots, \Delta A_1A_{n-1}A_n$. Тогда пусть у нас есть запросы: есть точка $P$. Найти какому треугольнику $\Delta A_1A_iA_{i+1}$ из этих она принадлежит или сказать, что точка лежит вне многоугольника.

Рассмотрим векторы $\overrightarrow{A_1A_2}, \overrightarrow{A_1A_3}, \ldots, \overrightarrow{A_1A_n}$. Заметим, что они отсортированы по углу. Рассмотрим вектор $\overrightarrow{A_1P}$ и найдем его *upper_bound* в этом массиве из $n-1$ векторов. Это можно сделать стандартным бинарным поиском. Тогда пусть мы нашли такое $2 \le i < n$, что $\angle(\overrightarrow{A_1A_i}, \overrightarrow{A_1P}) \ge 0$ и $\angle(\overrightarrow{A_1P}, \overrightarrow{A_1A_{i+1}}) > 0$. Просто проверим, попала ли точка $P$ в треугольник $\Delta A_1A_iA_{i+1}$ или нет. Если попала, то точка $P$ лежит внутри многоугольника и мы локализовали ее положение, иначе точка $P$ дежит снаружи многоугольника. При этом такой запрос мы обрабатываем за время $O(\log n)$.

## 2.5 Поиск касательных к выпуклому многоугольнику, параллельных данной прямой

Пусть у нас есть выпуклый многоугольник $A_1A_2 \ldots A_n$ (пусть обход многоугольника против часовой стрелки) и некоторая прямая. Мы хотим найти 2 прямые, которые касаются многоугольника и при этом параллельны данной прямой. Найдем точки, в которых эти прямые будут касаться многоугольника. Рассмотрим вектор $v$ — направляющий вектор данной прямой. Рассмотрим вектора $\overrightarrow{A_1A_2}, \overrightarrow{A_2A_3}, \ldots, \overrightarrow{A_nA_1}$ именно в таком порядке. Заметим, что они идут в порядке против часовой стрелки. Тогда если рассмотреть между какими соседними позициями в этом порядке находятся $v$ и $-v$, то эти позиции соответсвуют тем вершинам, в которых касательные касаются многоугольника. Для того, чтобы найти эти позиции надо сделать обычный *lower_bound*.

## 2.6 Поиск касательных из точки к выпуклому многоугольнику

Пусть у нас есть выпуклый многоугольник $A_1A_2\dots A_n$ (пусть обход многоугольника против часовой стрелки) и некоторая точка $P$. Мы хотим найти 2 точки $A_i$ и $A_j$, такие что прямые $PA_i$ и $PA_j$ это касательные к многоугольнику или сказать, что точка $P$ лежит внутри многоугольника. Для проверки того, что точка лежит внутри многоугольника, будем использовать проверку за логарифм.

Пусть мы поняли, что точка лежит снаружи. Назовем сторону $A_iA_{i+1}$ левой, если $\angle(\overrightarrow{PA_i}, \overrightarrow{PA_{i+1}}) > 0$ или $\angle(\overrightarrow{PA_i}, \overrightarrow{PA_{i+1}}) = 0$ и $PA_i < PA_{i+1}$ и правой, иначе. Заметим, что если пойдем вдоль границы многоугольника, то на нем будут подряд идущие левые стороны и подряд идущие правые стороны, при этом те вершины, в которых тип поменяется, будут подходить как ответ. Как найти границы, в которых тип меняется?

Рассмотрим луч $PA_1$. Найдем его вторую сторону которая пересекается с этим лучом. Для этого заметим, что точки $A_2, A_3, \dots, A_i$ лежат с одной стороны от этого луча, а $A_{i+1}, \dots, A_n$ с другой. Поэтому этот момент можно найти простым бинпоиском. Теперь посмотрим типы сторон начиная от $A_1A_2$ до $A_iA_{i+1}$ и на типы строк от $A_iA_{i+1}$ до $A_nA_1$. Заметим что на каждой из этой дуг теперь типы сначала одни, потом в точке касания меняются на другие и так до конца дуги. Поэтому на каждой из дуг можно сделать бинпоиск и найти точку касания.

## 2.7 Пересечение полуплоскостей

Заметим, что если у нас есть некоторая прямая, задаваемая уравнением $ax + by + c = 0$, то множества точек, удовлетворяющие неравенствам $ax+by+c \leq 0$ и $ax+by+c \geq 0$ это полуплоскости, на которые эта прямая разделяет плоскость. Будем задавать поплоскости уравнениями $ax + by + c \leq 0$, потому что для полуплоскостей второго вида можно поменять знаки всех коэффициентов. Пусть у нас задано много полуплоскостей. Мы хотим найти их пересечение, то есть как-то описать множество точек, удовлетворяющее всем неравенствам.

Для простоты предположим сначала, что нету вертикальных прямых. Рассмотрим множество прямых, полуплоскости которых "смотрят" вниз и полуплоскости которых "смотрят" вверх.

Пусть у нас есть множество полуплоскостей, которые "смотрят" в одну сторону. Пусть для простоты они смотрят вверх. Тогда рассмотрим все прямые, которые разделяют плоскость. Мы хотим найти такую верхнюю огибающую всех прямых, то есть выбрать в каждой $x$ координате максимальную точку с такой $x$ координатой, лежащую на одной из прямых. Этот метод также используется в очень известном методе $Convex\,hull\,trick$. Попробуем повторить что-то типа Алгоритма Грэхэма. Отсортируем все прямые по углу с осью $oX$. Будем перебирать прямые в таком порядке и в стеке поддерживать верхнюю огибающую для тех прямых, которые мы перебрали. Для каждой прямой будем поддерживать минимальную $x$ координату, начиная с которой эта прямая максимальная. При добавлении новой прямой мы должны будем выкинуть из стека несколько последних прямых и добавить новую. Пока стек непустой будем вынимать из него последнюю прямую. Рассмотрим точку пересечения этой прямой с нашей. Если $x$ координата этого пересечения будет не больше чем минимальная $x$ координата, начиная с которой последняя прямая из стека была максимальной, то последняя прямая в стеке больше никогда не будет максимальной и мы должны вынуть ее из стека. После этого надо добавить эту прямую в конец стека и минимальная $x$ координата начиная с которой эта прямая будет максимальной будет равна $x$ координате последней точки пересечения (во время того как мы вынимали последнюю прямую из стека).

Далее надо пересечь две цепочки: полуплоскостей смотрящих вниз и смотрящих вверх. Для этого переберем все пары звеньев верхней и нижней цепочки и попробуем их пересечь. Таким образом найдется $\leq 2$ точек пересечения, которые надо будет добавить в фигуру пересечения и удалить все звенья левее и правее добавленных точек. Можно перебирать лишь звенье верхней цепочки, а в нижней соответствующее искать бинпоиском.

Для того, чтобы разобраться с вертикальной прямой, можно попробовать пересечь ее с полученной конструкцией (что очень неприятно), а можно просто выбрать любую пару перпендикулярных векторов и сказать, что новые направления осей это эти вектора и в зависимости от этого выбрать какие полуплоскости смотрят вверх, а какие вниз.

Общее время работы составляет $O(n \log n)$.

## 2.8 Рандомизированный алгоритм поиска точки, лежащей в пересечении полуплоскостей за $O(n)$

TODO

## 2.9 Сумма Минковского многоугольников

TODO

# 3 Формула Пика

Пусть у нас есть многоугольник $A_1A_2\ldots A_n$, такой что координаты $A_i$ целые для всех $1 \le i \le n$. Обозначим за $S$ площадь многоугольника, за  количество целых точек **строго внутри** многоугольника, за  количество целых точек на границе многоугольника (сами вершины тоже считаются). Тогда утверждается, что выполнено следующее равенство $S = \; + \frac{}{2} - 1$.

Для доказательства заметим, что если формула верна для двух многоугольников, имеющих общую сторону, то для их объединения тоже верна. Это так, потому что $({}_1 + \frac{{}_1}{2} - 1) + ({}_2 + \frac{{}_2}{2} - 1) = (B_1 + B_2) + \frac{{}_1 + {}_2}{2} - 2$. Точки, которые были внутри одного из двух многоугольников, остались внутри. Точки, которые были на границе одного из двух многоугольников, если они не лежали на отрезке общей стороны, то останутся на границе и действительно посчитаются 1 раз. Точки, лежащие на общей стороне, которые не совпадают с концами отрезка, теперь станут внутренними, но заметим, что они посчитаются с коэфициентами $\frac{1}{2}$ в каждом из многоугольников, поэтому их коэфициент будет равен 1. Две точки, концы общего отрезка, останутся на границе, но теперь внесут вклад 2 в сумму, а должны 1. Но заметим, что у нас вычитается $-2$, и сократив 1, мы получим, что $S_1 + S_2 = \; + \frac{}{2} - 1$. Таким образом мы можем доказать для треугольника и тогда мы докажем для всех многоугольников, потому что любой многоугольник можно триангулировать. Чтобы доказать для треугольника можно заметить, что треугольник можно разбить на прямоугольные треугольники, для которых формула Пика очевидна.

Заметим, что теперь по многоугольнику мы можем посчитать количество целых точек внутри, потому что площадь мы можем вычислить за линейное время и остается вычислить количество точек на границе мноугольника. Заметим, что на отрезке, соединяющем точки с координатами $(x_1, y_1)$ и $(x_2, y_2)$ лежит $1 + gcd(|x_1 - x_2|, |y_1 - y_2|)$ точек. Поэтому мы можем просто просуммировать эти величины по всем сторонам (не забывайте, что концы посчитаются по 2 раза) и получим . Тогда из формулы Пика мы получим .