RUCODE festival

Дивизион D. День 3
Вычислительная геометрия

Лектор: Дмитрий Руденко

Базовые понятия и операции

Точка

```
struct Point {
    double x;
    double y;
}
```

Базовые понятия и операции

Точка

```
struct Point {
    double x;
    double y;
}
```

```
if (point1.x == point2.x) {
    // do some staff
} else {
    // do another staff
}
```

Базовые понятия и операции

Точка

```
struct Point {
    double x;
    double y;
}
```

```
if (poi          .x) {
    // do
} else {
    // d          ff
}
```

Базовые понятия и операции

Точка

```
if (fabs(point1.x - point2.x) < EPS) {
    // do some staff
} else {
    // do another staff
}
```

```
p1.x = 25.12857391413;
p2.x = 25.12857385032;

double EPS = 0.000001;

// fabs(p1.x - p2.x) = 0.000000...
```

Базовые понятия и операции

Точка (Лексикографический порядок)

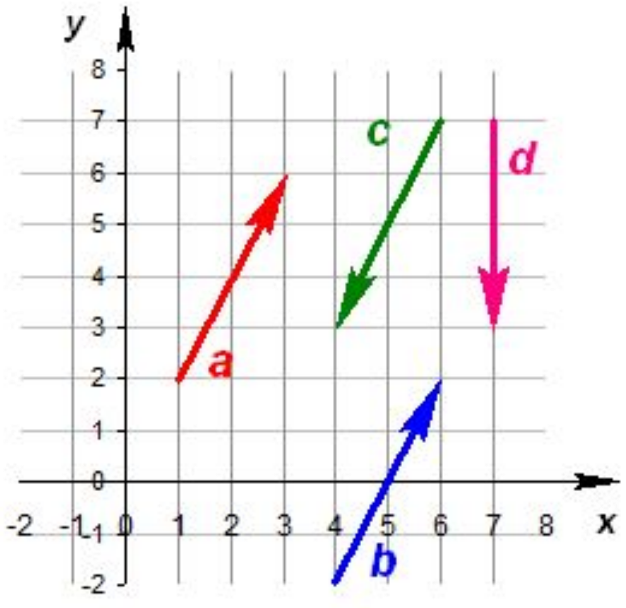
```
Точка A меньше B, если:
(a.x < b.x) || (a.x == b.x && a.y < b.y)
```

Базовые понятия и операции

Вектор

```
struct Vector {
    double x;
    double y;
}
```

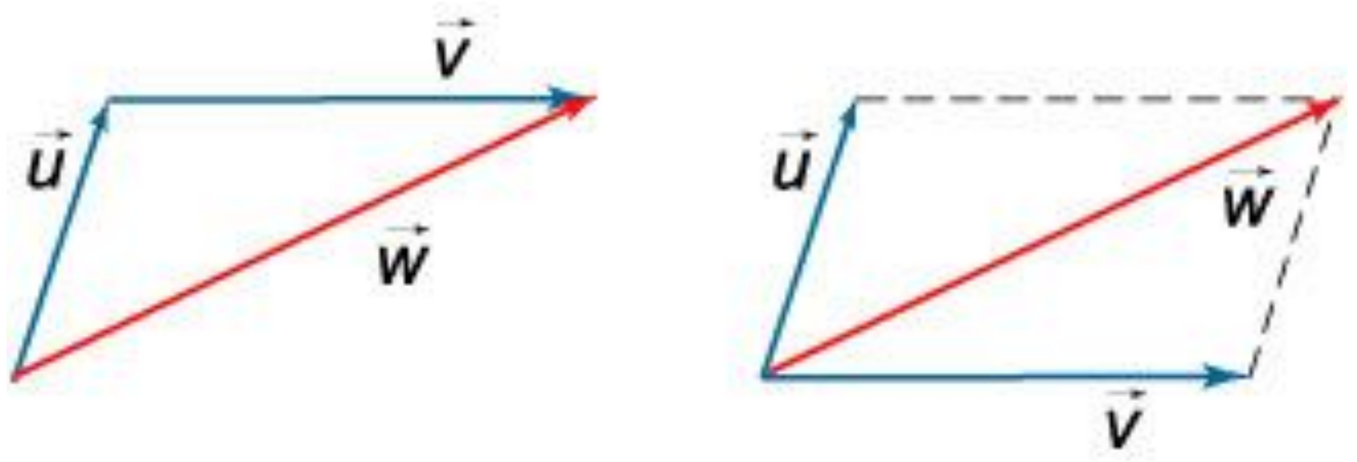
```
Vector(Point begin, Point end) {
    this->x = end.x - begin.x;
    this->y = end.y - begin.y;
}
```



Базовые понятия и операции

Вектор (Операции)

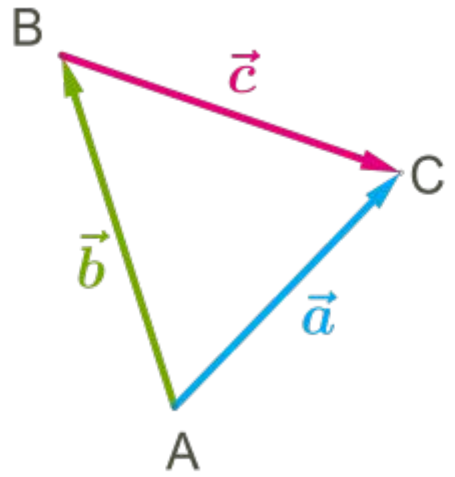
```
{ 5, 3 } + { 2, -1 } = { 7, 2 }
```



Базовые понятия и операции

Вектор (Операции)

```
{ 5, 3 } + { 2, -1 } = { 7, 2 }
{ 5, 3 } - { 2, -1 } = { 3, 4 }
```



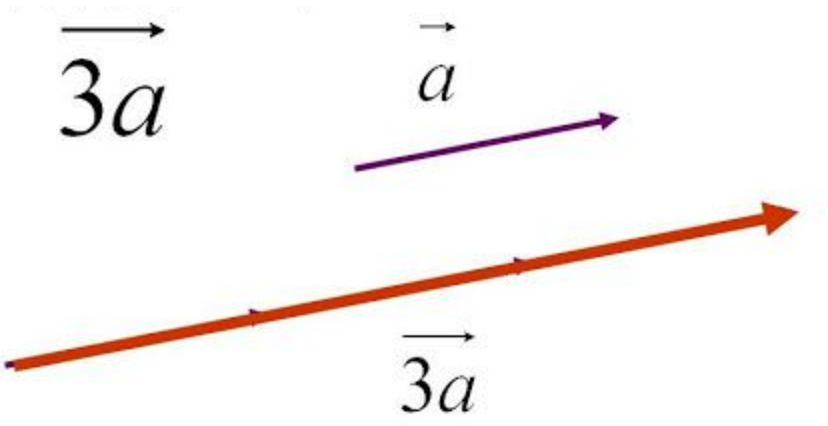
$$\vec{c} = \vec{a} - \vec{b}$$

$$\vec{a} = \vec{b} + \vec{c}$$

Базовые понятия и операции

Вектор (Операции)

```
{ 5, 3 } + { 2, -1 } = { 7, 2 }
{ 5, 3 } - { 2, -1 } = { 3, 4 }
{ 5, 3 } * 3 = { 15, 9 }
```



Базовые понятия и операции

Вектор (Реализация операций)

```
struct Vector {
    Vector(Point point) {
        this->x = point.x;
        this->y = point.y;
    }
    Vector(Point begin, Point end) {
        this->x = end.x - begin.x;
        this->y = end.y - begin.y;
    }
    double x, y;
}
```

Базовые понятия и операции

Вектор (Реализация операций)

```
Vector operator+(Vector left, Vector right) {
    Vector result;
    result.x = left.x + right.x;
    result.y = left.y + right.y;
    return result;
}

Vector a(5, 3);
Vector b(1, -2);
Vector c(6, 1);
//a + b == c -> true
```

Базовые понятия и операции

Вектор (Реализация операций)

```
Vector operator-(Vector left, Vector right) {
    Vector result;
    result.x = left.x - right.x;
    result.y = left.y - right.y;
    return result;
}

Vector a(5, 3);
Vector b(1, -2);
Vector c(4, 5);
//a + b == c -> true
```

Базовые понятия и операции

Вектор (Реализация операций)

```
Vector operator*(Vector left, double scalar) {
    Vector result;
    result.x = left.x * scalar;
    result.y = left.y * scalar;
    return result;
}

Vector a(5, 3);
Vector c(15, 9);
// a * 3 == c -> true
```

Базовые понятия и операции

Вектор (Реализация операций)

```
bool operator==(Vector left, Vector right) {
    return fabs(left.x - right.x) < EPS && fabs(left.y - right.y);
}
```

Базовые понятия и операции

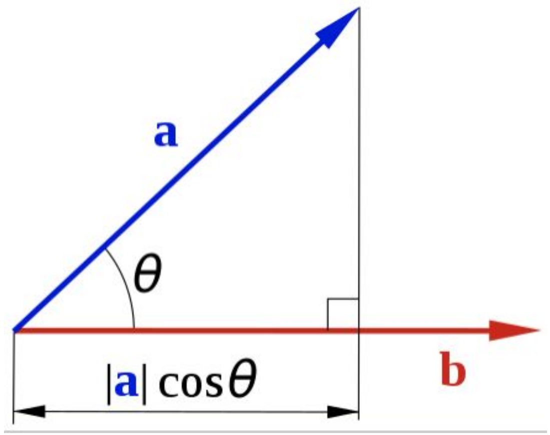
Вектор (Реализация операций)

```
double len(Vector vec) {
    return sqrt(vec.x * vec.x + vec.y * vec.y);
}
```

Базовые понятия и операции

Вектор (Скалярное произведение)

```
a * b = len(a) * len(b) * cos(a, b);
cos(a, b) - косинус угла между
векторами a и b


a * b = a.x * b.x + a.y * b.y
// Примечание: вектора a и b - ненулевые
```



Базовые понятия и операции

Вектор (Скалярное произведение)

```
a.x * b.x + a.y * b.y = len(a) * len(b) * cos(a, b);
cos(a, b) = (a.x + b.x + a.y * b.y) / (len(a) * len(b))
angle(a, b) = arccos(cos(a, b));
```

Базовые понятия и операции

Вектор (Свойства скалярного произведения)

```
1)  a * b = b * a - симметричность
2)  (a + b) * c = a * c + b * c - аддитивность
3)  (scalar * a) * b = scalar * (a * b) - однородность
4)  len(a) = sqrt(a * a)
```

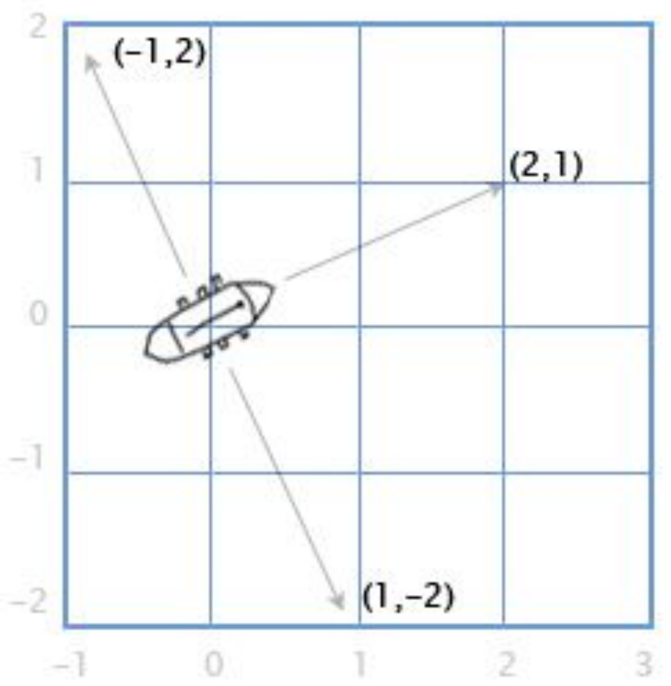
Базовые понятия и операции

Вектор (Реализация скалярного произведения)

```
double operator*(Vector left, Vector right) {
    return left.x * right.x + left.y * right.y;
}
// Угол в радианах
double angle(Vector left, Vector right) {
    return acos((left * right) / (len(a) * len(b)));
}
double radian_to_grad(double radian) {
    return radian * 180.0 / PI;
}
// const double PI = acos(-1.0);
```

Базовые понятия и операции

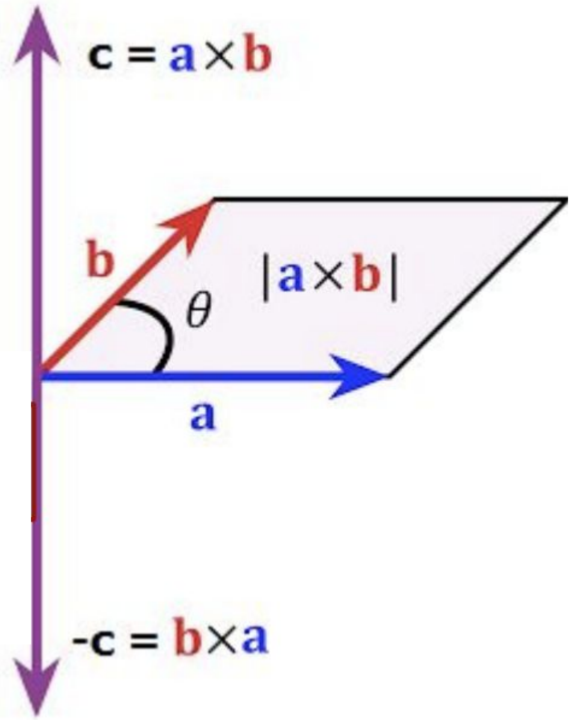
Вектор (Повернуть на 90 градусов)



Базовые понятия и операции

Вектор (Векторное произведение)

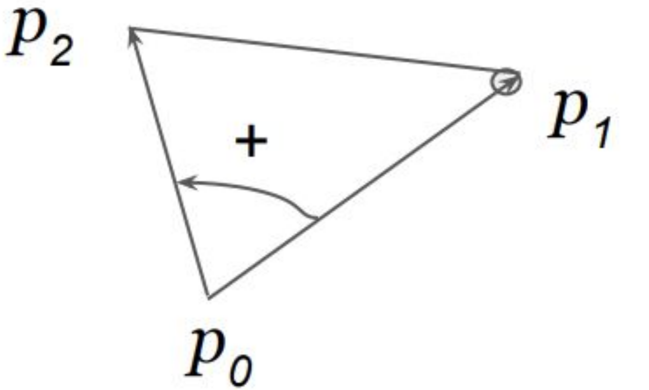
1) c = a x b
2) вектор c направлен в ту сторону, откуда **кратчайший поворот** от **a** к **b** виден **против часовой стрелки** – правая тройка векторов
3) a x b != b x a -> антикоммутативность
4) len(c) = len(a) * len(b) * sin(a, b)
5) len(c) = |a.x * b.y - a.y * b.y|

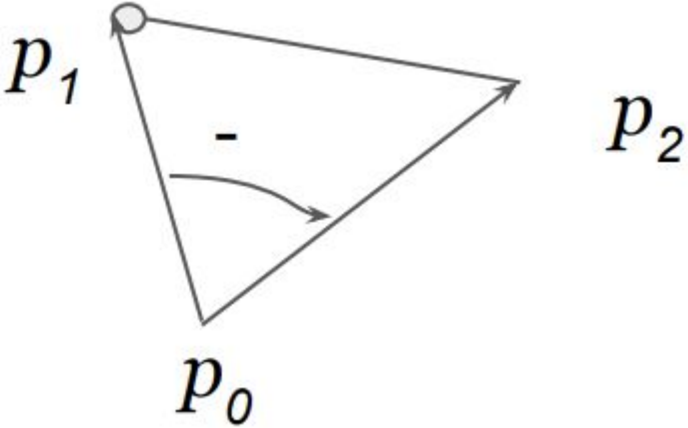


Базовые понятия и операции

Ориентированная площадь треугольника

```
1)  (p0 -> p1) x (p0 -> p2) = c > 0
2)  (p0 -> p1) x (p0 -> p2) = c < 0
```

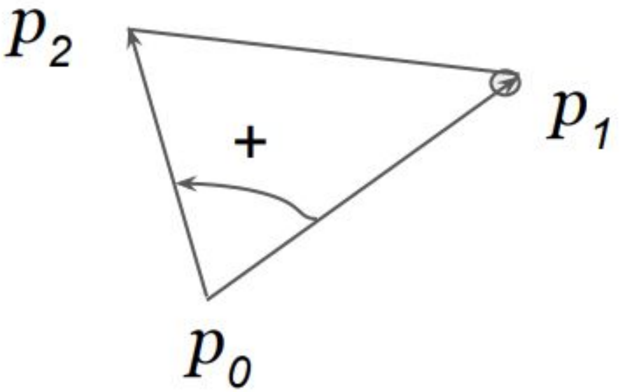


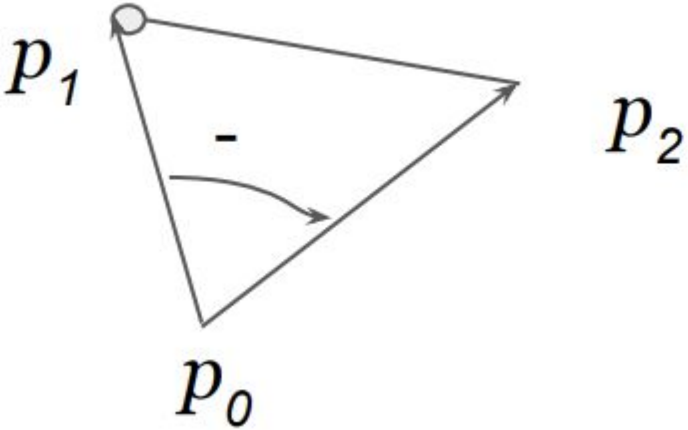


Базовые понятия и операции

Ориентированная площадь треугольника

```
double pseudo_scalar(Vector left, Vector right) {
    return left.x * right.y - left.y * right.x;
}
// (p1.x - p0.x) * (p2.y - p0.y) - (p1.y - p0.y) * (p2.x - p0.x) > 0.0
bool isLeftTurn(Point p0, Point p1, Point p2) {
    return pseudo_scalar(Vector(p0, p1), Vector(p0, p2)) > 0.0;
}
```

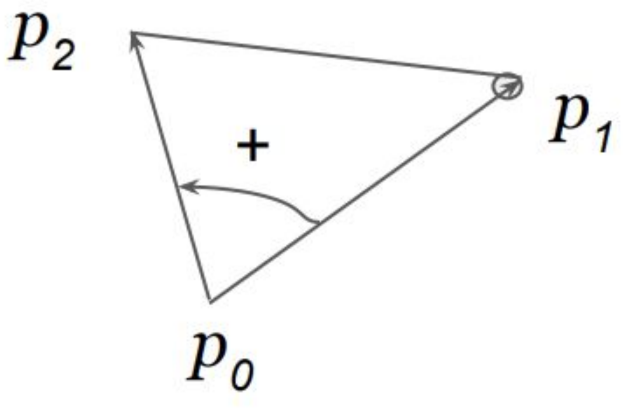


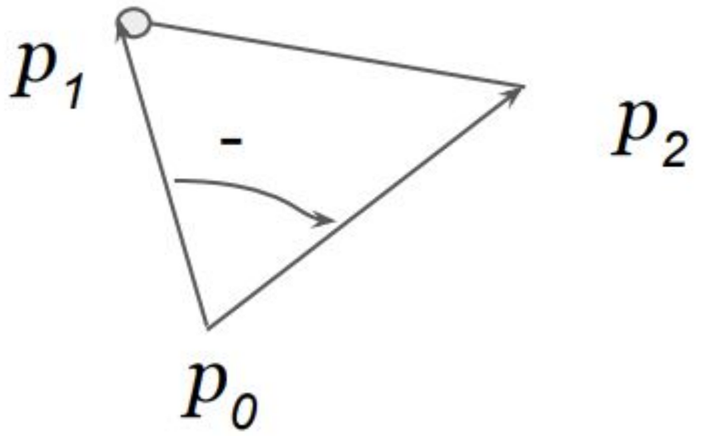


Базовые понятия и операции

Ориентированная площадь треугольника

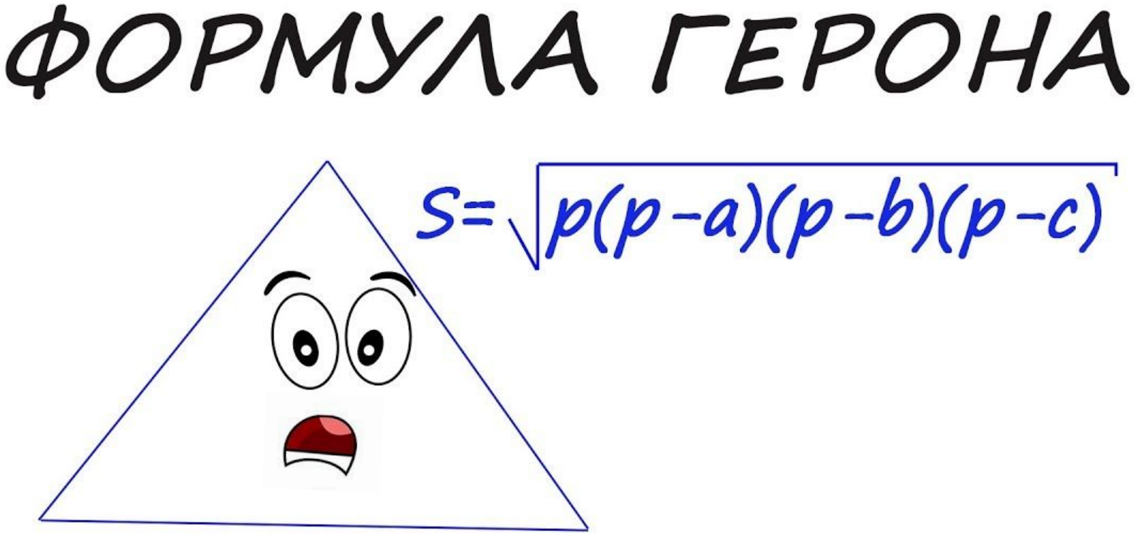
```
double pseudo_scalar(Vector left, Vector right) {
    return left.x * right.y - left.y * right.x;
}
// (p1.x - p0.x) * (p2.y - p0.y) - (p1.y - p0.y) * (p2.x - p0.x) > 0.0
double triangle_area(Point p0, Point p1, Point p2) {
    return fabs(pseudo_scalar(Vector(p0, p1), Vector(p0, p2))) / 2.0;
}
```





Базовые понятия и операции

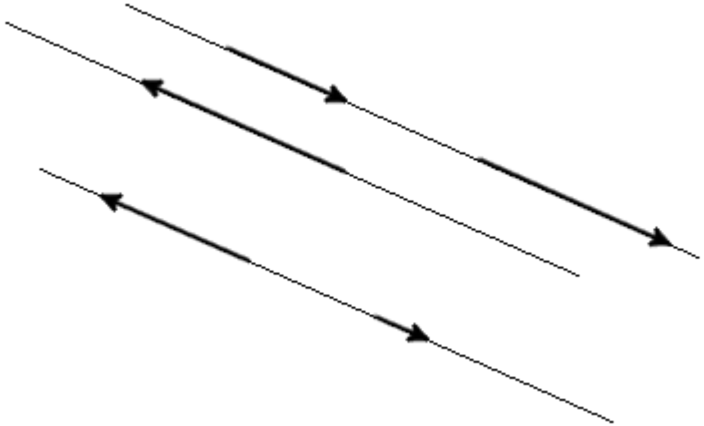
А как же формула Герона?



Базовые понятия и операции

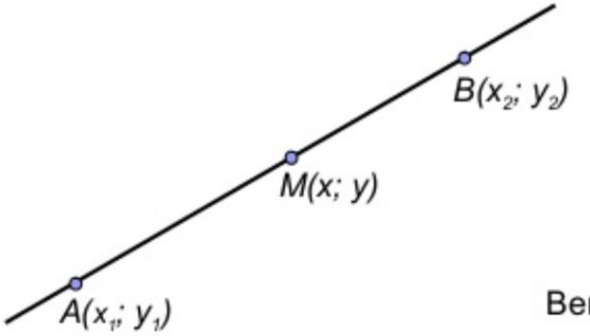
Вектор (Коллинеарность)

```
double pseudo_scalar(Vector left, Vector right) {
    return left.x * right.y - left.y * right.x;
}
bool is_collinear(Vector left, Vector right) {
    return pseudo_scalar(left, right) < EPS:
}
```



Базовые понятия и операции

Прямая (Через 2 точки)



$$\overrightarrow{AB} = \{x_2 - x_1; y_2 - y_1\}$$

$$\overrightarrow{AM} = \{x - x_1; y - y_1\}$$

Векторы $\overrightarrow{AB}$ и $\overrightarrow{AM}$ коллинеарны

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}$$

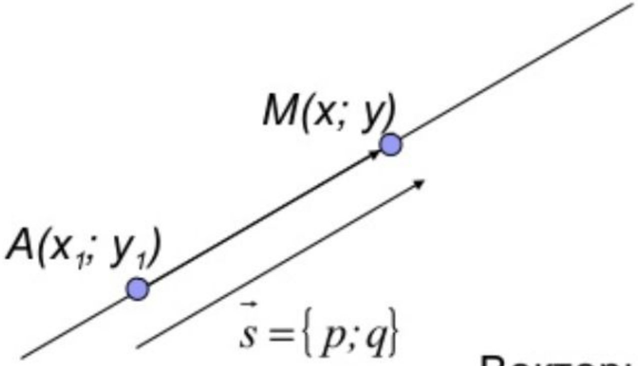
Базовые понятия и операции

Прямая (Нормальное уравнение прямой Ax + By + C = 0)

```
1)  (x - x1) / (x2 - x1) = (y - y1) / (y2 - y1)

2)  (x - x1) * (y2 - y1) = (y - y1) * (x2 - x1)

3)  x * (y2 - y1) - x1 * (y2 - y1) =

    y * (x2 - x1) - y1 * (x2 - x1)

4)  x * (y2 - y1) + y * (x1 - x2) + y1 * (x2 - x1) + x1 * (y1 - y2)

5)  A = (y2 - y1),

6)  B = (x1 - x2),

7)  C = y1 * (x2 - x1) + x1 * (y1 - y2)
```

Базовые понятия и операции

Прямая (Точка и направляющий вектор)



$$\overrightarrow{AM} = \{x - x_1; y - y_1\}$$

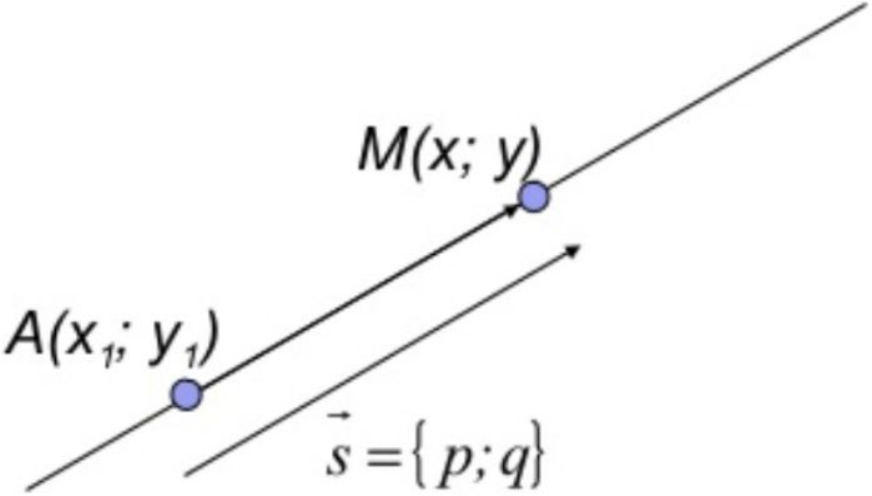
Векторы $\vec{s}$ и $\overrightarrow{AM}$ коллинеарны

$$\frac{x - x_1}{p} = \frac{y - y_1}{q}$$

Базовые понятия и операции

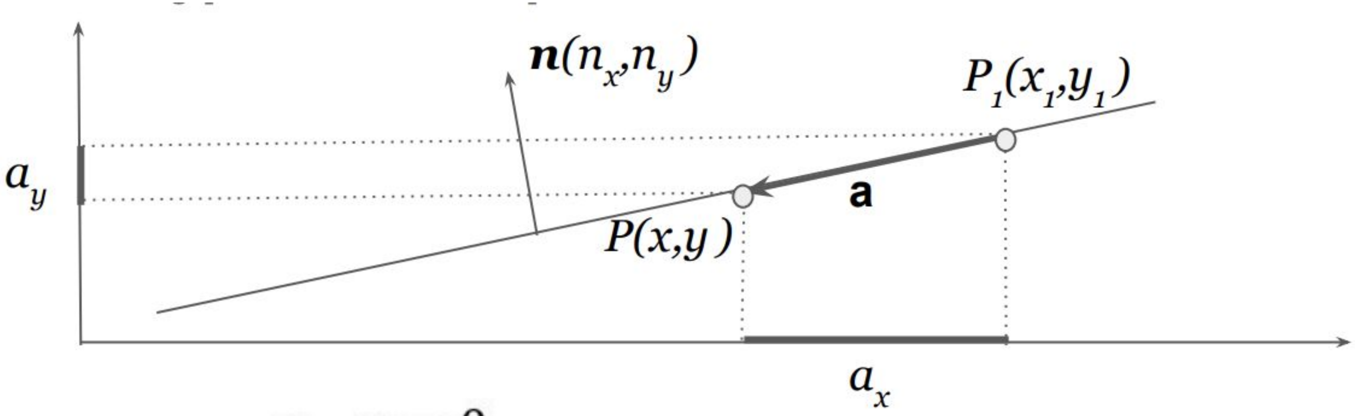
Прямая (Точка и направляющий вектор)

1) p + v * t | t Є R
2) p – точка через которую проходит прямая
3) v – направляющий вектор



Базовые понятия и операции

Прямая (Точка и нормаль)



$$\boldsymbol{n} \cdot \boldsymbol{a} = 0$$

$$n_x(x - x_1) + n_y(y - y_1) = 0$$

$$n_x x + n_y y + \left(-n_x x_1 - n_y y_1\right) = 0$$

Задачи

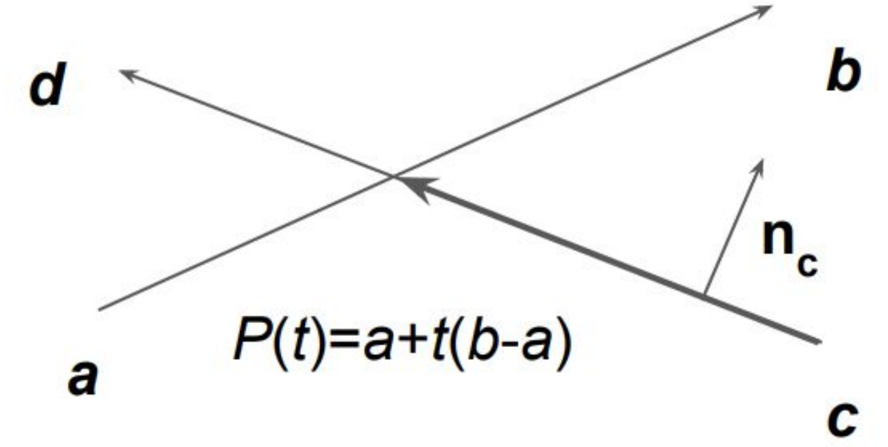
Пересечение двух прямых

1) Заданы 2 прямых (по двум точкам) - нужно найти точку их пересечения или сказать что ее не существует

Задачи

Пересечение двух прямых

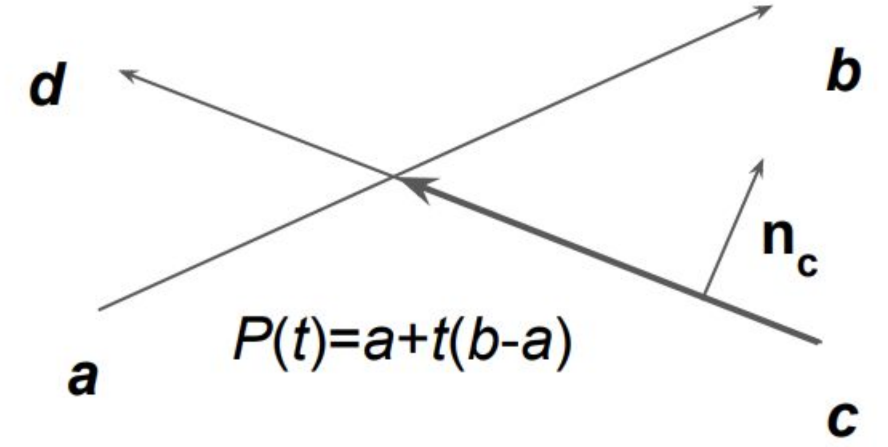
1) Пусть прямые проходят через точки (a, b) и (c, d) соответственно
1) Пусть точка пересечения прямых P(t)
2) Так как она лежит на первой прямой, то P(t) = a + t * (b - a)



Задачи

Пересечение двух прямых

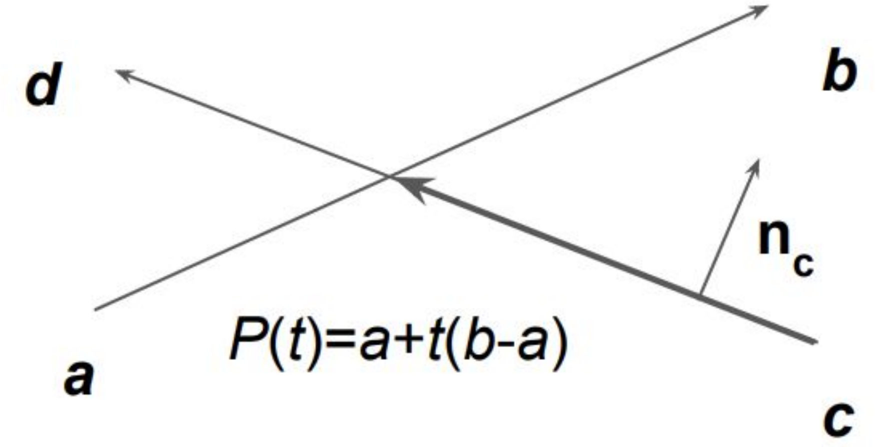
1) Пусть n_c - вектор нормали (можем получить путем поворота dc на 90 градусов)
2) Тогда P(t)c может быть направляющим вектором второй прямой



Задачи

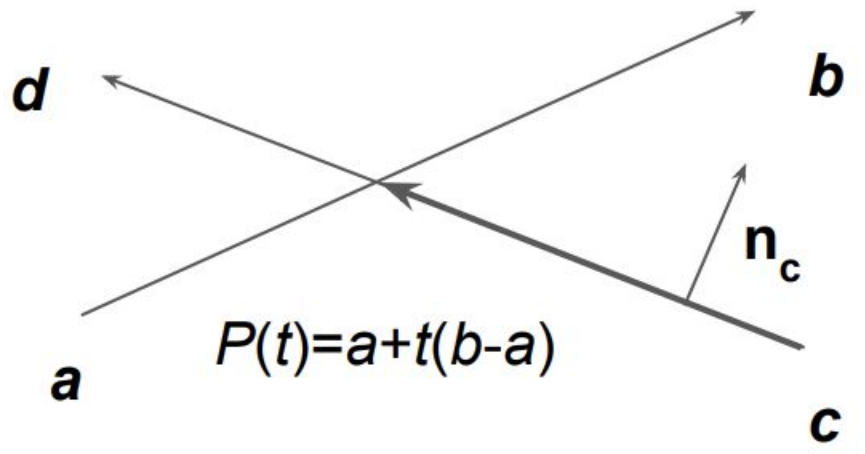
Пересечение двух прямых

1) Тогда P(t)c может быть направляющим вектором второй прямой
2) Направляющий вектор и нормаль перпендикулярны по определению
3) P(t)c * n_c = 0



Задачи

Пересечение двух прямых




```
1)  P(t)c * n_c = 0
1)  (P(t) - c) * n_c = 0
2)  (a + (b - a) * t - c) * n_c = 0
3)  n_c * (a - c) + t * (n_c * (b - a)) = 0
4)  t = n_c * (c - a) / n_c * (b - a)
```

*n_c вынести не можем - произведения скалярные

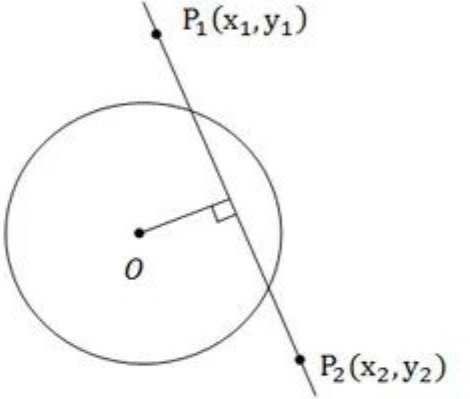
Задачи

Пересечение прямой и окружности

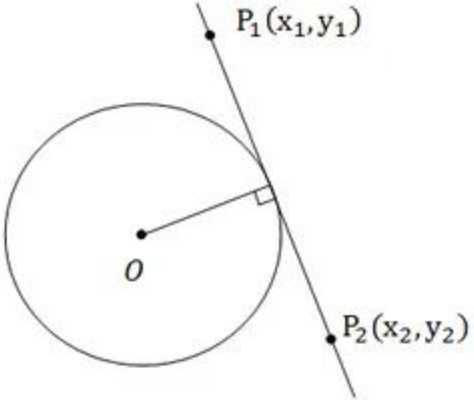
2) Дана прямая (по двум точкам a и b) и окружность с центром в точке c и радиусом r

Задачи

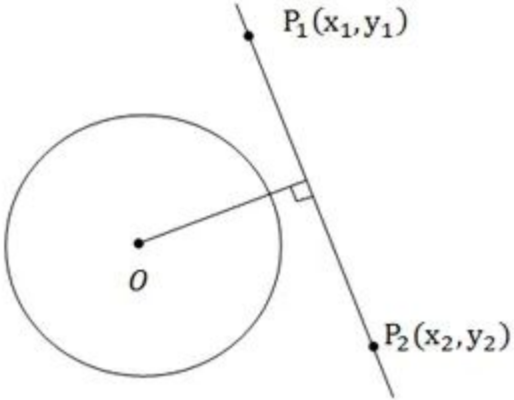
Пересечение прямой и окружности



Две точки пересечения



Одна точка касания



Точек пересечения нет

Задачи

Пересечение прямой и окружности

```
1)  Пусть P(t) пересечение прямой и окружности, тогда:
2)  P(t) = a + (b - a) * t
3)  Уравнение окружности (p - c) * (p - c) = r * r
 (p.x - c.x) * (p.x - c.x) + (p.y - c.y) * (p.y - c.y) = r * r
4)  (P(t) - c) * (P(t) - c) = r * r
5)  Подставляем (2) в (4)
```

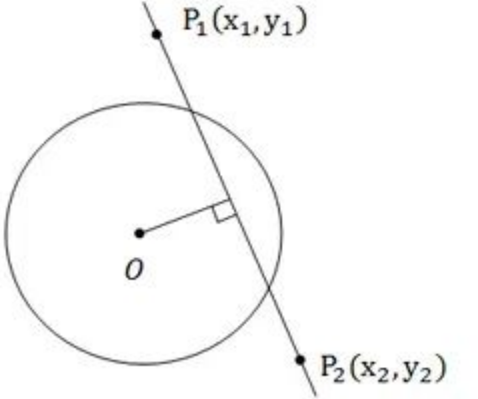
Задачи

Пересечение прямой и окружности

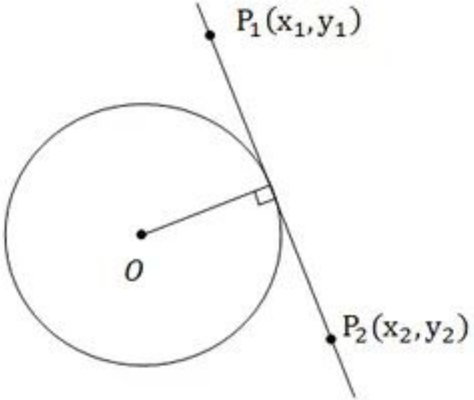
1) $(a + (b - a) * t - c) * (a + (b - a) * t - c) = r * r$
2) $((a - c) + t * (b - a)) * ((a - c) + t * (b - a)) = r * r$
3) $(a - c) * (a - c) + t * 2 * ((b - a) * (a - c)) + t^2 * (b - a) * (b - a) = r * r$
4) $t^2 * |(b - a)| + t * 2 * ((b - a) * (a - c)) + (a - c)^2 - r^2 = 0$
5) Решаем квадратное уравнение $a * t^2 + b * t + c = 0$

Задачи

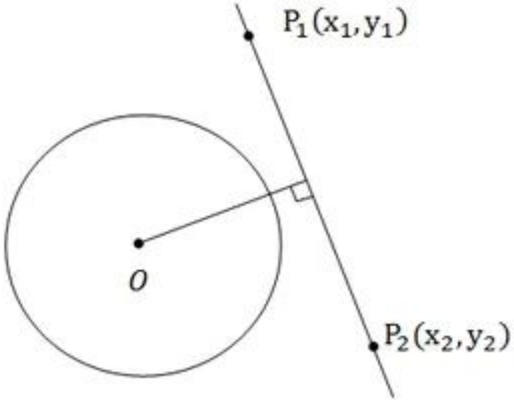
Пересечение прямой и окружности



Две точки пересечения



Одна точка касания



Точек пересечения нет

Задачи

Пересечение отрезков

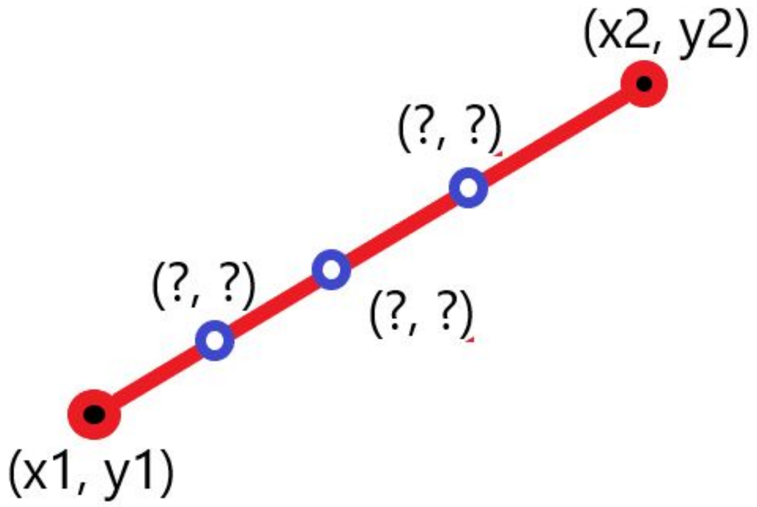
1) Давайте проверим пересекаются ли прямые, на которых лежат отрезки
2) Если нет, то отрезки тоже нет
3) Иначе проверим, что точка лежит на двух отрезках
4) Проверка:

```
min(a.x, b.x) <= p.x <= max(a.x, b.x) &&
min(a.y, b.y) <= p.y <= max(a.y, b.y)
```

Задачи

Проверка принадлежности точки отрезку

1) Если рассматриваем только целочисленные точки:
2) Находим НОД (x2 - x1) и (y2 - y1)
3) delta_x = (x2 - x1) / gcd
4) delta_y = (y2 - y1) / gcd



Задачи

Проверка принадлежности точки отрезку

1) Если рассматриваем все точки:
2) Проверим лежит ли точка на прямой (через скалярное произведение)
3) Аналогичная проверка:

```
min(a.x, b.x) <= p.x <= max(a.x, b.x) &&
min(a.y, b.y) <= p.y <= max(a.y, b.y)
```

Задачи

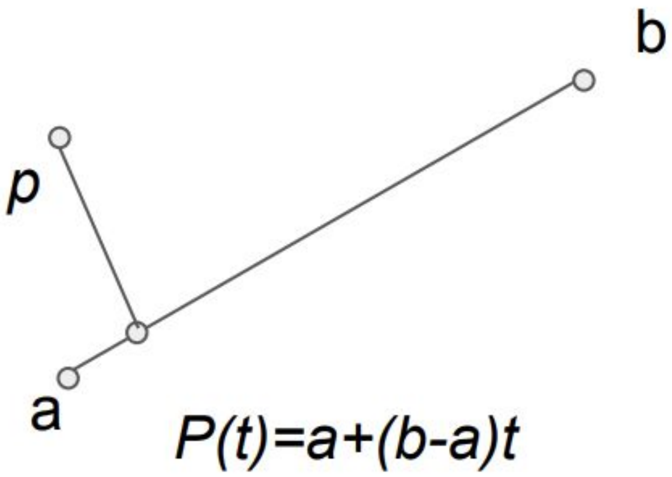
Расстояние от точки до прямой

1) Задана точка p и прямая (через 2 точки a и b) - требуется найти расстояние между ними

Задачи

Расстояние от точки до прямой

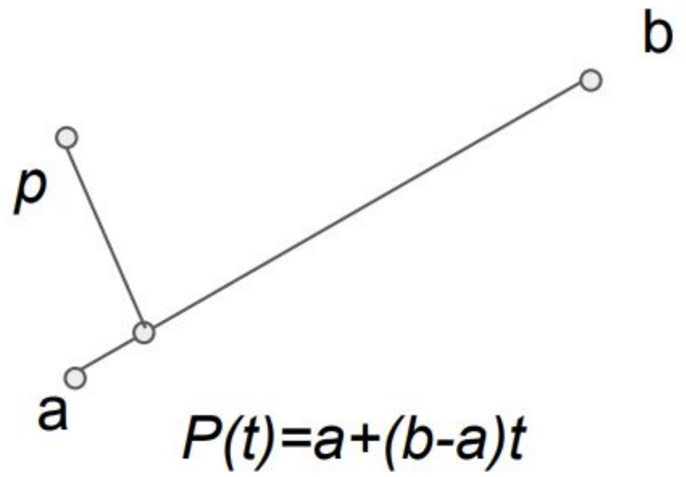
```
1)  (P(t) - p) * (b - a) = 0
2)  (a + (b - a) * t) * (b - a) = 0
3)  a * (b - a) + |(b - a)| * t = 0
4)  t = a * (a - b) / |(b - a)|
5)  Ответ: |p - P(t)|
```



Задачи

Расстояние от точки до отрезка

```
1)  (P(t) - p) * (b - a) = 0
2)  (a + (b - a) * t) * (b - a) = 0
3)  a * (b - a) + |(b - a)| * t = 0
4)  t = a * (a - b) / |(b - a)|
5)  Ответ: |(p - P(t))| если P(t)
    лежит на отрезке, иначе:
    min(|(p - a)|, |(p - b)|)
```



Задачи

Пересечение окружностей

1) Даны две окружности с радиусами r1 и r, а также центрами (x1, y1) (x2, y2) - найти их пересечение

Задачи

Пересечение окружностей

1) Сразу рассмотрим вырожденный случай, когда центры совпадают:
   a) радиусы совпадают, тогда решений бесконечное кол-во (окружности полностью совпадают)
   b) иначе решений нет

Задачи

Пересечение окружностей

2) Сместим центры окружностей так, чтобы центр первой был в точке (0, 0), потом к ответу обратно добавим это смещение

3) Имеем систему из 2 уравнений:

$$x^2 + y^2 = r_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2$$

4) Давайте вычтем из второго уравнения первое

Задачи

Пересечение окружностей

$$x^2 + y^2 = r_1^{\ 2}$$

$$x * -2x_2 + y * -2y_2 + (x_2^{\ 2} + y_2^{\ 2} + r_1^{\ 2} + r_2^{\ 2}) = 0$$

Теперь система отображает задачу пересечения окружности и прямой $Ax + By + C = 0$, где:

$A = -2x_2$

$B = 2y_2$

$C = (x_2^{\ 2} + y_2^{\ 2} + r_1^{\ 2} + r_2^{\ 2})$

Ее мы решать уже умеем

Примечание

Вывод с точностью P знаков после запятой в C++

```
#include <iomanip>
#include <cmath>

...

cout << fixed << setprecision(P) << number;
```