

Вычислительная геометрия

Базовые процедуры

Операции для векторов

Каждую точку плоскости можно считать **вектором**, начало которого находится в точке (0,0).

Обозначим координаты вектора $\mathbf{v}=(v_x, v_y)$, $\mathbf{w}=(w_x, w_y)$, $\mathbf{q}=(q_x, q_y)$.

Сумма: $\mathbf{q} = \mathbf{v} + \mathbf{w}$ $q_x = v_x + w_x$, $q_y = v_y + w_y$

Разность: $\mathbf{q} = \mathbf{v} - \mathbf{w}$ $q_x = v_x - w_x$, $q_y = v_y - w_y$

Умножение на число k: $\mathbf{q} = k * \mathbf{v}$ $q_x = k * v_x$, $q_y = k * v_y$

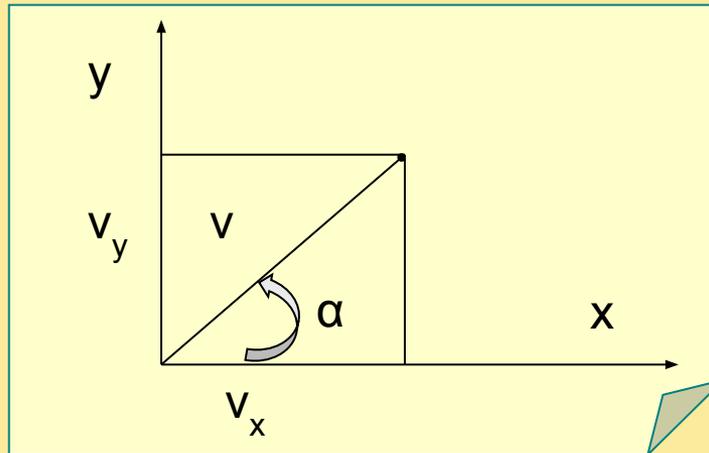
Скалярное произведение: $\mathbf{v} * \mathbf{w} = v_x * w_x + v_y * w_y$

Векторное произведение: $\mathbf{v} \times \mathbf{w} = (v_x * w_y - v_y * w_x) \mathbf{z}$

Полярная и декартова системы координат

Связь координат в полярной и декартовой системах координат

$$\begin{aligned}v_x &= v \cdot \cos(\alpha), & v_y &= v \cdot \sin(\alpha) \\ v &= \sqrt{v_x^2 + v_y^2}, & \tan(\alpha) &= v_y / v_x\end{aligned}$$



Скалярное произведение векторов (v, α) и (w, β)

$$\begin{aligned}\mathbf{v} \cdot \mathbf{w} &= v_x \cdot w_x + v_y \cdot w_y = \\ &= v \cdot \cos(\alpha) \cdot w \cdot \cos(\beta) + v \cdot \sin(\alpha) \cdot w \cdot \sin(\beta) = v \cdot w \cdot \cos(\alpha - \beta)\end{aligned}$$

Векторное произведение векторов (v, α) и (w, β)

$$\begin{aligned}\mathbf{v} \times \mathbf{w} &= (v_x \cdot w_y - v_y \cdot w_x) \mathbf{z} = \\ &= (v \cdot \cos(\alpha) \cdot w \cdot \sin(\beta) - v \cdot \sin(\alpha) \cdot w \cdot \cos(\beta)) \mathbf{z} = (v \cdot w \cdot \sin(\beta - \alpha)) \mathbf{z}\end{aligned}$$

СВОЙСТВА ВЕКТОРОВ

1 Скалярное произведение ненулевых векторов равно нулю тогда и только тогда, когда векторы перпендикулярны.

2 Векторное произведение ненулевых векторов равно нулю тогда и только тогда, когда векторы параллельны.

3 При общей начальной точке у двух векторов скалярное произведение больше нуля, если угол между векторами острый, и меньше нуля, если угол тупой.

4 При общей начальной точке у двух векторов их векторное произведение больше нуля, если второй вектор направлен влево от первого, меньше нуля, если вправо.

Прямая линия

Прямая линия на плоскости, проходящая через две точки (v_x, v_y) , (w_x, w_y) , определяется линейным уравнением от двух переменных $(w_x - v_x) * (y - v_y) = (w_y - v_y) * (x - v_x)$.

После преобразований и обозначений: $A * x + B * y + C = 0$.

Точка пересечения прямых, заданных уравнениями:

$$A_1 * x + B_1 * y + C_1 = 0 \text{ и } A_2 * x + B_2 * y + C_2 = 0$$



$$X = -(C_1 * B_2 - C_2 * B_1) / (A_1 * B_2 - A_2 * B_1)$$
$$Y = (A_2 * C_1 - A_1 * C_2) / (A_1 * B_2 - A_2 * B_1)$$

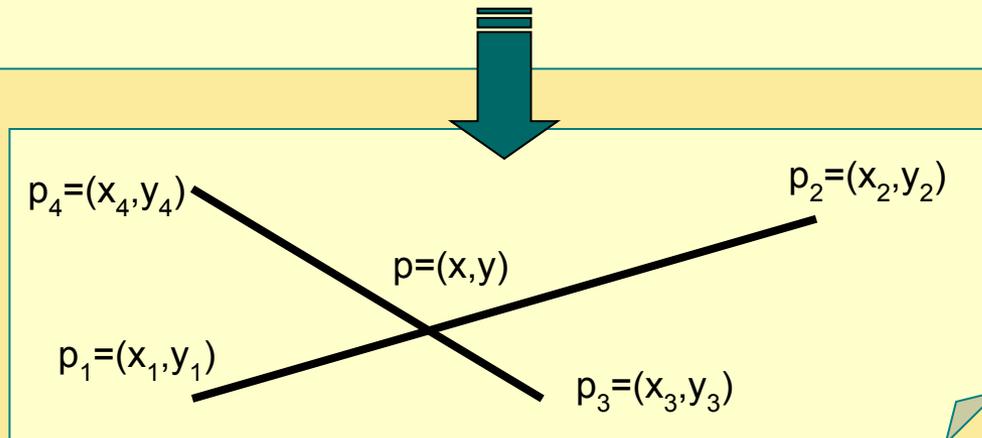
Координаты точек отрезка

$$x = v_x + (w_x - v_x) * t, \quad y = v_y + (w_y - v_y) * t$$

При $0 \leq t \leq 1$ точка (x, y) лежит на отрезке.

При $t < 0$ или $t > 1$ точка (x, y) лежит вне отрезка на прямой линии, продолжающей отрезок.

Пример. Даны два отрезка, заданные точками $p_1=(x_1, y_1)$ и $p_2=(x_2, y_2)$, $p_3=(x_3, y_3)$ и $p_4=(x_4, y_4)$.
Определить взаимное расположение отрезков на плоскости.



Взаимное расположение отрезков

$$v_1 = p_3 p_4 \times p_3 p_1, \quad v_2 = p_3 p_4 \times p_3 p_2$$

$$v_3 = p_1 p_2 \times p_1 p_3, \quad v_4 = p_1 p_2 \times p_1 p_4$$

$$v_1 v_2 < 0 \text{ и } v_3 v_4 < 0$$

Отрезки пересекаются

$$v_1 v_2 > 0 \text{ или } v_3 v_4 > 0$$

Отрезки не пересекаются

$$v_1 v_2 \leq 0, v_3 = 0, v_4 \neq 0$$

Точка p_3 лежит на отрезке $p_1 p_2$

$$v_1 v_2 \leq 0, v_4 = 0, v_3 \neq 0$$

Точка p_4 лежит на отрезке $p_1 p_2$

$$v_3 v_4 \leq 0, v_1 = 0, v_2 \neq 0$$

Точка p_1 лежит на отрезке $p_3 p_4$

$$v_3 v_4 \leq 0, v_2 = 0, v_1 \neq 0$$

Точка p_2 лежит на отрезке $p_3 p_4$

$$v_1 = 0, v_2 = 0, v_3 = 0, v_4 = 0$$

Отрезки $p_1 p_2$ и $p_3 p_4$ лежат на одной прямой.
Нужна проверка на их перекрытие.

Задача 1

Даны 3 точки A, B и C. Определить, лежат ли они на одной прямой.

Решение

Нужно выяснить, параллельны ли векторы AB и AC. Координаты этих векторов – разности координат точек.

Для того чтобы векторы с координатами (x_1, y_1) и (x_2, y_2) были параллельны, нужно, чтобы существовало такое число a , что

$$x_2 = a * x_1 \text{ и } y_2 = a * y_1.$$

Требуется проверить равенство

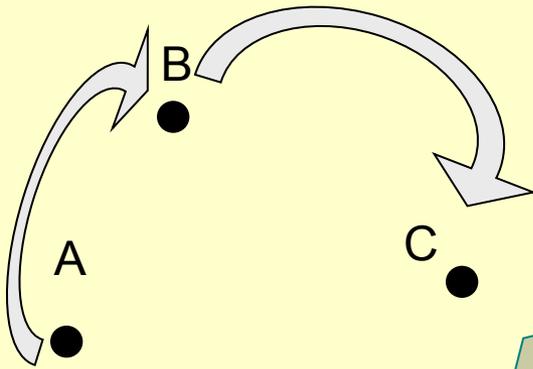
$$x_2 / x_1 = y_2 / y_1.$$

Так как x_1 или y_1 могут оказаться равными 0, то проверяем равенство

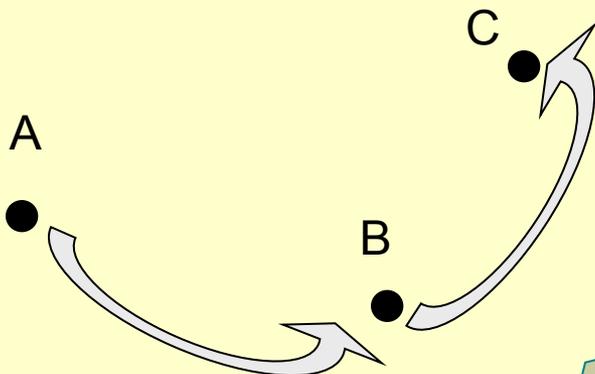
$$x_1 * y_2 = x_2 * y_1.$$

Задача 2

Даны 3 точки A, B и C, не лежащие на одной прямой.
Определить, является ли обход $A \rightarrow B \rightarrow C$ обходом по часовой стрелке или против часовой стрелки.



$((B.x - A.x) * (C.y - A.y) - (C.x - A.x) * (B.y - A.y) < 0)$ -
по часовой



Задача 3

Даны 4 точки A, B, C и D. Определить, пересекаются ли отрезки AC и BD.
Даны 4 точки A, B, C и D. Определить, является ли четырехугольник ABCD выпуклым.

Решение

Отрезки AC и BD пересекаются тогда и только тогда, когда четырехугольник ABCD выпуклый.

Для того, чтобы четырехугольник ABCD был выпуклым, необходимо и достаточно, чтобы все 4 обхода $A \rightarrow B \rightarrow C$, $B \rightarrow C \rightarrow D$, $C \rightarrow D \rightarrow A$, $D \rightarrow A \rightarrow B$ были обходами в одну сторону: либо по часовой стрелке, либо против.

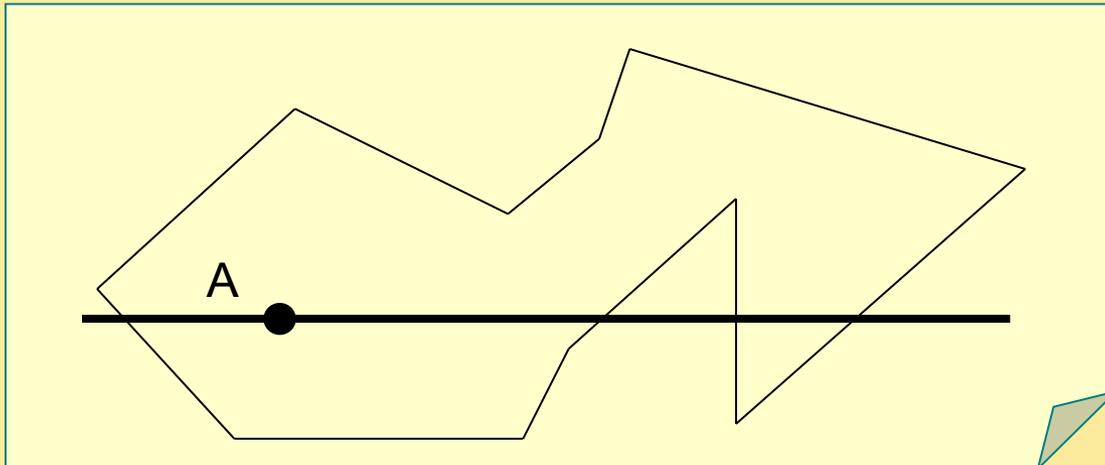
Задача 4-1

Лежит ли данная точка A внутри данного многоугольника $B_1B_2\dots B_n$?
Считаем многоугольник *простым*. Это значит, что его стороны не имеют общих точек, кроме вершин, причем в каждой вершине встречаются ровно 2 стороны.

Решение

Если провести через точку A прямую, параллельную оси X , то она пересечет стороны многоугольника в четном числе точек. При этом на прямой будут чередоваться интервалы, находящиеся вне многоугольника и внутри него.

Если по обе стороны от точки A на прямой окажется четное число точек пересечения со сторонами, то точка лежит вне многоугольника, а если нечетное – то внутри.



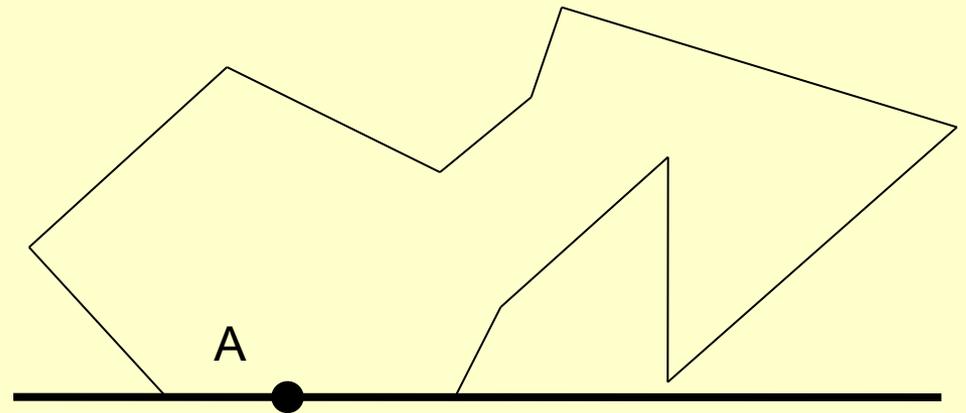
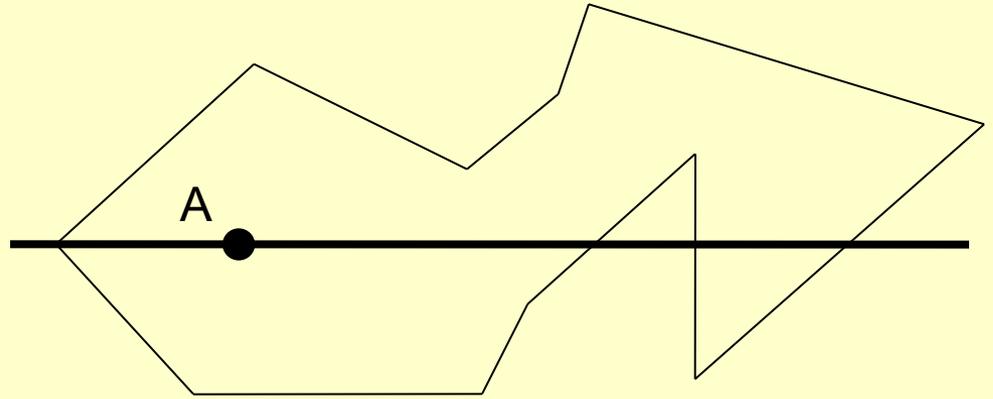
Задача 4-2

Алгоритм

1. Выбираем прямую.
2. Находим, с какими сторонами она пересекается и в каких точках.
3. Считаем количество таких точек с одной стороны от A.

Детали

1. Прямая может проходить через вершину многоугольника 2 разными способами – *пересекая* границу многоугольника или *касаясь* его границы.
2. Сторона прямоугольника может быть горизонтальной и лежать на прямой.



Задача 4-3

Точка А меняет свое положение, а многоугольник остается неизменным.
Порядок следования вершин многоугольника неизвестен.

Предварительная обработка

Данные, связанные с многоугольником, записываются в более удобном виде.

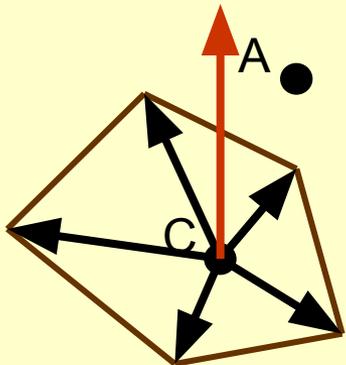


Алгоритм

1. Возьмем точку С внутри многоугольника.
2. Для всех лучей с началом в точке С, проходящих через вершины многоугольника, посчитаем углы, которые они образуют с вертикальным лучом, начинающимся в точке С. Углы считаем “против часовой стрелки”, так что они могут иметь значения от 0 до 360 градусов.
3. Теперь каждой вершине многоугольника соответствует число. Занумеруем вершины в порядке возрастания этих чисел: $V_1 V_2 \dots V_N$.



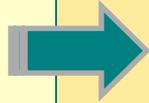
На это действие (сортировку) требуется время $O(N \log N)$.



Задача 4-4

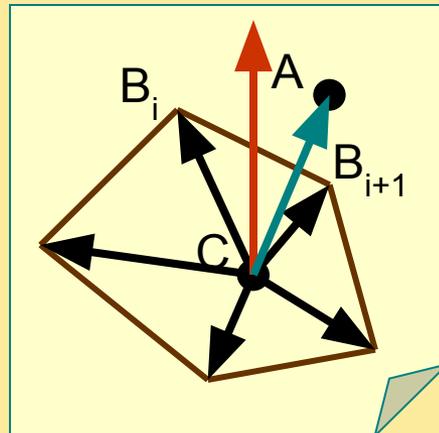
Ответ на вопрос задачи

Используя полученные в процессе предварительной обработки сведения, определяем для заданной точки, лежит ли она внутри многоугольника.



Алгоритм

1. Для заданной точки A проведем луч CA и посчитаем для него угол с вертикальным лучом.
2. Найдем, между какими двумя лучами CB_i и CB_{i+1} проходит луч CA . Это можно сделать за время $O(\log N)$ – используя бинарный поиск, поскольку лучи расставлены в порядке возрастания (или убывания) угла.
3. Выясним, лежит ли точка A внутри треугольника CB_iB_{i+1} .



Задача 5

Является ли данный многоугольник $V_1V_2\dots V_n$ простым?

Решение

Эту задачу можно решить за время $O(N^2)$, просто перебрав все возможные пары отрезков и проверив, не пересекаются ли они.

Задача 6-1

Можно ли мы из данного набора N точек выбрать его подмножество N' так, чтобы многоугольник, построенный *на* этих точках, был *выпуклым*?

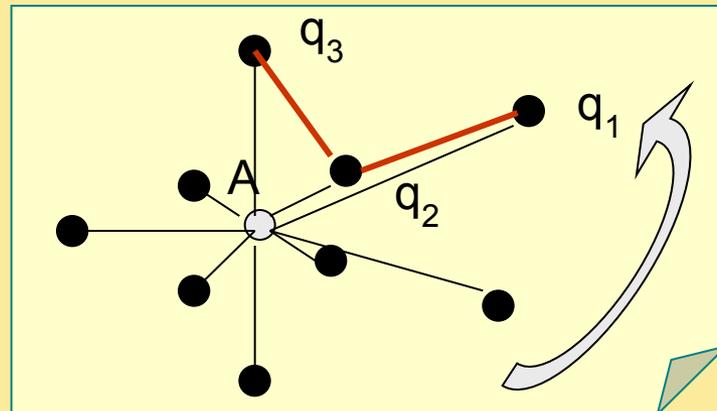
Решение

Будем выбирать подмножество точек-вершин N^{cover} так, чтобы получившийся многоугольник содержал *внутри все остальные точки* $N \setminus N^{\text{cover}}$.

У подмножества N^{cover} есть специальное название – **выпуклая оболочка**.

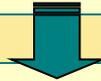
Алгоритм Грэхема

Проверять тройки последовательных точек в порядке обхода против часовой стрелки. Если угол $q_1q_2q_3$ больше или равен π , то точки образуют «правый поворот», иначе «левый поворот». Исключить точки, образующие «правый поворот».



Задача 6-2

Сортировка из N элементов может быть выполнена за время $O(N \cdot \log N)$.
Обход может быть выполнен за время, пропорциональное N .



Выпуклую оболочку можно построить за время, пропорциональное $N \cdot \log N$.

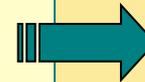
Алгоритм Грэхема

1. **Инициализация набора** из N точек с декартовыми координатами X_i и Y_i ($i=1:N$).

2. **Выбор внутренней точки** A будущего многоугольника.

Способ 1. Выбрать тройку точек исходного набора, образующих невырожденный треугольник; тогда любая его внутренняя точка нас устроит (по существу, берутся “первые попавшиеся” две точки, а третья отбирается при последовательном выборе, т.е. с трудоемкостью $O(N)$).

Способ 2. Найти средние арифметические значений координат X_i и Y_i всех точек и приписать их, соответственно, новой (внутренней!) точке – $A(X, Y)$; вычисления имеют ту же трудоемкость – $O(N)$.



Замечание. Можно найти самую левую верхнюю точку, которая заведомо принадлежит выпуклой оболочке. Значения углов вычислять относительно этой точки.

Задача 6-3

Алгоритм Грэхема

3. **Переопределение координат** всех точек исходного набора в полярной системе координат, принимая за начало координат точку A .

Нулевой угол отсчета можно связать с любым лучом, например, Aq_1 .

4. **Сортировка по неубыванию** точек исходного набора. В качестве ключа используется значение полярного угла. При равенстве углов “отсеивать” точки с меньшим расстоянием от начала координат. В результате получим набор $N_1 \in N$.

5. **Отсев точек**, которые не войдут в выпуклую оболочку. Для этого осуществим последовательный перебор троек “соседних” точек упорядоченного набора N_1 , отбрасывая лишние, до получения искомого набора N^{cover} .

Критерий отброса: для каждой очередной “тройки” проверяем значение угла между ними $p_{i-1}p_i p_{i+1}$. Если угол не меньше 180° , то точка p_i исключается из набора N_1 и новая тройка “начинается” с той же точки p_{i-1} . В противном случае индекс i увеличивается на 1. Перебор завершается, когда обход “по кругу” не приводит к отсеву точек: N^{cover} сформирована.

Задача 7

Даны 3 точки A , B и C , лежащие на одной прямой. Определить, какая из них лежит между двумя другими.

Решение

Будем решать задачу не для самих точек, а для их проекций на ось X . Для этого нужно выяснить, какое из чисел A_x , B_x и C_x лежит между двумя другими.

Следует иметь в виду, что эти числа могут совпасть (если точки лежат на прямой, перпендикулярной оси X). Тогда придется рассмотреть проекцию на другую ось.

Задача 8

Даны 4 точки A, B, C и D. Определить, в какой точке пересекаются прямые AC и BD.

Решение

Нужно найти точку, лежащую одновременно на двух прямых.

Условие принадлежности точки P прямой AC :

$$(C.x-A.x)*(P.y-A.y)-(P.x-A.x)*(C.y-A.y)=0.$$

Условие принадлежности точки P прямой BD :

$$(D.x-B.x)*(P.y-B.y)-(P.x-B.x)*(D.y-B.y)=0.$$

Решение системы этих уравнений:

$$P.x = \frac{((C.x-A.x)*(D.x*B.y-B.x*D.y)+(D.x-B.x)* \\ *(C.x*A.y-A.x*C.y))}{((C.x-A.x)*(B.y-D.y)-(C.y-A.y)* (B.x-D.x))};$$

$$P.y = \frac{((C.y-A.y)*(D.y*B.x-B.y*D.x)+(D.y-B.y)* \\ *(C.y*A.x-A.y*C.x))}{((C.y-A.y)*(B.x-D.x)-(C.x-A.x)* (B.y-D.y))}$$

Знаменатели у обеих дробей одинаковы, они равны 0, если исходные прямые параллельны или совпадают.

Задача 9-1

Дано конечное множество отрезков на плоскости. Координаты концов могут быть как целыми числами, так и дробными. Требуется найти все точки пересечения.

Пояснение

Используется алгоритм, основанный на **заметании плоскости вертикальной прямой**.

Его временная сложность $O((N+K)\log(N))$, где N - кол-во отрезков, K - кол-во точек пересечения.

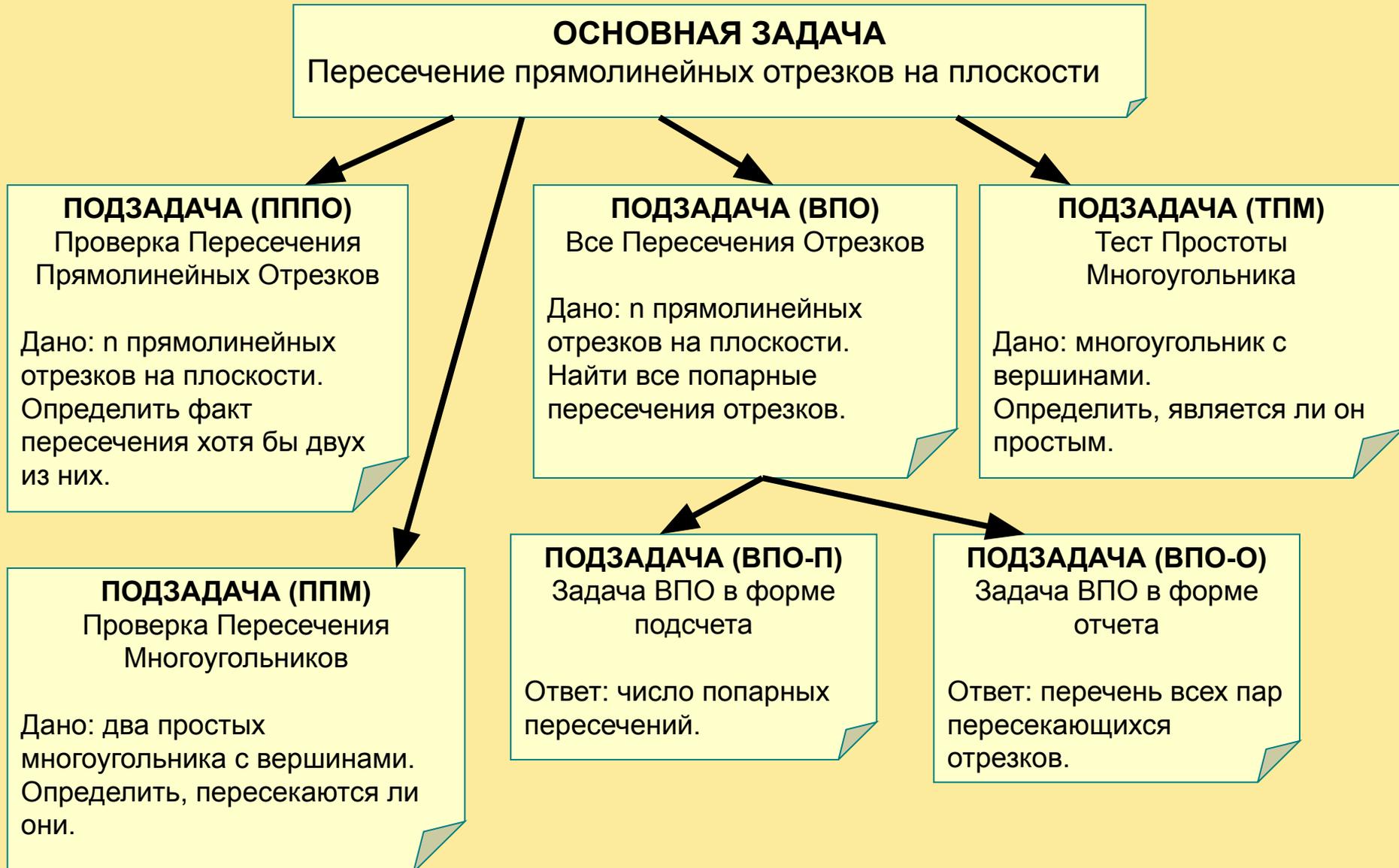
Алгоритм реализован на базе двоичного дерева отрезков, упорядоченных по значению ординат в конкретной абсциссе.

Замечание

Погрешность вычисления y -координаты отрезка в заданной абсциссе x может быть связана с тем, что отрезок близок к вертикальному (угол между вертикалью и отрезком менее 5 - 10 градусов). Это может привести к тому, что процедура поиска уходит не в ту ветку дерева.

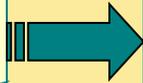
Ошибка связана с точностью представления вещественных чисел в компьютере.

Задача 9-2



Задача 9-3

Преобразование
задач



Задача А преобразуется в задачу В за время $O(T(n))$, если задачу А можно решить следующим образом:

1. Из исходных данных задачи А получить исходные данные к задаче В за время $O(T(n))$.
2. Решить задачу В.
3. Из результата решения задачи В получить результат решения задачи А за время $O(T(n))$.

ППМ \xrightarrow{n} ПЯПО

ТПМ \xrightarrow{n} ПЯПО

ПППО $\xrightarrow{1}$ ВЯО-П \xrightarrow{n} ВПО-О

Задача 9-4

**Проверка пересечения
пары отрезков**



Грубый тест

Определение факта пересечения ограничивающих прямоугольников каждого отрезка.

Два прямоугольника со сторонами, параллельными координатным осям, пересекаются тогда и только тогда, когда пересекаются их проекции на оси координат.

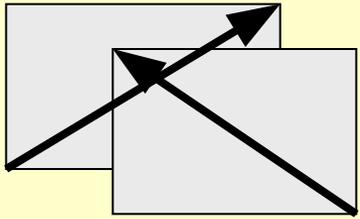


Точный тест

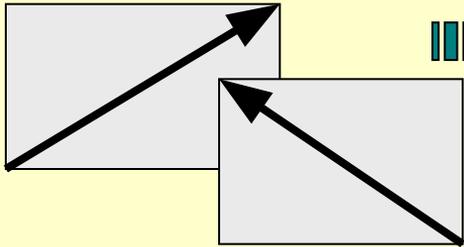
Определение факта пересечения отрезков.

Два отрезка пересекаются тогда и только тогда, когда каждый из отрезков пересекается с прямой, содержащей другой отрезок.

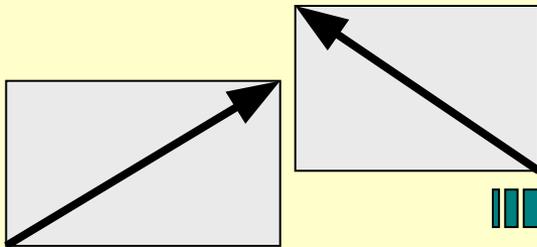
Задача 9-5



Из пересечения отрезков следует пересечение ограничивающих прямоугольников.



Из пересечения прямоугольников не следует пересечение отрезков.



Из непересечения прямоугольников следует непересечение отрезков.

Задача 9-6

Сравнение чисел с плавающей запятой

Даны **A** и **B** -- числа с плавающей запятой.

Для сравнения этих чисел вместо **A = B** нужно писать $(B - k) < A < (B + k)$, где **k** абсолютная погрешность вычислений числа **B**.

Нахождение точки пересечения отрезков

x_1, y_1 и x_2, y_2 -- координаты вершин первого отрезка.

x_3, y_3 и x_4, y_4 -- координаты вершин второго отрезка.

Для нахождения пересечения составляем уравнения:

$$(x-x_1)/(x_2-x_1) = (y-y_1)/(y_2-y_1)$$

$$(x-x_3)/(x_4-x_3) = (y-y_3)/(y_4-y_3)$$

Эти уравнения определяют прямую, проходящую через две точки.

Из уравнений находим x и y по следующим формулам:

$$x := ((x_1*y_2 - x_2*y_1)*(x_4 - x_3) - (x_3*y_4 - x_4*y_3)*(x_2 - x_1)) / ((y_1 - y_2)*(x_4 - x_3) - (y_3 - y_4)*(x_2 - x_1))$$

$$y := ((y_3 - y_4)*x - (x_3*y_4 - x_4*y_3)) / (x_4 - x_3)$$

Задача 9-7

Проверка условия существования точки пересечения

$((x1 \leq x) \text{ and } (x2 \geq x) \text{ and } (x3 \leq x) \text{ and } (x4 \geq x)) \text{ or } ((y1 \leq y) \text{ and } (y2 \geq y) \text{ and } (y3 \leq y) \text{ and } (y4 \geq y))$

Проверка параллельности отрезков при помощи угловых коэффициентов

$$k1 := (x2 - x1) / (y2 - y1)$$

$$k2 := (x4 - x3) / (y4 - y3)$$

где $k1$ и $k2$ – тангенсы угла наклона отрезков к положительному направлению оси Ox .

Если $k1 = k2$, то отрезки параллельны, а значит, не имеют точек пересечения.

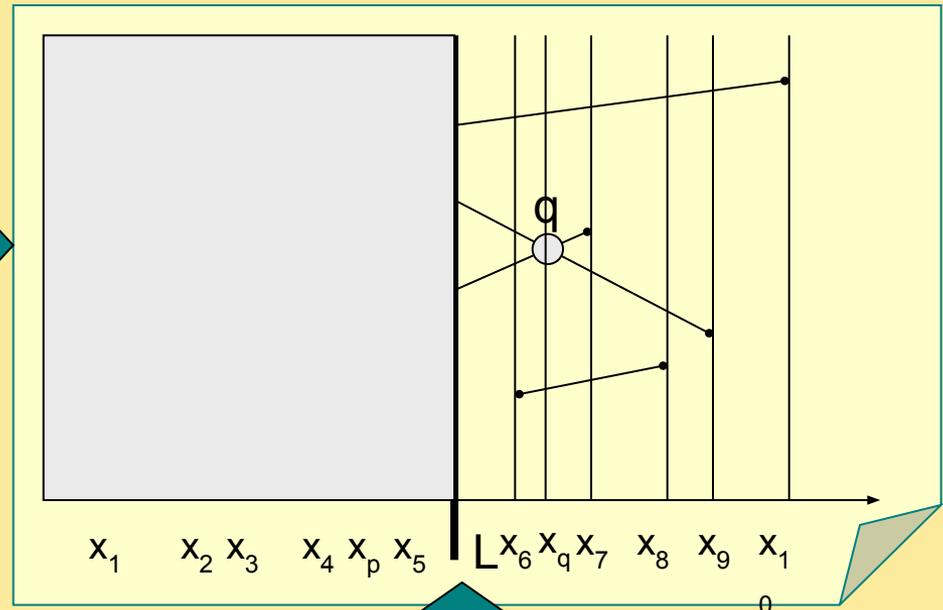
Задача 9-8

Идея плоского заметания

Возьмем вертикальную прямую L , которая разбивает плоскость на правую и левую полуплоскости.

Пусть каждая из полуплоскостей содержит концы отрезков.

Решение задачи может быть получено объединением решений для каждой полуплоскости.



Представим, что вертикальная прямая движется слева направо. По ходу движения будут пройдены все вертикальные сечения. Следовательно, будут выявлены все пары смежных отрезков и обнаружены все пересечения.

Искомое пересечение может произойти только между отрезками, чьи пересечения с некоторой вертикалью смежны.

Задача 9-9

Непрерывное заметание плоскости является метафорой.

Алгоритм, основанный на идее заметающей прямой, обрабатывает конечное множество «скачков» прямой по границам вертикальных полос.

Точки событий — точки, по которым «скачет» заметающая прямая.

Структуры данных, используемые в методе заметания

Список точек событий — последовательность абсцисс, упорядоченных слева направо и определяемых позициями «остановок» заметающей прямой.

Статус заметающей прямой — описание пересечения этой прямой с набором заданных отрезков. Статус должен обновляться в точках событий.

Задача 9-10

Упрощающие
предположения

Среди отрезков нет вертикальных.

Никакие три отрезка не имеют общих точек.

Отношение порядка на отрезках

s_1 выше s_2 в x , если s_1 и s_2 сравнимы в x , а точка пересечения отрезка s_1 с вертикалью x лежит выше точки пересечения отрезка s_2 с ней же.

Рассмотрим пару непересекающихся отрезков s_1 и s_2 .

Отрезки сравнимы в абсциссе x , если существует такая вертикаль, проходящая через x , которая пересекает оба отрезка.

