

Вычислительная геометрия, день 2

Борис Золотов Ольга Карманова

22 июня 2023 г.

Летняя школа МКН СПбГУ

Содержание

Диаграммы Вороного

Триангуляции Делоне

VD и DT — двойственные графы

Выпуклая оболочка в 3D

Gift Wrapping

Divide & Conquer

Lifting

Двойственность точек и прямых

Проблема локализации точки

Алгоритм Чена—Хана

Теорема Александрова

Склейки: важные примеры

Диаграммы Вороного

Диаграмма Вороного

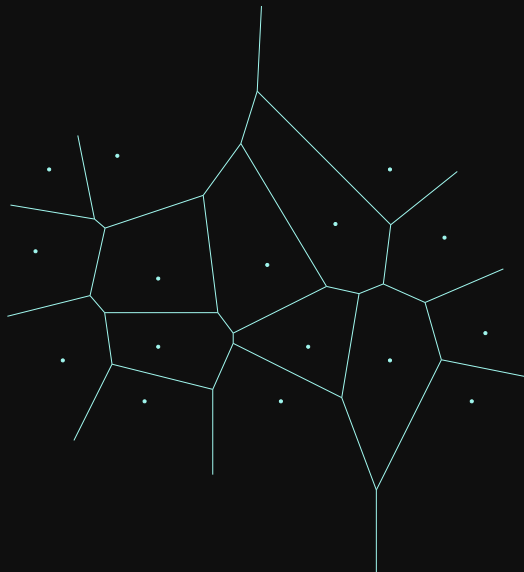
Определение

Пусть S — конечное подмножество \mathbb{R}^2 (его точки — сайты).

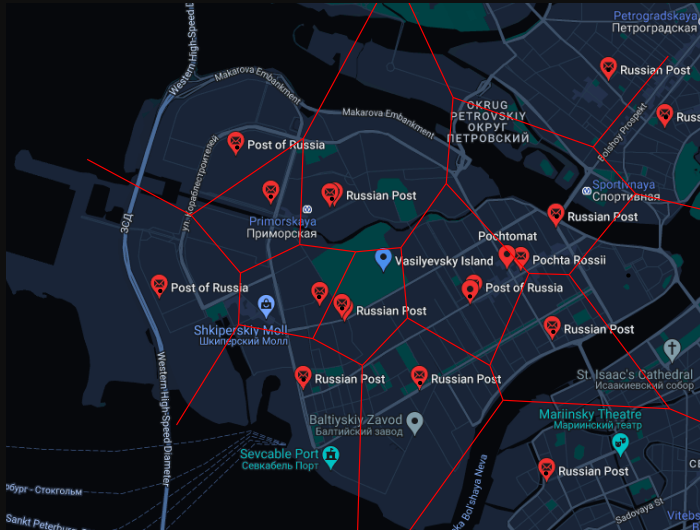
Диаграммой Вороного $\mathcal{VD}(S)$ называется разбиение плоскости на n ячеек $\{F_i\}$, при котором $x \in F_i \Leftrightarrow s_i \in S$ — ближайший к x сайт.

Мы рассматриваем $\mathcal{VD}(S)$ в рамках ограниченной области плоскости, содержащей все её вершины.

Диаграмма Вороного



Post office problem



Свойства диаграмм Вороного

- Серединный перпендикуляр к $s_i s_j$ делит плоскость на две гиперплоскости. Пусть $H(s_i s_j)$ — та из них, которая содержит s_i . Тогда

$$F_i = \bigcap_{j=1, j \neq i}^n H(s_i, s_j)$$

- Вершина — центр окружности, проходящей через 3 соседних с ней сайта.
- Клетка сайта s_i неограничена $\Leftrightarrow s_i \in \mathcal{CH}(S)$.

Свойства диаграмм Вороного

- $\mathcal{VD}(S)$ — планарный трёхсвязный 3-регулярный граф.
- Удобно хранить в *DCEL*.
- Если $|S| = n$, то $V(\mathcal{VD}(S)) \leq 2n - 5$, $E(\mathcal{VD}(S)) \leq 3n - 6$.
- Занимает линейный размер по памяти.

Построение диаграммы Вороного: Brute force

Идея: пересекать серединные перпендикуляры.

Построение ячейки для сайта s_i :

1. Проводим все серединные перпендикуляры между s_i и s_j , $i \neq j$.
2. Пересекаем попарно все серединные перпендикуляры, получаем $O(n^2)$ точек.
3. Каждую проверяем на принадлежность каждой из $n - 1$ полуплоскостей.

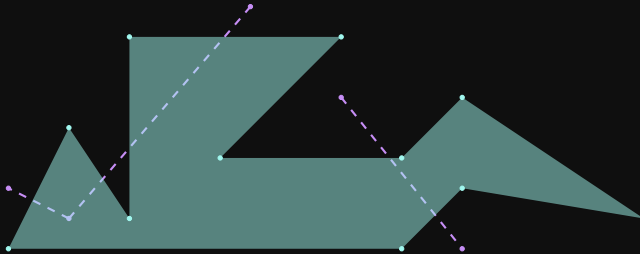
Время работы: $T(n) = n \cdot O(n^2) \cdot (n - 1) = O(n^4)$.

Построение $\mathcal{VD}(S)$: Пересечение полуплоскостей

Рассмотрим теперь более быстрый алгоритм, использующий геометрические идеи, связанные с пересечением полуплоскостей. Сначала мы реализуем несколько важных процедур, а после научимся более оптимально пересекать полуплоскости.

Принадлежность точки выпуклому многоугольнику

- Пускаем луч из точки в произвольном направлении.
- Если он пересекает границу многоугольника нечетное число раз, то точка внутри. Если четное — снаружи.



Пересечение выпуклых многоугольников

Пользуясь предыдущей процедурой, работающей за $O(n)$ (и $O(\log(n))$ на выпуклом), получаем наивный алгоритм:

- Пересекаем каждую сторону первого многоугольника с каждой стороной второго: получаем $O(m)$ точек I_{e_i, e_j} (макс. две на ребре), которые будут вершинами пересечения.

Пересечение выпуклых многоугольников

- Надо отсортировать I_{e_i, e_j} и вершины по аргументу относительно какой-либо точки пересечения (например, одной из них).

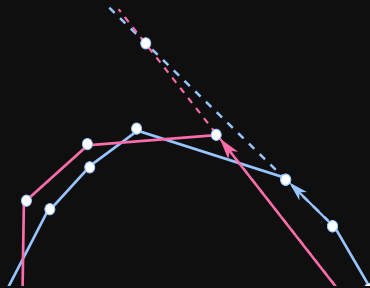
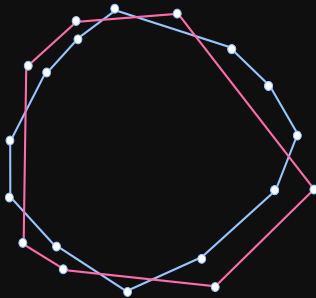
$$O((m + n) \cdot \log(m + n)).$$

- Из упорядоченного набора точек надо выкинуть не-вершины. Для этого проверяем *вершины многоугольников* (I_{e_i, e_j} не надо) на принадлежность обоим многоугольникам.

$$O((m + n) \cdot (\log m + \log n)).$$

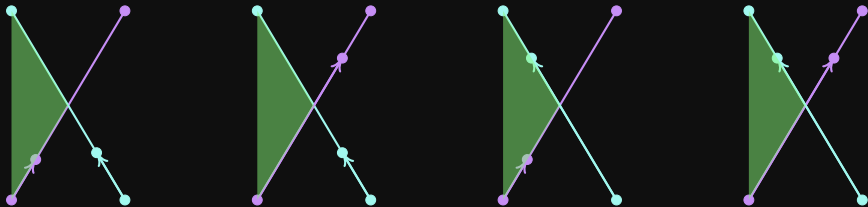
Пересечение вып. многоугольников: O'Rourke

- Вершины многоугольников упорядочены против часовой стрелки.
- Движемся по границам и на каждом шаге определяем, по какому многоугольнику переходить на одну вершину дальше.



Пересечение вып. многоугольников: O'Rourke

- Каждый раз получаем одну из восьми ситуаций:



- В случаях (1) и (2) двигаемся по второму многоугольнику, в случаях (3) и (4) — по первому.
Кроме того, для случая (4) необходимо вычислить и добавить точку пересечения.

Пересечение вып. многоугольников: O'Rourke

- В результате полного обхода получаем точки пересечения. Если каждый раз будем брать ломаные, лежащие левее в порядке обхода, получим пересечение.
- Интересно, что если каждый раз будем брать лежащую правее — получим объединение.

Время работы: $T(n, m) \leq 2(n + m) = O(n + m)$.

Построение $\mathcal{VD}(S)$: Пересечение полуплоскостей

Наконец, перейдем к построению диаграммы Вороного.

- Строим все серединные перпендикуляры к отрезкам, соединяющим сайт s_i с остальными сайтами, получаем $n - 1$ полуплоскость.
- Пересекаем все полуплоскости следующим образом: Сначала все пары, потом пары пересечений и так далее.
- Получаем клетку для выбранного сайта.

Время работы: $T(n) = O(n \cdot n \cdot \log(n)) = O(n^2 \log(n))$.

Недостатки этих алгоритмов

- Неоптимальное время работы — можно за $O(n \log(n))$.
- На выходе получаем список клеток (без всякой информации о их взаимосвязях).

Решение: алгоритм Форчуна с использованием DCEL.

Или алгоритм $D\&C$ за $O(n \log(n))$.

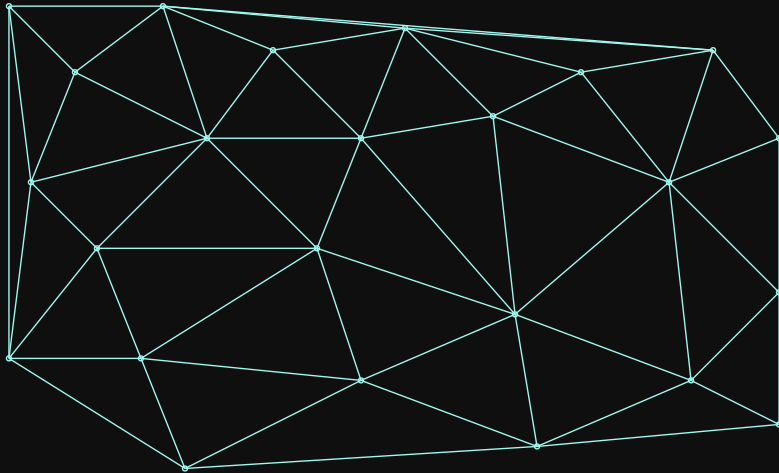
Триангуляции Делоне

Триангуляция Делоне

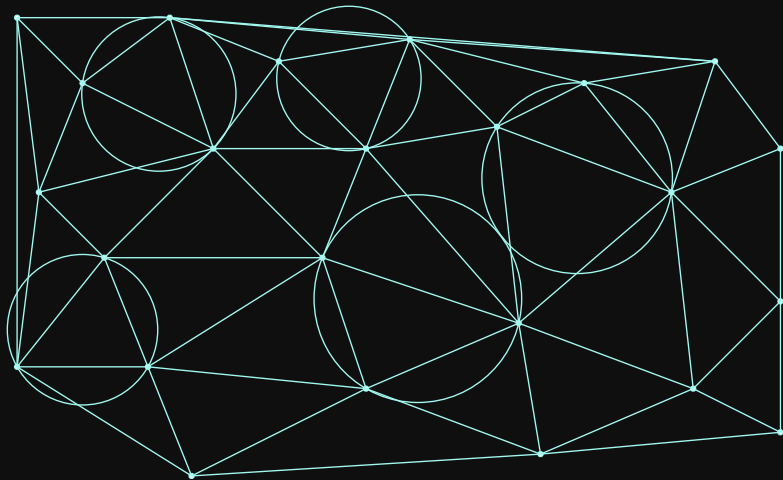
Определение

Пусть S — конечное подмножество \mathbb{R}^2 . Триангуляцией Делоне $\mathcal{DT}(S)$ называется такая триангуляция этого множества, при которой круг, описанный около каждого треугольника не содержит других вершин триангуляции.

Триангуляция Делоне



Триангуляция Делоне

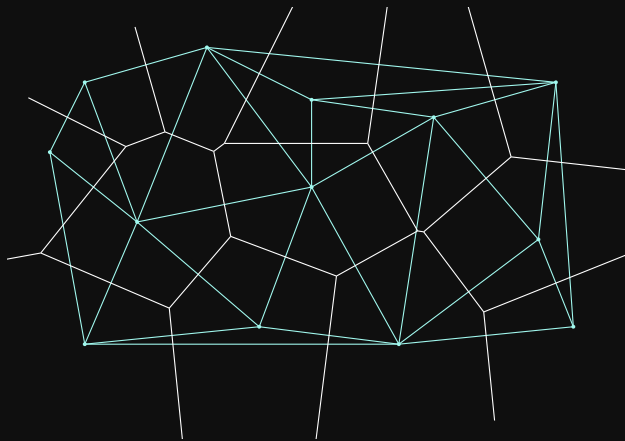


VD и DT — двойственные графы

$\mathcal{VD}(S)$ и $\mathcal{DT}(S)$ — двойственные графы

Теорема

$\mathcal{DT}(S)$ — двойственный граф к $\mathcal{VD}(S)$.



$\mathcal{VD}(S)$ и $\mathcal{DT}(S)$ — двойственные графы

Доказательство.

Рассмотрим сайты s_i, s_j, s_k из одного треугольника в триангуляции Делоне. Тогда вершина диаграммы Вороного — центр описанной окружности треугольника $\triangle s_i s_j s_k$.

Так как центр описанной окружности единственный, мы получили взаимнооднозначное соответствие.



$\mathcal{VD}(S)$ и $\mathcal{DT}(S)$ — двойственные графы

Следствие

При условии общего положения триангуляция Делоне множества точек единственна.

То есть, если мы умеем строить диаграмму Вороного и храним её в DCEL, то мы умеем строить и триангуляцию Делоне.

Свойства триангуляции Делоне

- Максимизирует минимальный угол.
- *MST* — подграф триангуляции Делоне (что позволяет относительно эффективно его считать).
- Так как триангуляция Делоне избегает узких (похожих на вырожденные) треугольников, её часто используют в самом разном моделировании. Например, ландшафтном.

Выпуклая оболочка в 3D

Постановка задачи

Даны точки $p_1, p_2, \dots, p_n \in \mathbb{R}^3$.

Построить **DCEL**, соответствующий граням выпуклой оболочки множества $S = \{p_1, \dots, p_n\}$, указывая p_i в записях вершин.

Граф любого многогранника планарен, поэтому количества рёбер, вершин и граней отличаются не более чем в константу раз. Отсюда количество записей и ссылок в итоговом DCEL будет $O(h)$.

Gift Wrapping

Gift wrapping

Обобщение алгоритма *Jarvis march* на трёхмерный случай.

Отрезок между самой нижней точкой и другой точкой, который имеет \min угол с горизонтальной плоскостью, будет ребром выпуклой оболочки.

Выбор следующей грани

while внешняя грань — не треугольник **do**

$p_i p_k p_l$ — грань, смежная с внешней

$p_k p_l$ — ребро внешней грани

for all $p \in S$ **do**

Найти угол $\angle((p_i p_k p_l), (p p_l p_k))$

end for

$p^* \in S$ — точка, для которой угол наибольший

Добавить грань $p^* p_l p_k$ в DCEL

end while

Выбор следующей грани

while внешняя грань — не треугольник **do**

$p_i p_k p_l$ — грань, смежная с внешней

$p_k p_l$ — ребро внешней грани

for all $p \in S$ **do**

Найти угол $\angle((p_i p_k p_l), (p p_l p_k))$

end for

$p^* \in S$ — точка, для которой угол наибольший

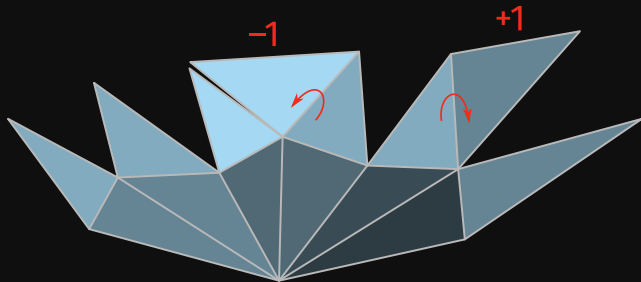
Добавить грань $p^* p_l p_k$ в DCEL

end while

▷ Время работы — $O(n \cdot h)$

Количество сторон внешней грани

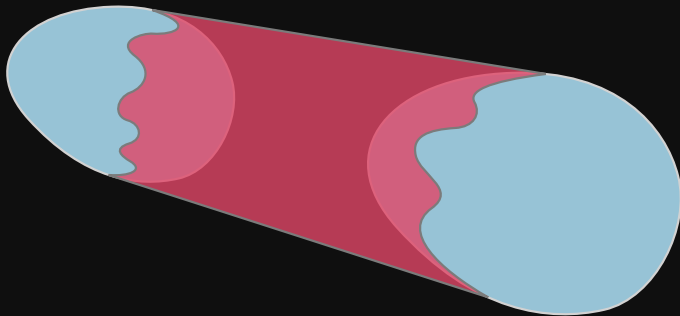
Количество сторон внешней грани может как увеличиться, так и уменьшиться при добавлении очередной грани выпуклой оболочки.



Divide & Conquer

Слияние двух выпуклых оболочек

При слиянии необходимо вычислить цилиндр, состоящий из граней, соединяющих выпуклые оболочки CH_1 и CH_2 .



Поиск первой грани

Рассмотрим нижнюю вершину и её соседа в \mathcal{CH}_1 и нижнюю вершину в \mathcal{CH}_2 . Проведём через них плоскость.

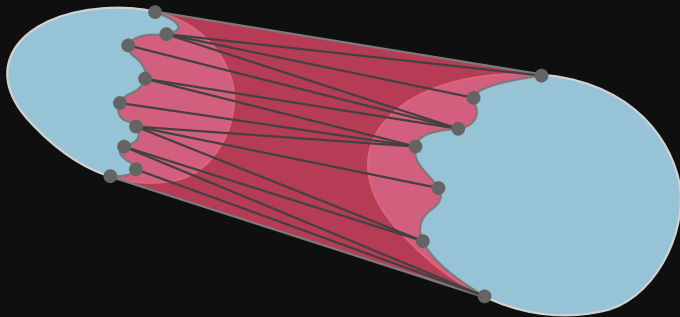
Если кто-то из соседей выбранных вершин лежит ниже этой плоскости, заменим одну из вершин на него.

Повторим процесс.

Картинка на доске

Вычисление цилиндра

Заметим, что рёбра, ограничивающие цилиндр, являются рёбрами выпуклых оболочек \mathcal{CH}_1 и \mathcal{CH}_2 .



Cylinder wrapping

Пусть p_i^1 и p_j^2 — последние вершины, добавленные к цилиндру из \mathcal{CH}_1 и \mathcal{CH}_2 соответственно;

$p_i^1 p_j^2 p_k^*$ — последняя известная грань цилиндра.

Рассмотрим всех соседей p_i^1 и p_j^2 в их выпуклых оболочках, добавим грань $p_k^* p_j^2 p_i^1$, которая образует наибольший угол с $p_i^1 p_j^2 p_k^*$, в цилиндр.

Повторим, пока не придём к изначальному ребру.

Время работы

Может показаться, что «проверяем всех соседей» — долго.

Однако заметим, что каждое ребро проверялось максимум дважды — при поиске как первой грани, так и каждой из последующих.

Отсюда сложность слияния линейна.

$T(n) = 2T(\frac{n}{2}) + O(n)$ даёт время $O(n \cdot \log n)$.

Lifting

Сведение DT к $3D\text{-}CH$

Покажем, что задача построения триангуляции Делоне на плоскости сводится к задаче построения выпуклой оболочки в \mathbb{R}^3 за линейное время.

Рассмотрим точки $\{p_1, \dots, p_n\} = S$ и поднимем их на параболоид:

$$(x, y) \mapsto (x, y, x^2 + y^2).$$

Теорема

Проекции граней выпуклой оболочки, нормаль которых направлена вниз, будут областями Делоне.

Пересечения с плоскостями и окружности

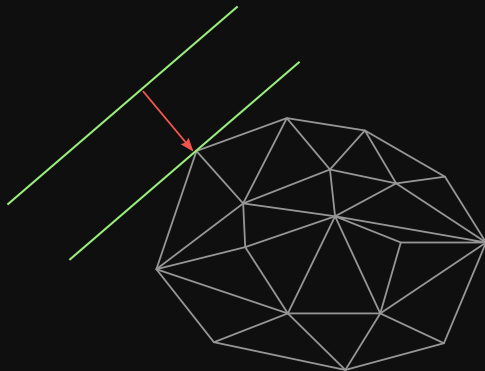
Лемма

Проекция пересечения параболоида $z = x^2 + y^2$ и плоскости $ax + by + cz = d$ на плоскость Oxy — окружность. Её центр не зависит от d .

Доказательство: раскроем скобки, коэффициенты при x^2 и при y^2 будут одинаковы; выделенные полные квадраты не будут меняться при изменении d .

Почему это работает

Вершина выпуклой оболочки $\mathcal{CH}(S)$ — это первая точка касания S и какой-то плоскости, придвигаемой снаружи.

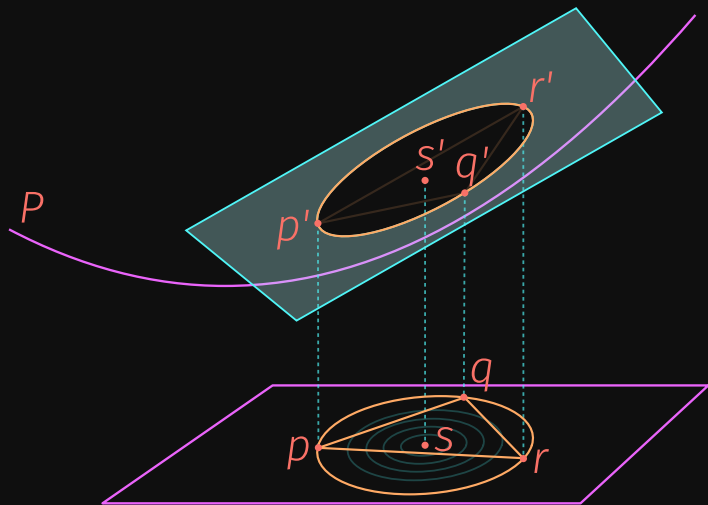


Ещё о касании

Если касание произошло сразу в трёх точках — значит, плоскость была параллельна грани выпуклой оболочки.

Будем поднимать плоскость, параллельную грани $СН$,
и одновременно смотреть, что происходит
на плоскости $Оху$.

Подъём плоскости



Подъём плоскости — обоснование

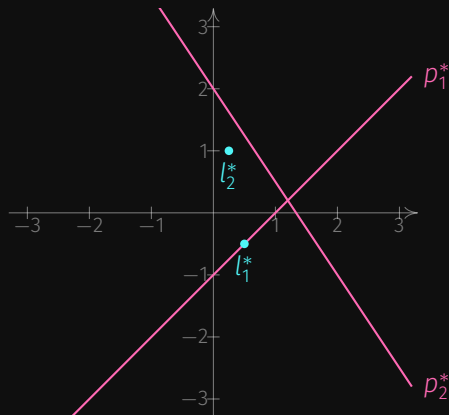
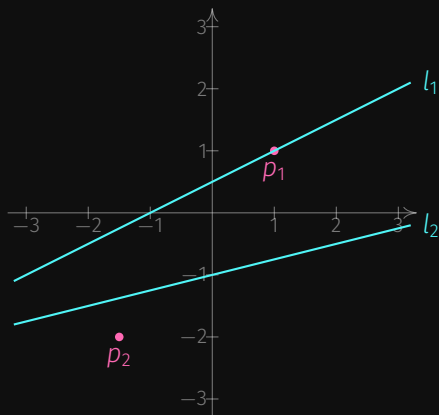
- Рассмотрим плоскость, которая касается P в точке s' — поднятом центре описанной окружности.
- Начнём поднимать её вверх. Коэффициент d растёт. Проекция пересечения с P — окружность с центром в s . Радиус увеличивается.
- Первые точки на P , которых коснётся плоскость, соответствуют первым точкам на Ox , через которые пройдёт расширяющаяся окружность.
- Треугольники Делоне соответствуют граням CH .

Диаграмма Вороного за $n \cdot \log n$

- Поднять точки на параболоид,
- Построить выпуклую оболочку,
- Спроецировать, получить триангуляцию Делоне,
- Перейти к двойственному графу, profit.

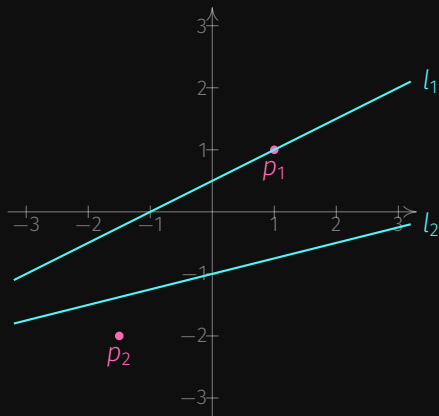
Двойственность точек и прямых

Определение

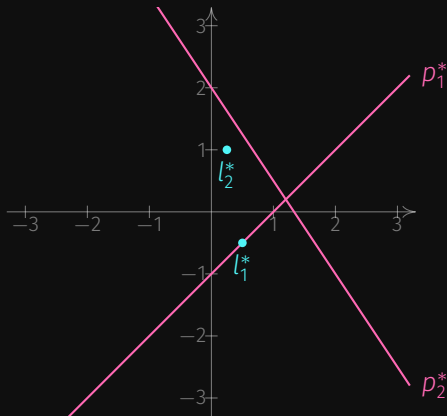


$$(a, b) \longleftrightarrow \{y = ax - b\}$$

Инцидентность и над / под

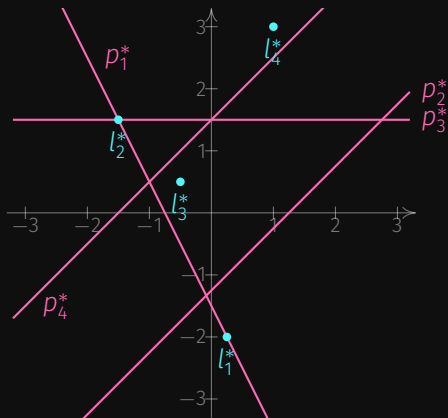
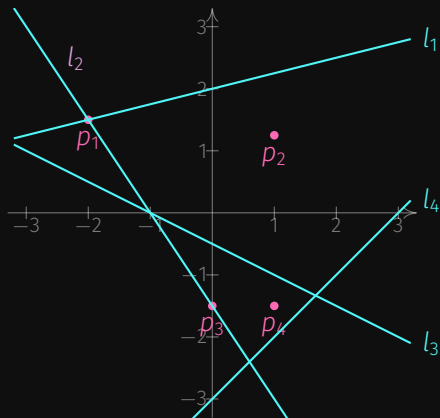


$$p \in l \iff l^* \in p^*$$



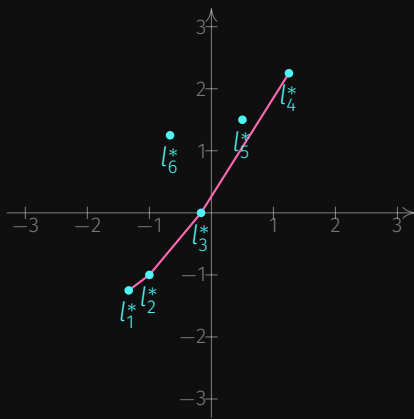
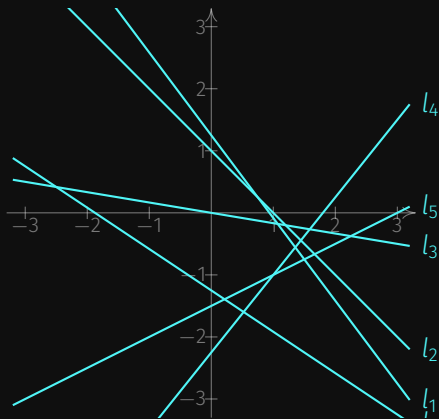
$$p \text{ над } l \iff l^* \text{ над } p^*$$

Ещё пример



$$l_1 \cap l_2 = p \iff p^* \text{ проходит через } l_1^*, l_2^*$$

Двойственность задач

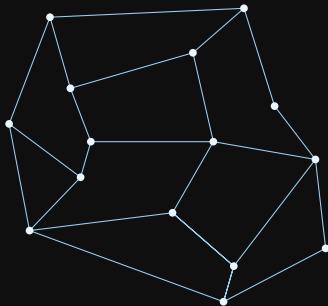


Верхнее пересечение полуплоскостей \iff Нижняя $СН$

Проблема локализации точки

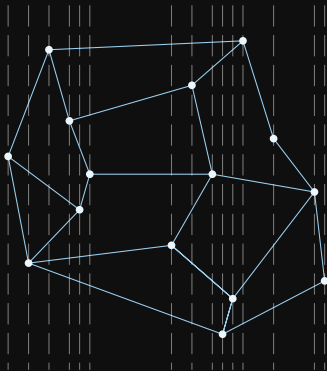
Постановка задачи

Дано разбиение плоскости на области, построить структуру данных, которая по точке будет указывать на её область.



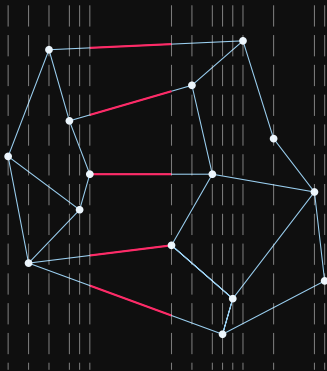
Vertical slabs

Проведём через каждую вершину вертикальную прямую, подразбив области. Внутри полосы набор областей фиксирован.



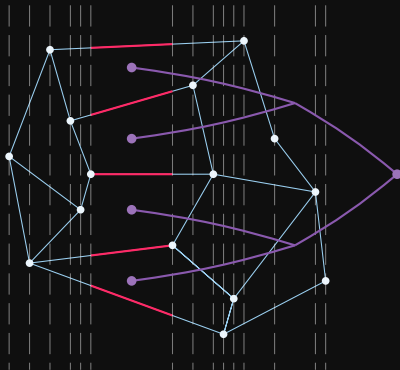
Разбиение внутри полосок

Как-нибудь (*sweeping line*) сохраним список отрезков, подразбивающих каждую область по вертикали.



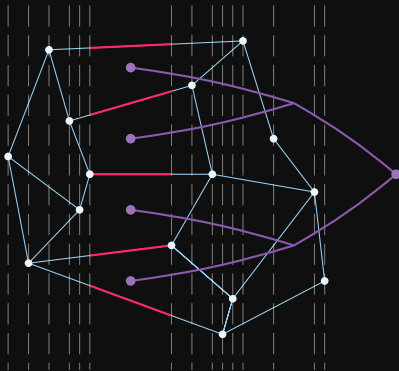
Двоичный поиск

Сначала по координате находим вертикальную полосу, потом, выясняя, выше или ниже точка отрезка, — область.



Анализ

Запрос — $O(\log n)$. На построение требуется $O(n^2)$ памяти и времени. Как улучшить — *persistent trees*.



Алгоритм Чена—Хана

Постановка задачи

Дана поверхность многогранника; все грани — треугольники.
Выбрана вершина S . Найти длины кратчайших
путей (*по поверхности многогранника*) от S
до каждой из остальных его вершин.

Устройство кратчайших путей

- Кратчайший путь переходит с грани на грань не более $n - 1$ раза,
- Кратчайшие пути не пересекаются, кроме как в S или в вершине назначения,
- При разгибании рёбер кратчайший путь превращается в отрезок прямой линии.

Дерево кратчайших путей

Грани многогранника хранятся в DCEL; путь определяется последовательностью пройденных граней.

Покажем, как пересчитываются пути при переходе на следующую грань и как формируется двоичное дерево.

One angle — one split

Лемма

Из двух узлов дерева, расположенных на одном полуребре, только у одного может быть два потомка, определяющих кратчайшие пути.

Эффективный алгоритм

Считаем двух детей листа и проверяем, кто теперь *оккупирует* вершину многогранника под ним. У того, кто (*теперь*) не оккупирует, удаляем одного ребёнка.

На каждой итерации у дерева линейное количество листьев: по одному на каждое полуребро, плюс по одному разделяющемуся на каждое полуребро.

Глубина дерева **не более n** .

Теорема Александрова

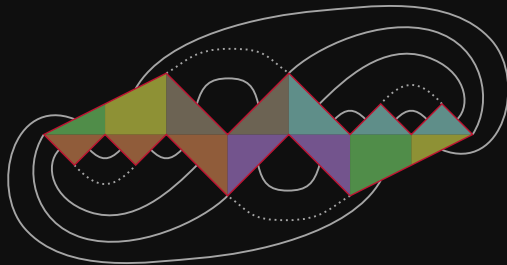
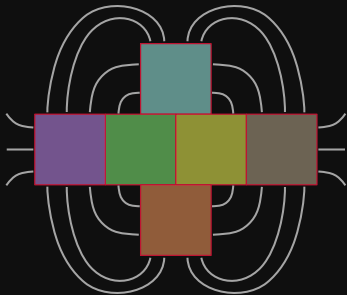
Отношение эквивалентности

Дан набор многоугольников. Разобьём $\bigcup \partial P_i$ на отрезки, пары отрезков одинаковой длины можно склеивать между собой.

Будем рассматривать **фактор-пространство**, получающееся при таком склеивании.

Теорема Александрова: склейки

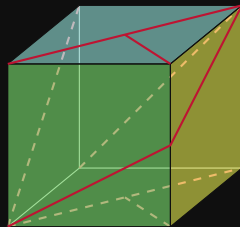
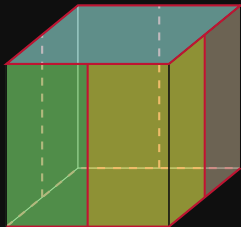
Нас интересуют склейки, которые гомеоморфны сфере и угол в каждой точке которых не превосходит 360° .



Теорема Александрова: многогранники

Теорема

Склейке, удовлетворяющей этим условиям, соответствует (изометричен) единственный выпуклый многогранник.



Склейки: важные примеры

Не существует замкнутой формулы

Координаты правильного N -угольника со стороной L не выражаются в радикалах.

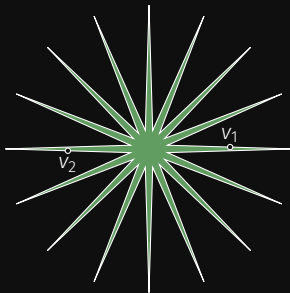


Отсюда координаты вершин бипирамиды (которая получается из набора треугольников с целыми сторонами по теореме Александрова) также не выражаются.

Экспоненциально много различных склеек

Теорема

Для любого чётного n существует многоугольник с n вершинами, из которого можно сделать не менее $2^{C \cdot n}$ комбинаторно различных **порёберных** склеек.



Спасибо за внимание!

Диаграммы Вороного

Триангуляции Делоне

VD и DT — двойственные графы

Выпуклая оболочка в 3D

Gift Wrapping

Divide & Conquer

Lifting

Двойственность точек и прямых

Проблема локализации точки

Алгоритм Чена—Хана

Теорема Александрова

Склейки: важные примеры