Содержание

1 Введение. Предикат поворота.

- Предикат поворота для двух векторов $v_1 = (x_1, y_1)$ и $v_2 = (x_2, y_2)$. Смотрим на знак $v_1 \times v_2 = x_1 \cdot y_2 - y_1 \cdot x_2$.
  - $v_1 \times v_2 < 0$. Если мы встанем в начале вектора v_1 и будем смотреть в сторону его конца, v_2 будет находиться в правой полуплоскости.
  - $v_1 \times v_2 = 0$. Вектора коллинеарны.
  - $v_1 \times v_2 > 0$. v_2 находится в левой полуплоскости относительно v_1.
- Примеры использования предиката поворота.
  - Проверка, что точки A и B находятся по разные стороны от прямой CD.
  - Проверка, что у двух отрезков AB и CD есть общая точка.
    - Если все четыре точки лежат на одной прямой, решаем одномерную задачу.
    - Иначе, проверяем что A и B не лежат по одну сторону от прямой CD. И что C и D не лежат по одну сторону от прямой AB.
  - Сортировка множества векторов по углу.
    - Разбиваем множество всех векторов на две части.
      1. Строго ниже OX и отрицательное направление OX.
      2. Строго выше OX и положительное направление OX.
    - Внутри одной части сортируем по предикату поворота.

- Проблемы с точностью при использовании чисел с плавающей точкой.
  - Оценка погрешности операций.
  - Использование интервальной арифметики.
  - Использование длинной арифметики и рациональных чисел.

2 Многоугольники. Планарные графы.

- Простой многоугольник.
- Выпуклый многоугольник.
  - Проверка простого многоугольника на выпуклость.
  - Проверка, что многоугольник простой и выпуклый.
- Определение направления обхода простого многоугольника.
- Проверка, что точка лежит внутри...
  - треугольника;
  - выпуклого многоугольника за $O(n)$;
  - выпуклого многоугольника, предподсчет $O(n)$, запрос $O(\log n)$;
  - невыпуклого многоугольника за $O(n)$.
- Плоский граф (но мы будем называть планарным).
- Граф, двойственный к планарному.
- Триангуляция многоугольника.
  - Диагональ многоугольника. Доказательство существования.
  - Доказательство существования триангуляции по индукции, разрезаем по диагонали.
- Площадь многоугольника.

3 Триангуляция многоугольника. Выделение граней связного планарного графа. DCEL.

- Ухо многоугольника. Доказательство, что у многоугольника есть хотя бы 2 уха.
- Ушная триангуляция за $O(n^3)$.
- Ушная триангуляция за $O(n^2)$.
- Doubly connected edge list (DCEL).
- Построение DCEL.
- Проверка, является ли грань внешней.

4 Выпуклая оболочка.

- Алгоритм Джарвиса (заворачивание подарка).
- Алгоритм Грэхема.
- Алгоритм Эндрю.
- Нахождение касательных из точки к многоугольнику.
- Алгоритм Чена.
- QuickHull.
- Выпуклая оболочка простого многоугольника, простой полилинии.

5 Динамическая выпуклая оболочка.

- Точки только добавляются.
- Оффлайн версия, все запросы известны заранее.
- Полная версия за $O(\log^2 n)$ на запрос
  - Строим верхнюю половину выпуклой оболочки. Точки на выпуклой оболочке расположены в порядке возрастания x.
  - Пусть есть два множества точек, разделенных вертикальной прямой. И нам известны выпуклые оболочки для обоих множеств. Тогда чтобы построить выпуклую оболочку для объединения, нужно найти общую касательную. Ее можно зайти за $O(\log n)$ параллельными бинпоисками по двум выпуклым оболочкам.
  - Все точки будем хранить в дереве поиска по X. Причем, сами точки будем хранить в листьях. Тогда, чтобы получить выпуклую оболочку для внутренней вершины, нужно взять выпуклые оболочки для её сыновей и найти у них общую касательную.
  - Чтобы это можно было сделать, нужно чтобы в каждой вершине дерева поиска хранилась выпуклая оболочка для её поддерева в каком-то виде. Мы будем хранить выпуклую оболочку в (другом) дереве поиска.
  - Однако, если в каждой вершине дерева поиска целиком хранить выпуклую оболочку точек из её поддерева, суммарная память может быть $\Omega(n^2)$. Поэтому, мы будем использовать для хранения выпуклой оболочки персистентное дерево поиска. Когда мы хотим вычислить выпуклую оболочку для вершины, и выпуклые оболочки для её сыновей уже известны, мы можем взять выпуклые оболочки для сыновей, бинпоиском найти общую касательную, посплитить и склеить их. Но так как выпуклые оболочки для сыновей хранились в персистентном дереве, в результате этой операции, они не испортились. Минус этого способа в том, что занимаемая структурой память не линейна от числа запросов (ее асимптотика равна времени).
  - Другой способ позволяет добиться линейной памяти. Давайте будем храни

6 Разное.

6.1 Сумма Минковского.

- Определение.
- Сумма Минковского двух выпуклых многоугольников.
- Построение за $O(n \log n)$.

- Построение за $O(n)$.
- Применение.

6.2 Две наиболее удаленные точки.

- Вращающиеся калиперы. $O(n)$.

6.3 Две наиболее близкие точки.

- Способ 1. Сканирующая прямая.
- Способ 2. Рандом шафл + Сетка.

6.4 Минимальный покрывающий круг.

- Алгоритм за $O(n^4)$.
- Рекурсивный алгоритм за $O(n^4)$.
- Доказательство линейного времени работы, если сделать рандом шафл в начале.

7 Локализация с помощью сканирующей прямой.

- Во всех задачах будем использовать сканирующую прямую слева направо, и поддерживать порядок объектов на этой прямой.
- Дано множество попарно непересекающихся окружностей. Построить дерево вложенностей.
- Дано множество отрезков.
  - Проверить, есть ли пара пересекающихся.
  - Найти все пары пересекающихся за $O((n + ans)\log(n + ans))$.
  - Найти все пары пересекающихся за время $O((n + ans)\log n)$ и память $O(n)$.
- Определить грань (связного) планарного графа, в которой лежит точка.
  - Все запросы даны заранее (оффлайн алгоритм).
  - Запросы поступают по-одному (онлайн алгоритм).
- Научиться строить грани для произвольного (не обязательно связного) планарного графа.

8 Пересечение полуплоскостей

- Полуплоскость задается неравенством $Ax + By + C \geqslant 0$.
- Пересечение за $O(n^2)$.
- Рандомизированный алгоритм для нахождения какой-нибудь точки в пересечении за $O(n)$.
- Пересечение полуплоскостей, с векторами нормалей внутри угла 180°. Объединение верхней и нижней половины.
- Пересечение полуплоскостей, если известна точка в пересечении. В частности, если у всех полуплоскостей $C \geqslant 0$.
- Общий случай пересечения полуплоскостей.

9 Триангуляция многоугольника за $O(n \log n)$.

- Триангуляция монотонного многоугольника за $O(n)$.
- Разбиение произвольного многоугольника на монотонные.
  - Используем метод сканирующей прямой слева направо.
  - Разбиваем все вершины на 5 типов: regular, begin, end, split, merge.
  - Нужно провести несколько диагоналей так, чтобы из каждой вершины merge была проведена диагональ направо, и в каждую вершину split была проведена диагональ слева.

10 Триангуляция Делоне.

11 Диаграмма Вороного.