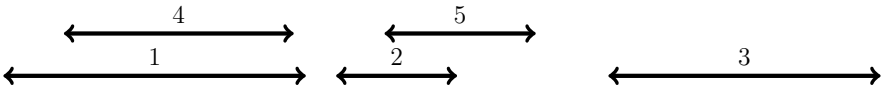
Алгоритмы и структуры данных

Лекция 24
Геометрия. Задача принадлежности. Пересечение полуплоскостей. Диаграмма Вороного. Триангуляция Делоне.

Сергей Леонидович Бабичев

Задача принадлежности.

Задача 1. [Задача принадлежности] Несамопересекающийся многоугольник M задан своими рёбрами. Дана точка P. Определить, принадлежит ли она внутренности M.

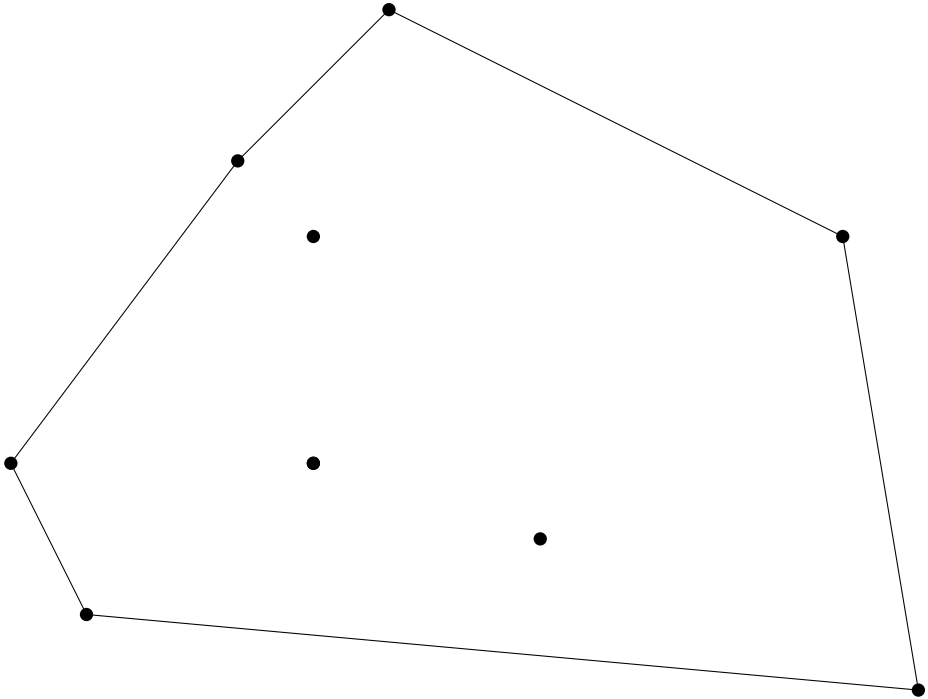


Задача принадлежности: способ 1

- Провести все углы между смежными вершинами и просуммировать их. Если сумма равна нулю — точка вне многоугольника.
  @рисунок
- Если $\pm 2\pi$, то внутри.
- Проблема — накапливающаяся погрешность. Вещественная арифметика даже в целочисленных координатах. Тригонометрия.

Второй способ: из точки выпустим горизонтальный луч вправо. Нечётное количество точек пересечения — как будто внутри. Если луч не проходит через какую-либо вершину — всё корректно. Тогда каждое пересечение со стороной есть изменение текущей области, где мы находимся. При движении по лучу всё чередуется. Проблема: вершины могут оказаться на луче, одно пересечение, но оно двойное. @@рисунок с плохим случаем.
Решение 1. Вместо горизонтального луча пускаем случайный. Либо выбираем тот, что не проходит через вершины. недостатки: могут появиться double, рандомизированный алгоритм.

Решение 2. Пусть все координаты имеют значение от 0 до A. Это можно обеспечить пересчётом. Пустим из P в точку $Q = (P + A, P + A + 1)$ луч. На нём нет целочисленных точек, кроме концов, так как $\gcd(A, A + 1) = 1$.
Недостаток: надо знать A.



Решение 3. Уметь обрабатывать вырожденные случаи. Оставим луч горизонтальным.

Пустим луч, переберём все стороны многоугольника. Пусть очередная сторона будет (a, b). Поменяем их местами так, чтобы $b.y \geqslant a.y$, т. е. b — более высокая.

@рисунок

Пропустив все такие стороны, в которых $b.y \leqslant p.y \vee a.y > p.y$. @рисунки.

Знак точного неравенства важен!

Теперь проверим, пересекает ли луч сторону:

$cross(p - a, b - a) < 0$ — условие пересечения.

Случаи: @@ рисунки.

Если a и b в разных полуплоскостях, то $cross$ отрицательный. После того, как это проделали для всех сторон, нечётный $count$ — точка в многоугольнике.
@@ рисунки случаев.

Итак, общий алгоритм.

```
int count = 0;
for (int i = 0; i < m.size(); m++) {
    point a = p[i], b = p[i+1];
    if (b.y < a.y) swap(a,b);
    if (b.y <= p.y || a.y > p.y) continue;
    if (cross(p-a, b-a) < 0) count++;
}
return count%2 == 1;
```

Почему это работает?
Мы пропускаем (a, b), если отрезок полностью в нижней полуплоскости или b на границе.
Замечание к алгоритму: можно сначала проверить, не лежит ли p на границе какого-либо отрезка, но это можно делать и параллельно с основным алгоритмтм.
Почему можно игнорировать часть отрезков?
@рисунки тех, с которыми ясно.
Опасные: Много рисунков.

Пересечение полуплоскостей.

Задача 2. Даны N полуплоскостей. Найти фигуру, образованную их пересечением.
@@Рисунок

Как выглядит эта фигура? Это выпуклый многоугольник, неограниченная выпуклая фигура или пустое множество.

Lemma

Пересечение выпуклых фигур — выпуклая фигура.

Доказательство.

Возьмём точки P и Q, лежащие в пересечении. Они лежат во всех пересекающихся фигурах. Фигуры — выпуклые $\rightarrow$ отрезок PQ лежит во всех фигурах. $\square$

Полуплоскости представляем неравенствами $ax + by + c \geqslant 0$. Нормаль удобно нормализовывать, чтобы привести параллельные прямые к единому виду.
Сам вектор нормали к точек x_0 на прямой лежит в полуплоскости:
$ax_0 + by_0 + c = 0$. Точка $(x_0 + a, y_0 + b)$ даёт
$a(x_0 + a) + b(y_0 + b) + c = ax_0 + a^2 + by_0 + b^2 + c = a^2 + b^2 > 0$.

Дополним плоскостями:

$$x \leqslant \infty$$
$$x \geqslant -\infty$$
$$y \leqslant \infty$$
$$y \geqslant -\infty$$

Это называется *bounding box*.
Смысл: если в пересечении есть BB, то осторожно: все пересекающиеся прямые должны быть в BB.

Алгоритм за $O(N^2)$.
Построим BB. Каждой очередной прямой отрезаем от BB куски. @рисунки.

Храним многоугольник промежуточного ответа и сужаем его пересечением м i-й плоскостью.
На каждом шаге $O(N)$ вершин, N шагов $\to O(N^2)$.

Как найти пересечение M с полуплоскостью.
Алгоритм должен дать список вершин многоугольника в порядке обхода, включая вершины на границе BB.
Случай 1: пересечений нет. Проверка: если полуплоскость внутри — ничего не меняется.
Для любого $v \in M : av_x + bv_y + c \geqslant 0$ — ничего не делаем. < 0 — $\emptyset$.

Случай 2: пересечения есть. @рисунок
Для каждого i рассмотрим сторону p_i, p_j.
Если $ap_{1x} + bp_p1y + c > 0 \wedge ap_{2x} + bp_p2y + c > 0$ или оба < 0 — они по одну сторону.
Если сторона целиком < 0 — в список ничего не добавлять.
Если сторона целиком > 0 — добавить p_{i+1}.
Если раньше не лежали — добавить q и p_{i+1}. @рисунок.
Если раньше лежали, но перестали лежать — добавляем только q. @рисунок.
В алгоритме замыкаем p_n с p_1.
Все добавленные точки добавляем в новый вектор Q.

Алгоритм 2 за $O(N \log N)$.
Разобьём полуплоскости на два класса.

1. — вектор нормали $[0..\pi)$, @рисунок
2. — вектор нормали $\pi, 2\pi)$. @рисунок

Сортируем оба класса отдельно по увеличению угла нормали, затем сливаем в единый массив. Сортировка обоих сразу не прошла бы.

Имеется отсортированный список полуплоскостей. Потребуется deque.
@рисунок
Что меняется в деке, когда приходит новая полуплоскость?
Экстремальный случай — удаляется несколько полуплоскостей с начала дека и с конца дека. Несущественные неравенства (то, что на пойдёт в финальный результат) убираются.
Пока первые плоскости несущественны — удаляем их pop_front. Пока последние плоскости несущественны — плоскостей удаляем их pop_back. В выбранном нами порядке обхода всегда будут отрезаться насколько первых и несколько последних плоскостей.