

Геометрия. Базовые алгоритмы

Вадим Бездушный

Факультатив Алгоритмики, 2017
Подготовка к обласной олимпиаде, 2018

План лекции

Дробные числа в памяти компьютера

Векторы и операции с ними

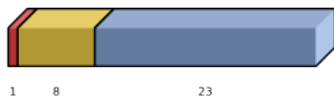
Прямые и отрезки

Многоугольники

Задачи

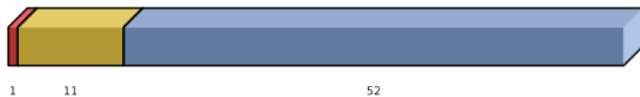
Представление числа с плавающей запятой

Single



- sign bit
- exponent
- mantissa

Double



Extended



Типы дробных чисел в C++

float	double	long double
32 бита (4 байта)	64 бита (8 байт)	80 бит (10 байт)
23 бит мантииссы	52 бита мантииссы	52 бита мантииссы
$\frac{\log(2^{24})}{\log(10)} \approx 7.22$	$\frac{\log(2^{53})}{\log(10)} \approx 15.95$	$\frac{\log(2^{64})}{\log(10)} \approx 19.27$
Недостаточная точность для задач геометрии!	Точность достаточная для большинства задач	Нужна иногда (когда нужно вывести числа порядка 10^9 с точностью 10^{-9})

Сравнение дробных чисел

```
float a = 0.15 + 0.15  
float b = 0.1 + 0.2  
if(a == b) // can be false!  
if(a >= b) // can also be false!
```

Сравнение дробных чисел

```
float a = 0.15 + 0.15
float b = 0.1 + 0.2
if(a == b) // can be false!
if(a >= b) // can also be false!

const double EPS = 1e-6;
...
bool equal(double a, double b)
{
    return abs(a - b) < EPS;
}
```

Дробные ноль и бесконечность

```
double x = 0.0, inf = 1/x;  
cout << 1/x << " " << -1/x; // inf -inf  
cout << inf * 0 << " " << -inf * 0; // -nan -nan
```

Ввод и вывод дробных чисел в C++

```
#include <iomanip>
...
float fx = 1./25;
double dx = 1./25;
cout << setprecision(15);
cout << fx; // 0.0399999991059303
cout << dx; // 0.04

cout << setprecision(20) << fixed;
cout << fx; // 0.03999999910593032837
cout << dx; // 0.0400000000000000000083
```


Функции библиотеки <cmath>

```
cout << fixed << setprecision(15);  
double angle = 30;  
double angle_rad = 30 * M_PI/180;  
double back_to_angle = angle_rad * 180/M_PI;  
// 29.999999999999996
```

Функции библиотеки <cmath>

```
cout << fixed << setprecision(15);  
double angle = 30;  
double angle_rad = 30 * M_PI/180;  
double back_to_angle = angle_rad * 180/M_PI;  
// 29.999999999999996  
  
double x = 1e9;  
cout << x << endl; // 1000000000.0000000000  
cout << sqrt(x) * sqrt(x); // 999999999.9999998808  
if(1e4 == sqrt(1e4) * sqrt(1e4)) // true  
if(1e5 == sqrt(1e5) * sqrt(1e5)) // false
```

Plan

Дробные числа в памяти компьютера

Векторы и операции с ними

Прямые и отрезки

Многоугольники

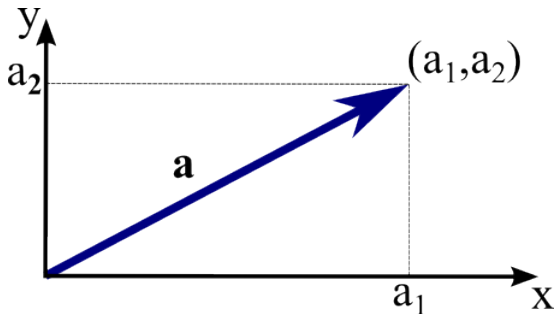
Задачи

Понятие вектора

Вектор - направленный отрезок прямой.

Будем считать, что начало отрезка всегда лежит в точке $(0, 0)$

Вектор будет задаваться координатами точки конца (x, y) .

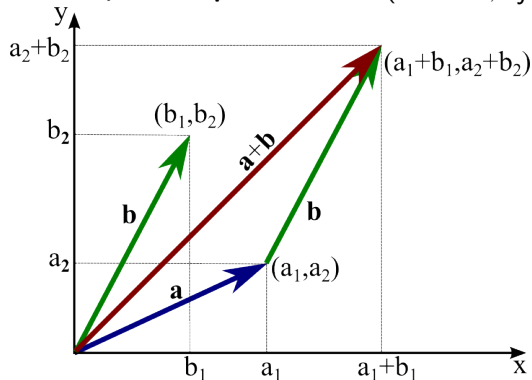


Свойства вектора

Длина вектора: $|a| = \sqrt{a_x^2 + a_y^2}$

Сумма векторов: $a + b = (a_x + b_x, a_y + b_y)$

Разница векторов: $a - b = (a_x - b_x, a_y - b_y)$



Скалярное произведение векторов

- ▶ $\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta$
- ▶ $\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y$
- ▶ Быстрый способ найти $\cos \theta = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|}$
- ▶ Быстрый способ найти информацию об угле (тупой, острый или прямой). Достаточно сравнить $\mathbf{a} \cdot \mathbf{b}$ и 0.

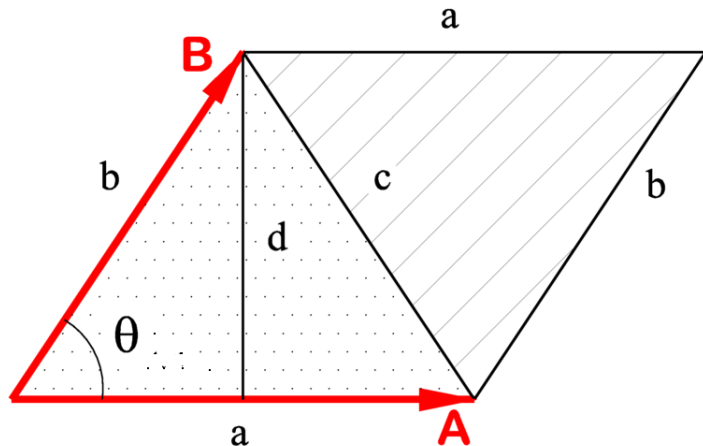
Псевдоскалярное произведение

- ▶ $\mathbf{a} \wedge \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}| \sin \theta$
- ▶ $\theta = \angle(\mathbf{a}, \mathbf{b})$
угол вращения (против часовой стрелки) от \mathbf{a} к \mathbf{b}
- ▶ $\mathbf{a} \wedge \mathbf{b} = \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix} = a_x \cdot b_y - b_x \cdot a_y$

Псевдоскалярное произведение

- ▶ $\mathbf{a} \wedge \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}| \sin \theta$
- ▶ $\theta = \angle(\mathbf{a}, \mathbf{b})$
угол вращения(против часовой стрелки) от \mathbf{a} к \mathbf{b}
- ▶ $\mathbf{a} \wedge \mathbf{b} = \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix} = a_x \cdot b_y - b_x \cdot a_y$
- ▶ Если сравнить $\mathbf{a} \wedge \mathbf{b}$ с нулём, то можно узнать расположение вектора \mathbf{a} относительно вектора \mathbf{b}
- ▶ $\mathbf{a} \wedge \mathbf{b}$ равен площади параллелограмма, натянутого на эти вектора
- ▶ ** Угол между двумя векторами: $\text{atan2}(\mathbf{a} \wedge \mathbf{b}, \mathbf{a} \cdot \mathbf{b})$

Псевдоскалярное произведение



C++ Vector

```
struct Vec {  
    double x, y;  
  
    Vec(double x, double y) :x(x), y(y) {}  
    Vec() :x(0), y(0) {}  
  
    inline double len_sqr() {  
        return x*x + y*y;  
    }  
    inline double len() {  
        return sqrt(x*x + y*y);  
    }  
    ...  
};
```

C++ Vector

```
...  
friend Vec operator*(const Vec & a, double k) {  
    return Vec(a.x*k, a.y*k);  
}  
inline Vec normal() {  
    if (len_sqr() == 0) return *this;  
    double inv = 1 / len();  
    return Vec(x*inv, y*inv);  
}  
inline Vec rot(double angle) {  
    double sn = sin(angle);  
    double cs = cos(angle);  
    return Vec(x*cs - y*sn, x*sn + y*cs);  
}  
...
```

C++ Vector

```
...  
friend Vec operator+(const Vec & a, const Vec & b) {  
    return Vec(a.x + b.x, a.y + b.y);  
}  
friend Vec operator-(const Vec & a, const Vec & b) {  
    return Vec(a.x - b.x, a.y - b.y);  
}  
friend double operator*(const Vec & a, const Vec & b) {  
    return a.x*b.x + a.y*b.y;  
}  
friend double operator^(const Vec & a, const Vec & b) {  
    return a.x*b.y - a.y*b.x;  
}  
}
```

Plan

Дробные числа в памяти компьютера

Векторы и операции с ними

Прямые и отрезки

Многоугольники

Задачи

Пересечение двух прямых через уравнения прямых



Пересечение двух прямых через уравнения прямых

$$\begin{cases} A_1x + B_1y + C_1 = 0 \\ A_2x + B_2y + C_2 = 0 \end{cases}$$

$$\begin{cases} x = -\frac{\begin{vmatrix} C_1 & B_1 \\ C_2 & B_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}}, y = -\frac{\begin{vmatrix} A_1 & C_1 \\ A_2 & C_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}} \end{cases}$$

Эта система имеет решения, если прямые (1) и (2) не параллельные.

Пересечение двух прямых через уравнения прямых

1. Строим прямые по отрезкам
2. Проверяем пересекаются ли прямые
3. Находим точку пересечения прямых
4. Проверяем принадлежит ли точка отрезкам

Plan

Дробные числа в памяти компьютера

Векторы и операции с ними

Прямые и отрезки

Многоугольники

Задачи

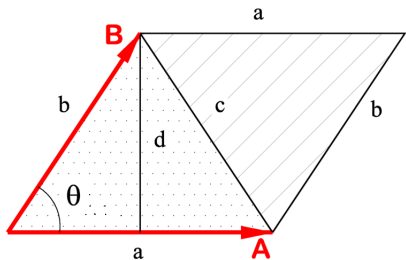
Площадь треугольника

Как найти площадь треугольника по координатам?

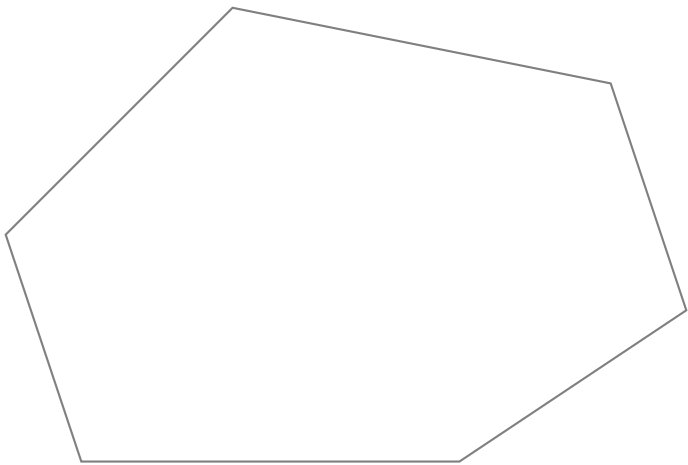
Площадь треугольника

Как найти площадь треугольника по координатам?

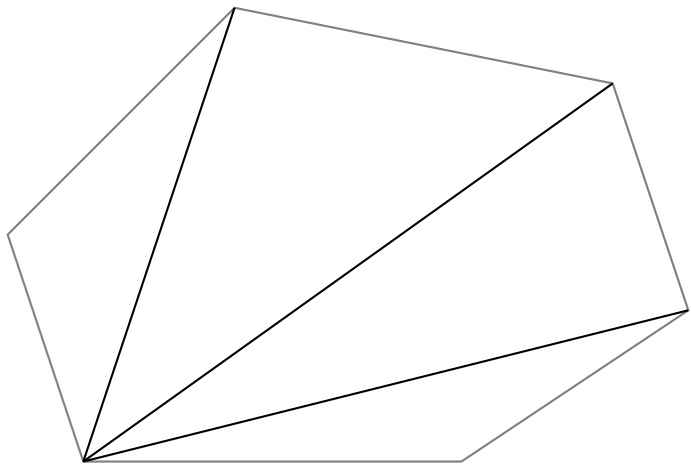
$$S(A, B, C) = \left| \frac{1}{2} (\vec{AB} \wedge \vec{AC}) \right|$$



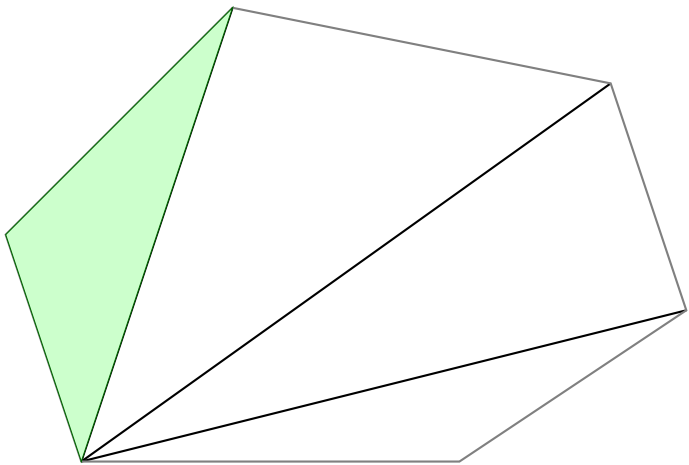
Площадь многоугольника



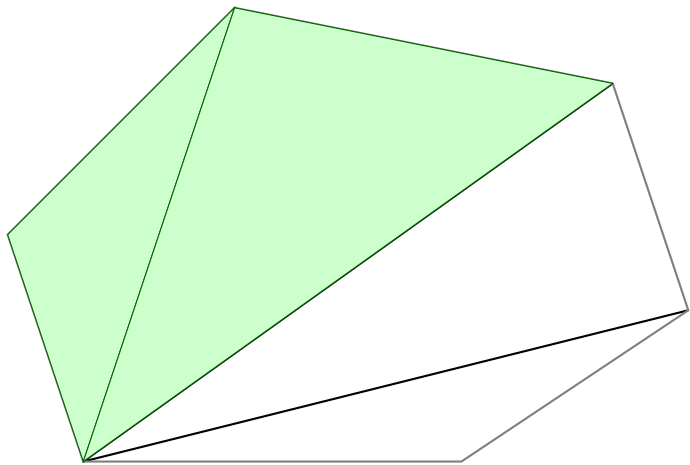
Площадь многоугольника



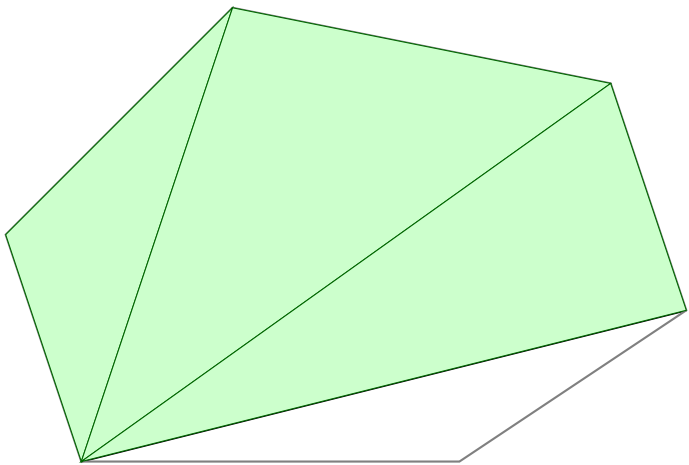
Площадь многоугольника



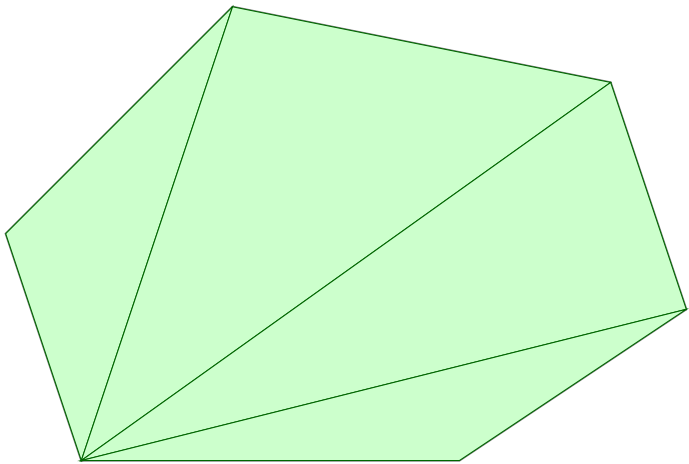
Площадь многоугольника



Площадь многоугольника

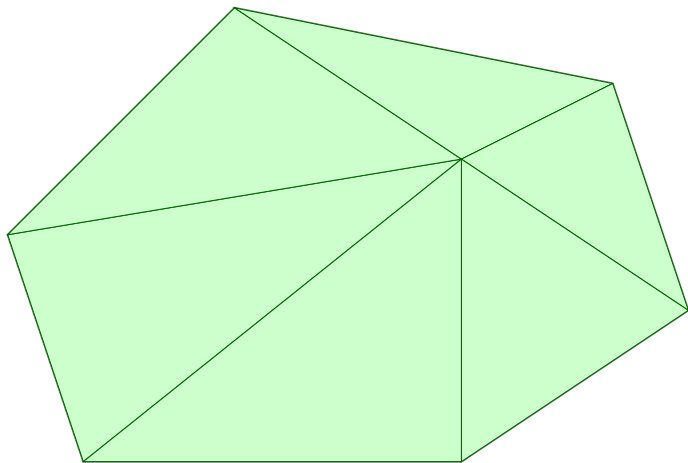


Площадь многоугольника

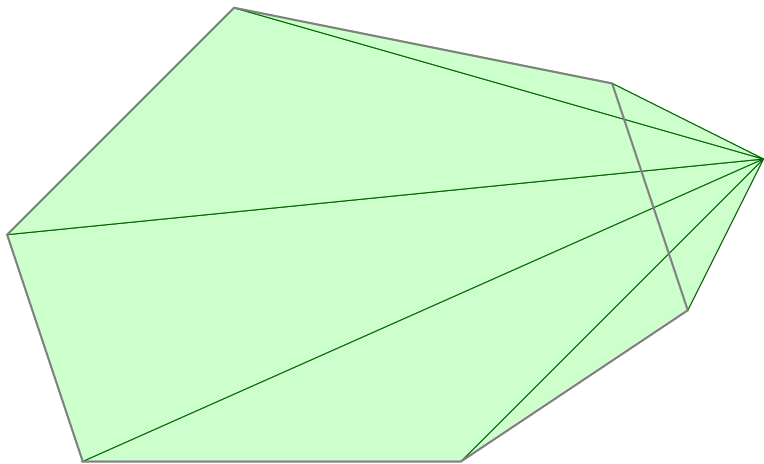


Принадлежность точки многоугольнику

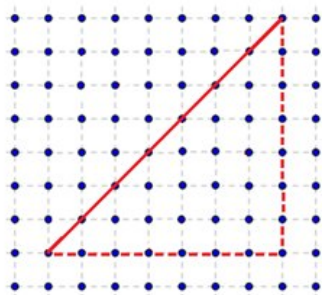
Как проверить принадлежит ли точка многоугольнику?



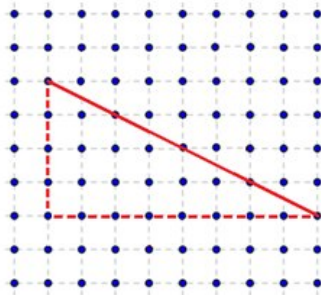
Принадлежность точки многоугольнику



Целые точки



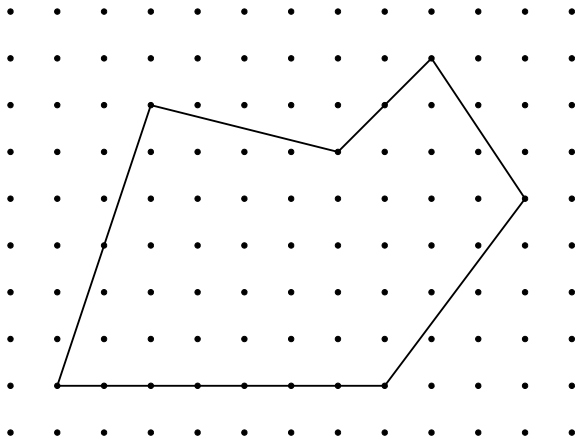
$$\text{НОД}(7, 7) + 1 = 8$$



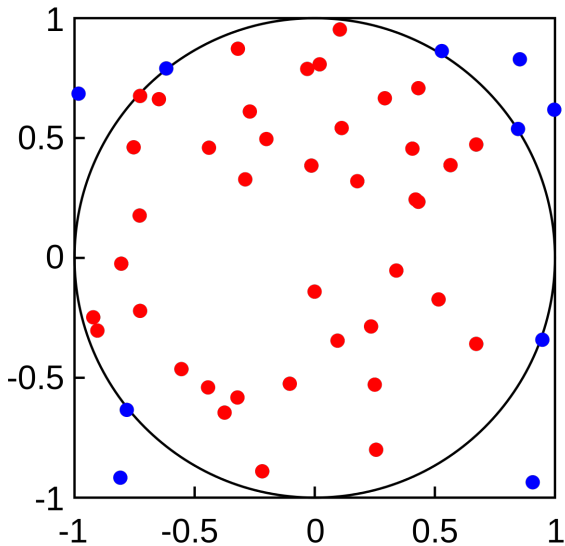
$$\text{НОД}(4, 8) + 1 = 5$$

Теорема Пика

$$S = N_{inside} + \frac{N_{bound}}{2} - 1$$



Метод Монте-Карло



Plan

Дробные числа в памяти компьютера

Векторы и операции с ними

Прямые и отрезки

Многоугольники

Задачи

Задача №1

Четыре точки задают прямоугольник. Дано три точки,
найти 4-ую

<https://goo.gl/xE17Bm>

Задача №2

На лесной опушке растет дружная семейка грибов. Местоположение каждого гриба задается координатами X , Y , а шляпка гриба имеет радиус R .

Когда идет дождь, радиус шляпки каждого гриба непрерывно и равномерно увеличивается со скоростью 1 сантиметр в минуту.

Когда дождь заканчивается (а он идет не более T минут), шляпки прекращают расти. Если во время дождя шляпки двух грибов соприкоснулись, то они немедленно перестают расти, чтобы не навредить друг другу.

Грибы очень дружные, поэтому если перестают расти два гриба, то и все остальные тоже не растут.

Требуется посчитать, на сколько сантиметров увеличился радиус шляпки каждого гриба после завершения дождя. http://acmp.ru/index.asp?main=task&id_task=699

Задача №3

У вас есть выпуклый многоугольник P , вершины которого расположены в n различных точках p_1, p_2, \dots, p_n .

Точка p_i имеет координаты (x_i, y_i) на плоскости.

Точки перечислены в порядке обхода по часовой стрелке.

Задана точка (x_0, y_0) . Найти кратчайшее расстояние от точки (x_0, y_0) до многоугольника P .

Задача №4

На поле боя размещены N имперских штурмовиков. Поле боя представляет собой плоскость с прямоугольной системой координат. Каждый штурмовик задан своими координатами (x, y) на этой плоскости.

У Хана Соло есть новейшая двусторонняя лазерная пушка для сражения с этими штурмовиками. Она расположена в точке с координатами (x_0, y_0) . За один выстрел она способна поразить всех штурмовиков, находящихся на некоторой прямой, проходящей через точку (x_0, y_0) .

Требуется определить, за какое минимальное количество выстрелов Хан Соло сможет уничтожить всех штурмовиков.

<http://codeforces.com/problemset/problem/514/B>

Задача №5

У вас есть выпуклый многоугольник P , вершины которого расположены в n различных точках p_1, p_2, \dots, p_n . Точка p_i имеет координаты (x_i, y_i) на плоскости. Точки перечислены в порядке обхода по часовой стрелке.

Вы можете выбрать вещественное число D и передвинуть каждую из вершин многоугольника из начального положения в любую точку на расстоянии не больше D .

Найдите максимальное значение D такое, что независимо от того, как вы передвинете вершины, многоугольник не пересечет сам себя и останется выпуклым.

<http://codeforces.com/problemset/problem/772/B>

Задача №6

На нескінченному полі знаходиться будівля, що має форму випуклого багатокутника. До однієї з вершин цього багатокутника на ланцюгу довжини L прив'язують вівцю. Знайти площу на якій зможе пастись вівця. Стіни будівлі є суцільними, вівця не може потрапити всередину будівлі. Вважається, що вівця може випасатись в будь-якій точці строго зовні будівлі, до якої може дійти.

https://netoi.org.ua/index_ua.php?lng=ua&cid=1593

Задача №7*

Дано N разных точек $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

Гарантируется, что никакие три точки не лежат на одной прямой.

Выбрать из них 4 точки, так чтобы площадь четырехугольника построенного на этих точках была максимальна

<http://codeforces.com/problemset/problem/340/B>

Задача №8*

Недавно Роман, зайдя в класс, увидел, что на доске нарисовано N точек. Разумеется, он сразу задумался, сколько существует троек из этих точек, которые являются вершинами равнобедренных треугольников.

<https://www.e-olymp.com/ru/problems/461>

Задача №9*

Задано поле N на M (из 0 и 1). На нем нарисована фигура (квадрат, круг или треугольник (произвольный)). Фигура состоит из единиц. Определить тип фигуры.

[http:](http://acm.timus.ru/problem.aspx?num=1378&locale=ru)

[//acm.timus.ru/problem.aspx?num=1378&locale=ru](http://acm.timus.ru/problem.aspx?num=1378&locale=ru)

Вопросы!

Вопросы?

Почитайте дома

Если много свободного времени:

<https://www.e-olymp.com/ru/contests/8947>

<https://www.e-olymp.com/ru/contests/9009>

Более сложные алгоритмы и их реализация:

<http://algolist.manual.ru/maths/geom/>

Лекции по линейной алгебре[Eng]:

https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab